

Springer Tracts in Nature-Inspired Computing

Xin-She Yang *Editor*

Engineering Applications of AI and Swarm Intelligence



Springer

Springer Tracts in Nature-Inspired Computing

Series Editors

Xin-She Yang, School of Science and Technology, Middlesex University, London, UK

Nilanjan Dey, Department of Computer Science and Engineering, Techno International New Town, Kolkata, India

Simon Fong, Faculty of Science and Technology, University of Macau, Macau, Macao

The book series is aimed at providing an exchange platform for researchers to summarize the latest research and developments related to nature-inspired computing in the most general sense. It includes analysis of nature-inspired algorithms and techniques, inspiration from natural and biological systems, computational mechanisms and models that imitate them in various fields, and the applications to solve real-world problems in different disciplines. The book series addresses the most recent innovations and developments in nature-inspired computation, algorithms, models and methods, implementation, tools, architectures, frameworks, structures, applications associated with bio-inspired methodologies and other relevant areas.

The book series covers the topics and fields of Nature-Inspired Computing, Bio-inspired Methods, Swarm Intelligence, Computational Intelligence, Evolutionary Computation, Nature-Inspired Algorithms, Neural Computing, Data Mining, Artificial Intelligence, Machine Learning, Theoretical Foundations and Analysis, and Multi-Agent Systems. In addition, case studies, implementation of methods and algorithms as well as applications in a diverse range of areas such as Bioinformatics, Big Data, Computer Science, Signal and Image Processing, Computer Vision, Biomedical and Health Science, Business Planning, Vehicle Routing and others are also an important part of this book series.

The series publishes monographs, edited volumes and selected proceedings.

Xin-She Yang
Editor

Engineering Applications of AI and Swarm Intelligence

 Springer

Editor

Xin-She Yang
Department of Design Engineering
and Mathematics
Middlesex University
London, UK

ISSN 2524-552X

ISSN 2524-5538 (electronic)

Springer Tracts in Nature-Inspired Computing

ISBN 978-981-97-5978-1

ISBN 978-981-97-5979-8 (eBook)

<https://doi.org/10.1007/978-981-97-5979-8>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.

The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

If disposing of this product, please recycle the paper.

Preface

Many problems in engineering and industrial applications can be formulated as optimization problems with complicated objectives, subject to multiple highly nonlinear constraints. Solving such optimization problems requires sophisticated optimization algorithms and techniques. Traditional algorithms such as gradient-based methods are efficient, but they use multiple derivatives and calculations of these derivatives in high-dimensional space can have high computation costs. In addition, such derivative-based methods are typically local search methods and the final solutions may depend on the starting points if optimization problems are highly nonlinear and multimodal. An alternative and recent trend is to use derivative-free, nature-inspired algorithms. Nature-inspired algorithms, especially those based on swarm intelligence (SI), have been successfully applied to solve a variety of optimization problems in real-world applications, and thus, their popularity has also increased significantly in recent years. Furthermore, many problems in engineering applications are closely related to artificial intelligence (AI) and machine learning. Many techniques in AI are optimization techniques that use learning algorithms and training data to carry out more advanced tasks.

This edited book reviews and summarizes the state-of-the-art developments in AI and swarm intelligence with the emphasis on engineering applications. Topics include the comprehensive review of metaheuristic algorithms, swarm intelligence algorithms and their engineering applications, AI applications in civil engineering, AI-driven predictive maintenance of industrial machinery, AI in production systems with lean manufacturing, transmit beamforming designs in wireless communications by the firefly algorithm, optimum designs of reinforcement frames and optimum designs of seismic base isolator using the modified flower pollination algorithm, multi-objective metaheuristic algorithm for the design of digital differentiators with zero-phase response, modelling of quality-of-services in 5G networks, enhancement of SPECT 2D reconstruction with improved image quality, complex-valued neural networks, solution of numerical benchmarks, optimal camera placement for the maximum coverage of polygons with holes using SI-based algorithms and other applications.

This book is a state-of-the-art update on various engineering applications of AI and swarm intelligence. As a timely snapshot of the latest developments, this book will be interested to students, researchers, lecturers and professionals in engineering designs, AI, machine learning and swarm intelligence and can thus serve as an additional reference for graduates and researchers in computer science, evolutionary computing, computational intelligence and engineering applications, as well as researchers in various disciplines and industrial applications.

London, UK
May 2024

Xin-She Yang

Contents

1	Transmit Beamforming Designs in Wireless Communications Using the Firefly Algorithm	1
	Tuan Anh Le and Xin-She Yang	
2	Structural Optimization of Reinforced Concrete Frames with a Modified Flower Pollination Algorithm	33
	Panagiotis E. Mergos and Xin-She Yang	
3	Swarm Intelligence Algorithms and Their Engineering Applications	51
	Adam Slowik, Krzysztof Cpalka, Absalom Ezugwu, and Ali Wagdy Mohamed	
4	Convergence Catalysts: Exploring the Fusion of Embedded Systems, IoT, and Artificial Intelligence	69
	Cosmina Mihaela Rosca	
5	Investigation of the Effect of Lateral Displacement and Damping Limits of Optimum Designed Seismic Base Isolators with Flower Pollination Algorithm on Structural Control	89
	Ayla Ocak, Sinan Melih Nigdeli, Gebrail Bekdaş, and Xin-She Yang	
6	Review of Recent Advances on AI Applications in Civil Engineering	107
	Yaren Aydın, Gebrail Bekdaş, and Sinan Melih Nigdeli	
7	Meta-heuristic Algorithms as an Optimizer: Prospects and Challenges (Part I)	131
	Ata Jahangir Moshayedi, Seyed Taha Mousavi Nasab, Zeashan Hameed Khan, and Amir Sohail Khan	

8	Meta-heuristic Algorithms as an Optimizer: Prospects and Challenges (Part II)	155
	Ata Jahangir Moshayedi, Seyed Taha Mousavi Nasab, Zeashan Hameed Khan, and Amir Sohail Khan	
9	Depth Perspective-Aware Multiple Object Tracking	181
	Kha Gia Quach, Pha Nguyen, Chi Nhan Duong, Tien Dai Bui, and Khoa Luu	
10	AI-Driven Predictive Maintenance	207
	Ramazan Ünlü and İsmet Söylemez	
11	Artificial Intelligence in Production Systems and Integration with Lean Manufacturing	235
	İsmet Söylemez and Ramazan Ünlü	
12	A Multiobjective Metaheuristic Algorithm Perspective to Design Riesz Digital Differentiator Exhibiting Zero-Phase Response	253
	Chandan Nayak	
13	Utilizing Cobb-Douglas Production Function in the Modeling of Joint Security and Quality of Service (QoS) in 5G Network	285
	Tiew On Ting, Su Fong Chien, and Arash Bozorgchenani	
14	Enhancing Low-Dose SPECT 2D Reconstruction from One-Dimensional Projections By Using Conditional Generative Adversarial Networks for Improved Image Quality	301
	Adis Alihodzic	
15	Performance of Weightless Swarm Algorithm on Numerical Benchmark Functions	323
	Yu Dou and Tiew On Ting	
16	On Complex Neural Networks	343
	Momin Jamil Sandhu and Xin-She Yang	
17	Self-adaptive Big Bang-Big Crunch Algorithm for Engineering Designs	365
	Jéssica Cristiane Andrade, Claudemir Mota da Cruz, Fran Sérgio Lobato, Gustavo Barbosa Libotte, and Gustavo Mendes Platt	
18	Optimizing Camera Placement for Maximum Coverage of Simple Polygons with Holes: Deterministic Approaches and Swarm Intelligence Algorithms	389
	Adis Alihodzic, Eva Tuba, and Milan Tuba	

Editor and Contributors

About the Editor

Xin-She Yang obtained his D.Phil. in Applied Mathematics from the University of Oxford. He then worked at Cambridge University and National Physical Laboratory (UK) as a senior research scientist. Now, he is a reader at Middlesex University London and an elected Fellow of Institute of Mathematics and its Applications (FIMA). He was the IEEE CIS chair for the Task Force on Business Intelligence and Knowledge Management (2015–2020). He was also an editor-in-chief of International Journal of Mathematical Modelling and Numerical Optimisation (IJMMNO) (2009–2022). With more than 20 years of teaching and research experience, he has authored more than ten books and edited more than 15 books. He has published more than 250 peer-reviewed research papers with nearly 86,000 citations. According to Clarivate Analytics/Web of Sciences, he has been on the prestigious list of highly cited researchers for eight consecutive years (2016–2023).

Contributors

Adis Alihodžić Department of Mathematical and Computer Sciences, University of Sarajevo, Sarajevo, Bosnia and Herzegovina

Jéssica Cristiane Andrade Federal University of Uberlândia, Uberlândia, Brazil

Yaren Aydın Department of Civil Engineering, Istanbul University—Cerrahpaşa, Avcılar, Istanbul, Turkey

Gebrail Bekdaş Department of Civil Engineering, Istanbul University—Cerrahpaşa, Avcılar, Istanbul, Turkey

Arash Bozorgchenani School of Computing, University of Leeds, Leeds, UK

Tien Dai Bui Computer Science and Software Engineering Department, Concordia University, Montreal, QC, Canada

Su Fong Chien Multidisciplinary Research and Innovation Centre, MIMOS Berhad, Kuala Lumpur, Malaysia

Krzysztof Cpalka Department of Intelligent Computer Systems, Czestochowa University of Technology, Czestochowa, Poland

Claudemir Mota da Cruz State University of Santa Cruz, Ilhéus, Brazil

Yu Dou School of Engineering, University of Leicester, Leicester, UK

Chi Nhan Duong pdActive Inc., Longmont, USA

Absalom Ezugwu School of Computer Science and Information Systems, North-West University, Potchefstroom, North West, South Africa

Amir Sohail Khan School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi, China

Zeashan Hameed Khan Interdisciplinary Research Center for Intelligent Manufacturing and Robotics (IRC-IMR), King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia

Tuan Anh Le School of Science and Technology, Middlesex University, London, UK

Gustavo Barbosa Libotte Rio de Janeiro State University, Nova Friburgo, Brazil

Fran Sérgio Lobato Federal University of Uberlândia, Uberlândia, Brazil

Khoa Luu Electrical Engineering and Computer Science, University of Arkansas, Fayetteville, USA

Panagiotis E. Mergos Department of Engineering, School of Science and Technology, City, University of London, London, UK

Ali Wagdy Mohamed Department of Operations Research and Management, Cairo University, Giza, Egypt

Ata Jahangir Moshayedi School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi, China

Seyed Taha Mousavi Nasab Department of Mathematics, Iran University of Science and Technology, Tehran, Iran

Chandan Nayak School of Electronics Engineering, VIT-AP University, Amara-vati, Andhra Pradesh, India

Pha Nguyen Electrical Engineering and Computer Science, University of Arkansas, Fayetteville, USA

Sinan Melih Nigdeli Department of Civil Engineering, Istanbul University—Cerrahpaşa, Avcılar, Istanbul, Turkey

Ayla Ocak Department of Civil Engineering, Istanbul University—Cerrahpaşa, Avcılar, Istanbul, Turkey

Gustavo Mendes Platt Federal University of Rio Grande, Santo Antônio da Patrulha, Brazil

Kha Gia Quach pdActive Inc., Longmont, USA

Cosmina Mihaela Rosca Petroleum—Gas University of Ploiesti, Ploiesti, Romania

Momin Jamil Sandhu Harman Becker Automotive Systems GmbH, Karlsbad, Germany

Adam Slowik Department of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland

İsmet Söylemez Industrial Engineering Department, Abdullah Gul University, Kayseri, Turkey

Tiew On Ting Department of Digital Media and Technology, Stony Brook Institute at Anhui University (SBIAHU), Anhui University, Hefei, Anhui Province, People's Republic of China

Eva Tuba Trinity University, San Antonio, TX, USA

Milan Tuba Singidunum University, Belgrade, Serbia

Ramazan Ünlü Industrial Engineering Department, Abdullah Gul University, Kayseri, Turkey

Xin-She Yang School of Science and Technology, Middlesex University, London, UK;
Design Engineering and Mathematics, Middlesex University London, London, UK;
Department of Design Engineering and Maths, Middlesex University, London, BT, UK

Chapter 1

Transmit Beamforming Designs in Wireless Communications Using the Firefly Algorithm



Tuan Anh Le and Xin-She Yang

1 Introduction

In wireless communications, antennas are used to transmit and receive radio frequency (RF) signals conveying information between a transmitter and its mobile users. The antennas of individual elements can be classified as isotropic, omnidirectional, and directional depending on their radiation characteristic [1]. An isotropic antenna equally emits its electromagnetic power to all directions where an omnidirectional has an isotropic pattern in a given plane, e.g., horizontal/azimuth plane. On the other hand, a directional antenna radiates higher gains in certain directions or angular regions. Omnidirectional antennas are normally used if there is no specific information of user directions. Consequently, using omnidirectional antennas are considered as non-efficient transmission approaches whereas directional antennas are more energy efficient than their counterparts [1–3].

Phased array antennas are formed by an array of single elements. Having multiple antennas offers spatial dimensions in which the signals transmitted from or received at all antennas are processed or combined to form desired radiation patterns [1]. The ability of controlling the phases and amplitudes of individual array elements allows an operator to generate power-efficient RF beams towards desired mobile receivers while imposing minimum or non interference on undesired mobile users. As a result, array antennas enhance the performance of wireless communication systems. The process of controlling or combining RF signals, i.e., phases and amplitudes, of the antenna elements are known as a beamforming design which is either performed by hardware, e.g., mechanically moving the array, changing the combinations of various coaxial cable lengths, switching between different combinations of beams [2],

T. A. Le (✉) · X.-S. Yang
School of Science and Technology, Middlesex University, London NW4 4BT, UK
e-mail: t.le@mdx.ac.uk

X.-S. Yang
e-mail: x.yang@mdx.ac.uk

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025
X.-S. Yang (ed.), *Engineering Applications of AI and Swarm Intelligence*, Springer Tracts
in Nature-Inspired Computing, https://doi.org/10.1007/978-981-97-5979-8_1

or carried out digitally, i.e., via digital signal processing algorithms or software [1–4]. This chapter focuses on digital beamforming designs.

Transmit beamforming designs are usually formulated as optimization problems with design variables being beamforming vectors collecting users' phases and amplitudes at all array elements. The data or inputs for transmit beamforming designs are channel state information (CSI) of the transmitter and the mobile receivers, the requirements posed by operators and mobile users. CSI can be provided in either instantaneous (see, e.g., [5–7]), or statistical forms (see, e.g., [8–10]). Requirements from operators and users are captured by objective functions and constraints in transmit beamforming designs which are metrics used in wireless communications, such as signal-to-interference-plus-noise-ratio (SINR) [9, 11–15], transmit power, system sum throughput/rate, secrecy rate [16, 17], interference level [9, 12], energy efficiency [18], data transmission security, power transfer reliability, and data transmission reliability [13–15].

There are two fundamental transmit beamforming problems. The first fundamental problem minimizes the overall transmit power, subject to SINR constraints [6, 7, 10, 12, 19–24]. The second fundamental problem either maximizes the lowest SINR level [25, 26] or maximizes a given utility function of the SINRs [27] subject to a total transmit power constraint. The other version of the second transmit beamforming problem aims to maximize the sum throughput, subject to a total transmit power constraint [28–30]. It has been shown in [27, 31] that these two fundamental transmit beamforming problems are equivalent.

Different constraints can also be included into these two fundamental transmit beamforming problems to present various scenarios/applications in wireless communications. For example, a constraint on the power transfer level was introduced for simultaneous-wireless-information-and-power-transfer systems [17, 32, 33] where radio frequency signal is utilized to convey both information and energy. An interference threshold constraint was included for cognitive radio applications to ensure that the operations of any opportunistic-spectrum-utilization system do not affect the users of primary system owning the radio frequency resource [9, 12, 34–36]. Constraints in transmit beamforming can be either deterministic, e.g., [9, 12, 34, 36–41], to capture perfect CSI scenarios, or probabilistic, e.g., [15, 42–50], to formulate practical problems dealing with errors in the estimations of CSI.

Most of beamforming problems are non-convex, due to the characteristics of their objectives or constraints. For example, SINR, secrecy rate, and probabilistic constraints are non-convex functions of beamforming vectors. Therefore, transmit beamforming problems involving such constraints are known to be NP-hard, i.e., the optimal solution cannot be sought in polynomial time. Several techniques have been introduced to solve non-convex transmit beamforming problems. Exploiting uplink-downlink duality, iterative algorithms were developed for transmit power minimization problems in [6, 19, 36, 51] or for the sum rate maximization problem in [30]. The authors of [7] introduced an elegant framework to transform these two fundamental transmit beamforming problems into second-order cone programming forms, which are convex and hence can be effectively solved by a standard interior point method (IPM). Other method is to convert transmit beamforming problems

realizing in quadratically constrained quadratic programs (QCQPs), which are mostly non-convex [8, 9, 52, 53], into convex semidefinite programmings (SDPs). This method is known as semidefinite relaxation technique as it relaxes rank-one constraint during the transformation.

Non-convexity of transmit beamforming problems are also attributed to the fact that they may have more than one types of beamforming vectors as optimization or design variables. For instance, planar arrays of programmable and passive reflecting elements, i.e., elements whose just reflect impinging radio frequency signals on their surfaces in controllable ways, have been proposed for 6G wireless networks as an effective approach to enhance the wireless propagation environments, hence improving the wireless networks' performances. Such programmable arrays are known as intelligent reflective surfaces (IRS) or reconfigurable intelligent surfaces (RIS), see, e.g., [54–68]. Particularly, an RIS is placed between a conventional transmitter and its mobile receivers to assist the communication. The reflecting signals introduced by all the elements of an RIS are designed by controlling their phase shifts to achieve a specific target, e.g., to focus the resulting reflected radio frequency signals on certain directions. The designed phase shifts of an RIS are collected in a so-called passive beamforming vector whereas the phases and amplitudes of the antenna elements of the conventional transmitter, i.e., also known as a base station or an access point, form an active beamforming vector. Two types of beamforming vectors in an RIS-aided system are independent however they need to be jointly designed/optimized.

To tackle such problems in RIS-aided systems, a commonly adopted approach is to alternatively solve two sub problems where, in each sub problem, one optimization variable is fixed while solving for the other. This approach is also known as alternating optimization (AO) approach [60]. Another method approximates the original non-convex problem by utilizing first-order Taylor expansion, which is referred to as a successive convex approximation (SCA) approach [69]. Sub-optimization problems in an AO approach frequently appear in non-convex QCQP forms that require further approximation techniques. The effort of decoupling two independent variables in the original non-convex problem by tackling two non-convex sub-optimization problems results in the fact that the global optimal solutions may not be attainable for the active and passive beamforming vectors. On the other hand, the SCA approach usually obtains a lower or upper bounded solution [52].

Barrier methods, such as IPMs, are based on gradient algorithms. They have a good exploitation ability, i.e., the ability of utilizing information from the original problem of interest to find better solutions than the current ones [70]. Therefore, they are effective methods for solving convex optimization problems [71]. As mentioned earlier in this chapter, the majority of transmit beamforming designs are non-convex optimization problems. Hence, they require methods with better exploration ability, i.e., the capability to efficiently explore the domain space to generate sufficiently diverse and far solutions from the existing ones [70] so that they can escape from being trapped in local optimums.

The firefly algorithm (FA) is a nature-inspired algorithm, which is good at both exploitation and exploration. Therefore, FA appears as a proper choice for tackling non-convex problems in transmit beamforming designs [52]. FA was developed by

Xin-She Yang in 2008, based on the behaviour and flashing character of tropical fireflies [70, 72]. FA is straightforward to implement and yet flexible to deal with multimodal optimization problems. Although FA has been gaining widely adoptions in many applications, see, e.g., [73, 74] and references therein, there were only three studies applying FA to transmit beamforming designs. The first work proposed FA to solve the throughput maximization problem [75], whereas the second work applied FA to tackle the power minimization problem [76]. Recently, a generalized FA was introduced by Le and Yang in [52] as a general optimization framework in which the objective function and constraints are multivariate functions of independent variables. The proposed general optimization framework and its associated FA can capture the majority problems in transmit beamforming in wireless communications.

In this chapter, the optimization framework and its associated generalized FA in [52] are summarized in Sect. 2. Conventional transmit beamforming problem with a soft-shaping interference constraint is stated in Sect. 3. The optimization problem formulations outlined in Sect. 3 have only one type of optimization variable, which is the set of beamforming vectors for mobile users. SDR and FA approaches are discussed to solve the problem with a brief complexity analysis. The RIS-aided transmit beamforming problem, which involves passive and active beamforming vectors as two independent variables, is given in Sect. 4, followed by their solutions in SDR and FA with their corresponding complexity analysis. Evaluations for the two transmit beamforming problems with SDR and FA approaches are presented in Sect. 5. Finally, concluding remarks are given in Sect. 6.

Notations: Lower and upper case letters y and Y are used to present a scalar, whereas a bold lower case letter, such as $\mathbf{y}$, denotes a column vector. A bold upper case letter, such as $\mathbf{Y}$, represents a matrix; $\|\cdot\|$ denotes the Euclidean norm; $(\cdot)^T$ presents the transpose operator; $(\cdot)^H$ indicates the complex conjugate transpose operator. The operator $\text{Tr}(\cdot)$ is the trace operator, whereas $\mathbf{Y} \succeq \mathbf{0}$ is utilized to denote that $\mathbf{Y}$ is a positive semidefinite matrix; $\mathbf{I}_N$ is an $N \times N$ identity matrix; $\mathcal{O}$ indicates the big O notation; $\mathbb{C}^{M \times 1}$ presents the set of all $M \times 1$ vectors with complex elements; $\mathbb{H}^{M \times M}$ is the set of all $M \times M$ Hermitian matrices. In addition, $y \sim \mathcal{CN}(0, \sigma^2)$ means y is a circularly symmetric complex Gaussian random variable with a zero mean and a variance of σ^2 ; $\text{diag}(\mathbf{y})$ denotes a diagonal matrix with diagonal elements being the entries of vector $\mathbf{y}$. Finally, $\text{diag}(\mathbf{Y})$ represents a vector whose entries are the diagonal elements of matrix $\mathbf{Y}$.

2 Main Steps of the Generalized Firefly Algorithm

The FA was introduced with the following three idealized assumptions [70, 72]:

- Fireflies are unisex and attract others among their flock.
- The flashing brightness of a firefly proportionally determines its attractiveness to its mates. Furthermore, the attractiveness and brightness increase when the distance between two fireflies decreases. The less brighter firefly will always fly towards

the brighter one. A firefly will randomly fly when it does not see any brighter ones in the neighbourhood.

- The landscape of an objective function under consideration can be encoded with the brightness of fireflies.

The works in [70, 72] were recently generalized in [52, 77] in which the objective and constraints of the optimization framework are multivariate functions of independent variables. In this section, the optimization framework proposed in [52] is presented. Consider the following optimization problem:

$$\begin{aligned} & \underset{\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z}}{\text{minimize}} && f(\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z}), \\ & \text{subject to} && g_l(\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z}) \leq 0, \quad l \in \{1, 2, \dots, L\}, \\ & && h_k(\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z}) = 0, \quad k \in \{1, 2, \dots, K\}, \end{aligned} \quad (1)$$

where $\mathbf{A} \in \mathbb{C}^{M_a \times N_a}$, $\mathbf{B} \in \mathbb{C}^{M_b \times N_b}$, $\dots$, $\mathbf{Z} \in \mathbb{C}^{M_z \times N_z}$ are decision variables, i.e., optimization variables, with their dimensions of $M_a \geq 1$, $N_a \geq 1$, $M_b \geq 1$, $N_b \geq 1$, $\dots$, $M_z \geq 1$, $N_z \geq 1$. The optimization variables can be scalars, vectors, matrices, or the combination of all depending on the values of $M_a, N_a, M_b, N_b, \dots, M_z, N_z$.

In order to solve (1), the penalty method [70, 72] is utilized to convert the original constrained optimization problem with multiple constraints into the following unconstrained problem:

$$\underset{\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z}}{\text{minimize}} \quad f(\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z}) + P(\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z}), \quad (2)$$

where $P(\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z})$ is the constraint-violation-penalty term, a.k.a, penalty term, given as:

$$\begin{aligned} P(\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z}) = & \sum_{l=1}^L \lambda_l \max\{0, g_l(\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z})\}^2 \\ & + \sum_{k=1}^K \rho_k \{h_k(\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z})\}^2. \end{aligned} \quad (3)$$

In (3), $\lambda_l > 0$, $\forall l$, and $\rho_k > 0$, $\forall k$, capture the importance of the constraints, a.k.a., penalty constants or coefficients.

Let $\{\mathbf{A}_i, \mathbf{B}_i, \dots, \mathbf{Z}_i\}$ present the i -th firefly in the flock of N fireflies, i.e., $i \in \{1, 2, \dots, N\}$. As the considered optimization framework is a minimization, according to the third assumption of the FA, the brightness of the i -th firefly is defined as follows:¹

$$I_i(\mathbf{A}_i, \mathbf{B}_i, \dots, \mathbf{Z}_i) = \frac{1}{f(\mathbf{A}_i, \mathbf{B}_i, \dots, \mathbf{Z}_i) + P(\mathbf{A}_i, \mathbf{B}_i, \dots, \mathbf{Z}_i)}. \quad (4)$$

¹ It is worth pointing out that (2) can be written as

$$\underset{\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z}}{\text{minimize}} \quad -f(\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z}) + P(\mathbf{A}_i, \mathbf{B}_i, \dots, \mathbf{Z}_i)$$

when (1) is a maximization problem.

According to the second assumption of the FA, the brightest firefly is the most attractive one and all fireflies will fly toward the brightest one. In other words, the most attractive firefly in the flock represent the optimal solution of the optimization problem. For any two fireflies $i, j \in \{1, 2, \dots, N\}$, if their light intensities satisfy $I_j(\mathbf{A}_j, \mathbf{B}_j, \dots, \mathbf{Z}_j) > I_i(\mathbf{A}_i, \mathbf{B}_i, \dots, \mathbf{Z}_i)$, then firefly i will fly towards firefly j at the next generation [i.e., the $(n + 1)$ -th generation] as:

$$\mathbf{A}_i^{(n+1)} = \mathbf{A}_i^{(n)} + \beta_{a,0} e^{-\gamma_a (r_{a,ij}^{(n)})^2} (\mathbf{A}_j^{(n)} - \mathbf{A}_i^{(n)}) + \alpha_a^{(n)} \mathbf{\Lambda}_{a,i}^{(n)}, \quad (5)$$

$$\mathbf{B}_i^{(n+1)} = \mathbf{B}_i^{(n)} + \beta_{b,0} e^{-\gamma_b (r_{b,ij}^{(n)})^2} (\mathbf{B}_j^{(n)} - \mathbf{B}_i^{(n)}) + \alpha_b^{(n)} \mathbf{\Lambda}_{b,i}^{(n)}, \quad (6)$$

$$\begin{aligned} & \vdots \\ \mathbf{Z}_i^{(n+1)} &= \mathbf{Z}_i^{(n)} + \beta_{z,0} e^{-\gamma_z (r_{z,ij}^{(n)})^2} (\mathbf{Z}_j^{(n)} - \mathbf{Z}_i^{(n)}) + \alpha_z^{(n)} \mathbf{\Lambda}_{z,i}^{(n)}, \end{aligned} \quad (7)$$

where $r_{a,ij}^{(n)} = \|\mathbf{A}_j^{(n)} - \mathbf{A}_i^{(n)}\|$, $r_{b,ij}^{(n)} = \|\mathbf{B}_j^{(n)} - \mathbf{B}_i^{(n)}\|$, $\dots$, $r_{z,ij}^{(n)} = \|\mathbf{Z}_j^{(n)} - \mathbf{Z}_i^{(n)}\|$ are the Cartesian distances. In other application, non-Euclidean distances may be more appropriate². Here, $\beta_{a,0}, \beta_{b,0}, \dots, \beta_{z,0}$ are, respectively, the attractiveness at $r_{a,ij}^{(n)} = 0, r_{b,ij}^{(n)} = 0, \dots, r_{z,ij}^{(n)} = 0$, whereas $\gamma_a, \gamma_b, \dots, \gamma_z$ are the variations of attractiveness.

The second terms on the right-hand sides of (5), (6), and (7) represent the attractions. The third terms in (5), (6), and (7) capture randomization. The randomization factors $\alpha_a^{(n)}, \alpha_b^{(n)}, \dots, \alpha_z^{(n)}$ and randomization matrices

$$\mathbf{\Lambda}_{a,i}^{(n)} \in \mathbb{C}^{M_a \times N_a}, \mathbf{\Lambda}_{b,i}^{(n)} \in \mathbb{C}^{M_b \times N_b}, \dots, \mathbf{\Lambda}_{z,i}^{(n)} \in \mathbb{C}^{M_z \times N_z}$$

are, respectively, scalars and matrices of random numbers which are drawn from an uniform or a Gaussian distribution.

The pseudocode presented in Algorithm 1 summarizes the generalized FA as an optimization framework for finding the optimal solution (1). Here, in Algorithm 1, T represents the maximum generations (i.e., iterations, of the algorithm).

3 Transmit Beamforming

3.1 Optimization Problem Formulation

Let us now consider a system of wireless communications where a base station (BS), equipped with M_t antennas, communicates with U local mobile receivers, each having a single antenna, while imposing a tolerable interference level of $I_{\text{to},k}$ on each of K neighbouring users, where each user is also equipped with a single antenna.

² These distances can be any metric which effectively capture the quantities of interest of the under-investigated optimization problem.

Algorithm 1 Pseudocode of the Generalized Firefly Algorithm to solve (1) [52]

```

1: Input: The parameter settings for FA:  $N, T, \lambda_t, \rho_k, \beta_{a,0}, \beta_{b,0}, \dots, \beta_{z,0}, \gamma_a, \gamma_b, \dots, \gamma_z$ ; Optimization data:  $f(\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z})$ ,  $g_l(\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z})$ ,  $h_k(\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z})$ ;
2: Randomly initialize the locations of  $N$  fireflies  $\{\{\mathbf{A}_1, \mathbf{B}_1, \dots, \mathbf{Z}_1\}, \{\mathbf{A}_2, \mathbf{B}_2, \dots, \mathbf{Z}_2\}, \dots, \{\mathbf{A}_N, \mathbf{B}_N, \dots, \mathbf{Z}_N\}\}$ ;
3: Calculate the brightness of  $N$  fireflies as (4);
4: Sort the firefly population in the descending order of the brightness  $I_i(\mathbf{A}_i, \mathbf{B}_i, \dots, \mathbf{Z}_i)$ ;
5: Update the current brightest firefly, i.e., the current best solution vector:  $I^* := I_1(\mathbf{A}^*, \mathbf{B}^*, \dots, \mathbf{Z}^*)$ ;  $\{\mathbf{A}^*, \mathbf{B}^*, \dots, \mathbf{Z}^*\} := \{\mathbf{A}_1, \mathbf{B}_1, \dots, \mathbf{Z}_1\}$ ;
6: for  $n = 1 : T$  do
7:   for  $i = 1 : N$  do
8:     for  $j = 1 : N$  do
9:       if  $I_i(\mathbf{A}_i, \mathbf{B}_i, \dots, \mathbf{Z}_i) > I^*$  then
10:         $I^* := I_i(\mathbf{A}_i, \mathbf{B}_i, \dots, \mathbf{Z}_i)$ ;  $\{\mathbf{A}^*, \mathbf{B}^*, \dots, \mathbf{Z}^*\} := \{\mathbf{A}_i, \mathbf{B}_i, \dots, \mathbf{Z}_i\}$ ;
11:      end if
12:      if  $I_j(\{\mathbf{A}_j, \mathbf{B}_j, \dots, \mathbf{Z}_j\}) > I^*$  then
13:         $I^* := I_j(\mathbf{A}_j, \mathbf{B}_j, \dots, \mathbf{Z}_j)$ ;  $\{\mathbf{A}^*, \mathbf{B}^*, \dots, \mathbf{Z}^*\} := \{\mathbf{A}_j, \mathbf{B}_j, \dots, \mathbf{Z}_j\}$ ;
14:      end if
15:      if  $I_j(\mathbf{A}_j, \mathbf{B}_j, \dots, \mathbf{Z}_j) > I_i(\mathbf{A}_i, \mathbf{B}_i, \dots, \mathbf{Z}_i)$  then
16:        Move firefly  $i$  towards firefly  $j$  as (5)-(7);
17:      end if
18:      Vary the attractiveness according to distances via
          
$$e^{-\gamma_a \left(r_{a,ij}^{(n)}\right)^2}, e^{-\gamma_b \left(r_{b,ij}^{(n)}\right)^2}, \dots, e^{-\gamma_z \left(r_{z,ij}^{(n)}\right)^2};$$

19:      Calculate new solutions and update the brightness of  $N$  fireflies as (4);
20:    end for
21:  end for
22:  Sort the firefly population in the descending order of the brightness  $I_i(\mathbf{A}_i, \mathbf{B}_i, \dots, \mathbf{Z}_i)$ ;
23:  Update the current brightest firefly, i.e., the current best solution:  $I^* := I_1(\mathbf{A}^*, \mathbf{B}^*, \dots, \mathbf{Z}^*)$ ;  $\{\mathbf{A}^*, \mathbf{B}^*, \dots, \mathbf{Z}^*\} := \{\mathbf{A}_1, \mathbf{B}_1, \dots, \mathbf{Z}_1\}$ ;
24: end for
25: return the optimal solution  $\{\mathbf{A}^*, \mathbf{B}^*, \dots, \mathbf{Z}^*\}$ .

```

This kind of system model is also known as cognitive transmit beamforming, see, e.g., [34, 36, 52] and references therein. The signal arrived at the t -th local user, $t \in \{1, \dots, U\}$, is

$$y_t = \mathbf{h}_t^H \mathbf{w}_t x_t + \sum_{j=1, j \neq t}^U \mathbf{h}_t^H \mathbf{w}_j x_j + n_t, \quad (8)$$

where

- $\mathbf{h}_t^H \in \mathbb{C}^{1 \times M_t}$ represents the channel coefficient, i.e., CSI, of the wireless link between the t -th local user and the BS. Specifically, $\mathbf{h}_t = [h_{t,1}, h_{t,2}, \dots, h_{t,M_t}]^T$. The entry of $h_{t,v}$ of vector $\mathbf{h}_t$ captures the channel coefficient between antenna v -th of the BS, i.e., $v \in \{1, 2, \dots, M_t\}$, and t -th local user;
- x_t , i.e., $\mathbb{E}[|x_t|^2] = 1$, is the data symbol for the local user i ;

- $\mathbf{w}_t = [w_{t,1}, w_{t,2}, \dots, w_{t,M_t}]^T \in \mathbb{C}^{M_t \times 1}$ is the beamforming vector for the t -th local user in which $w_{t,v}$, i.e., $v \in \{1, 2, \dots, M_t\}$, represents the complex weighting factor for the data symbol of the t -th local user applied on the v -th antenna of the BS. It is worth pointing out that $\mathbf{w}_t^H \mathbf{w}_t$ is the transmit power that the BS spends on the transmission for the t -th local user;
- $s_t \sim \mathcal{CN}(0, 1)$ presents the data symbol intended for the t -th local user;
- and $n_t \sim \mathcal{CN}(0, \sigma_t^2)$ is the circularly symmetric complex Gaussian (ZMCSCG) noise with a zero mean and a variance of σ_t^2 , at the t -th local user.

It can be seen from Eq. (8) that the radio frequency signal delivered to the t -th local user is the product of the channel coefficient (i.e., $\mathbf{h}_t^H$), with the linear combination of the data symbols for all U local users and their associated beamforming vectors. That is

$$\sum_{j=1}^U \mathbf{w}_j x_j.$$

The first term on the right-hand side of (8) captures the intended signal for the t -th local user, whereas the second term represents the interference caused by the transmission to the other $U - 1$ local users, i.e., the intra-system interference.

From (8) with the notes that $\mathbb{E}[|x_t|^2] = 1$ and $s_t \sim \mathcal{CN}(0, 1) \forall t \in \{1, 2, \dots, U\}$, the signal-to-interference-plus-noise-ratio (SINR) at the t -th local user can be written as

$$\text{SINR}_t = \frac{|\mathbf{h}_t^H \mathbf{w}_t|^2}{\sum_{j=1, j \neq t}^U |\mathbf{h}_t^H \mathbf{w}_j|^2 + \sigma_t^2} \quad (9)$$

$$= \frac{\mathbf{w}_t^H \mathbf{H}_t \mathbf{w}_t}{\sum_{j=1, j \neq t}^U \mathbf{w}_j^H \mathbf{H}_t \mathbf{w}_j + \sigma_t^2}, \quad (10)$$

where $\mathbf{H}_t = \mathbb{E}(\mathbf{h}_t \mathbf{h}_t^H)$ for the case of statistical CSI and $\mathbf{H}_t = \mathbf{h}_t \mathbf{h}_t^H$ for the case of instantaneous CSI. The level of SINR determines the quality of service (QoS), i.e., the throughput or data rate, at the local user. The higher the level SINR is, the better the QoS.

Let $\mathbf{q}_k^H \in \mathbb{C}^{1 \times M_t}$ be the channel coefficient of the wireless link between the k -th neighbouring user (where $k \in \{1, \dots, K\}$) and the serving BS, where $\mathbf{q}_k = [q_{k,1}, q_{k,2}, \dots, q_{k,M_t}]^T$. The interference level at the k -th neighbouring user can be expressed as

$$y_k = \sum_{j=1}^U \mathbf{q}_k^H \mathbf{w}_j x_j + n_k, \quad (11)$$

where n_k is the ZMCSCG noise at the user. The total interference power imposed by the BS on the k -th neighbouring user is

$$y_k = \sum_{j=1}^U |\mathbf{q}_k^H \mathbf{w}_j|^2. \quad (12)$$

Denoting $\mathbf{Q}_k = \mathbb{E}(\mathbf{q}_k \mathbf{q}_k^H)$ for the case of statistical CSI and $\mathbf{Q}_k = \mathbf{q}_k \mathbf{q}_k^H$ for the case of instantaneous CSI, the total interference power inflicted on the k -th neighbouring user by the serving BS can also be written as

$$\sum_{j=1}^U \mathbf{w}_j^H \mathbf{Q}_k \mathbf{w}_j. \quad (13)$$

The aim is to find the beamforming vectors for the local users, which minimize the serving BS's total transmit power while ensuring the required SINR for all local users. At the same time, it is required to guarantee the total interference level inflicted on each neighbouring user below its tolerable threshold. Hence, the optimization problem can be rewritten as

$$\begin{aligned} \min_{\mathbf{w}_t} \quad & \sum_{t=1}^U \mathbf{w}_t^H \mathbf{w}_t \\ \text{s. t.} \quad & \frac{\mathbf{w}_t^H \mathbf{H}_t \mathbf{w}_t}{\sum_{j=1, j \neq t}^U \mathbf{w}_j^H \mathbf{H}_t \mathbf{w}_j + \sigma_t^2} \geq \eta_t, \quad \forall t \in \{1, \dots, U\}, \\ & \sum_{j=1}^U \mathbf{w}_j^H \mathbf{Q}_k \mathbf{w}_j \leq I_{\text{to},k}, \quad \forall k \in \{1, \dots, K\}, \end{aligned} \quad (14)$$

where η_t is the required SINR of the t -th local user.

3.2 Semidefinite Programming Relaxation Approach

Utilizing the identity $\mathbf{x}^H \mathbf{X} \mathbf{x} = \text{Tr}(\mathbf{X} \mathbf{x} \mathbf{x}^H)$, (14) can be rewritten as

$$\begin{aligned} \min_{\mathbf{w}_t} \quad & \sum_{t=1}^U \text{Tr}(\mathbf{w}_t \mathbf{w}_t^H) \\ \text{s. t.} \quad & \frac{\text{Tr}(\mathbf{H}_t \mathbf{w}_t \mathbf{w}_t^H)}{\sum_{j=1, j \neq t}^U \text{Tr}(\mathbf{H}_t \mathbf{w}_j \mathbf{w}_j^H) + \sigma_t^2} \geq \eta_t, \quad \forall t, \\ & I_{\text{to},k} - \sum_{j=1}^U \text{Tr}(\mathbf{Q}_k \mathbf{w}_j \mathbf{w}_j^H) \geq 0, \end{aligned} \quad (15)$$

where $t \in \{1, \dots, U\}$, and $k \in \{1, \dots, K\}$.

In order to solve (15), a new set of optimization variables $\mathbf{F}_t = \mathbf{w}_t \mathbf{w}_t^H$ are introduced where $\mathbf{F}_t \succeq \mathbf{0}$, $\mathbf{F}_t \in \mathbb{H}^{M_t \times M_t}$, and $\mathbf{F}_t$ is a rank-one matrix. Briefly speaking, a rank-one matrix is a matrix containing only one linearly independent row/column. With some mathematical manipulations on the SINR constraint, (15) can be rewritten as

$$\begin{aligned}
& \min_{\mathbf{F}_t \in \mathbb{H}^{M \times M}} \sum_{t=1}^U \text{Tr}(\mathbf{F}_t) \\
& \text{s. t.} \quad \left(1 + \frac{1}{\eta_t}\right) \text{Tr}(\mathbf{H}_t \mathbf{F}_t) - \sum_{j=1}^U \text{Tr}(\mathbf{H}_t \mathbf{F}_j) - \sigma_t^2 \geq 0, \quad \forall t, \\
& \quad I_{\text{to},k} - \sum_{j=1}^U \text{Tr}(\mathbf{Q}_k \mathbf{F}_j) \geq 0, \quad \forall k, \\
& \quad \mathbf{F}_t \succeq \mathbf{0}, \quad \forall t,
\end{aligned} \tag{16}$$

where $t \in \{1, \dots, U\}$, and $k \in \{1, \dots, K\}$.

Problem (16) is in the semidefinite programming (SDP) form. Therefore, its optimal solution can be attained using a general purpose interior point method. For example, there is a powerful Matlab based optimization package CVX for constructing and solving convex optimization problems [78]. For the formulation (16), the rank-one condition on $\mathbf{F}_t$, $\forall t$, has been relaxed. Fortunately, (16) is a special case of a rank-one optimization framework introduced in [24], which will yield an optimal rank-one solution if the problem is feasible.

3.3 Firefly Algorithm Approach

Here, the generalized FA presented in Algorithm 1 is used to find the solution for (14). After rearranging the constraints, problem (14) can be cast as

$$\begin{aligned}
& \min_{\mathbf{W}} f(\mathbf{W}) \\
& \text{s. t.} \quad \phi_t(\mathbf{W}) \leq 0, \quad \forall t \in \{1, \dots, U\}, \\
& \quad \varphi_k(\mathbf{W}) \leq 0, \quad \forall k \in \{1, \dots, K\},
\end{aligned} \tag{17}$$

where

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_U] \in \mathbb{C}^{M_t \times U}, \quad f(\mathbf{W}) = \sum_{t=1}^U \mathbf{w}_t^H \mathbf{w}_t,$$

$$\phi_t(\mathbf{W}) = \eta_t \sum_{j=1, j \neq i}^U \mathbf{w}_j^H \mathbf{H}_t \mathbf{w}_j + \eta_t \sigma_t^2 - \mathbf{w}_t^H \mathbf{H}_t \mathbf{w}_t, \quad \varphi_k(\mathbf{W}) = \sum_{j=1}^U \mathbf{w}_j^H \mathbf{Q}_k \mathbf{w}_j - I_{\text{to},k}.$$

To solve problem (17) with constraints, its constraints are imposed using the penalty method and the problem is thus converted to an unconstrained optimization problem as

$$\min_{\mathbf{W}} f(\mathbf{W}) + P(\mathbf{W}), \tag{18}$$

where $P(\mathbf{W})$ represents the penalty terms of all constraints

$$P(\mathbf{W}) = \sum_{t=1}^U \lambda_t \max\{0, \phi_t(\mathbf{W})\}^2 + \sum_{k=1}^K \rho_k \max\{0, \varphi_k(\mathbf{W})\}^2, \tag{19}$$

with $\lambda_t > 0$ and $\rho_k > 0$ are penalty constants.

To proceed, let $\mathbf{W}_i = [\mathbf{w}_1^i, \mathbf{w}_2^i, \dots, \mathbf{w}_U^i] \in \mathbb{C}^{M_t \times U}$ be the firefly i . A population of N fireflies $\mathbf{W}_i, i \in \{1, 2, \dots, N\}$, are initialized and the brightness of the firefly $\mathbf{W}_i$ is given by

$$I_i(\mathbf{W}_i) = \frac{1}{f(\mathbf{W}_i) + P(\mathbf{W}_i)}. \quad (20)$$

For any two fireflies i and j in the swarm, if the firefly j is brighter than the firefly i , i.e., $I_j(\mathbf{W}_j) > I_i(\mathbf{W}_i)$, then the firefly i will fly towards the firefly j and its location is updated by

$$\mathbf{W}_i^{(n+1)} = \mathbf{W}_i^{(n)} + \beta_0 e^{-\gamma (r_{ij}^{(n)})^2} (\mathbf{W}_j^{(n)} - \mathbf{W}_i^{(n)}) + \alpha^{(n)} \mathbf{V}, \quad (21)$$

where

$$r_{ij}^{(n)} = \|\mathbf{W}_j^{(n)} - \mathbf{W}_i^{(n)}\|$$

denotes the Cartesian distance, and β_0 represents the attractiveness at the distance $r_{ij}^{(n)} = 0$. The coefficient γ controls the variation of the attractiveness. The second term on the right-hand side of equation (21) represents the attraction between firefly i and firefly j . The last term on the right-hand side of equation (21) represents a randomization perturbation comprising of a random factor $\alpha^{(n)}$ and a matrix of random numbers $\mathbf{V} \in \mathbb{C}^{M_t \times U}$. The randomization factor $\alpha^{(n)}$ and the entries of matrix $\mathbf{V}$ are drawn from either an uniform or more often a Gaussian distribution.

It is clear that problem (17) is a special realization of the framework (1) in which its objective and constraints are functions of a single matrix of variables, i.e., $\mathbf{W}$. Therefore, the firefly algorithm to solve (17) includes similar steps as in Algorithm 1 except four steps, i.e., 3, 16, 18, and 19. The differences are provided in Algorithm 2.

Algorithm 2 Procedure of the generalized FA to solve (14)

Input: The parameter settings for FA: $N, T, \lambda_t, \rho_k, \beta_0, \gamma$; Data: $\mathbf{H}_t, \mathbf{Q}_k, \sigma_t^2, \eta_t, I_{t0,k}$;

Step 3: Calculate the brightness of N fireflies as (20);

Step 16: Update the location of firefly i towards firefly j as (21);

Step 18: Vary the attractiveness strength according to distance via $e^{-\gamma (r_{ij}^{(n)})^2}$;

Step 19: Calculate new solutions and update the brightness of N fireflies $I_i(\mathbf{W}_i)$ as (20);

return Optimal solution $\mathbf{W}^*$.

3.4 Complexity Analysis

This section briefly outlines the computational complexities of the generalized FA to solve (16), and the procedure is based on those similar to the analysis of IPM and the standard FA.

Definition 1 Given $\varepsilon > 0$, $\mathbf{F}_t^\varepsilon$, $\forall t \in \{1, 2, \dots, U\}$ is called an ε -solution of problem (16) (i.e., a solution with the acceptable accuracy of ε) if the following condition is satisfied:

$$\sum_{t=1}^U \text{Tr}(\mathbf{F}_t^\varepsilon) \leq \min_{\mathbf{F}_t \in \mathbb{H}^{M \times M}} \sum_{t=1}^U \text{Tr}(\mathbf{F}_t) + \varepsilon. \quad (22)$$

Here, M_t^2 is the number of decision variables of problem (16). The computational complexity of (16) can be derived as follows:

Lemma 1 Obtaining an ε -solution for problem (16) requires the computational complexity on the order of [52]:

$$\ln(\varepsilon^{-1}) \sqrt{U(M_t + 1) + K} \left[(M_t^2 + 1)(U + K) + U M_t^2 (M_t^2 + M_t) + M_t^4 \right] M_t^2. \quad (23)$$

Proof It is clear that problem (16) contains $(U + K)$ constraints in the form of linear matrix inequalities (LMI) of size 1 and U constraints in the LMI form of size M_t . Following the same procedure as described in [48, Section V-A], the following conclusions can be drawn:

- The iteration complexity is of the order of

$$\ln(\varepsilon^{-1}) \sqrt{U(M_t + 1) + K}. \quad (24)$$

- The complexity of each iteration is on the order of:

$$\left[(M_t^2 + 1)(U + K) + U M_t^2 (M_t^2 + M_t) + M_t^4 \right] M_t^2. \quad (25)$$

Combining (24) and (25) will lead to (23). $\square$

Lemma 2 The FA described in Algorithm 2 requires the computational complexity on the order of [52]:

$$\begin{aligned} T N^2 \left[M_t^2 + N U M_t (1 + U M_t + K M_t) \right] + T N \log N + N M_t U \\ + N U M_t (1 + U M_t + K M_t) + N \log N. \end{aligned} \quad (26)$$

Proof The major computational complexity to perform Algorithm 2 are contributed by steps 2, 3, 4, 16, 19, and 22.

In step 2, generating an $M_t \times U$ matrix requires a complexity of $M_t U$. Hence, to generate N matrix, the complexity is in the order of $N M_t U$.

In steps 3 and 19, computing $\phi_t(\mathbf{W})$ or $\varphi_k(\mathbf{W})$ needs the complexity of UM_t^2 . On the other hand, UM_t is the order of complexity to compute $\sum_{t=1}^U \mathbf{w}_t^H \mathbf{w}_t$. Therefore, evaluating the brightness for the flock of N firefly demands the complexity in the order of $N(UM_t + U^2M_t^2 + KUM_t^2) = NUM_t(1 + UM_t + KM_t)$.

In step 16, M_t^2 is the order of complexity to move a firefly.

In steps 4 and 22, ranking all N fireflies requires the complexity in the order of $N \log N$.

In the worst case scenario where step 16 is required for all inner loops of the FA algorithm, using the above observations with some mathematical manipulations, expression (26) can be obtained. $\square$

4 Reconfigurable Intelligent Surface-Aided Transmit Beamforming

4.1 Problem Formulation

Now let us consider a system of wireless communications consisting of a base station (BS) with M_t -antennas and U single-antenna mobile users. Due to high buildings and other physical obstacles, there is no direct communication link between the BS and the mobile users [44]. To tackle such kind of problem, a passive N_t -reflective-element reconfigurable intelligent surface (RIS) is deployed to aid the communication.

Let $h_{k,v}$ (where $k \in \{1, 2, \dots, N_t\}$ and $v \in \{1, 2, \dots, M_t\}$) be the complex channel coefficient between the k -th element of the RIS and the v -th antenna of the BS. Thus, $\mathbf{h}_k = [h_{k,1}, h_{k,2}, \dots, h_{k,M_t}]^T$ represents the channel between the k -th element of the RIS and M_t antennas of the BS. The coefficients of the wireless channels between the N_t -element RIS and the M_t -antenna BS can be expressed in a matrix form as

$$\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_{N_t}] \in \mathbb{C}^{M_t \times N_t}.$$

Let $\mathbf{g}_i = [g_{i,1}, \dots, g_{i,N_t}]^T \in \mathbb{C}^{N_t \times 1}$ represent the coefficients of the wireless channel between the i -th user and the RIS where $g_{i,k}$ is the complex coefficient of the wireless channel between the i -th user and the k -th element of the RIS. In addition, x_i is used to denote the data symbol sent to the i -th user with $\mathbb{E}[|x_i|^2] = 1$. Furthermore, $\mathbf{w}_i = [w_{i,1}, w_{i,2}, \dots, w_{i,M_t}]^T \in \mathbb{C}^{M_t \times 1}$ is used to represent the active beamforming vector for the i -th user in which $w_{i,v}$ (where $v \in \{1, 2, \dots, M_t\}$) represents the complex weighting factor for the data symbol of the i -th user applied on the v -th antenna of the BS. It is worth pointing out that $\mathbf{w}_i^H \mathbf{w}_i$ is the transmit power that the BS spends on the transmission for communicating with the i -th local user.

In essence, each element of the RIS introduces a suitable phase shift so that the overall reflecting radio frequency signals produced by the RIS can assist the BS in the communication with its mobile users. Let θ_k such that $|\theta_k| \leq 1$ and $\arg(\theta_k) \in [-\pi, \pi)$ (for $\forall k = 1, \dots, N_t$) represent the phase-shift coefficient introduced by the k -th

reflective element. Thus, the vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_{N_t}]^T$ stacks all coefficients to form a so-called passive beamforming vector. The signal delivered at the i -th user can be expressed as:

$$\begin{aligned} y_i &= \mathbf{g}_i^H \text{diag}(\boldsymbol{\theta})^H \mathbf{H}^H \mathbf{w}_i x_i + \mathbf{g}_i^H \text{diag}(\boldsymbol{\theta})^H \mathbf{H}^H \sum_{j=1, j \neq i}^U \mathbf{w}_j x_j + n_i, \\ &= \boldsymbol{\theta}^H \mathbf{G}_i^H \mathbf{w}_i x_i + \boldsymbol{\theta}^H \mathbf{G}_i^H \sum_{j=1, j \neq i}^U \mathbf{w}_j x_j + n_i, \end{aligned} \quad (27)$$

where the Gaussian white noise $n_i \sim \mathcal{CN}(0, \sigma^2)$ is additive at the i -th user. Here, $\mathbf{G}_i^H = \text{diag}(\mathbf{g}_i^*) \mathbf{H}^H \in \mathbb{C}^{N_t \times M_t}$ is the concatenated channel coefficients between the BS and i -th user, which is the concatenation of the channel from the BS to the RIS and the channel from the RIS to the user. The first term on the right-hand side of (27) is the intended signal for i -th user, whereas the second term is essentially the interference on the i -th user due to the concurrent transmission to the other $U - 1$ users.

Now let $\{\mathbf{w}_i\} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_U\}$ be the set of all active beamforming vectors where $\text{SINR}_i(\{\mathbf{w}_i\}, \boldsymbol{\theta})$ denotes the SINR at the i -th user. Thus, the SINR at the i -th user can be expressed as

$$\text{SINR}_i(\{\mathbf{w}_i\}, \boldsymbol{\theta}) = \frac{|\boldsymbol{\theta}^H \mathbf{G}_i^H \mathbf{w}_i|^2}{\sum_{j=1, j \neq i}^U |\boldsymbol{\theta}^H \mathbf{G}_i^H \mathbf{w}_j|^2 + \sigma_i^2}. \quad (28)$$

Consider the first fundamental transmit beamforming problem discussed in the Introduction, the optimization is cast as

$$\begin{aligned} \min_{\{\mathbf{w}_i\}, \boldsymbol{\theta}} \quad & \sum_{i=1}^U \mathbf{w}_i^H \mathbf{w}_i \\ \text{s. t.} \quad & \text{SINR}_i(\{\mathbf{w}_i\}, \boldsymbol{\theta}) \geq \eta_i, \forall i, \\ & |\theta_k| \leq 1, \forall k, \end{aligned} \quad (29)$$

where η_i is the required SINR of the i -th user. As two sets of optimization variables $\mathbf{w}_i$ and $\boldsymbol{\theta}$ are coupled in the SINR function, subject to inequality constraints, problem (29) is thus non-convex.

4.2 Alternating Approach and Formulation

A widely used approach to find the solution to problem (29) is to alternatively solve for $\mathbf{w}_i$ and $\boldsymbol{\theta}$ [44, 60, 79, 80]. Using the fact that

$$|\boldsymbol{\theta}^H \mathbf{G}_i^H \mathbf{w}_i|^2 = \boldsymbol{\theta}^H \mathbf{G}_i^H \mathbf{w}_i \mathbf{w}_i^H \mathbf{G}_i \boldsymbol{\theta} = \text{Tr}(\mathbf{G}_i^H \mathbf{w}_i \mathbf{w}_i^H \mathbf{G}_i \boldsymbol{\theta} \boldsymbol{\theta}^H) \quad (30)$$

$$= \text{Tr}(\mathbf{w}_i \mathbf{w}_i^H \mathbf{G}_i \boldsymbol{\theta} \boldsymbol{\theta}^H \mathbf{G}_i^H) = \text{Tr}(\mathbf{G}_i \boldsymbol{\theta} \boldsymbol{\theta}^H \mathbf{G}_i^H \mathbf{w}_i \mathbf{w}_i^H), \quad (31)$$

and

$$\mathbf{w}_i^H \mathbf{w}_i = \text{Tr}(\mathbf{w}_i \mathbf{w}_i^H), \quad (32)$$

and introducing two new variables $\mathbf{F}_i = \mathbf{w}_i \mathbf{w}_i^H$ and $\boldsymbol{\Theta} = \boldsymbol{\theta} \boldsymbol{\theta}^H$ such that $\text{rank}(\mathbf{F}_i) = 1$ and $\text{rank}(\boldsymbol{\Theta}) = 1$, some straightforward manipulations on the SINR constraint lead to an equivalent form for (29). That is

$$\begin{aligned} \min_{\{\mathbf{F}_i\}, \boldsymbol{\Theta}} \quad & \text{Tr}\left(\sum_{i=1}^U \mathbf{F}_i\right) \\ \text{s. t.} \quad & \text{Tr} \frac{\mathbf{G}_i \boldsymbol{\Theta} \mathbf{G}_i^H \mathbf{F}_i}{\eta_i \sigma_i^2} - \sum_{j=1, j \neq i}^U \text{Tr} \frac{\mathbf{G}_i \boldsymbol{\Theta} \mathbf{G}_j^H \mathbf{F}_j}{\sigma_i^2} - 1 \geq 0, \forall i, \\ & \mathbf{F}_i \succeq \mathbf{0}, \forall i \in \{1, \dots, U\}, \\ & \text{rank}(\mathbf{F}_i) = 1, \forall i \in \{1, \dots, U\}, \\ & \text{diag}(\text{diag}(\boldsymbol{\Theta})) \leq \mathbf{I}_{N_t}, \\ & \boldsymbol{\Theta} \succeq \mathbf{0}, \\ & \text{rank}(\boldsymbol{\Theta}) = 1. \end{aligned} \quad (33)$$

Since the newly derived SINR constraint is still coupled by two variables $\mathbf{F}_i$ and $\boldsymbol{\Theta}$, problem (33) will be solved by an alternative approach. To proceed, dropping the rank-one constraints on both $\mathbf{F}_i$ and $\boldsymbol{\Theta}$, starting with any initial values of the passive beamforming matrix $\boldsymbol{\Theta}^{(0)}$, the active beamforming matrix $\mathbf{F}_i^{(p)}$ can be found by solving the following relaxed problem at the p -th iteration:

$$\begin{aligned} \min_{\{\mathbf{F}_i\}} \quad & \text{Tr}\left(\sum_{i=1}^U \mathbf{F}_i\right) \\ \text{s. t.} \quad & \text{Tr} \frac{\mathbf{G}_i \boldsymbol{\Theta}^{(p-1)} \mathbf{G}_i^H \mathbf{F}_i}{\eta_i \sigma_i^2} - \sum_{j=1, j \neq i}^U \text{Tr} \frac{\mathbf{G}_i \boldsymbol{\Theta}^{(p-1)} \mathbf{G}_j^H \mathbf{F}_j}{\sigma_i^2} - 1 \geq 0, \forall i, \\ & \mathbf{F}_i \succeq \mathbf{0}, \forall i \in \{1, \dots, U\}. \end{aligned} \quad (34)$$

Having $\mathbf{F}_i^{(p)}$ from solving (34), the passive beamforming matrix $\boldsymbol{\Theta}^{(p)}$ can be updated by solving the following problem at the p -th iteration [44]:

$$\begin{aligned} \min_{\boldsymbol{\Theta}} \quad & \text{Tr}(\boldsymbol{\Theta}) \\ \text{s. t.} \quad & \text{Tr} \frac{\boldsymbol{\Theta} \mathbf{G}_i^H \mathbf{F}_i^{(p)} \mathbf{G}_i}{\eta_i \sigma_i^2} - \sum_{j=1, j \neq i}^U \text{Tr} \frac{\boldsymbol{\Theta} \mathbf{G}_j^H \mathbf{F}_j^{(p)} \mathbf{G}_j}{\sigma_i^2} - 1 \geq 0, \forall i, \\ & \text{diag}(\text{diag}(\boldsymbol{\Theta})) \leq \mathbf{I}_{N_t}, \\ & \boldsymbol{\Theta} \succeq \mathbf{0}. \end{aligned} \quad (35)$$

The alternating optimization (AO) method repetitively solves two problems (34) and (35) in n_0 iterations to attain the solution for the original problem (29). In essence, the AO method utilizes the semidefinite relaxation technique [8] by dropping/relaxing the rank-one constraints on $\mathbf{F}_i$ and $\boldsymbol{\Theta}$. Fortunately, (34) and (35) are two special cases

of a rank-one optimization framework introduced in [24] which will return rank-one optimal solutions if these problems are feasible. On the other hand, the AO method approximates the original non-convex optimization problem (29) by solving two sub-problems (34) and (35). Due to the nature of such approximations, solutions obtained by the AO method may not be the globally optimal solution. However, they serve as upper bounds for the solutions of the original problem (29).

In the rest of this work, the FA approach is used to concurrently solve $\mathbf{w}_i$ and $\boldsymbol{\theta}$ for the original problem (29).

4.3 Firefly Algorithm Approach

The original optimization problem (29) can be written equivalently as

$$\begin{aligned} \min_{\{\mathbf{W}, \boldsymbol{\theta}\}} \quad & f(\mathbf{W}) \\ \text{s. t.} \quad & \phi_i(\{\mathbf{W}, \boldsymbol{\theta}\}) \leq 0, \forall i, \\ & \varphi_k(\theta_k) \leq 0, \forall k, \end{aligned} \quad (36)$$

where

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_U] \in \mathbb{C}^{M_t \times U}, \quad f(\mathbf{W}) = \sum_{i=1}^U \mathbf{w}_i^H \mathbf{w}_i,$$

$$\phi_i(\mathbf{W}, \boldsymbol{\theta}) = \eta_i \frac{\sum_{j=1}^U \mathbf{w}_j^H \mathbf{G}_i \boldsymbol{\theta} \boldsymbol{\theta}^H \mathbf{G}_i^H \mathbf{w}_j}{\sigma_i^2} + \eta_i - (1 + \eta_i) \frac{\mathbf{w}_i^H \mathbf{G}_i \boldsymbol{\theta} \boldsymbol{\theta}^H \mathbf{G}_i^H \mathbf{w}_i}{\sigma_i^2}, \quad (37)$$

and $\varphi_k(\theta_k) = |\theta_k| - 1$. Utilizing the penalty approach for constraints, (36) becomes

$$\min_{\{\mathbf{W}, \boldsymbol{\theta}\}} \quad f(\mathbf{W}) + P(\mathbf{W}, \boldsymbol{\theta}), \quad (38)$$

where $P(\mathbf{W}, \boldsymbol{\theta})$ represents the penalty term given by

$$P(\mathbf{W}, \boldsymbol{\theta}) = \sum_{i=1}^U \lambda_i \max\{0, \phi_i(\{\mathbf{W}, \boldsymbol{\theta}\})\}^2 + \sum_{k=1}^{N_t} \rho_k \max\{0, \varphi_k(\theta_k)\}^2, \quad (39)$$

with $\lambda_i > 0$ and $\rho_k > 0$ are penalty constants.

Let $\{\mathbf{W}_t, \boldsymbol{\theta}_t\} = \{[\mathbf{w}_1^t, \mathbf{w}_2^t, \dots, \mathbf{w}_U^t], \boldsymbol{\theta}_t\}$ be the firefly t . A population of N fireflies $\{\mathbf{W}_t, \boldsymbol{\theta}_t\}$, $t \in \{1, 2, \dots, N\}$ are initialized and the brightness of the firefly t $\{\mathbf{W}_t, \boldsymbol{\theta}_t\}$, is defined as

$$I_t(\mathbf{W}_t, \boldsymbol{\theta}_t) = \frac{1}{f(\mathbf{W}_t) + P(\mathbf{W}_t, \boldsymbol{\theta}_t)}. \quad (40)$$

Given any two fireflies t and l among the population, if the firefly t is brighter than the firefly l , i.e., $I_t(\mathbf{W}_t, \boldsymbol{\theta}_t) > I_l(\mathbf{W}_l, \boldsymbol{\theta}_l)$, then the firefly l will fly towards the firefly t by

$$\mathbf{W}_l^{(n+1)} = \mathbf{W}_l^{(n)} + \beta_0 e^{-\gamma \left(r_{w,tl}^{(n)}\right)^2} \left(\mathbf{W}_t^{(n)} - \mathbf{W}_l^{(n)}\right) + \alpha^{(n)} \mathbf{V}, \quad (41)$$

$$\boldsymbol{\theta}_l^{(n+1)} = \boldsymbol{\theta}_l^{(n)} + \beta_0 e^{-\gamma \left(r_{\theta,tl}^{(n)}\right)^2} \left(\boldsymbol{\theta}_t^{(n)} - \boldsymbol{\theta}_l^{(n)}\right) + \alpha^{(n)} \mathbf{v}, \quad (42)$$

where $r_{w,tl}^{(n)} = \|\mathbf{W}_t^{(n)} - \mathbf{W}_l^{(n)}\|$ and $r_{\theta,tl}^{(n)} = \|\boldsymbol{\theta}_t^{(n)} - \boldsymbol{\theta}_l^{(n)}\|$ are the Cartesian distances, β_0 represents the attractiveness at $r_{w,tl}^{(n)} = 0$ and $r_{\theta,tl}^{(n)} = 0$. Here, γ represents the variations of attractiveness. The second terms on the right-hand sides of equations (41) and (42) are the firefly's attractions. The third terms on the right-hand sides of equations (41) and (42) represent randomization in which the randomization factor $\alpha^{(n)}$ controls the strength of the randomization. The entries of $\mathbf{v}$ and $\mathbf{V}$ are drawn from either a Gaussian distribution or an uniform distribution.

In essence, problem (36) is a special case of the general problem (1) whose objective and constraints depend on two sets of variables $\mathbf{W}$ and $\boldsymbol{\theta}$. Hence, the procedure using the FA method for RIS-aided beamforming possesses is similar to the steps as in Algorithm 1 except four steps 3, 16, 18 and 19. The different steps are described in Algorithm 3.

Algorithm 3 Procedure of generalized FA to solve (29)

Input: Parameters for FA: $N, T, \lambda_i, \rho_n, \beta_0; \gamma$; Optimization data: $\mathbf{H}, \mathbf{g}_i, \sigma_i^2, \eta_i, I_{to}$;
 Step 3: Calculate the brightness of N fireflies as (40);
 Step 16: Update the location of firefly i towards firefly j as (41) and (42);
 Step 18: Vary the attractiveness strength according to distances via $e^{-\gamma \left(r_{w,ji}^{(n)}\right)^2}$ and $e^{-\gamma \left(r_{\theta,ji}^{(n)}\right)^2}$;
 Step 19: Calculate new solutions and update the brightness of N fireflies $I_i(\mathbf{W}_i, \boldsymbol{\theta}_i)$ as (40);
return Optimal solution $\mathbf{W}^*, \boldsymbol{\theta}^*$.

4.4 Complexity Analysis for AO and FA

The computational complexities of utilizing AO and FA to solve problem (29) are analyzed in this section.

Lemma 3 The computational complexity that the AO method requires to solve (29) is on the order of [52]:

$$n_o (\tau_1 + \tau_2), \quad (43)$$

where

$$\tau_1 = \ln(\varepsilon^{-1})\sqrt{U(M_t + 1)} \left[(M_t^2 + 1)U + U M_t^2(M_t^2 + M_t) + M_t^4 \right] M_t^2, \quad (44)$$

$$\tau_2 = \ln(\varepsilon^{-1})\sqrt{U + 2N_t} \left[(N_t^2 + 1)(U + 2N_t^2) + N_t^4 \right] N_t^2. \quad (45)$$

Proof First, consider sub-problem (34). It can be seen that (34) has U constraints in the LMI form of size 1 and U constraints in the LMI form of size M_t . Following similar steps as described in [48, Section V-A], the following conclusions can be obtained:

- The complexity for iterations is of the order

$$\ln(\varepsilon^{-1})\sqrt{U(M_t + 1)}. \quad (46)$$

- The complexity of each iteration is of the order

$$\left[(M_t^2 + 1)U + U M_t^2(M_t^2 + M_t) + M_t^4 \right] M_t^2. \quad (47)$$

Combining (46) and (47), the overall computational complexity to solve (34) can be written as

$$\tau_1 = \ln(\varepsilon^{-1})\sqrt{U(M_t + 1)} \left[(M_t^2 + 1)U + U M_t^2(M_t^2 + M_t) + M_t^4 \right] M_t^2. \quad (48)$$

Now, consider sub-problem (35) with N_t^2 being the number of decision variables of (35). Given $\varepsilon > 0$, Θ^ε is defined as an ε -solution to the optimization problem (35) when

$$\text{Tr}(\Theta^\varepsilon) \leq \min_{\Theta} \text{Tr}(\Theta) + \varepsilon.$$

Similar observation as in the previous sub-problem leads to the following facts. Optimization problem (35) contains U constraints in the LMI form of size 1 and 2 constraints in the LMI form of size N_t . Hence, similar conclusions can be drawn

- The complexity for iterations is of the order

$$\ln(\varepsilon^{-1})\sqrt{U + 2N_t}. \quad (49)$$

- The computational complexity of each iteration is of the order

$$\left[(N_t^2 + 1)(U + 2N_t^2) + N_t^4 \right] N_t^2. \quad (50)$$

Combining (49) and (50), the computational complexity to solve (35) can be expressed as

$$\tau_2 = \ln(\varepsilon^{-1})\sqrt{U + 2N_t} \left[(N_t^2 + 1)(U + 2N_t^2) + N_t^4 \right] N_t^2. \quad (51)$$

As the AO method alternatively solves (34) and (35) in n_o iterations, its complexity is of the order of $n_o (\tau_1 + \tau_2)$. $\square$

Lemma 4 *The FA presented in Algorithm 3 requires the complexity of the order of [52]*

$$\begin{aligned} & TN^2 [M_t^2 + N_t + N (UM_t + U(N_t^2 + M_t N_t) + N_t)] \\ & + TN \log N + NM_t U + N_t N + N \log N \\ & + N (UM_t + U(N_t^2 + M_t N_t) + N_t). \end{aligned} \quad (52)$$

Proof The major computational complexity to perform Algorithm 3 are dominated by steps 2, 3, 4, 16, 19, and 22.

In step 2, randomly generate an $M_t \times U$ matrix and N_t scalars, respectively requires the complexity of $NM_t U$ and NN_t . Hence, generating N firefly requires the complexity on the order of $NM_t U + N_t N$.

In steps 3 and 19, computing $\phi_i(\mathbf{W}, \boldsymbol{\theta})$, $\varphi_k(\theta_k)$, and $\sum_{i=1}^U \mathbf{w}_i^H \mathbf{w}_i$, respectively, require the complexities on the order of $U(N_t^2 + M_t N_t)$, N_t , and UM_t . Therefore, evaluating the brightness of N fireflies requires the complexity of the order of $N (UM_t + U(N_t^2 + M_t N_t) + N_t)$.

In step 16, the computational cost to move a firefly is on the order of $M_t^2 + N_t$.

In steps 4 and 22, the computational cost to order/rank N firefly is $N \log N$.

Finally, consider a worst case where step 16 is required in all inner loops of the FA algorithm, (52) can be attained after some mathematical manipulations. $\square$

5 Performance Evaluations

The performance of the proposed FA approaches for transmit beamforming and RIS-aided transmit beamforming has been evaluated and consequently compared with their SDR counterparts in this section.

The CVX optimization package [78] has been used to attain the solutions for the SDR method for problem (16) and the AO method for the RIS-aided downlink beamforming. Two SDP sub-problems (34) and (35) in the AO method are iteratively solved for $n_0 = 10$ iterations. The following parameters are used for the FAs. The attractiveness variation, i.e., γ , and the attractiveness at zero distance, i.e., β_0 , are all set to be 1. The same penalty coefficient value is used for all penalty constants, however, this value dynamically varies with the generation/iteration index n in Algorithm 1 as

$$\lambda_i = \rho_k = n^2, \quad \forall i, k.$$

The initial randomization factor is set at $\alpha^{(0)} = 0.9$ but its value is changing at the n -th generation or iteration as $\alpha^{(n)} = \alpha^{(0)} 0.9^n$.

5.1 Evaluation for Transmit Beamforming

This experiment considers the following scenario. Three local mobile users located at -40° , 10° , 30° , and two neighbouring users located at 0° and 50° , with respect to the BS's antenna array broadside. These two neighbouring users, respectively, have their tolerable interference levels of $I_{to,1} = 0.001$ and $I_{to,2} = 0.0001$. The required SINR of the local users is set to 1 and the noise variance is 0.1 for all local users.

The channel covariance matrices between the BS and its local mobile user t , i.e., $\mathbf{H}_t = \mathbf{R}(\zeta_t, \delta_a)$, as well as between the BS and its neighbouring user k , i.e., $\mathbf{Q}_k = \mathbf{R}(\zeta_k, \delta_a)$, are complex functions of the standard deviation of the angular spread δ_a , the angles of departure ζ_t and ζ_k . The (m, n) th entry of $\mathbf{R}(\zeta, \delta_a)$ can be expressed as, [8]:

$$e^{\frac{j2\pi\Delta}{\psi}[(n-m)\sin\zeta]} e^{-2\left[\frac{\pi\Delta\delta_a}{\psi}\{(n-m)\cos\zeta\}\right]^2}, \quad (53)$$

where ψ captures the carrier wavelength. Here, $\sigma_a = 2^\circ$ and the BS's antenna elements are equally spaced at the distance of $\Delta = \psi/2$.

Figure 1a displays the BS's radiation patterns of the SDR method in (16) whereas Fig. 1b indicates the BS's radiation patterns of the FA method presented in Algorithm 2. The number of the BS's antenna elements is 8. It is clear that the FA provides

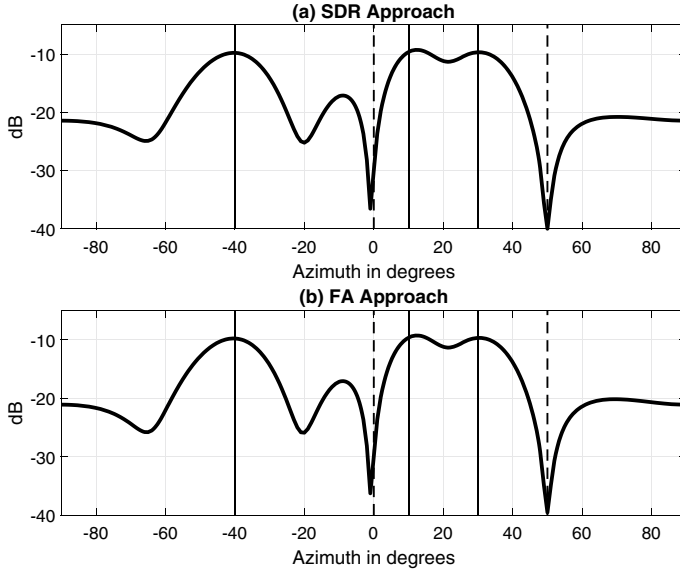


Fig. 1 Radiation pattern of the BS with 8 antenna elements: **a** SDR method; **b** FA method with the population size of $N = 100$. Three solid vertical lines indicate the locations of the local mobile users, whereas two dashed vertical lines show the locations of the neighbouring users. The number of maximum generation is $T = 150$

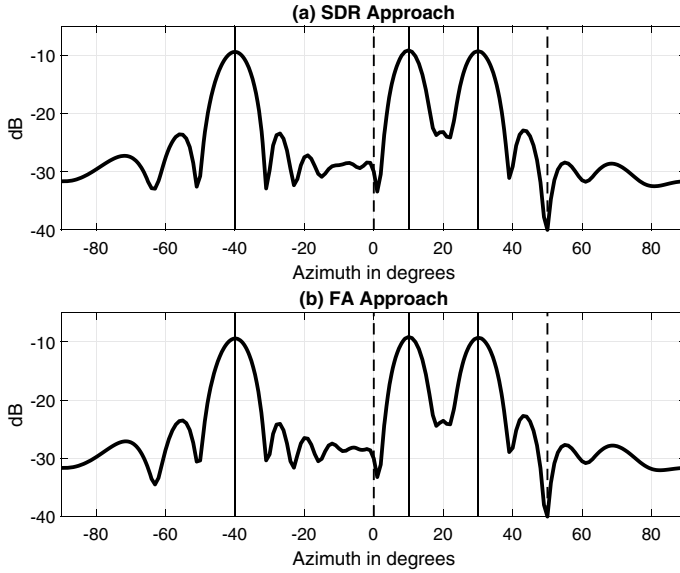


Fig. 2 Radiation pattern of the BS with 16 antenna elements: **a** SDR method; **b** FA method with the population size of $N = 100$. Three solid vertical lines indicate the locations of the local mobile users, whereas two dashed vertical lines show the locations of the neighbouring users. The number of maximum generation is $T = 150$

the same radiation pattern as its SDR counterpart does. Both FA and SDR methods are able to steer peaks and nulls, respectively, towards the directions of the BS's local mobile users and neighbouring users.

Figure 2a and b illustrate the BS's radiation patterns of the SDR and FA, respectively, with 16 antennas. As the number of antenna elements increases, sharper patterns can be formed by both approaches. Again, it can be observed that FA and SDR attain the same radiation patterns with the desired peaks and nulls.

The radiation patterns of the SDR and FA approaches with 24 antenna elements are shown in Fig. 3a and b, respectively. It is confirmed here again that both approaches deliver the same optimal solutions hence having the same radiation patterns.

In Fig. 4a, b, the radiation patterns of the SDR and FA are respectively, provided for 32 antennas. Due to higher number of antenna elements available at the BS, higher radiation pattern resolution can be realized by both approaches. This is because of the fact that, increasing the number of antenna elements at the BS provides higher degree of freedom for the two approaches, hence they can attain better solution. From the results shown in Figs. 1, 2, 3, and 4, it can be seen that FA obtains the same optimal solution as the interior-point method does for the SDR counterpart. This result essentially confirms that the proposed generalized FA can effectively handle highly nonlinear and multimodal optimization problems.

Consider the setups in Fig. 1, i.e., $M_t = 8, 16, 24, 32$, $U = 3$, $K = 2$, $N = 100$, and $T = 150$, it can be observed from Lemmas 1 and 2 that the FA method demands

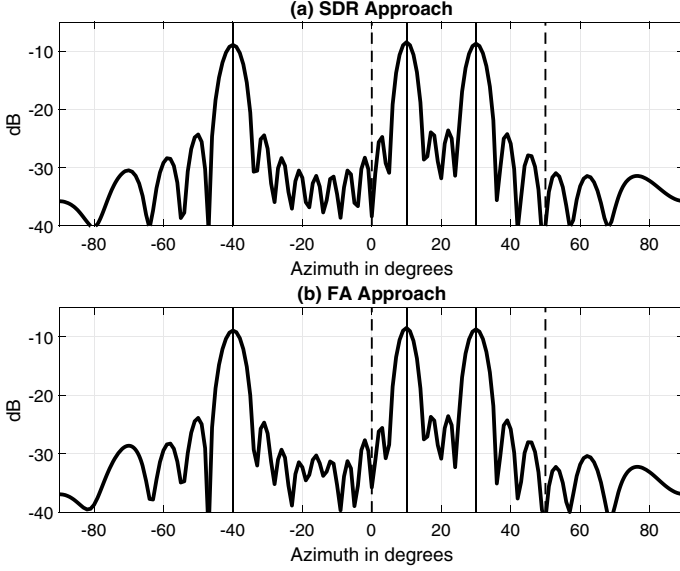


Fig. 3 Radiation pattern of the BS with 24 antenna elements: **a** SDR method; **b** FA method with the population size of $N = 100$. Three solid vertical lines indicate the locations of the local mobile users while two dashed vertical lines show the locations of the neighbouring users. The number of maximum generation $T = 150$

higher computational complexity than its SDR counterpart does. Interestingly, FA seems performs better than the SDR at a large-antenna regime. It can be verified the fact that the dominant term in (23) is $M_t^{6\frac{1}{2}}$. On the other hand, assuming the worst case when $T = N = M_t$, the dominant term in (26) is M_t^6 . Comparing these two results, i.e., $\mathcal{O}(M_t^{6\frac{1}{2}})$ and $\mathcal{O}(M_t^6)$, it is clear that the computational complexity of an IPM to solve (16) is slightly higher than that of the FA in Algorithm 2 [52].

Figure 5 depicts the total transmit power of the FA method versus the population size with various BS's antennas. The FA converges after the maximum number of population of $N = 50$ with all setups for the size of BS's antenna arrays. It can also be observed that the size of the population at the converged transmit power increases as the size BS's antenna arrays increases. For example, with $M = 8, 16, 24$, and 32 , the FA approach converges with the population size of $N = 20, 30, 40$, and 50 , respectively. Moreover, the total transmit power decreases as the size of BS's antenna arrays increases. These results can be reasoned as follows. As evidenced in Figs. 1, 2, 3, and 4, the resolution of the BS's radiation beam patterns increases with the number of BS's antenna elements. Therefore, sharper beams can be formed and the wasted power in the sides-lobes is reduced leading to the reduction of the required transmit power. Furthermore, the size of the problem increases with a higher

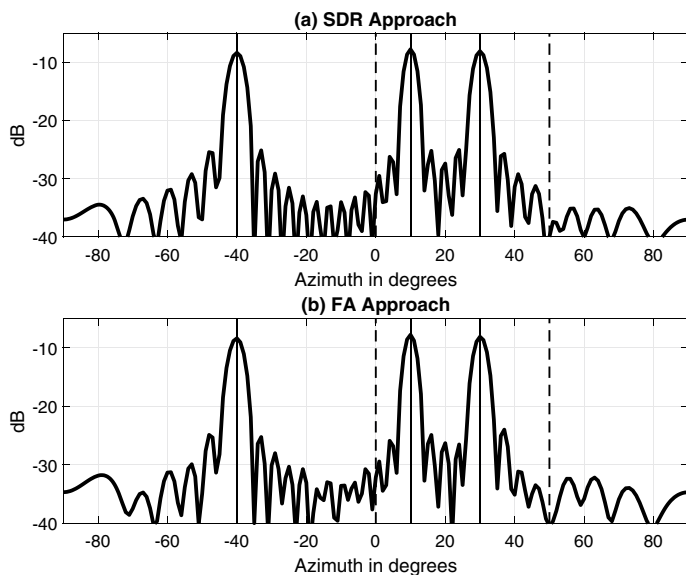


Fig. 4 Radiation pattern of the BS with 32 antenna elements: **a** SDR method; **b** FA method with the population size $N = 100$. Three solid vertical lines indicate the locations of the local mobile users, whereas two dashed vertical lines show the locations of the neighbouring users. The number of maximum generation is $T = 150$

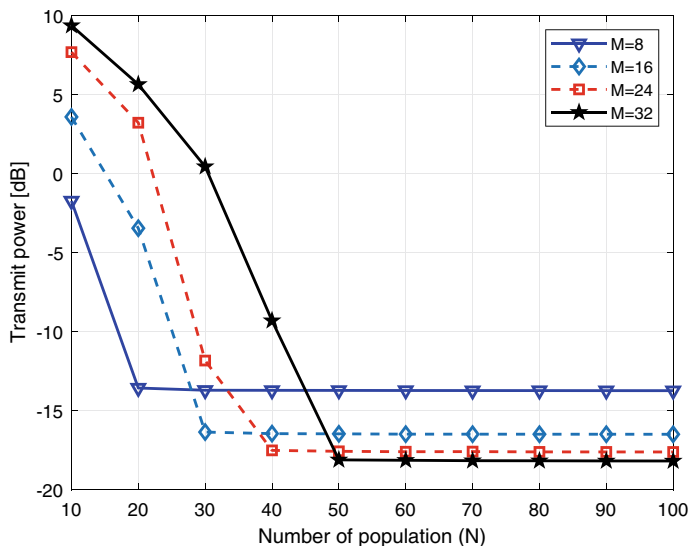


Fig. 5 Total transmit power obtained by the FA method versus the population size with varying numbers of BS's antennas. The maximum number of generations is $T = 150$

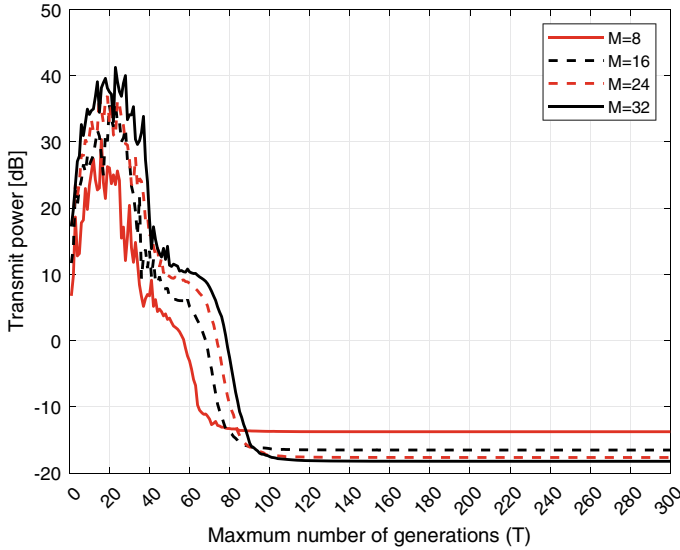


Fig. 6 Total transmit power obtained by the FA method versus the number of maximum generations with varying numbers of BS's antenna elements. The population size of $N = 100$ is used

number of BS's antenna elements, hence, demanding a larger firefly population to ensure sufficient diversification for the FA to explore.

Figure 6 illustrates the total transmit power of the FA method versus the maximum number of generations with various numbers of BS's antenna elements. The same trend as presented in Fig. 5 can be seen in Fig. 6. The total transmit power required by the FA method converges with all numbers of BS's antenna elements. As the number of BS's antenna element increases, the problem's dimension enlarges. Hence, the number of generations has to increase to provide better exploitation. For example, the total transmit power, respectively, reaches a stable level at around 80 and 90 generations for $M = 8$ and 16. Interestingly, the difference in the required number of generations for $M = 24$ and $M = 32$ is negligible as the number of generations for those setups is around 100.

5.2 Evaluations for RIS-Aided Transmit Beamforming

In this experiment, an RIS-aided wireless communication system consisting of one RIS, one BS, and two users (i.e., $U = 2$) is considered. The distance between the RIS and the BS is 500 m. Mobile users are located randomly with a distance of 50 m from the RIS. Line-of-sight channels are assumed between BS and RIS as well as between RIS and mobile users where the signal attenuation following the free-space

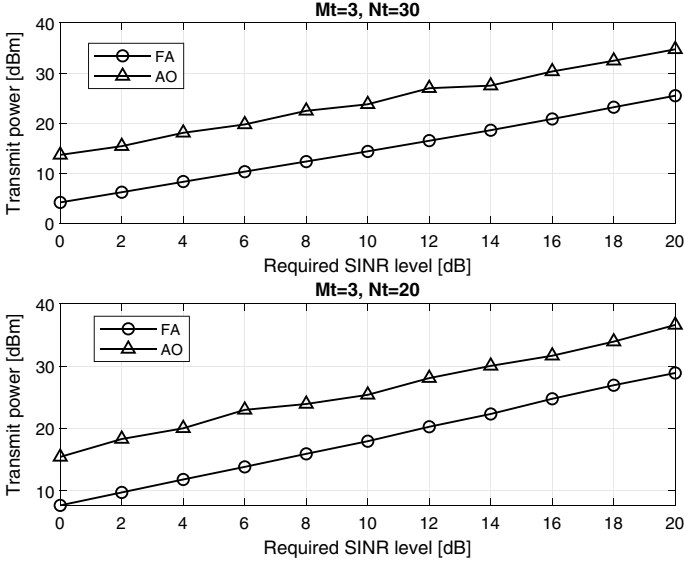


Fig. 7 Total transmit power of the BS versus the required SINR level with varying numbers of RIS's reflective elements. Here, $M_t = 3$ and $N = 120$ are used, together with the maximum number of generations $T = 50$

path loss model, i.e., $\left(\frac{4\pi d\psi}{c}\right)^{-2}$ [81] where d is the distance between the RIS and the BS or between the RIS and a mobile user, c is the speed of light, and $\psi = 950$ MHz is the carrier frequency. The variance of the noise measured at each mobile user is -124 dBm. Monte Carlo simulations are performed over channel realizations where each channel realization is the combination of a random user location and a random fading coefficient with a zero mean and the variance of 1.

Figure 7 shows the total transmit power versus the required SINR with various numbers of RIS's reflective elements. The number of BS' antenna elements is 3. It is clear from the figure that the FA can outperform the AO approach in consuming less power to deliver a required SINR. The same behaviour can also be seen in Fig. 8. The results can be reasoned as follows. The original non-convex problem (29) in the AO method is approximated by two convex problems (34) and (35). As a result of the approximation, the solution provided by the AO method is not necessarily the global optimal solution to the original problem (29). Whereas, having both exploration and exploitation abilities, the FA effectively solves such non-convex problem and attain much better solution than the AO does. The results, shown in both Figs. 7 and 8, again confirm the capability of the FA of solving highly nonlinear and multimodal optimization problems.

Next, the computational complexities of both AO and FA methods for the experiments are compared, as shown in Figs. 7 and 8. Since $N_t > M_t$, it can be derived from Lemma 3 and Lemma 4 that $n_0 N_t^{6\frac{1}{2}}$ and $T N^3 N_t^2$ are, respectively, the domi-

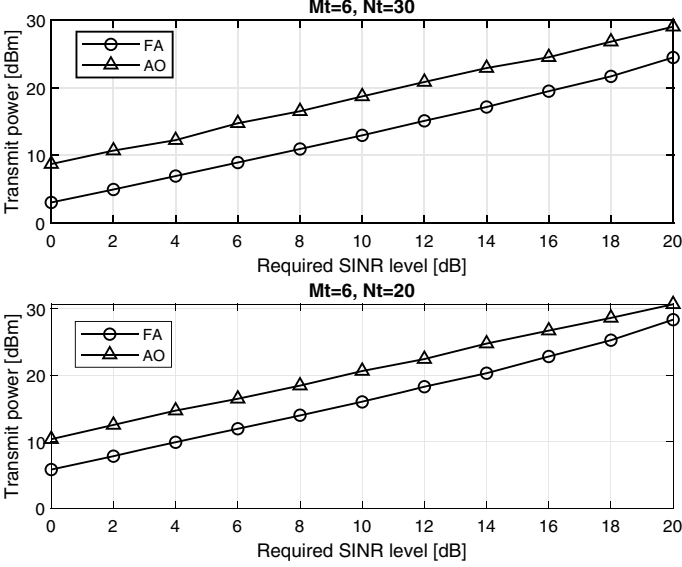


Fig. 8 Total transmit power of the BS versus the required SINR level with varying numbers of RIS's reflective elements. Both $M_t = 6$ and $N = 120$ are used for the maximum number of generations $T = 50$

nant terms of the complexity of the AO and FA methods. Substituting for $N_t = 30$, $n_0 = 10$, $N = 120$ and $T = 50$ into $n_0 N_t^{6\frac{1}{2}}$ and $T N^3 N_t^2$ leading to the conclusion that the FA and AO methods have the same complexity order of $\mathcal{O}(10^{10})$.

In a large-antenna regime, i.e., M_t and N_t are large, by letting $N_t = n_0 = M_t$ in (43), it is straightforward to verify that the dominant term of the computational complexity of the IPM method is $M_t^{7\frac{1}{2}}$. Furthermore, the dominant term of the computational complexity of the FA method in (52) is M_t^6 when assuming $T = N = N_t = M_t$. Comparing $M_t^{7\frac{1}{2}}$ with M_t^6 leads to a conclusion that the computational complexity of the IPM approach in solving (29) is higher than that of the FA method in Algorithm 3.

Figure 9 shows the total transmit power of the BS versus the maximum number of generations T of the FA method presented in Algorithm 3 with various BS's antenna elements and the RIS's elements. Clearly, the results show that the FA method needs around 40 to 50 generations to obtain the optimal solution for all antenna-element setups.

Figure 10 plots out the total transmit power versus the number of population N with various RIS's elements and BS's antenna elements. Again, the results indicate that the FA can attain better solutions when the population size of the firefly increases. For example, when the firefly population grows from 20 to 120, the total transmit power reduces by around 5 dB for the setups of $(M_t = 6, N_t = 30)$, $(M_t = 6, N_t = 20)$, $(M_t = 3, N_t = 30)$ and around 2.5 dB for $(M_t = 3, N_t = 20)$. On the other hand, the total transmit power drops by around 0.5 dB when the population size grows from 120 to 200 for all setups.

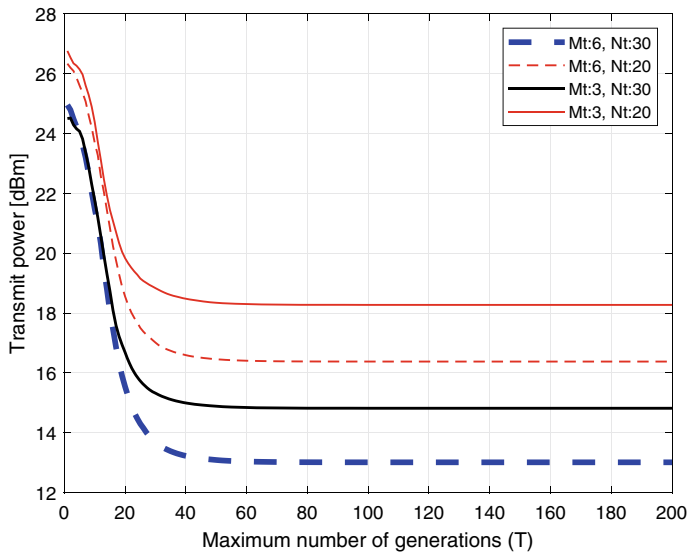


Fig. 9 Total transmit power of the BS versus the number of maximum generations with varying numbers of BS's antennas as well as RIS's reflective elements. Here, the population size $N = 50$ is used, together with the required SINR 10 dB

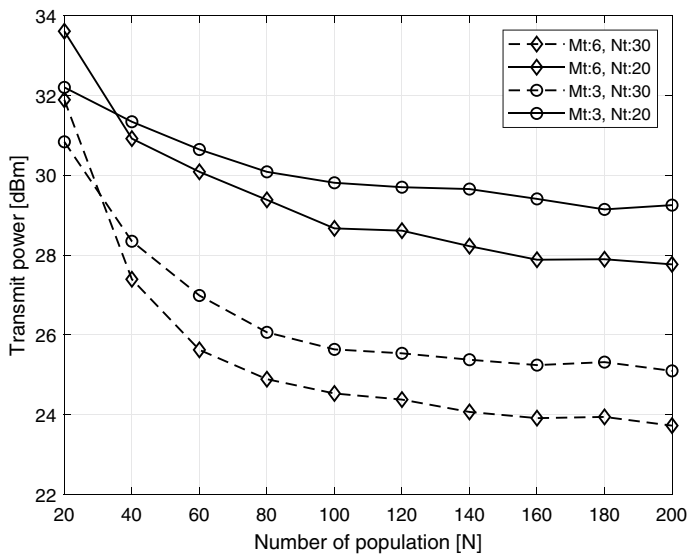


Fig. 10 Total transmit power of the BS versus the number of populations with varying numbers of BS's antennas as well as RIS's reflective elements. The maximum number of generations is set to $T = 50$, whereas the required SINR is set to 20 dB

6 Conclusion

This chapter summarizes the latest work using the nature-inspired firefly algorithm in solving transmit beamforming problems in wireless communications. A generalized FA is presented for solving an optimization framework with an objective function and nonlinear constraints being in forms of multivariate functions of matrix inputs/variables. Two transmit beamforming problems have been used to show how both FA and semidefinite relaxation technique perform. Simulations reveal that FA obtains the same optimal solution as the interior point method does for a hidden convex problem, whereas FA can outperform its gradient-based counterpart in solving a non-convex optimization problem with multivariate matrix inputs/variables. Interestingly, for problems in the large-antenna regimes, the proposed FA methods require less computational complexity than their interior-point counterparts do.

References

1. Balanis CA, Ioannides PI (2007) *Introduction to Smart Antennas*. Morgan and Claypool Publishers
2. Godara LC (2004) *Smart antennas*. CRC Press
3. Zooghyby AE (2005) *Smart antenna engineering*. Artech House
4. Gross F (2005) *Smart antennas for wireless communications*. McGraw-Hill
5. Hayssam D, Yu W (2010) Coordinated beamforming for the multicell multi-antenna wireless system. *IEEE Trans Wireless Commun* 9(5):1748–1759
6. Rashid-Farrokhi F, Liu KJR, Tassiulas L (1998) Transmit beamforming and power control for cellular wireless systems. *IEEE J Sel Areas Commun* 16(8):1437–1450
7. Ami W, Yonina EC, Shlomo S (2006) Linear precoding via Conic optimization for fixed MIMO receivers. *IEEE Trans Sig Proc* 54(1):161–176
8. Bengtsson M, Ottersten B (1999) Optimal downlink beamforming using Semidefinite optimization. In: *Proceeding 37th Annual Allerton Conference on Communication, Control*, pp 987–996
9. Huang Y, Palomar DP (2010) Rank-constrained separable semidefinite programming with applications to optimal beamforming. *IEEE Trans Signal Process* 58(2):664–678
10. Le TA, Nakhai MR (2013) Downlink optimization with interference pricing and statistical CSI. *IEEE Trans Commun* 61(6):2339–2349
11. Chien TV, Bjoernson E, Larsson EG, Le TA (2018) Distributed power control in downlink cellular massive MIMO systems. In: *WSA 2018; 22nd international ITG workshop on smart antennas*
12. Huang Yongwei, Li Qiang, Ma Wing-Kin, Zhang Shuzhong (2012) Robust multicast beamforming for spectrum sharing-based cognitive radios. *IEEE Trans. Signal Process.* 60(1):527–533 January
13. Le TA, Nguyen HX, Vien Q-T, Karamanoglu M (2015) Secure information transmission and power transfer in cellular networks. *IEEE Commun Lett* 19(9):1532–1535
14. Le TA, Nguyen HX, Vien QT, Karamanoglu M (2015) Secure information transmission in the presence of energy-harvesting eavesdroppers in multi-cell networks. In: *IEEE Global Communication of Conference*, pp 1–6
15. Le TA, Vien Q-T, Nguyen HX, Ng DWK, Schober R (2017) Robust chance-constrained optimization for power-efficient and secure SWIPT systems. *IEEE Trans Green Commun Network* 1(3):333–346

16. Chu Z, Le TA, Nguyen HX, Karamanoglu M, Zhu Z, Nallanathan A, Ever E, Yazici A (2017) Robust design for miso swipt system with artificial noise and cooperative jamming. In: GLOBE-COM 2017–2017 IEEE Global Communications Conference, pp 1–6
17. Kwan NDW, Ernest LS, Robert S (2014) Robust beamforming for secure communication in systems with wireless information and power transfer. *IEEE Trans Wireless Commun* 13(8):4599–4615
18. Kwan NDW, Ernest LS, Robert S (2013) Wireless information and power transfer: energy efficiency optimization in OFDMA systems. *IEEE Trans Wirel Commun* 12(12):6352–6370
19. Hayssam D, Yu W (2011) Multicell interference mitigation with joint beamforming and common message decoding. *IEEE Trans Commun* 59(8):2264–2273
20. Le TA, Nakhai MR (2011) An iterative algorithm for downlink multi-cell beam-forming. In: Proceeding of IEEE global telecommun Conference, pp 1–6
21. Le TA, Nakhai MR (2012) Coordinated beamforming using semidefinite programming. In: Proceeding of IEEE international conference communication, pp 1–5
22. Le TA, Nakhai MR (2012) A decentralized downlink beamforming algorithm for multicell processing. In: Proceeding IEEE global telecommunication conference, pp 3924–3929
23. Le TA, Nasser S, Zarrebini-Esfahani A, Mills A, Nakhai MR (2011) Power-efficient downlink transmission in multicell networks with limited wireless backhaul. In: IEEE wireless communications magazine, special issue on technologies for green radio communication networks, vol 18, no 5, pp 82–88
24. Le TA, Ng DWK, Yang X-S (2023) A rank-one optimization framework and its applications to transmit beamforming. *IEEE Trans Veh Technol* 73(1):620–636
25. Martin S, Holger B (2004) Solution of the multiuser downlink beamforming problem with individual SINR constraints. *IEEE Trans Veh Technol* 53(1):18–28
26. Yang W, Xu G (1998) Optimal downlink power assignment for smart antenna systems. In: Proceedings of the 1998 IEEE international conference on acoustics, speech and signal processing, ICASSP '98 (Cat. No.98CH36181), vol 6, pp 3337–3340
27. Emil B, Mats B, Björn O (2014) Optimal multiuser transmit beamforming: a difficult problem with a simple solution structure [lecture notes]. *IEEE Sig Proc Magaz* 31(4):142–148
28. Ramya B, Heath Robert W (2011) Adaptive limited feedback for sum-rate maximizing beamforming in cooperative multicell systems. *IEEE Trans Sig Proc* 59(2):800–811
29. Chu Z, Le TA, To D, Nguyen HX (2017) Sum throughput optimization for wireless powered sensor networks. In: Globecom 2017–2017 IEEE global communications conference, pp 1–6
30. Gan Z, Kai-Kit W, Tung-Sang N (2008) Throughput maximization in linear multiuser MIMO-OFDM downlink systems. *IEEE Trans Veh Technol* 57(3):1993–1998
31. Hammarwall D, Bengtsson M, Ottersten B (2006) On downlink beamforming with indefinite shaping constraints. *IEEE Trans Sig Proc* 54(9):3566–3580
32. Clerckx B, Zhang R, Schober R, Wing Kwan Ng D, Kim DI, Poor HV (2019) Fundamentals of wireless information and power transfer: from RF energy harvester models to signal and system designs. *IEEE J Select Areas Commun* 37(1):4–33
33. Khandaker MRA, Wong K-K (2015) Robust secrecy beamforming with energy-harvesting eavesdroppers. *IEEE Wireless Commun Lett* 4(1):10–13
34. Le TA, Navaie K (2015) On the interference tolerance of the primary system in cognitive radio networks. *IEEE Wireless Commun Lett* 4(3):281–284
35. Le TA, Navaie K (2014) Downlink beamforming design for cognitive cellular systems. In: 2014 IEEE wireless communications and networking conference (WCNC), pp 1188–1193
36. Le TA, Navaie K (2014) Downlink beamforming in underlay cognitive cellular networks. *IEEE Trans Commun* 62(7):2212–2223
37. Le TA, Nakhai MR, Navaie K (2015) A robust transmission strategy for multi-cell interference networks. In: Proceedings of IEEE Global Communicational Conference (GLOBECOM), pp 1–6
38. Le TA, Navaie K, Vien Q-T, Nguyen HX (2016) Beamforming in coexisting wireless systems with uncertain channel state information. In: 2016 IEEE 84th vehicular technology conference (VTC-Fall), pp 1–5

39. Nasser S, Le TA, Nakhai MR (2013) Robust and power efficient interference management in downlink multi-cell networks. In: 2013 IEEE 24th annual international symposium on personal, indoor, and mobile radio communications (PIMRC), pp 1097–1101
40. Navaie K, Le TA (2014) Fundamental performance trade-offs in coexisting wireless networks. In: 1st international conference on 5G for ubiquitous connectivity, pp 246–251
41. Song E, Shi Q, Sanjabi M, Sun R, Luo ZQ (2011) Robust sinr-constrained MISO downlink beamforming: when is semidefinite programming relaxation tight? In: IEEE Int Conf Acoustics, Speech and Signal Process. (ICASSP), pp 3096–3099
42. Chu Z, Xing H, Johnston M, Le Goff S (2016) Secrecy rate optimizations for a MISO secrecy channel with multiple multiantenna eavesdroppers. *IEEE Trans Wireless Commun* 15(1):283–297
43. Khandaker MRA, Wong KK, Zhang Y, Zheng Z (2016) Probabilistically robust SWIPT for secrecy MISOME systems. *IEEE Trans Inf Forens Secur* 1–16. Early access
44. Le TA, Chien TV, Renzo MD (2021) Robust probabilistic-constrained optimization for IRS-Aided MISO communication systems. *IEEE Wirel Commun Lett* 10(1):1–5
45. Le TA, Vien Q-T, Nguyen HX, Ng DWK, Schober R (2016) Robust optimization with probabilistic constraints for power-efficient and secure swipt. In: 2016 IEEE global communications conference (GLOBECOM), pp 1–7
46. Nasser S, Nakhai M, Le TA (2016) Chance constrained robust downlink beamforming in multicell networks. *IEEE Trans Mobile Comput* 15(11):2682–2691
47. Feng W, Tao P, Yongwei H, Xin W (2015) Robust transceiver optimization for power-splitting based downlink MISO SWIPT systems. *IEEE Signal Proc Lett* 22(9):1492–1496
48. Wang K-Y, So AM-C, Chang T-H, Ma W-K, Chi C-Y (2014) Outage constrained robust transmit optimization for multiuser MISO downlinks: tractable approximations by conic optimization. *IEEE Trans Signal Proc* 62(21):5690–5705
49. Zarrebini-Esfahani A, Le TA, Nakhai MR (2013) Robust cognitive beamforming for cell-edge coverage in multicell networks with probabilistic constraints. In: 2013 IEEE global communications conference (GLOBECOM), pp 1253–1258
50. Zhou F, Li Z, Cheng J, Li Q, Si J (2016) Robust an-aided beamforming and power splitting design for secure MISO cognitive radio with SWIPT. February 2016. [Online]. Available: <http://arxiv.org/pdf/1602.06913v1.pdf>
51. Yu W, Tian L (2007) Transmitter optimization for the multi-antenna downlink with per-antenna power constraints. *IEEE Trans Sig Proc* 55(6):2646–2660
52. Le TA, Yang X-S (2023) Generalized firefly algorithm for optimal transmit beamforming. *IEEE Transactions on Wireless Communications*, pp 1–15. Early Access
53. Luo Z-Q, Ma W-K, So AM-C, Ye Y, Zhang S (2010) Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Proc Mag* 27(3):20–34
54. Abeywickrama S, Zhang R, Wu Q, Yuen C (2020) Intelligent reflecting surface: practical phase shift model and beamforming optimization, [Online]. Available: <https://arxiv.org/pdf/2002.10112.pdf>
55. Basar E, Di Renzo M, De Rosny J, Debbah M, Alouini M, Zhang R (2019) Wireless communications through reconfigurable intelligent surfaces. *IEEE Access* 7:116753–116773
56. De-Ming C, Feng-Ji C, Yu-Chen C, Chao-Kai W, Wu C-H, Fu-Kang W, Kai-Kit W, Chan-Byoung C (2024) Active RIS-assisted MIMO-OFDM system: analyses and prototype measurements. *IEEE Commun Lett* 28(1):208–212
57. Renzo MD, Debbah M, Phan-Huy D-T, Zappone A, Alouini M-S, Yuen C, Sciancalepore V, Alexandropoulos GC, Hoydis J, Gacanin H, De Rosny J, Bounceur A, Leroose G (2019) Fink M (2019) Smart radio environments empowered by reconfigurable AI meta-surfaces: an idea whose time has come. *EURASIP J Wireless Commun Network* 1:1–20
58. Renzo MD, Zappone A, Debbah M, Alouini M-S, Yuen C, de Rosny J, Tretyakov S (2020) Smart radio environments empowered by reconfigurable intelligent surfaces: how it works, state of research, and road ahead. *ArXiv preprint arXiv:2004.09352*
59. Gong S, Xing C, Yue P, Zhao L, Quek TQS (2023) Hybrid analog and digital beamforming for RIS-assisted mmWave communications. *IEEE Trans Wireless Commun* 22(3):1537–1554

60. Zhangjie P, Zhiwei C, Cunhua P, Gui Z, Hong R (2022) Robust transmission design for RIS-aided communications with both transceiver hardware impairments and imperfect CSI. *IEEE Wirel Commun Lett* 11(3):528–532
61. Qian X, Renzo MD, Liu J, Kammoun A, Alouini M-S (2020) Beamforming through reconfigurable intelligent surfaces in single-user MIMO systems: SNR distribution and scaling laws in the presence of channel fading and phase noise, 2020. [Online]. Available: <https://arxiv.org/pdf/2005.07472.pdf>
62. Shahiri V, Behroozi H, Kuhestani A, Wong K-K (2024) Reconfigurable intelligent surface-assisted secret key generation under spatially correlated channels in quasi-static environments. *IEEE Internet Things J* 1–15
63. Kewei W, Nan Q, Xin G, Qingjiang S, Ming X, Shi J, Kai-Kit W (2023) Transmit/passive beamforming design for multi-IRS assisted cell-free MIMO networks. *IEEE Syst J* 17(4):6282–6291
64. Wu Q, Zhang R (2020) Towards smart and reconfigurable environment: intelligent reflecting surface aided wireless network. *IEEE Commun Mag* 58(1):106–112
65. Xiao H, Hu X, Li A, Wang W, Su Z, Wong K-K, Yang K (2024) STAR-RIS enhanced joint physical layer security and covert communications for multi-antenna mmWave systems. *IEEE Trans Wireless Commun* 1–15
66. Zappone A, Di Renzo M, Shams F, Qian X, Debbah M (2020) Overhead-aware design of reconfigurable intelligent surfaces in smart radio environments. [Online]. Available: <https://arxiv.org/abs/2003.02538>
67. Zheng B, Zhang R (2019) Intelligent reflecting surface-enhanced OFDM: channel estimation and reflection optimization 9(4):518–522
68. Zhou G, Pan C, Ren H, Wang K, Di Renzo M, Nallanathan A (2019) Robust beamforming design for intelligent reflecting surface aided MISO communication system
69. Wu Q, Rui Z (2020) Weighted sum power maximization for intelligent reflecting surface aided SWIPT. *IEEE Wireless Commun Lett* 9(5):586–590
70. Yang X-S (2008) Nature-inspired metaheuristic algorithms. Luniver Press
71. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press
72. Yang X-S (2009) Engineering optimisation: an introduction with metaheuristic applications. Wiley
73. Vien Q-T, Le TA, Yang X-S, Duong TQ (2019) Enhancing security of MME handover via fractional programming and firefly algorithm. *IEEE Trans Commun* 67(9):6206–6220
74. Yang X-S (2020) Chapter 13: firefly algorithm: variants and applications. In: *Swarm intelligence algorithms*, pp 175–186. CRC Press
75. Yamanaka T, Higuchi K (2017) Transmitter beamforming control based on firefly algorithm for massive MIMO systems with per-antenna power constraint. In: *23rd Asia-Pacific Conference Communication (APCC)*, pp 1–6
76. Le TA, Yang X-S (2023) Firefly algorithm for beamforming design in RIS-aided communications systems. In: *Proceeding of IEEE vehicular technology conference (VTC 2023-Spring)*, pp 1–5
77. Le TA, Yang X-S (2023) Firefly algorithm for beamforming design in RIS-aided communication systems. In: *2023 IEEE 97th vehicular technology conference (VTC2023-Spring)*, pp 1–5
78. CVX Research Inc (2015) CVX: Matlab software for disciplined convex programming, academic users. <http://cvxr.com/cvx>
79. Perović NS, Tran L-N, Renzo MD, Flanagan MF (2021) Optimization of RIS-aided MIMO systems via the cutoff rate. *IEEE Wireless Commun Lett* 10(8):1692–1696
80. Yu H, Tuan H, Nasir AA, Duong TQ, Poor HV (2020) Joint design of reconfigurable intelligent surfaces and transmit beamforming under proper and improper gaussian signaling. *IEEE JSAC* 38(11):2589–2603
81. Goldsmith A (2005) Wireless communications. Cambridge University Press

Chapter 2

Structural Optimization of Reinforced Concrete Frames with a Modified Flower Pollination Algorithm



Panagiotis E. Mergos and Xin-She Yang

1 Introduction

Reinforced concrete (RC) is the most common structural material in built environment. However, its construction is associated with high economic costs and environmental footprint [1–5]. Hence, designing efficiently RC structures is expected to play a key role in sustainable construction [1–5]. Concrete structures employ a wide range of structural systems. Amongst them, RC frames represent the most frequent one as they combine economy, aesthetics, versatility in configuration and reliability in structural response. Simultaneously, the efficient design of concrete frames remains a challenging task due to the numerous design variables and the complex interactions when calculating both the design objectives and constraints [6].

Therefore, the use of numerical algorithms is promising in material efficient design of concrete frames. Gradient-based optimization algorithms are known to converge faster to optimum solutions. Nevertheless, their application necessitates calculation of gradients that are not readily available in RC frames design [7]. As a result, the use of evolutionary algorithms is gaining more ground in the automation of RC frames design, such as the Genetic Algorithm (GA) [8], Simulated Annealing [9], Particle Swarm Optimization (PSO) [10], Flower Pollination Algorithm [11] and many others. Despite the advantages of evolutionary algorithms when optimizing RC frames, they still lack certainty in tracking global optimum solutions and often demonstrate slow rates of convergence. The latter drawback combined with the high

P. E. Mergos (✉)

Department of Engineering, School of Science and Technology, City, University of London, London EC1V 0HB, UK

e-mail: panagiotis.mergos.1@city.ac.uk

X.-S. Yang

Design Engineering and Mathematics, Middlesex University London, The Burroughs, London, UK

computational costs for the design of complex RC frames turns the structural optimization of real-world RC frames not applicable in standard engineering practice. It is therefore essential that efficient evolutionary algorithms are developed to tackle structural optimization of RC frames.

Numerous research efforts examine the application of evolutionary algorithms to the optimum design of 2D concrete frames (e.g. [12–17]). However, the number of studies optimizing realistic 3D RC frames is still rather limited [18–28]. The latter observation could be attributed to the additional computational cost and complexity of numerical procedures associated to the 3D concrete frames structural design. The first author of this study examined structural optimization of 3D RC frames by using the FPA [27]. It was observed that FPA provides similar quality results to other well-known algorithms when applied to small-scale RC frames, but with slightly slower convergence rates. Furthermore, it outperforms other optimization algorithms when applied to large and complex RC frames due to its high capacity of exploring the search space.

The FPA was originally developed by the second author of this chapter [11] as a metaheuristic algorithm that imitates evolution of flowers. The main advantages of FPA are its high computational efficiency, mostly in terms of exploration potential, and the formulation simplicity. Consequently, FPA has been extensively used in literature to optimize numerous and diverse optimization problems in different scientific fields (e.g. [29–38]). Furthermore, a significant number of hybridized and modified FPA versions have been developed to enhance its performance for various optimization tasks. For example, Yang et al. [39] developed a multi-objective version of FPA. Abdel-Raouf et al. [40] enhanced computational performance of FPA by employing chaotic maps. In their study, Rodrigues et al. [41] formulated a binary variant of FPA to tackle combinatorial optimization tasks. Wang et al. [42] combined FPA with a bee-pollinator to conduct cluster analysis. Tamilselvan and Jayabarathi [43] formulated a multi-objective FPA variant to solve the capacitor placement problem. Abdel-Basset and Hezam [44] developed a hybrid FPA variant using a local search algorithm. Hezam et al. [45] formulated another hybrid FPA with tabu search. Nigdeli et al. [46] used a hybrid FPA with harmony search for optimizing the tuning of mass dampers in structural systems. Zhou et al. [47] formulated a discrete greedy FPA that was applied to the spherical travelling salesman problem. Abdel-Basset et al. [48] used crossover to modify FPA and apply it to the knapsack challenge. In [49], the strategy of island population is employed for mitigating early FPA convergence. In [50], a new FPA version is constructed by generating global orientations of all flowers in the population. Ozsoydan and Baykasoglu [51] enhanced FPA by chaos and intensification. Moreover, Ozsoydan and Baykasoglu [52] developed a species-based FPA with enhanced intensification.

In [53], Mergos and Yang developed an FPA variant, named FPAPA, considering an extra feature of flower pollination in nature called *pollinator attraction* [54, 55]. The latter feature addresses the observation that flowers are evolving to attract pollinators and spread their pollen in the population. Consistently, FPAPA offers fitter flowers an enhanced probability to manage transfer of pollen through biotic

pollination. FPAPA was tested against the benchmark suite of single-objective, real-parameter optimization problems specified in IEEE-CEC'13 [56], and it was found to offer a statistically important improvement in relation to the original version [53].

In the current study, FPAPA is used to the structural optimization of small-scale benchmark RC frames, and its computational performance is evaluated against competitor algorithms. All RC frames are duly designed according to the specifications of the Eurocodes [57–59]. Furthermore, the global optimal solutions of the RC frames are pre-known from exhaustive search. Hence, the efficiency of the algorithms is measured in actual errors against the global optima.

In the following sections, the optimal structural design problem of concrete frames is first explained. Then, the algorithmic formulation of FPAPA is described. Next, a set of benchmark RC frames is introduced with their known global optimal solutions. Finally, FPAPA is applied to optimize the benchmark RC frames and its performance is ranked against other algorithms. Useful conclusions and future recommendations are made.

2 Optimization Problem

The main objective of the present optimization problem is to reduce the materials cost of real-world RC frames in a way that satisfies the structural design requirements set by current guidelines like the Eurocodes [57–59]. More specifically, the optimization problem herein searches for the optimal combination of RC members' cross sections that leads to minimum material economic cost. In this effort, external loads, frame geometry, boundary conditions and material properties are taken as pre-determined. Hence, the design variables vector $\mathbf{x}$ consists of cross sections attributed to different structural members groups in the concrete frames. Moreover, following real-world applications, cross sections are taken from discrete, pre-known lists. The afore-described optimization task is expressed as

$$\begin{aligned}
 &\text{Minimize : } F(\mathbf{x}) \\
 &\text{Subject to : } g_j(\mathbf{x}) \leq 0, j = 1 \text{ to } m, \\
 &\text{where } \mathbf{x} = (x_1, x_2, \dots, x_d) \\
 &x_i \in \mathbf{D}_i = (1, 2, \dots, k_i), i = 1 \text{ to } d.
 \end{aligned} \tag{1}$$

In this equation, $F(\mathbf{x})$ represents the objective function aiming to minimize, and $\mathbf{x}$ stands for the design variables vector that includes d number of variables x_i ($i = 1$ to d), indicating member cross sections. Hence, d coincides with the number of groups with possibly different cross sections in the RC frames. The variables x_i assume values from sets of integer values $\mathbf{D}_i = (1, 2, \dots, k_i)$ indicating the locations of cross sections in their allocated lists, where k_i stands for the length of the corresponding list attributed to group i . Moreover, the solution should satisfy m design constraints $g_j(\mathbf{x}) \leq 0$ ($j = 1$ to m).

The objective function $F(\mathbf{x})$ is calculated from Eq. (2), where V_c (m^3) denotes concrete volume and m_s (kg) denotes the mass of rebars including both the longitudinal and transverse reinforcement. In Eq. (2), f_{co} and f_{so} are material costs of concrete per m^3 and rebars per kg, respectively. The material unit prices adopted in this study are $f_{co} = 100\text{€}/\text{m}^3$ and $f_{so} = 1\text{€}/\text{kg}$.

$$F(\mathbf{x}) = V_c(\mathbf{x}) \cdot f_{co} + m_s(\mathbf{x}) \cdot f_{so}. \quad (2)$$

In Eq. (2), the calculation of $V_c(\mathbf{x})$ is straight-forward from the design vector $\mathbf{x}$ (i.e. member cross sections) and frame geometry. However, for the same $\mathbf{x}$ vector, the steel reinforcement mass $m_s(\mathbf{x})$ is calculated by structural analysis and design procedures. The latter procedures are nonlinear and complex working essentially like black-box functions in the optimization problem.

More particularly, herein, the design rules and specifications of Eurocodes [57–59] are employed to determine $m_s(\mathbf{x})$. To facilitate this task, structural analysis and design is performed with the computer software SAP2000 [60]. A detailed description of the structural analysis and design procedures followed in SAP2000 for concrete frames can be found in [27, 61]. It is also noted that an in-house application in MATLAB [62], called STROLAB (i.e. STRuctural Optimization LABoratory) [27] has been developed to interact with SAP2000 for the purposes of this optimization framework.

Following the afore-described procedures, concrete columns and beams are first designed for axial loads, bending, torsion and shear at different locations along their lengths. The outcomes of structural design are the areas of required longitudinal and transverse reinforcement at the respective member cross sections. These areas are multiplied by their application lengths and then are summed to calculate $V_c(\mathbf{x})$ and finally $m_s(\mathbf{x})$. In this procedure, it is taken for simplicity that the provided rebar areas at the design cross sections match exactly the required ones and that these areas are applied up to the neighbouring member sections with smaller steel reinforcement requirements.

The constraints $g_j(\mathbf{x}) \leq 0$ ($j = 1$ to m) stand for the design checks prescribed by the Eurocodes [57–59] at the Ultimate and Serviceability and Limit States (ULS and SLS). For the ULS design checks for local flexural and shear resistance, a constraint is considered violated when the respective checks are not satisfied by any permitted area of rebars inside the cross sections. The afore-described approach is followed since only the member cross sections act as design variables. Hence, if ρ_w and ρ_l are the required transverse and longitudinal reinforcing steel volumetric ratios and $\rho_{w, \max}$ and $\rho_{l, \max}$ their respective maximum limits according to Eurocodes and construction practice, then the following inequalities should apply:

$$\rho_l - \rho_{l, \max} \leq 0. \quad (3)$$

$$\rho_w - \rho_{w, \max} \leq 0, \quad (4)$$

Moreover, the design constraints at the ULS against diagonal compression failure are considered not to be met when the combination of torsion and shear demands exceeds the maximum strength of diagonal compressive struts in concrete members. The latter constraints are expressed as below, where V_{Ed} and T_{Ed} represent acting shear forces and torsional moments, respectively, and $V_{Rd,max}$ and $T_{Rd,max}$ denote the respective resistances of the diagonal compression struts for an angle $\theta = 45^\circ$.

$$\frac{V_{Ed}}{V_{Rd,max}} + \frac{T_{Ed}}{T_{Rd,max}} - 1 \leq 0. \quad (5)$$

For the Serviceability Limit State (SLS), design checks ensure that displacements and deformations do not exceed permissible values. The corresponding design constraints are written as follows, where Δ represents the displacement/deformation demand for the SLS load combination and Δ_{max} the respective permissible value.

$$\Delta - \Delta_{max} \leq 0. \quad (6)$$

From the previous, it is inferred that design constraints work also as black-box functions since they are calculated from complex structural design and analysis procedures in the RC frames. This enhances further the complexity of the optimization task. To deal with these constraints, in the current study, the penalty function approach is adopted.

3 FPAPA Algorithm

FPA is imitating how flowers are reproduced in nature. This process is so successful that flowers have dominated the landscape of planet earth [63]. Similar to all biological organizations, flowers aim at reproduction. The latter is achieved through pollination. Pollination is either biotic or abiotic [11]. In the former case, the transfer of pollen is achieved via pollinators (i.e. insects and animals). Pollinators can make long flights. Therefore, biotic pollination works like global search [11]. Moreover, pollinator flights resemble Lévy flights [64]. Abiotic pollination entails pollen transfer via the wind or water. The reproduction of grass is a typical example of abiotic pollination [11, 63]. Abiotic pollination, typically, happens locally. Hence, it can be modelled as local search [11]. Another flower pollination feature is the so-called constancy. With constancy, pollinators prefer specific species bypassing others reducing the risks coming from exploring unknown flower species [11].

The previous observations have been idealized in the following basic rules of the original FPA:

1. Biotic pollination is assumed as a global search mechanism where pollinators perform Lévy flights.
2. Abiotic pollination is treated as a local search procedure.

3. The reproduction probability is proportional to the similarity of flowers involved due to flower constancy.
4. The pollination procedure (global vs local) is selected by a probability switch value p that belongs in $[0, 1]$.

According to the 1st and 3rd FPA rules, the biotic pollination mechanism can be idealized as follows:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \gamma \cdot L(\lambda) \cdot (\mathbf{g}^* - \mathbf{x}_i^t), \quad (7)$$

where $\mathbf{x}_i^t$ reflects flower i in the population at iteration t and $\mathbf{g}^*$ stands for the fittest flower in the population, λ is a positive constant that is suggested to be taken as $3/2$ [11], γ represents a step size scaling factor, and $L(\lambda) > 0$ stands for the size of the flight step. More specifically, $L(\lambda)$ follows a Lévy distribution shown below:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\frac{\pi \lambda}{2})}{\pi} \cdot \frac{1}{s^{1+\lambda}}, \quad (s > 0), \quad (8)$$

where $\Gamma(\lambda)$ represents gamma function with $s > 0$. On the other hand, the 2nd and 3rd rules are idealized by Eq. (9), where $\mathbf{x}_j^t$ and $\mathbf{x}_k^t$ represent other flowers in the same population and ε is a random number withdrawn in $[0, 1]$ assuming uniform distribution. Finally, following the 4th rule of the original FPA, the type of flower pollination is determined by comparing a random number in $[0, 1]$ against p . If the random number is smaller than p , the global pollination mechanism is invoked.

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \varepsilon \cdot (\mathbf{x}_j^t - \mathbf{x}_k^t). \quad (9)$$

Clearly, parameters p and γ should be selected so that they maximize the computational performance of FPA. The authors have conducted a parameters tuning study revealing the optimal FPA parameters values for different optimization problems [65]. In that study, it is suggested that γ ranges between 0.1 and 1 and p ranges between 0.2 and 0.6.

Another mechanism of flower pollination, observed in nature and not considered in the original FPA, is that flower species try to evolve so that they attract pollinators and transfer pollen through biotic pollination [54, 55]. This evolution mechanism is observed in many ways. For example, flowers evolve to offer nutritious rewards such as nectar and pollen to entice pollinators (e.g. honeybees). Other flower species develop shapes facilitating access to pollinators (e.g. bowl-shaped flowers) or bright colours attracting pollinators with colour vision (e.g. birds and bees) or even special scents, such as the case of some flowers species relying on night pollinators (e.g. bats) [54, 55]. Perhaps the most characteristic example of pollinator attraction is the case of orchids mimicking female bees in appearance and scent to trick and attract male bees [54, 55]. In all cases, pollinator attraction aims at succeeding pollen transfer and reproduction via biotic pollination.

FPAPA is an FPA variant developed by the authors to account explicitly for the pollinator attraction evolution mechanism [53]. The novel feature of FPAPA is that it increases the chances of fitter flowers to manage pollen transfer through biotic pollination. Consistently, FPAPA assumes that p_i (i.e. switch probability of flower i in the population) varies in accordance with the flower rank within the population. For simplicity, a linear variation of p_i between two limit values p_L and p_U in $[0, 1]$ is assumed, where p_L corresponds to the least fit flower and p_U to the fittest flower. Therefore, if f_{sh} denotes the sorted (i.e. assuming descending order), vector of objective functions in the previous population and ξ_i represents the position of flower i in f_{sh} , then p_i in the new population is given by Eq. (10), where n denotes the size of the flowers population.

$$p_i = \frac{1}{n-1} [(p_U - p_L) \cdot \xi_i + np_L - p_U]. \quad (10)$$

It is clarified that in Eq. (10), it should hold $p_U > p_L$ (i.e. maximum probability to conduct biotic pollination for the fittest flower) for consistency with the pollinator attraction rule. Nevertheless, FPAPA works with any p_L and p_U values. As a matter of fact, in the special case where $p_U = p_L$, the proposed variant coincides with the original version of FPA.

Figure 1 illustrates the flowchart of FPAPA. It is observed that the variant follows closely the original FPA apart from the calculation of different switch probabilities for the flowers of the population based on the feature of pollinator attraction. Finally, it is noted that the FPAPA version applied herein rounds the design solutions to their closest integer values to address the discrete nature of the optimization problem under investigation.

4 Benchmark RC Frames

Six benchmark RC frames are considered herein originally developed by the first author in [66]. The benchmarks are named by the letter B followed by a number. The geometric configurations of the frames are illustrated in Fig. 2 and the configurations of the concrete sections are shown in Fig. 3. The benchmark frames have up to three spans and between two and six storeys. Span lengths are equal and either 5 or 6 m. The heights of storeys are all 3 m. Concrete grades are from C20/25 to C30/37 as specified in Eurocode 2 (EC2) [57]. Steel reinforcement grade is B500. Cover of concrete to the centroid longitudinal rebars is 50 mm. The frames belong to RC buildings that are designed against dead, live and either earthquake or wind loads of different magnitudes. The RC frames are designed to EC2 [57] and EC8 [58] for the Low Ductility Class (DCL) [58]. More details about the structural configuration, material properties, design loads and modelling assumptions of the benchmark RC frames can be found in [66].

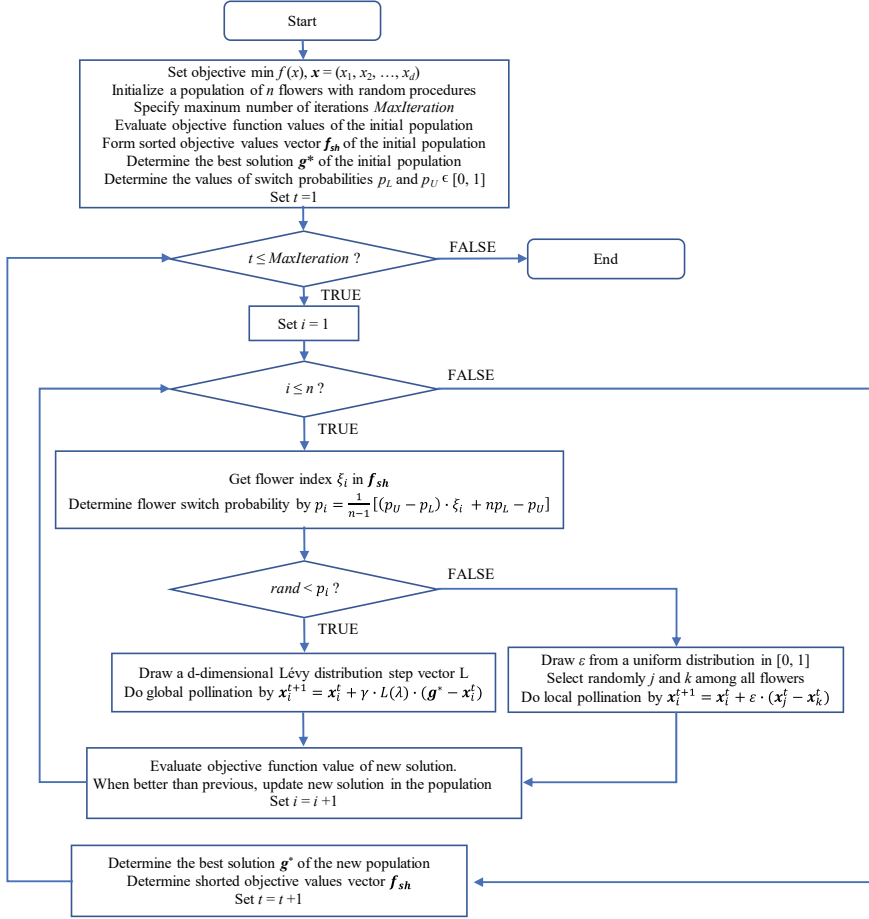


Fig. 1 FPAPA flow chart

The concrete frames are divided into groups of column and beam members of the same cross sections. For each group, cross sections come from pre-determined lists. Therefore, the total number Ω of potential design solutions (i.e. search space) is calculated from the number of member groups and the size of the allocated lists of cross sections. Finally, the minimum material costs F_{best} are derived from previous exhaustive search analyses. Therefore, it is guaranteed that they represent the global optima of the concrete frames. Table 1 presents the number of design variables, size of search space and minimum material costs of the six RC frames. More information regarding the arrangement of design variables, member groups and available lists of cross sections can be found in [66].

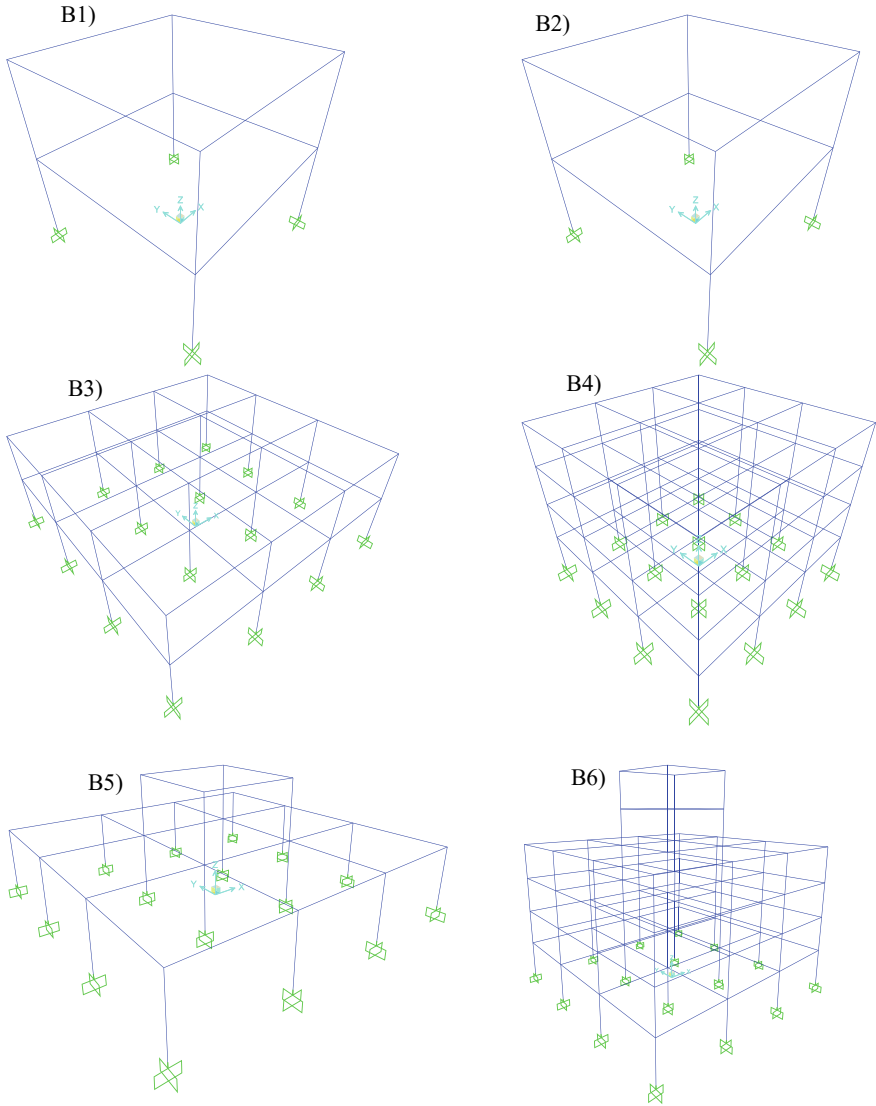


Fig. 2 Benchmark RC frames [66]

As discussed in the previous sections, the objective function $F(\mathbf{x})$ values (i.e. material costs of the RC frames) for each candidate design vector $\mathbf{x}$ should be determined by following a sequence of structural analysis and design procedures for the RC frames that collectively resemble black-box functions. To facilitate future optimization studies on the same benchmark problems, the objective function values of all designs are given in a supplementary file of [66] following an exhaustive search analysis by the first author of this chapter.

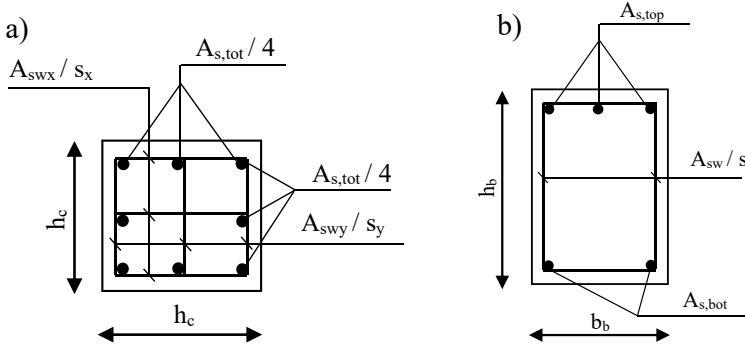


Fig. 3 Member cross sections and configuration of steel reinforcement **a** columns, **b** beams

Table 1 Benchmark RC frames optimization problem properties [66]

Benchmark frame	d	Ω	F_{best} (Euros)
B1	2	10^2	1350.8
B2	3	10^3	1476.5
B3	4	10^4	10,918
B4	6	7^6	13,553
B5	5	9^5	4964.3
B6	7	6^7	30,060

5 Performance of FPAPA Algorithm

Herein, the performance of FPAPA in optimizing the concrete frames discussed above is evaluated. The performance is measured in errors (%) from F_{best} , given in Table 1. In the following analyses with FPAPA, switch probability values of $p_L = 0.2$ and $p_U = 0.8$ are assumed. Furthermore, a population size of $n = 10$ individuals is used for B1, B2 and of $n = 20$ for the other frames. Twenty independent trial runs are used for each frame to account for the stochastic nature of FPAPA.

Figure 4 presents the mean optimization histories of FPAPA for all benchmark RC frames and for three different step sizes (i.e. $\gamma = 0.1, 1.0$ and 10.0). It is first observed that all average histories converge practically to zero within a small number of evaluations. This is important since function evaluations are computationally expensive in RC frames design. As expected, optimization histories converge more rapidly for B1, B2 and less quickly for benchmarks B4-B6 due to the larger size of the latter search spaces. Nevertheless, in all cases, the optimization histories reach 2% errors in less than $70 \cdot d$ function evaluations, where d stands for the total number of design variables. Figure 4 shows also that the step size γ does not greatly affect the convergence response. A step size of $\gamma = 1.0$ seems to offer the best response for the most challenging benchmark problem B6 and therefore is used in the rest of this study.

By using $\gamma = 1.0$, FPAPA reaches on average 2% errors within only 40- d function evaluations for all benchmark problems.

Furthermore, Figs. 5, 6 and 7 compare the response of the FPAPA algorithm against the original FPA assuming two different p values (i.e. $p = 0.8$ and $p = 0.2$) that match the values of p_L and p_U assumed for FPAPA in this study. It is recalled that the FPAPA has been found to offer statistically significant improvements with respect to the original FPA when tested against the set of IEEE-CEC'13 [56] challenging and complex optimization problems [53].

Figure 5 compares the average convergence histories coming from 20 independent runs for each algorithm and for all benchmarks. It is noted that all algorithmic options offer similar convergence efficiency. This could be attributed to the rather simple

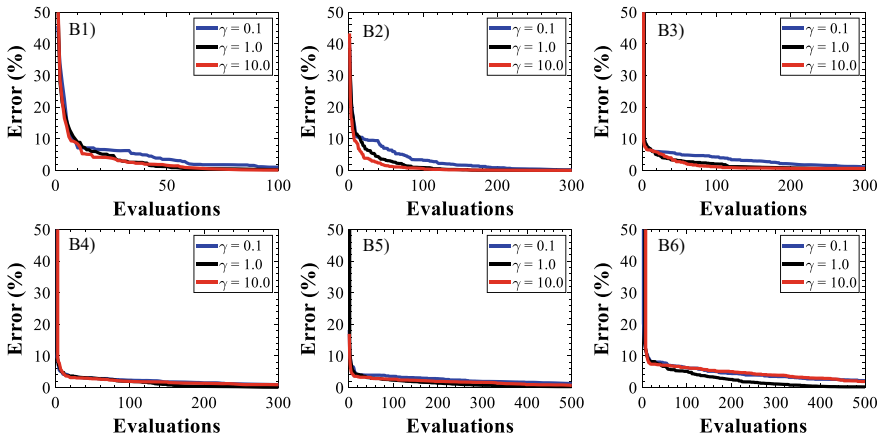


Fig. 4 FPAPA average optimization histories with different step sizes

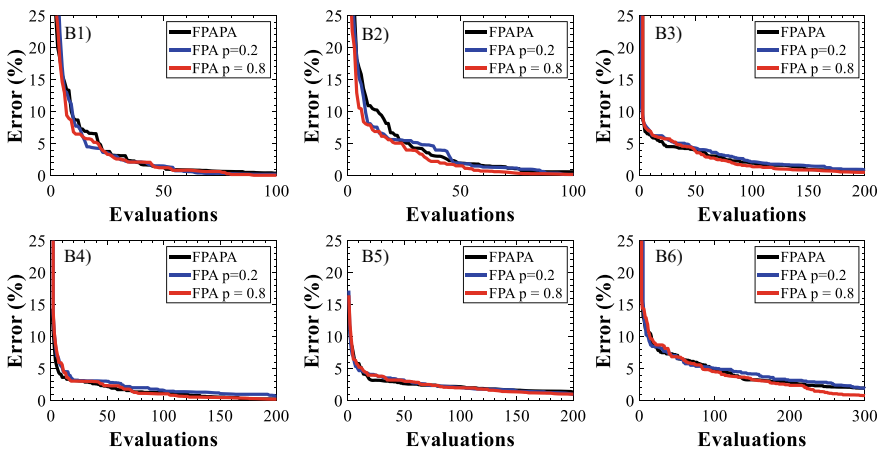


Fig. 5 Average optimization histories of FPAPA and the original FPA with $p = 0.2$ and $p = 0.8$

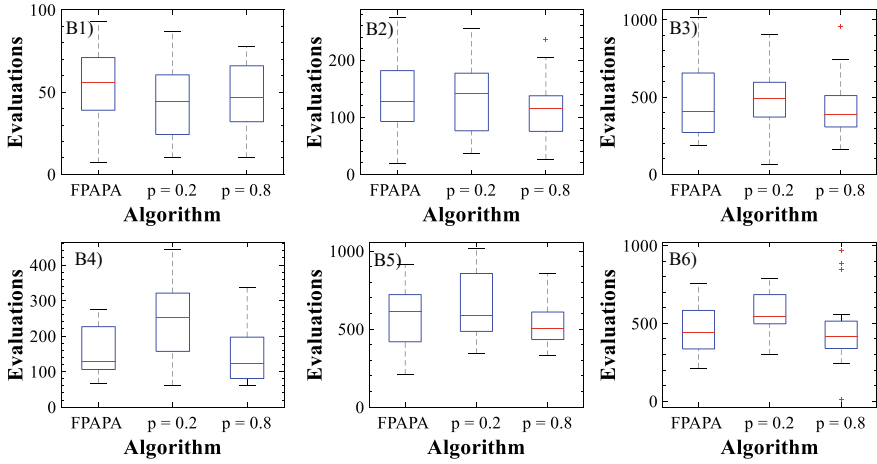


Fig. 6 Number of function evaluations till reaching global optimum of FPAPA and the original FPA with $p = 0.2$ and $p = 0.8$

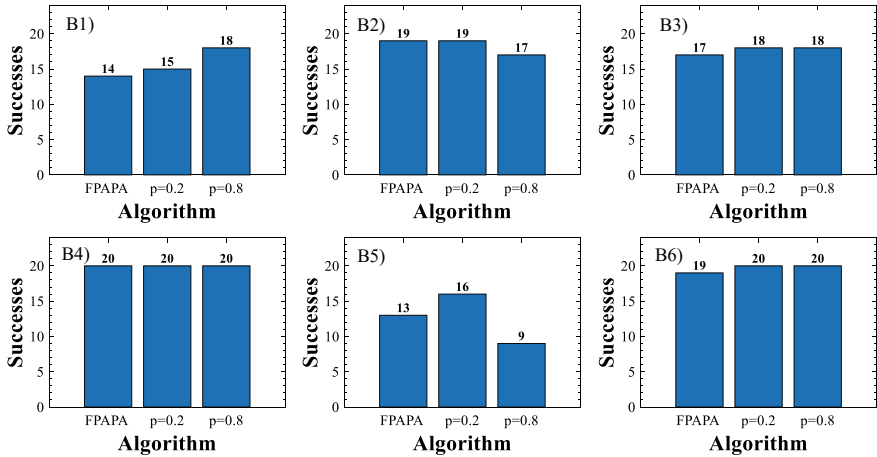


Fig. 7 Number of runs reaching global optimum (i.e. successes) out of 20 of FPAPA and the original FPA with $p = 0.2$ and $p = 0.8$

landscape of the examined optimization problems for which the switch probability values of FPA seem to not influence significantly computational performance.

Furthermore, Fig. 6 shows the box plots of function evaluations required to converge (i.e. success) to the global optima obtained by the different runs of the FPA variants. Successful runs are only shown in this figure. A run is treated as successful when the algorithm tracks the global best solution in less than 1000 function evaluations for all benchmarks problems apart from B1, where only 100 function evaluations are used. The limit of 1000 evaluations is generally considered as a reasonable

computational budget to afford for optimizing an RC frame in everyday practice. For B1, the limit of 100 function evaluations matches the size of the search space. Like in Fig. 5, very small differences amongst the different algorithms are noticed in Fig. 6. The total number of successes, out of 20, of all FPA variants and for all benchmark problems are presented in Fig. 7. It is observed that marginally the FPA with $p = 0.2$ has a higher number of successes within the affordable computational budget. This could be partly explained by the increase in exploitation capacity of the FPA when the switch probability value decreases, but the differences are rather small to make solid conclusions.

Moreover, Figs. 8, 9 and 10 compare the performance of FPAPA against some of the most established evolutionary algorithms in literature like the GA [8], SA [9] and PSO [10]. GA simulates Darwin's evolution theory. SA resembles the annealing process in metallurgy to evolve a candidate design solution. Finally, PSO is a famous evolutionary algorithm that takes inspiration from fish schools and bird flocks. It is noted that the SA, PSO and GA are formulated and implemented as described in [67]. Furthermore, the GA and PSO assume the same population sizes as FPAPA for the different benchmark problems herein. The rest of the algorithm parameters for GA, SA and PSO assume the default values recommended in [67] that seem to maximize their computational performance.

Figure 8 demonstrates the average optimization histories of the various algorithms out of 20 independent runs. It is seen that the PSO and SA tend to converge quicker than other algorithms for the small-scale optimization problems B1-B3, but they become slower for the larger-scale problems B4-B6. The FPAPA algorithm exhibits similar convergence rates to GA for all benchmarks apart from B6, where the latter algorithm converges more rapidly. Furthermore, Figs. 9 and 10 show the numbers of function evaluations required for tracking global optimum solutions (i.e.

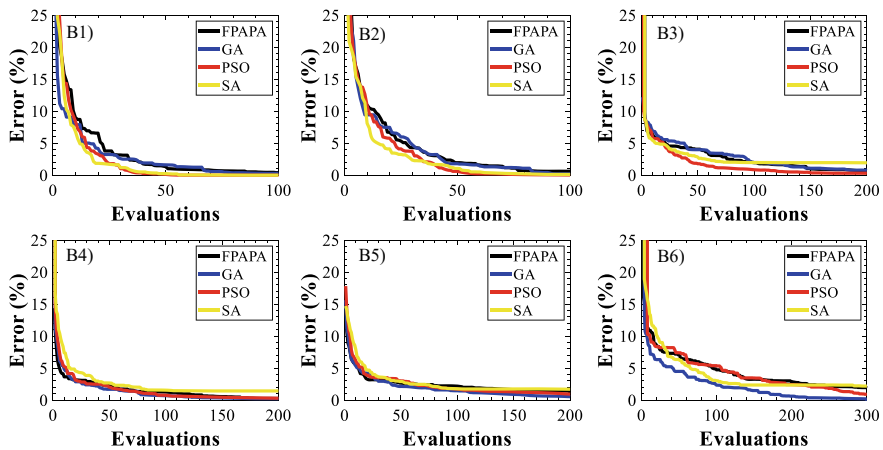


Fig. 8 Average optimization histories of FPAPA and other algorithms

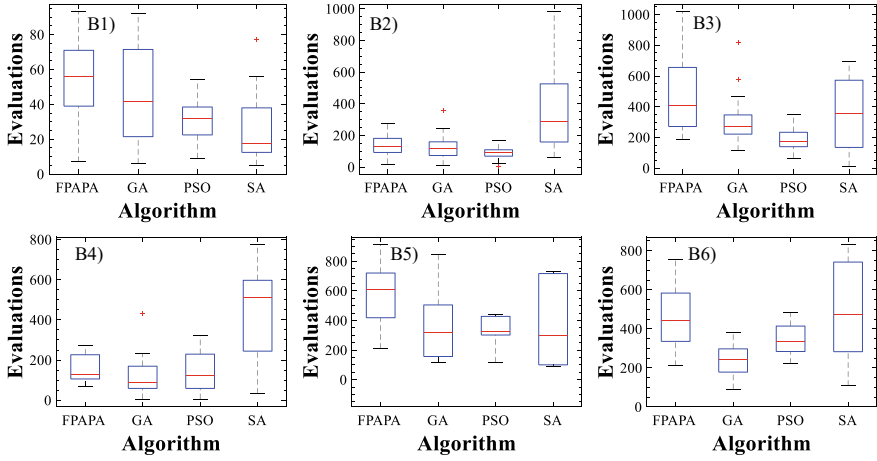


Fig. 9 Number of function evaluations till reaching global optimum of FPAPA and other algorithms

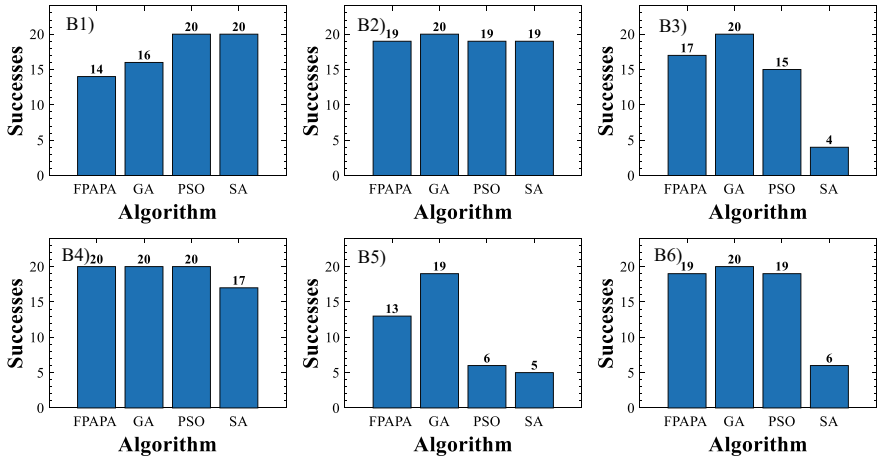


Fig. 10 Number of runs reaching global optimum (i.e. successes) out of 20 of FPAPA and other algorithms

success) by all algorithms as well as the number of successes out of the 20 independent runs within the afore-described computational budget. It is observed that the PSO requires the smallest or second smallest numbers of function evaluations for tracking the global optima in Fig. 9. This observation hints at the high exploitation ability of this algorithm. Nevertheless, in Fig. 10, it is observed that the FPAPA and GA algorithms are more successful at tracking F_{best} than the SA and PSO algorithms for all benchmark problems apart from B1. This indicates that the former algorithms are more capable of exploring the search space than the latter. The high capability of

the FPA algorithm and its variants to explore complex search landscapes is a common observation in many similar studies [e.g. 11, 27, 65].

6 Conclusions

Optimum structural design of RC frames represents an important challenge in the road of de-carbonising the built environment. Due to the complex nature of this design problem, computationally efficient optimization algorithms are required. FPA is one of the most competitive metaheuristic algorithms imitating the successful reproduction process of flowers on earth. In this study, a novel FPA variant is examined, named FPAPA, that accounts for a new evolutionary feature in the natural pollination process called Pollinator Attraction. This evolutionary characteristic describes the fact that flowers evolve to become more attractive to pollinators and thereby achieve reproduction by biotic pollination. Consistently, FPAPA provides fitter flowers with a higher probability of conducting biotic pollination that is a global search mechanism. FPAPA has been found to offer statistically significant improvements to the original FPA when applied to complex and multimodal objective functions.

In the present study, FPAPA is applied to the structural optimization of concrete frames. More specifically, the performance of FPAPA is tested against a set of six benchmark structural optimization problems of concrete frames with known global optimum solutions. It is found that FPAPA manages to track near global optima within a computational budget that could be easily afforded in standard engineering practice. In this category of optimization tasks, FPAPA seems to offer similar computational efficiency to the original FPA algorithm. Furthermore, comparisons with other well-established optimization algorithms, show that FPAPA offers competitive convergence rates and more robust performance at tracking global optimum solutions.

Closing this chapter, it is recommended that further studies investigate the efficiency of FPAPA and other alternative algorithms in the structural optimization of large-scale and complex real-world RC frames with numerous design variables.

References

1. IEA (2020) Energy technology perspectives 2020. International Energy Agency
2. IPCC (2021) Climate change 2021: the physical science basis. Cambridge University Press, Cambridge, UK and New York, USA, Intergovernmental Panel on Climate Change
3. Lagaros ND (2020) The environmental and economic impact of structural optimization. *Struct Multidiscip O* 49:1047–1066
4. Mergos PE (2018) Contribution to sustainable seismic design of reinforced concrete members through embodied CO₂ emissions optimization. *Struct Concrete* 19:454–462
5. Mergos PE (2018) Seismic design of reinforced concrete frames for minimum embodied CO₂ emissions. *Energ Build* 162:177–186

6. Sarma KC, Adeli H (1997) Cost optimization of concrete structures. *J Struct Eng* 124:570–578
7. Lagaros ND (2014) A general purpose real-world structural design optimization computing platform. *Struct Multidiscip O* 49:1047–1066
8. Holland JH (1975) *Adaptation in natural and artificial systems. An introductory analysis with application to biology, control and artificial intelligence.* University of Michigan Press, Ann Arbor, MI
9. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220:671–680
10. Kennedy J (2001) Particle swarm optimization. In: *Encyclopedia of machine learning*, Springer, pp 760–766
11. Yang XS (2012) Flower pollination algorithm for global optimization. *Unconven Comput Nat Comput* 7445:240–249
12. Paya-Zaforteza I, Yepes V, Gonzalez-Vidoso F (2008) Hospitaler A. Multiobjective optimization of concrete frames by simulated annealing. *Comput-Aided Civ Inf* 23:596–610
13. Kaveh A, Sabzi O (2011) A comparative study of two metaheuristic algorithms for optimal design of planar RC frames. *Int J Civ Eng* 9:193–206
14. Akin A, Saka MP (2015) Harmony search algorithm based optimum detailed design of reinforced concrete plane frames subject to ACI 318–05 provisions. *Comput Struct* 147:75–95
15. Mergos PE (2018) Efficient optimum seismic design of reinforced concrete frames with nonlinear structural analysis procedures. *Struct Multidiscip O* 58:2565–2581
16. Mergos PE (2020) Minimum cost performance-based seismic design of reinforced concrete frames with pushover and nonlinear response-history analysis. *Struct Concrete* 21:599–609
17. Kaveh A, Izadifard RA, Mottaghi L (2020) Optimal design of planar RC frames considering CO₂ emissions using ECBO, EVPS and PSO metaheuristic algorithms. *J Build Eng* 28:101014
18. Fadaee MJ, Grierson DE (1996) Design optimization of 3D reinforced concrete structures. *Struct. Optimization* 12:127–134
19. Balling RJ, Yao X (1997) Optimization of reinforced concrete frames. *J Struct Eng ASCE* 123:193–202
20. Sahab MG, Ashour AF, Toropov VV (2005) Cost optimization of reinforced concrete flat slab buildings. *Eng Struct* 27:313–322
21. Govindaraj V, Ramasany JV (2007) Optimum detailed design of reinforced concrete frames using genetic algorithms. *Eng Optimiz* 39:471–494
22. Sharafi P, Muhammad NS, Hadi M (2012) Heuristic approach for optimum cost and layout design of 3D reinforced concrete frames. *J Struct Eng* 138:853–863
23. Kaveh A, Behnam AF (2013) Design optimization of reinforced concrete 3D structures considering frequency constraints via a charged system search. *Sci Ira Trans A* 20:387–396
24. Esfandiari MJ, Urgessa GS, Sheikholarefin S, Dehghan Manshadi SH (2018) Optimum design of 3D reinforced concrete frames using the DMPSO algorithm. *Adv Eng Softw* 115:149–160
25. Dehnavipour H, Mehrabani M, Fakhriyat A, Jakubczyk-Galczyńska A (2019) Optimization-based design of 3D reinforced concrete structures. *J Soft Comput Civil Eng* 3:95–106
26. Martins A, Simões L, Negrão J, Lopes A (2020) Sensitivity analysis and optimum design of reinforced concrete frames according to Eurocode 2. *Eng Optimiz* 52:2011–2032
27. Mergos PE (2021) Optimum design of 3D reinforced concrete building frames with the flower pollination algorithm. *J Build Eng* 44:102935
28. Mergos PE (2022) Surrogate-based optimum design of 3D reinforced concrete building frames to Eurocodes. *Developm Built Environ* 11:100079
29. Bekdas G, Nigdeli SM, Yang XS (2015) Sizing optimization of truss structures using flower pollination algorithm. *Appl Soft Comput* 37:322–331
30. Abdelaziz A, Ali E, Elazim SA (2016) Combined economic and emission dispatch solution using flower pollination algorithm. *Int J Electr Power Energy Syst* 80:264–274
31. Abdelaziz A, Ali E, Elazim SA (2016) Implementation of flower pollination algorithm for solving economic load dispatch and combined economic emission dispatch problems in power systems. *Energy* 101:506–518

32. Singh U, Salgotra R (2016) Synthesis of linear antenna array using flower pollination algorithm, *Neural Comput Appl*, 1–11
33. Singh U, Salgotra R (2016) Synthesis of linear antenna array using flower pollination algorithm, *Neural Comput Appl* 1–11
34. Nigdeli SM, Bekdaş G, Yang XS (2016) Application of the flower pollination algorithm in structural engineering. In: *Metaheuristics and optimization in civil engineering*, Springer, pp 25–42
35. Abdel-Raouf O, El-Henawy I, Abdel-Basset M (2014) A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles. *Int J Mod Educ Comput Sci* 6:38
36. Heng J, Wang C, Zhao X, Xiao L (2016) Research and application based on adaptive boosting strategy and modified CGFPA algorithm: a case study for wind speed forecasting. *Sustainability* 8:235
37. Zhou Y, Zhang S, Luo Q, Wen C (2016) Using flower pollination algorithm and atomic potential function for shape matching. *Neural Comput Appl* 29:21–40
38. Abdel-Basset M, Shawky LA (2018) Flower pollination algorithm: a comprehensive review. *Artif Intell Rev* 52:2533–2557
39. Yang XS, Karamanoglu M, He X (2014) Flower pollination algorithm: a novel approach for multi-objective optimization. *Eng Optim* 46:1222–1237
40. Abdel-Raouf O, Abdel-Baset M et al (2014) A new hybrid flower pollination algorithm for solving constrained global optimization problems. *Int J Appl Oper Res-An Open Access J* 4:1–13
41. Rodrigues D, Yang XS, De Souza AN, Papa JP (2015) Binary flower pollination algorithm and its application to feature selection. In: *Recent advances in swarm intelligence and evolutionary computation*, Springer, pp 85–100
42. Wang R, Zhou Y, Qiao S, Huang K (2016) Flower pollination algorithm with bee pollinator for cluster analysis. *Inf Proc Lett* 116:1–14
43. Tamilselvan V, Jayabarathi T (2016) Multi-objective flower pollination algorithm for solving capacitor placement in radial distribution system using data structure load flow analysis. *Arch Electr Eng* 65:203–220
44. Abdel-Basset M, Hezam I (2016) A hybrid flower pollination algorithm for engineering optimization problems. *Int J Comput Appl* 140:10–23
45. Hezam IM, Abdel-Baset M, Hassan B (2016) A hybrid flower pollination algorithm with tabu search for unconstrained optimization problems. *Inf Sci Lett* 5:29–34
46. Nigdeli SM, Bekdaş G, Yang XS (2017) Optimum tuning of mass dampers by using a hybrid method using harmony search and flower pollination algorithm. In: *International conference on harmony search algorithm*, Springer, pp 222–231
47. Zhou Y, Wang R, Zhao C, Luo Q, Metwally MA (2017) Discrete greedy flower pollination algorithm for spherical traveling salesman problem. *Neural Comput Appl* 31:2155–2170
48. Abdel-Basset M, El-Shahat D, El-Henawy I, Sangaiah AK (2018) A modified flower pollination algorithm for the multidimensional knapsack problem: human-centric decision making. *Soft Comput* 22:4221–4239
49. Al-Betar MA, Awadallah MA, Doush IA, Hammouri AI, Mafarja M, Alyasseri ZAA (2019) Island flower pollination algorithm for global optimization. *J Supercomput* 75:5280–5532
50. Fouad A, Gao X-Z (2019) A novel modified flower pollination algorithm for global optimization. *Neural Comput Appl* 31:3875–3908
51. Ozsoydan FB, Baykasoglu A (2021) Chaos and intensification enhanced flower pollination algorithm to solve mechanical design and unconstrained function optimization problems. *Expert Syst Appl* 225:107125
52. Ozsoydan FB, Baykasoglu A (2021) A species-based flower pollination algorithm with increased selection pressure in abiotic local pollination and enhanced intensification. *Knowl Based Syst* 225:107125
53. Mergos PE, Yang XS (2023) Flower pollination algorithm with pollinator attraction. *Evol Intell* 16:873–889

54. Glover BJ (2011) Pollinator attraction: the importance of looking good and smelling nice. *Curr Biol* 21:R307–R309
55. Wikipedia contributors, Flower, Wikipedia (2018) The free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Flower&oldid=861525932>
56. Liang J, Qu B, Suganthan P, Hernandez-Daz AG (2013) Problem definitions and evaluation criteria for the CEC2013 special session on real-parameter optimization. Zhengzhou University, Zhengzhou China and Nanyang Technological University, Singapore, Technical Report, Computational Intelligence Laboratory, p 20121
57. CEN (2000) Eurocode 2: Design of concrete structures. Part 1–1: general rules and rules for buildings, Brussels, European Standard EN 1992-1-1
58. CEN (2004) Eurocode 8: design of structures for earthquake resistance. Part 1: general rules, seismic actions and rules for buildings, Brussels,; European Standard EN 1990-1
59. CEN (2002) Eurocode 0: Basis of structural design. Brussels, European Standard EN 1992:2002
60. CSI (2022) <https://www.csiamerica.com/products/sap2000>
61. CSI (2016) Concrete frame design manual: Eurocode 2–2004 with Eurocode 8–2004 for SAP2000. ISO SAP091415M29 Rev. 1, USA
62. MathWorks (2022) https://uk.mathworks.com/products/matlab.html?s_tid=hp_products_matlab
63. Glover BJ (2007) Understanding flowers and flowering: an integrated approach. Oxford University Press, UK
64. Pavlyukevich I (2007) Lévy flights, non-local search and simulated annealing. *J Comput Phys* 226:1830–1844
65. Mergos PE, Yang XS (2021) Flower pollination algorithm parameters tuning. *Soft Comput* 25:14429–14447
66. Mergos PE (2024) Benchmark problems in optimum structural design of 3D reinforced concrete frames. *J Build Eng* 91:109554
67. MathWorks (2022b) MATLAB R2022b—global optimization toolbox. Natick, MA, USA

Chapter 3

Swarm Intelligence Algorithms and Their Engineering Applications



Adam Slowik[✉], Krzysztof Cpalka[✉], Absalom Ezugwu[✉],
and Ali Wagdy Mohamed[✉]

1 Introduction

Faced with increasing technological, economic and social challenges, modern engineering is placing particular attention on the optimization process as a key element in improving a variety of systems, products and processes. Engineering optimization is a comprehensive set of methods, tools and strategies that aim to maximize efficiency and minimize costs. The optimization process in engineering involves analyzing, modeling, simulating and improving systems, keeping in mind a variety of criteria such as performance, durability, safety or cost. The basis of optimization is the search for the best solutions in the context of specific constraints and requirements. The origins of engineering optimization can be traced back to the development of applied mathematics and numerical methods, but modern approaches to the subject also include advanced technologies such as artificial intelligence, nature-based optimization algorithms including evolutionary algorithms (EAs) and SIAs. These modern tools allow engineers to analyze large amounts of data, create advanced sim-

A. Slowik (✉)

Department of Electronics and Computer Science, Koszalin University of Technology,
ul. Sniadeckich 2, 75-452 Koszalin, Poland
e-mail: adam.slowik@tu.koszalin.pl

K. Cpalka

Department of Intelligent Computer Systems, Czestochowa University of Technology,
Al. Armii Krajowej 36, 42-202 Czestochowa, Poland
e-mail: krzysztof.cpalka@pcz.pl

A. Ezugwu

School of Computer Science and Information Systems, North-West University,
11 Hoffman Street Potchefstroom 2531, Potchefstroom, North West, South Africa
e-mail: absalom.ezugwu@nwu.ac.za

A. W. Mohamed

Department of Operations Research and Management, Cairo University,
5 Ahmed Zewail Street, Giza 12613, Egypt

ulation models and rapidly explore the solution space. Engineering optimization is used in a wide range of fields, for example in the design of building structures, manufacturing, logistics and the development of modern technologies. For example, in manufacturing, the consumption of materials is minimized, while in logistics, transport routes and supply chain management are improved. Furthermore, engineering optimization is important in the field of energy, where it enables the efficient design of energy networks, the optimized use of renewable sources and the minimization of energy losses. In the field of computer science, algorithms are optimized to reduce the running time of systems, and in electronics, the performance of electronic circuits is improved. As mentioned earlier, one of the tools used in engineering optimization are optimization algorithms based on nature. Among these, we can mention techniques such as evolutionary algorithms [1–7] and swarm intelligence algorithms [8, 9].

EAs are an exciting area in the field of artificial intelligence, taking inspiration from evolutionary processes observed in the natural world. Their development and application in a variety of fields has made them an integral tool in solving complex optimization problems, and searching solution spaces for the best possible results. Taking inspiration from the process of natural selection, evolutionary algorithms introduce an innovative approach to designing, improving systems and finding optimal solutions. In their basic conception, EAs are based on the process of biological evolution, where a population of individuals undergoes successive iterations, as a result of which it evolves and adapts to its environment. Similarly, in the context of EAs, solutions are represented by individuals in a population that are evolved through genetic operations such as a crossover and a mutation. The operating principle is based on creating a diversity of a population of solutions (individuals) and evaluation of their fitness for the given problem, and then selection, crossover and mutation of the best adapted individuals to produce a new generation. In this way, successively better solutions are obtained.

In the case of swarm intelligence algorithms, their main inspiration comes from the social behavior of organisms in the natural environment. These algorithms are also an exciting branch of artificial intelligence that has attracted the attention of researchers and engineers because of their effectiveness in solving complex optimization problems. Based on the principle of collective behavior of individuals, these algorithms simulate the interactions between agents in a group, leading to an efficient search through the solution space. The idea of a swarm intelligence in a computer science has gained popularity, inspired by observations of social organizations in nature, such as swarms of bees or schools of fish. These observations were the source of the concept of a collective optimization, where individuals in a group operate locally, but their cooperation leads to global, positive results. The basic idea of swarm intelligence algorithms is to model interaction processes between agents that move through the solution space. Typical elements of these algorithms are the agents' movement rules, communication mechanisms, and ways of responding to changes in the environment. One of the key application areas for swarm intelligence algorithms is the optimization of complex problems, where traditional methods often encounter difficulties. Thanks to their ability to simultaneously search through multiple solutions and dynamically

adapt to changing conditions, these algorithms find applications in a logistics, a route planning, a resource optimization or a structure design, etc.

This chapter focuses precisely on swarm intelligence algorithms and their engineering applications. The chapter is structured as follows: Sect. 2 presents a general scheme of nature-based optimization methods including a detailed description of one of the basic swarm intelligence algorithms which is the particle swarm optimization (PSO) algorithm; Sect. 3 analyses the popularity of selected methods and their applications in selected engineering fields; Sect. 4 presents selected practical applications of the previously indicated algorithms in selected engineering fields; Sect. 5 presents a summary of this chapter.

2 Nature-Based Optimization Methods

2.1 General Idea

The general idea of nature-based optimization methods is presented in Fig. 1.

Figure 1 consists of nine steps as follows. In step one, all the parameters of the algorithm are determined. In step two, the objective function is defined. In step

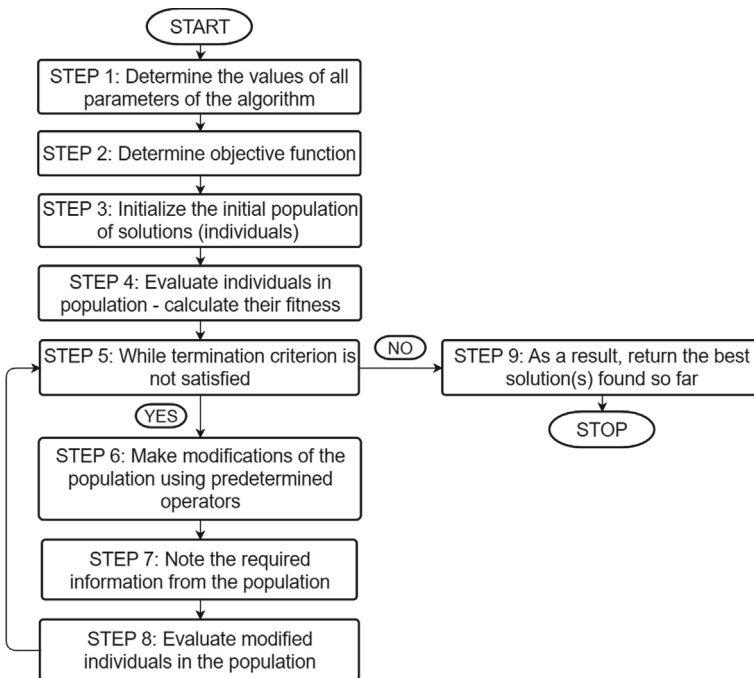


Fig. 1 General idea of nature-based optimization methods

three, the population of initial solutions, often known as individuals, is initialized. Step four consists of the evaluation of the individuals (solutions) in the population. Their fitness value for the objective function is calculated here. In step five, the main loop of the algorithm begins. It is executed until the stopping conditions of the algorithm are met. In step six, population modification operations are performed using predefined operators (e.g., crossover and mutation in genetic algorithms). In step seven, all necessary information is saved (e.g., saving the best individual found so far). In step eight, an evaluation of the modified individuals is performed using the objective function. In step nine, the best solution found so far during the execution of the algorithm is output as the result of the algorithm.

2.2 PSO Algorithm

The PSO algorithm consists of the following steps. In step one, a D -dimensional objective function $OF(.)$ is defined. In step two, the variability ranges $P_{i,j}^{\min}$ and $P_{i,j}^{\max}$ for the j -th variable are assigned. In step three, all the parameters of the PSO algorithm such as the learning rates c_1 and c_2 and the number of particles (solutions) N in the swarm are set. In step four, a P population of N particles (each particle is a D -dimensional vector) is randomly created. In step five, a D -dimensional vector $Gbest$ representing the best solution found so far by the algorithm is created. In steps six and seven, for each i -th particle P_i of the population P , vectors are created: the best particle found so far $Pbest_i$ (step 6) and the velocity vector V_i (step 7). In step eight, each particle P_i from the population P is evaluated using the objective function $OF(.)$. In step nine, the best particle P_i from the population P is assigned to the variable $Gbest$. In step ten, the main loop of the algorithm starts, which is executed until its stop condition is met (when stop condition is met then the algorithm is jump to step 19). Then, for each particle P_i from the population P and for each of its dimensions j , operations such as: updating its velocity vector $V_{i,j}$ (step 11) using the relation presented in Eq. 1; updating the values stored in particle P_i for dimension j (step 12) using the relation shown in Eq. 2; checking for each j th dimension of particle P_i (step 13) whether its value is within the range of variability (assumed in step two)—see Eqs. 3, and 4.

$$V_{i,j} = V_{i,j} + c_1 \cdot r_1 \cdot (Pbest_{i,j} - P_{i,j}) + c_2 \cdot r_2 \cdot (Gbest_j - P_{i,j}) \quad (1)$$

$$P_{i,j} = P_{i,j} + V_{i,j} \quad (2)$$

$$P_{i,j} = P_j^{\min} \text{ when } P_{i,j} < P_j^{\min} \quad (3)$$

$$P_{i,j} = P_j^{\max} \text{ when } P_{i,j} > P_j^{\max} \quad (4)$$

In step 14, each particle P_i is evaluated using the objective function $OF(.)$. In step 15, if the value of $OF(P_i)$ is better than the value of $OF(Pbest_i)$ then the particle

P_i is assigned to the particle P_{best_i} . In step 16, the best particle is selected from the population P and then it is assigned to the particle T . In step 17, if the value of $OF(T)$ is better than the value of $OF(G_{best})$ then the particle T is assigned to the particle G_{best} . In step 18 the algorithm jump to the step 10. In step 19 the particle G_{best} is returned as the result of the algorithm.

2.3 Other Swarm Intelligence Algorithms

Of course, besides the PSO algorithm, which is one of the fundamental swarm intelligence algorithms, there are many other algorithms. One of the also fundamental swarm algorithms is the Ant Colony Optimization (ACO) algorithm [12]. This algorithm takes its inspiration from the natural behavior of an ant colony in finding the shortest path to a food source. It was introduced by Marco Dorigo. In nature, ants move between the food source and the nest, leaving pheromones along their route. Pheromones are chemicals that, when left by one ant, influence the behavior of others. The more intense the pheromones along a particular route, the greater the chance that other ants will choose that route. In this way, the ant colony can coordinate its activities and find the most efficient route to the food source. The ACO uses this mechanism in the context of combinatorial optimization. The algorithm starts with a random distribution of artificial ants in the search space. Each ant moves from point to point, based on the pheromone rate on the route. As time passes, the pheromones evaporate, simulating the natural degradation of pheromones in the environment. However, if the ant has found a better solution, it increases the amount of pheromones on the route it has taken. Initially, the ACO algorithm may explore a larger space of potential solutions, but over time, it focuses on better and better solutions due to the accumulated pheromones. The process is iterative, with the ant colony progressively moving toward finding the optimal solution.

Besides the two most popular swarm intelligence algorithms such as the PSO and the ACO, there are also algorithms such as the Bat Algorithm (BA) [13], the Firefly Algorithm (FA) [14], the Cuckoo Search (CS) [15], the Social Spider Optimization (SSO) [16] and many others. Some part of these algorithms is presented in the Sect. 3.

3 Popularity of Swarm Intelligence Algorithms in Engineering

In order to check the popularity of swarm algorithms, it was checked how many papers using them are indexed in the various scientific databases. The algorithms selected for checking are presented in Table 1.

The considered scientific databases are: Springer, Google Scholar, Science Direct, IEEE Explore and Web of Science. The results obtained are presented in Table 2.

Table 1 Swarm intelligence algorithms selected from literature

Algorithm full name	Shortcut	References	Author(s)	Year
Particle swarm optimization	PSO	[11]	Kennedy et al.	1995
Ant colony optimization	ACO	[12]	Dorigo	1991
Bat algorithm	BA	[13]	Yang	2010
Firefly algorithm	FA	[14]	Yang	2009
Cuckoo search	CS	[15]	Yang et al.	2009
Social spider optimization	SSO	[16]	Cuevas et al.	2013
Krill herd algorithm	KHA	[17]	Gandomi et al.	2012
Crow search algorithm	CSA	[18]	Askarzadeh	2016
Cat swarm optimization	CSO	[20]	Chu et al.	2006
Artificial bee colony	ABC	[19]	Karaboga et al.	2007
Moth flame optimization	MFO	[21]	Mirjalili	2015
Artificial fish school algorithm	AFSA	[22]	Hu et al.	2010
Elephant herding optimization	EHO	[23]	Wang et al.	2015
Whale optimization algorithm	WOA	[24]	Mirjalili et al.	2016
Salp swarm algorithm	SSA	[25]	Mirjalili et al.	2017
Ant lion optimizer	ALO	[26]	Mirjalili	2015
Grey wolf optimizer	GWO	[27]	Mirjalili et al.	2014
Bacterial foraging optimization	BFO	[28]	Passino	2002
Bacterial colony chemotaxis algorithm	BCCA	[29]	Li et al.	2005
Fast bacterial swarming algorithm	FBSA	[30]	Chu et al.	2008
Glowworm swarm optimization	GSO	[31]	Krishnanand et al.	2009
Cockroach swarm optimization	CRSO	[32]	Chen et al.	2010
Hunting search	HS	[33]	Oftadeh et al.	2010
Bacterial colony optimization	BCO	[34]	Niu et al.	2012
Blind naked mole rats algorithm	BNMR	[35]	Taherdangkoo et al.	2012
Lion's algorithm	LA	[36]	Rajakumar	2012
Wolf search	WS	[37]	Tang et al.	2012
Fruit fly optimization	FFO	[38]	Xing et al.	2013
Dolphin swarm algorithm	DSA	[39]	Wu et al.	2013
Monarch butterfly optimization	MBO	[40]	Wang et al.	2015
Grasshopper optimization	GO	[41]	Saremi et al.	2017
Spotted hyena optimizer	SHO	[42]	Dhiman et al.	2017

From Table 2, it can be seen that most articles that contain one of the swarm intelligence algorithms have been published for the PSO algorithm. Table 3 shows the ranking of the different algorithms according to their popularity in the selected scientific databases. Number 1 indicates the best value number 32 indicates the worst value.

Table 2 The number of articles on different swarm intelligence algorithms indexed in scientific databases (February 2024)

Algorithm	Google Scholar	Springer	IEEE Explore	Science Direct	Web of Science	Total
PSO	784000	57360	39365	57916	74199	1012840
ACO	267000	23536	14439	16688	13480	335143
BA	41100	5135	1653	3436	2675	53999
FA	53200	7589	1327	5498	4430	72044
CS	58600	7598	1914	5754	4523	78389
SSO	2790	515	54	321	188	3868
KHA	6360	776	84	514	539	8273
CSA	8260	1201	149	932	675	11217
CSO	7410	1121	145	643	479	9798
ABC	105000	12416	2315	10379	7853	137963
MFO	11800	1720	234	1527	962	16243
AFSA	1020	98	34	35	47	1234
EHO	3220	646	56	416	250	4588
WOA	32700	4680	1837	3914	3034	46165
SSA	13100	1824	261	1694	1213	18092
ALO	7330	1106	176	1317	468	10397
GWO	39000	3523	3036	6801	2661	55021
BFO	12600	1804	326	1188	827	16745
BCCA	213	12	15	7	31	278
FBSA	141	23	2	6	2	174
GSO	4940	623	129	383	368	6443
CRSO	268	623	5	7	27	930
HS	1020	305	19	106	38	1488
BCO	415	115	11	36	28	605
BNMRA	28	9	1	5	0	43
LA	108	341	0	200	11	660
WS	1150	283	31	100	41	1605
FFO	10700	1152	227	1070	732	13881
DSA	391	47	7	25	14	484
MBO	2770	494	34	346	160	3804
GO	11600	1717	204	1549	890	15960
SHO	1720	320	14	244	156	2454
Total	1489954	138712	68104	123057	121001	1940828

Table 3 Popularity ranking of particular swarm intelligence algorithms in selected scientific databases (February 2024)

Algorithm	Google Scholar	Springer	IEEE Explore	Science Direct	Web of Science	Total
PSO	1	1	1	1	1	1
ACO	2	2	2	2	2	2
BA	6	6	7	8	7	6.8
FA	5	5	8	6	5	5.8
CS	4	4	5	5	4	4.4
SSO	20	21	20	21	20	20.4
KHA	17	17	18	17	15	16.8
CSA	14	13	15	15	14	14.2
CSO	15	15	16	16	16	15.6
ABC	3	3	4	3	3	3.2
MFO	11	11	11	11	10	10.8
AFSA	24	28	21	27	23	24.6
EHO	19	18	19	18	19	18.6
WOA	8	7	6	7	6	6.8
SSA	9	9	10	9	9	9.2
ALO	16	16	14	12	17	15
GWO	7	8	3	4	8	6
BFO	10	10	9	13	12	10.8
BCCA	29	31	25	29	26	28
FBSA	30	30	30	31	31	30.4
GSO	18	19	17	19	18	18.2
CRSO	28	20	29	30	28	27
HS	25	25	24	24	25	24.6
BCO	26	27	27	26	27	26.6
BNMRA	32	32	31	32	32	31.8
LA	31	23	32	23	30	27.8
WS	23	26	23	25	24	24.2
FFO	13	14	12	14	13	13.2
DSA	27	29	28	28	29	28.2
MBO	21	22	22	20	21	21.2
GO	12	12	13	10	11	11.6
SHO	22	24	26	22	22	23.2

From Table 3, it can be seen that the three best algorithms in terms of their popularity ranking are: the PSO, ACO and ABC algorithms. In contrast, the three worst algorithms are: the BNMRA, the FBSA and the DSA. Figure 2 graphically shows the average ranking value for each algorithm.

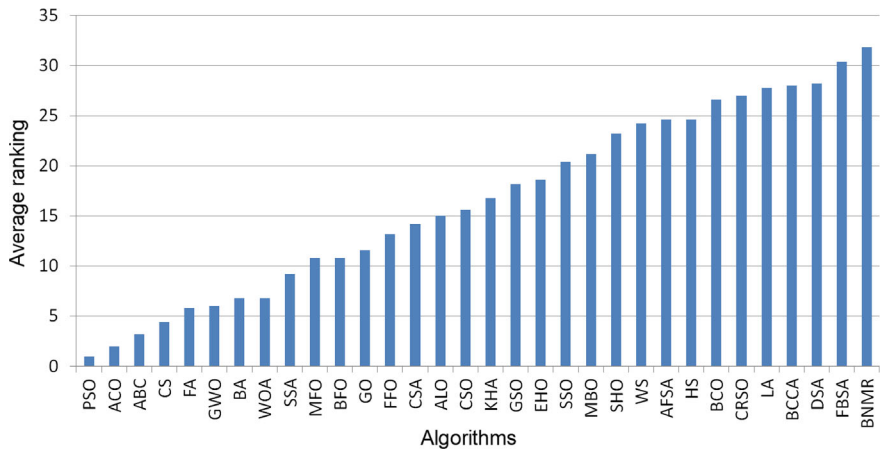


Fig. 2 Average ranking for selected algorithms (February 2024)

Based on Fig. 2, algorithms with a ranking value less than or equal to the value of 10 were selected for a further consideration. The selected algorithms are: the PSO, the ACO, the ABC, the CS, the FA, the GWO, the BA, the WOA and the SSA.

Table 4 shows the 10 most popular application areas for each of the selected algorithms. From Table 4, it can be seen that for the seven application areas, all of the previously mentioned algorithms have published papers. These areas are: “Engineering Electrical Electronic”, “Energy and Fuels”, “Engineering Multidisciplinary”, and the areas related to “Computer Science” such as “Artificial Intelligence”, “Theory Methods”, “Information Systems” and “Interdisciplinary Applications”. Therefore, Sect. 4 will present the applications of the algorithms from Table 4 in the previously mentioned areas. Also for a better visual illustration, the data from Table 4 are presented in graphical form in Figs. 3 and 4.

In Table 4 *Highly Cited Papers* means that an article received enough citations to place it in the top 1% most cited articles for the field and publication year; *Hot Papers* means that the selected article is in the top 0.1% most cited articles for the field and year publication.

4 Engineering Applications of Selected Swarm Intelligence Algorithms

This section presents the selected papers on the application of swarm intelligence algorithms in the most popular industrial fields. The papers were selected from the Web of Science database based on their popularity.

Table 4 Number of papers in most published areas for selected swarm intelligence algorithms (February 2024)

Area of application	PSO	ACO	ABC	CS	FA	GWO	BA	WOA	SSA	Total
Engineering Electrical Electronic	23278	3493	2253	1261	1212	739	732	792	310	34070
Computer Science Artificial Intelligence	17111	4117	2212	1149	1128	625	748	607	264	27961
Computer Science	8326	2647	1102	582	603	220	388	247	99	14214
Computer Science Information Systems	8035	2013	1034	598	619	348	394	452	182	13675
Computer Science Interdisciplinary Applications	6855	1688	1004	455	518	304	283	269	140	11516
Energy Fuels	6846	1514	498	440	331	314	184	322	126	10575
Telecommunications	6650	–	713	433	449	270	288	350	133	9286
Automation Control Systems	6592	1051	500	284	213	142	154	152	–	9088
Engineering Multidisciplinary	5115	762	624	389	386	247	209	266	128	8126
Operations Research Management Science	3097	1054	426	–	–	136	107	–	–	4820
Computer Science Software Engineering	–	687	–	195	165	–	–	–	–	1047
Instruments Instrumentation	–	–	–	–	–	–	–	138	–	138
Environmental Sciences	–	–	–	–	–	–	–	–	59	59
Green Sustainable Science Technology	–	–	–	–	–	–	–	–	55	55
Total	91905	19026	10366	5786	5624	3345	3487	3595	1496	144630
Highly Cited Papers	671	102	109	66	65	85	48	87	40	1273
Hot Papers	26	6	3	0	1	0	0	2	0	38
Total	697	108	112	66	66	85	48	89	40	1311

4.1 Engineering Electrical Electronic

In paper [43], the authors present a hybrid cluster head selection algorithm for wireless sensor networks using artificial bee colony optimization and monarch butterfly optimization. In this method, the worker bee phase from artificial bee colony optimization is replaced by a mutant butterfly adjustment operator from monarch butterfly optimization. The goal of this adaptation is to avoid early trapping of solutions in local optima and delayed convergence by maintaining a balance between exploitation and exploration. In addition, the proposed approach eliminates the problem of overloaded cluster heads, which can lead to rapid depletion of sensor nodes during implementation. Simulation results showed that the number of active nodes in the network is around 18% higher than the compared cluster head selection methods.

4.2 Computer Science

4.2.1 Artificial Intelligence

Zhang et al. [44] presented a multi-objective PSO algorithm for selecting significant features from a training dataset. Selection of significant features from the learning set is very important for large data sets. As is well known, reducing the number of features directly affects the time required to train a model (such as an artificial neural network model). The algorithm described in this paper includes such novelties as rapid nondominated sorting technique and five single candidate solution gener-

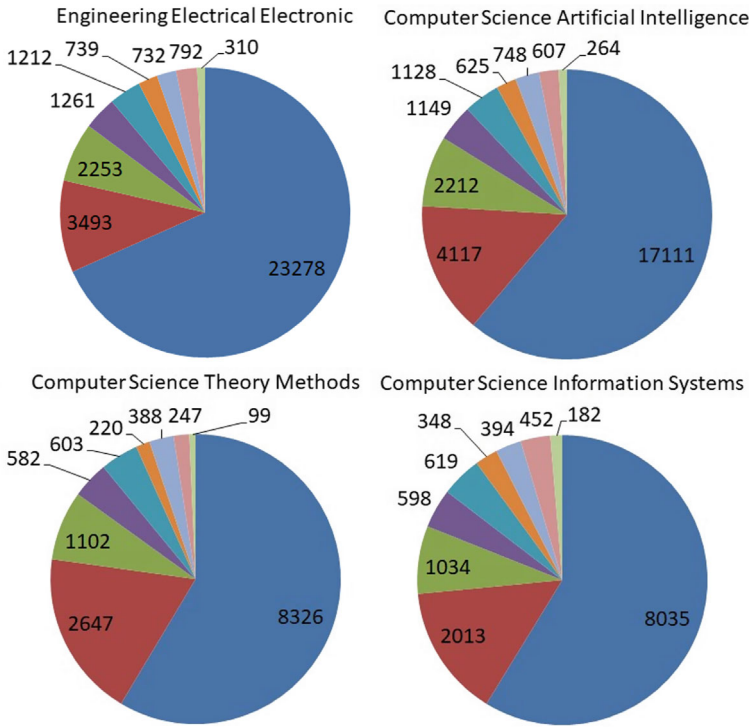


Fig. 3 Number of papers in each application area for selected algorithms (February 2024)

ation strategies. The effectiveness of the presented approach was tested on ten test datasets selected from the literature. The results show a significant improvement in the classification error in relation to the number of reduced features.

4.2.2 Theory Methods

The authors in [45] presented a hybrid algorithm that combines properties of such algorithms as crow search and grey wolf optimizer. In the developed method, the GWO algorithm was responsible for exploration of the space of potential solutions. The CSA algorithm, on the other hand, was responsible for local exploration of the space of potential solutions. The algorithm was developed to optimize problems with a huge number of dimensions. The effectiveness of the described method was tested on 15 test functions selected from the CEC 2015 collection and on 4 real engineering design problems. The results obtained were shown to be better than those obtained by other methods selected from the literature.

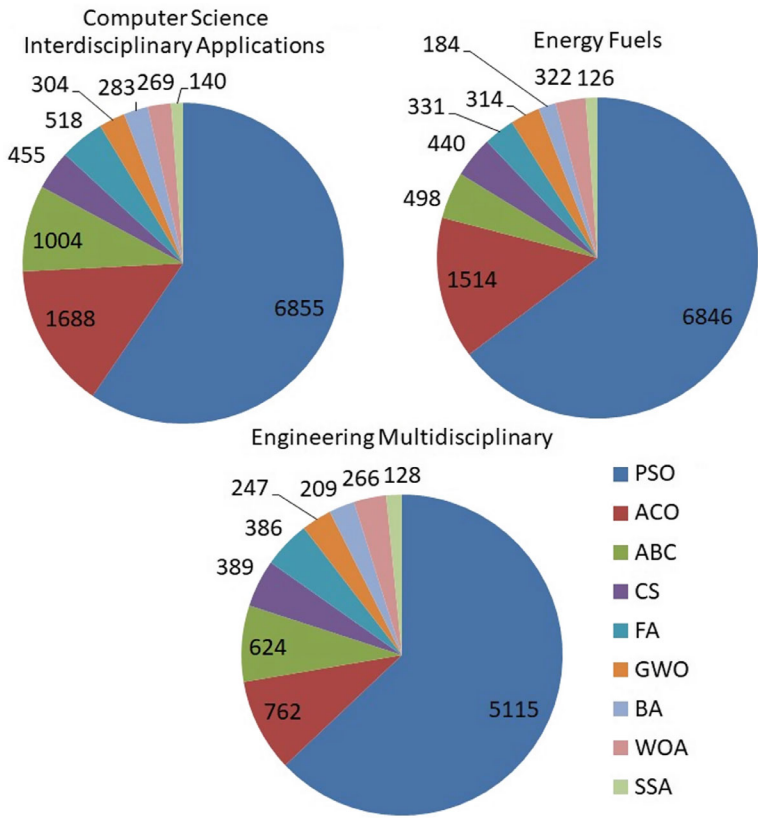


Fig. 3 (continued)

4.2.3 Information Systems

In the paper [46], the authors proposed a modified bat algorithm for solving the open shop scheduling problem (OSSP). As is well known, OSSP depends on the execution of assumed tasks on a limited number of machines. The main goal of the optimization is to find such a schedule for the scheduling of all tasks on individual machines that its execution time is as small as possible. For this purpose, in this work bat algorithm was modified by adding ColReuse and substitution meta-heuristic functions. These functions are proposed in order to increase the convergence of the algorithm to the optimal solution. The described algorithm was tested on standard open shop benchmarks, and the results obtained were compared with those obtained by other methods. In the end, the proposed method showed better performance than previously developed methods and found the optimal solution for all test cases.

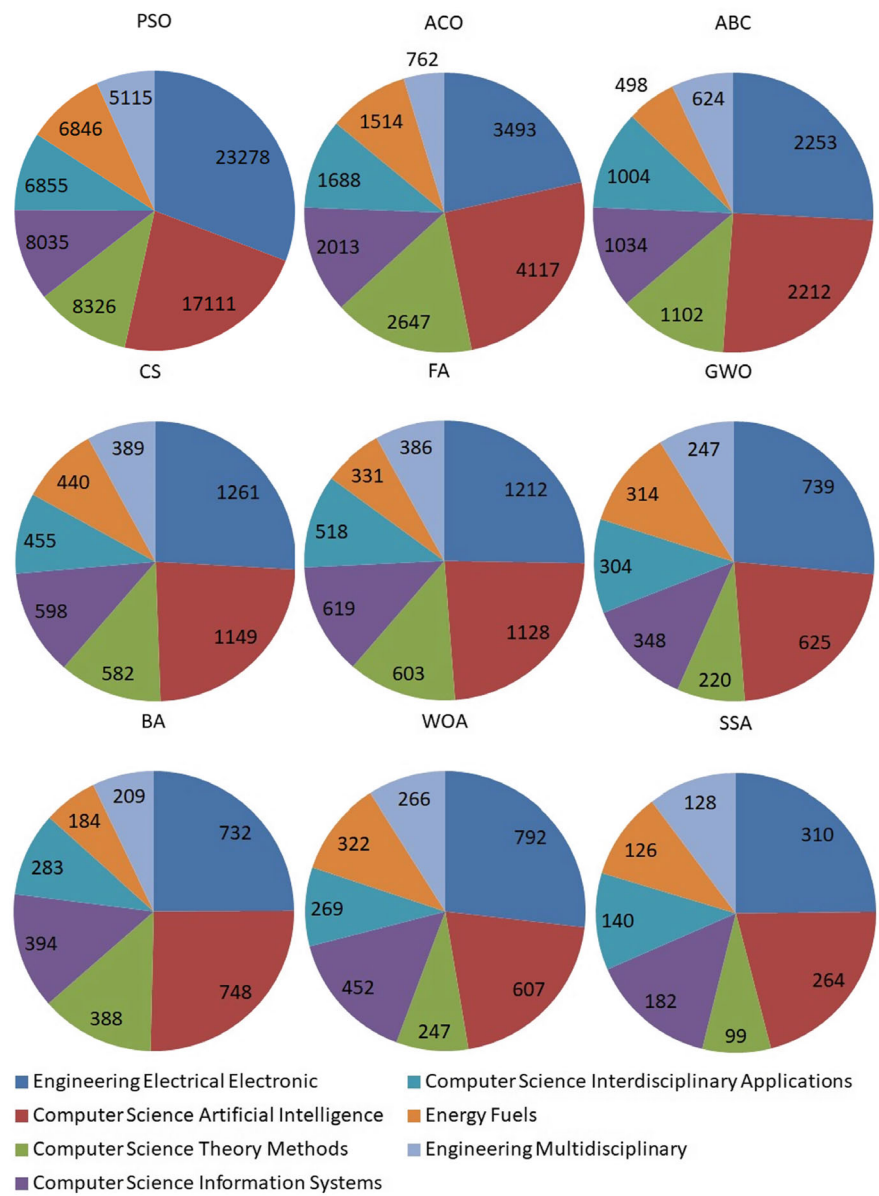


Fig. 4 Number of papers for each algorithm in selected application areas (February 2024)

4.2.4 Interdisciplinary Applications

The paper by Qi et al. [47] presents an innovative COVID-19 X-ray image segmentation method based on a modified ant colony optimization algorithm. Improvements were made to the algorithm by improving the directional crossover and mutation strategies to speed up the convergence process and avoid local optima. In addition, an image segmentation model was developed using various techniques such as 2D histograms, 2D Kapura entropy and non-local average strategy. The application of this model to COVID-19 X-ray image segmentation showed its superior stability and efficiency compared to existing methods at different threshold levels.

4.3 Energy and Fuels

The article [48] presents a method based on a multi-objective version of the grey wolf optimizer algorithm. The proposed method was applied to forecast electricity consumption. The main concept of the presented algorithm was to simultaneously minimize the number of features in the dataset and minimize the forecast error of electricity consumption. In addition, the developed algorithm introduced individual regressor selection, which was selected during the operation of the algorithm. This selection was made between such regressors as random forest, extra trees, decision tree and K-nearest neighbors. The results obtained by the described method were compared with those obtained by the multi-objective PSO (MOPSO) algorithm and by the NSGA II algorithm. In many cases, the obtained results were better than those obtained by the MOPSO and NSGA II algorithms.

4.4 Engineering Multidisciplinary

The paper [49] presents a hybrid algorithm (PSO-CS) that combines algorithms such as PSO and CS. This algorithm was used to train a deep neural network as a replacement for the error back-propagation algorithm. Such a concept was used because the back-propagation algorithm has limitations due to the speed of convergence and getting stuck in local minima. The trained deep neural network was applied to a depression recognition task. In addition, the authors compared the classification efficiency of their method with such methods as decision trees, K-nearest neighbor, visual geometry group, support vector regression, residual neural network and simple neural network. As a result of the comparison, it was shown that the presented PSO-CS method outperformed all other models receiving a high degree of classification accuracy of 99.5%.

5 Conclusions

It can be seen from the presented chapter that swarm intelligence algorithms have very wide engineering applications. The most commonly used algorithm is the PSO algorithm, which is described in detail in this chapter. In addition to the PSO algorithm, there are many other popular SIAs such as the ACO algorithm or the CS algorithm. Newer and newer algorithms are also currently being developed in this area. Further research directions in this area may focus on creating mechanisms such as adaptively changing the parameters of a given algorithm depending on the progress of the optimization process or changes in the optimization problem. Such a procedure may enhance the algorithm's ability to deal with a variety of engineering problems. Another issue may be combining swarm intelligence algorithms with other advanced optimization algorithms or other artificial intelligence algorithms, which can lead to hybrid optimization solutions. For example, the hybridization with gradient algorithms may result in a more powerful final algorithm. Another goal may be to adapt swarm intelligence algorithms to handle multi-criteria problems. This is a relevant issue in the context of more complex optimization problems. The development of efficient methods for dealing with multiple objectives simultaneously will allow for a more extensive application of swarm intelligence algorithms in engineering tasks.

References

1. Goldberg D (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Company Inc., Boston
2. Michalewicz Z (1992) Genetic algorithms + data structures = evolution programs. Springer, Berlin
3. Slowik A, Slowik J (2008) Multi-objective optimization of surface grinding process with the use of evolutionary algorithm with remembered Pareto set. *Int J Adv Manuf Technol* 37(7–8):657–669. <https://doi.org/10.1007/s00170-007-1013-0>
4. Slowik A, Bialko M (2004) Design and optimization of combinational digital circuits using modified evolutionary algorithm. In: *Lecture notes in artificial intelligence*, vol 3070. Springer, pp 468–473
5. Slowik A, Bialko M (2004) Modified version of roulette selection for evolution algorithms—the fan selection. In: *Lecture notes in artificial intelligence*, vol 3070. Springer, pp 474–479
6. Slowik A, Bialko M (2006) Partitioning of VLSI circuits on subcircuits with minimal number of connections using evolutionary algorithm. In: *Lecture notes in computer science*, vol 4029, ICAISC. Springer, pp 470–478
7. Slowik A (2010) Steering of balance between exploration and exploitation properties of evolutionary algorithms—mix selection. In: *Lecture notes in artificial intelligence*, vol 6114, ICAISC. Springer, pp 213–220
8. Slowik A (ed) (2020) Swarm intelligence algorithms: a tutorial. CRC Press, Taylor & Francis Group, Boca Raton, USA
9. Slowik A (ed) (2020) Swarm intelligence algorithms: modifications and applications. CRC Press, Taylor & Francis Group, Boca Raton, USA
10. Slowik A, Cpalka K (2022) Hybrid approaches to nature-inspired population-based intelligent optimization for industrial applications. *IEEE Trans Ind Inf* 18(1):546–558. <https://doi.org/10.1109/TII.2021.3067719>

11. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, pp 1942–1948
12. Colomi A, Dorigo M, Maniezzo V (1991) Distributed optimization by ant colonies. In: Proceedings of the European conference on artificial life, pp 134–142
13. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: Proceedings of nature inspired cooperative strategies for optimization (NICSO), pp 65–74
14. Yang XS (2009) Firefly algorithms for multimodal optimization. In: Proceedings of international symposium on stochastic algorithms, LNCS, vol 5792, pp 169–178
15. Yang, XS, Deb S (2009) Cuckoo search via levy flights. In: Proceedings of IEEE world congress on nature and biologically inspired computing (NaBIC), pp 210–214
16. Cuevas E, Cienfuegos M, Zaldivar D, Perez-Cisneros M (2013) A swarm optimization algorithm inspired in the behaviour of the social-spider. *Expert Syst Appl* 40(16):6374–6384
17. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. *Commun Nonlinear Sci Numer Simul* 17(12):4831–4845
18. Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Comput Struct* 169:1–12
19. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Glob Optim* 39(3):459–471
20. Chu SA, Tsai PW, Pan JS (2006) Cat swarm optimization. In: Proceedings of Pacific Rim international conference on artificial intelligence, LNCS, vol 4099, pp 854–858
21. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl Based Syst* 89:228–249
22. Hu J, Zeng X, Xiao J (2010) Artificial fish school algorithm for function optimization. In: Proceedings of IEEE 2nd international conference on information engineering and computer science, pp 1–4
23. Wang GG, Deb S, Coelho LDS (2015) Elephant herding optimization. In: Proceedings of 3rd international symposium on computational and business intelligence (ISCBI), pp 1–5
24. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
25. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191
26. Mirjalili S (2015) The ant lion optimizer. *Adv Eng Softw* 83:80–98
27. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
28. Passino K (2002) Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst* 22(3):52–67
29. Li WW, Wang H, Zou ZJ (2005) Function optimization method based on bacterial colony chemotaxis. *J Circ Syst* 10:58–63
30. Chu Y, Mi H, Liao H, Ji Z, Wu QH (2008) A fast bacterial swarming algorithm for high-dimensional function optimization. In: Proceedings of IEEE world congress on computational intelligence (CEC), pp 3135–3140
31. Krishnanand KN, Ghose D (2009) Glowworm swarm optimisation: a new method for optimising multi-modal functions. *Int J Comput Intell Studi* 1(1):93–119
32. Chen ZH, Tang HY (2010) Cockroach swarm optimization. In: Proceedings of 2nd international conference on computer engineering and technology, pp 652–655
33. Oftadeh R, Mahjoob MJ, Shariatpanahi M (2010) A novel meta-heuristic optimization algorithm inspired by group hunting of animals: hunting search. *Comput Mathe Appl* 60(7):2087–2098
34. Niu B, Wang H (2012) Bacterial colony optimization. *Discrete Dyn Nature Soc* 2012:1–28
35. Taherdangkoo M, Shirzadi MH, Bagheri MH (2012) A novel meta-heuristic algorithm for numerical function optimization: blind, naked mole-rats (BNMR) algorithm. *Sci Res Essays* 7(41):3566–3583
36. Rajakumar BR (2012) The lion's algorithm: a new nature-inspired search algorithm. *Procedia Technol* 6:126–135

37. Tang R, Fong S, Yang XS, Deb S (2012) Wolf search algorithm with ephemeral memory. In: Proceedings of seventh international conference on digital information management (ICDIM), pp 165–172
38. Xing B, Gao WJ (2013) Fruit fly optimization algorithm. In: Innovative computational intelligence: a rough guide to 134 clever algorithms, pp 167–170
39. Wu T, Yao M, JH (2016) Dolphin swarm algorithm. *Front Inf Technol Electron Eng* 17(8):717–729
40. Wang G-G, Deb S, Cui Z (2019) Monarch butterfly optimization. *Neural Comput Appl* 31:1995–2014
41. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. *Adv Eng Softw* 105:30–47
42. Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. *Adv Eng Softw* 114:48–70
43. Rambabu B, Reddy AV, Janakiraman S (2022) Hybrid Artificial Bee Colony and Monarchy Butterfly Optimization Algorithm (HABC-MBOA)-based cluster head selection for WSNs. *J King Saud Univ Comput Inf Sci* 34(5):1895–1905
44. Zhang C, Xue Y, Neri F, Cai X, Slowik A (2024) Multi-objective self-adaptive particle swarm optimization for large-scale feature selection in classification. *Int J Neural Syst* 34(3), Article ID: 2450014
45. Rizk-Allah RM, Slowik A, Hassanien AE (2020) Hybridization of grey wolf optimizer and crow search algorithm based on dynamic fuzzy learning strategy for large-scale optimization. *IEEE Access* 8:161593–161611
46. Shareh MB, Bargh SH, Hosseinabadi AAR, Slowik A (2021) An improved bat optimization algorithm to solve the tasks scheduling problem in open shop. *Neural Comput Appl* 33:1559–1573
47. Qi AL, Zhao D, Yu FH, Heidari AA, Wu ZD, Cai ZN, Alenezi F, Mansour RF, Chen HL, Chen MY (2022) Directional mutation and crossover boosted ant colony optimization with application to COVID-19 X-ray image segmentation. *Comput Biol Med* 148, Article ID: 105810
48. Moldovan D, Slowik A (2021) Energy consumption prediction of appliances using machine learning and multi-objective binary grey wolf optimization for feature selection. *Appl Soft Comput* 111, Article ID: 107745
49. Jawad K, Mahto R, Das A, Ahmed SU, Aziz RM, Kumar P (2023) Novel cuckoo search-based metaheuristic approach for deep learning prediction of depression. *Appl Sci*, 13(9), Article ID: 5322

Chapter 4

Convergence Catalysts: Exploring the Fusion of Embedded Systems, IoT, and Artificial Intelligence



Cosmina Mihaela Rosca 

1 Literature Survey

The Internet of Things (IoT) represents a network interlinking physical devices, appliances, and objects that are integrated with sensors, software, and network connectivity, facilitating data collection and exchange. The fundamental objective of the IoT is to facilitate communication and interaction among devices through the Internet, promoting seamless data sharing and automated processes.

The Internet functions as a communication network linking people and information, whereas the IoT expands connectivity to objects capable of sensing and processing data. Each IoT object has a unique cyberspace address, facilitating seamless operations and communications. IoT application domains are vast, incorporating Machine Learning (ML) and Artificial Intelligence (AI) for anomaly detection systems in behavioral analysis [1]. The IoT, a phase in the Internet's evolution, connects ordinary devices globally, with around 30.7 billion smart devices integrated by 2020, and the progression continued with 35.8 billion by 2021, 42.6 billion by 2022, 51.1 billion by 2023, with an estimated figure of 62.1 billion by 2024 and 75.4 billion by 2025 [2].

AI has emerged as a prominent force across diverse research domains, offering applications that either substitute human operators or streamline users' everyday tasks. Consequently, AI-specific technologies find integration in a myriad of IoT applications, ranging from multilingual text recognition systems employing Tesseract and EasyOCR services [3] to consumer behavior analysis on social media platforms [4]. The paramount application realm of AI within the IoT context manifests in the medical field, encompassing initiatives like the acquisition, analysis, and processing of medical signal data collected by wearable devices [5], innovative techniques for the segmentation of leukemia-affected cell images achieving 94% accuracy in nucleus

C. M. Rosca (✉)
Petroleum—Gas University of Ploiesti, Ploiesti, Romania
e-mail: cosmina.rosca@upg-ploiesti.ro

segmentation and 78% in cytoplasm segmentation [6], or the establishment of an e-health platform utilizing the IoT for global connectivity, facilitating remote health-care, early disease detection, and cost-effective preventive measures [7]. Furthermore, the implementation of machine learning techniques takes center stage in the early and efficient detection of inherited diseases, with a specific focus on autism spectrum disorder through real-time questionnaire-based diagnosis [8], or brain intracerebral hemorrhage classification using head CT scan for stroke analysis [9].

The development of IoT systems is driven by the integration of technological and infrastructure progressions centered around the Internet. Key factors include Industry 4.0, digitalization, the proliferation of sensors, affordability, and seamless integration. Industry 4.0, a transformation in manufacturing through technologies like automation, robotics, and AI, aligns with IoT's core principles. Smartphones, acting as sensor-rich gateways, play a pivotal role, providing a massive user base and connectivity. Cloud computing enables global IoT systems without upfront server investments, offering scalable capacity. AI and ML advancements further enhance IoT, providing actionable insights from data. The Internet serves as the foundational pillar, offering standardized communication protocols, global reach, and interoperability, fostering innovation and creating a competitive marketplace [10].

An embedded system (ES) is a dedicated computing device created to execute particular functions or tasks as part of a broader system. It typically consists of a combination of specialized hardware components and software, often programmed to execute a predetermined set of functions. ESs are embedded into other devices or systems and operate with real-time constraints. For this reason, a smartphone device might be considered an embedded system.

ESs, characterized by bare-metal configurations, lack dedicated hardware security provisions. Integrating an operating system (OS), be it general-purpose or real-time, offers a structured environment, resource management, and common functions through API calls for multiple applications. Virtualization further enhances security by providing stronger separation between applications, particularly useful for loosely coupled ones. Containerization, a packaging mechanism bundling applications, and dependencies, is employed in systems with sufficient resources and proper OS support. Containers simplify deployment and portability, running directly on the supporting OS or within virtualized environments. Widely used in cloud systems, containers also play a crucial role in managing IoT applications on edge nodes, fog nodes, and gateways, optimizing function placement dynamically [10].

Conceptually, an IoT infrastructure comprises edge components and cloud components. At the edge, there are embedded systems, smart devices, gateways, and fog nodes. These edge-level components are responsible for real-time data collection and processing. In the cloud, there are large-scale storage and processing services where data collected from the edge is aggregated and analyzed to provide valuable insights.

As embedded systems predominantly operate with proprietary firmware, the burgeoning interest in the IoT domain has fueled a heightened cyber threat landscape. This surge underscores the imperative for robust security measures. One viable

approach involves integrating edge computing into IoT devices to enhance security for devices with immutable firmware [11].

ESs use microprocessors or microcontrollers. These computing systems require, in addition to the CPU, memory, I/O devices, and interconnection elements among them. These systems can fulfill a wide range of functions.

In contrast to microprocessors, a microcontroller is a compact computing system on a single chip, having a similar internal structure as a microprocessor. However, its primary objective is to accomplish a limited set of functions, guided by a pre-defined functionality of the dedicated application running on the microcontroller.

Systems containing either microprocessors or microcontrollers are also referred to as single-chip systems or System-on-a-Chip (SoC). SoCs integrate multiple components of a computing system onto a single chip. A typical SoC includes the following components in a single device: processor, RAM, Flash memory, network ports, peripheral ports, power ports, etc. [12].

Choosing a processor for IoT must consider power consumption, software tools, area, and performance. A study comparing 8-bit and 32-bit SoCs, showing the latter’s higher power consumption but better performance and larger area is presented in the paper [13]. The paper [14] examines SoCs for power electronics control and presents a design and experimental results of an SoC-based controller. Additionally, it reviews SoC-based platforms developed by researchers in this field.

The paper [15] proposes embedding a minimal virtual machine (VM) on a SoC to address the challenge of modifying SoCs without hardware changes. This solution facilitates dynamic program updates without reprogramming ROM, contingent upon the SoC having network capabilities.

Fundamental differences between microprocessors and microcontrollers exist, as illustrated in Table 1, emphasizing the microcontroller’s significantly simplified nature compared to a processor.

By analyzing Table 1, the following observations can be made:

Table 1 Differences between microprocessors and microcontrollers

Characteristic	Microprocessor	Microcontroller
Objective	Designed to perform complex computing and data processing operations	Designed for control, monitoring, and resource management tasks in a dedicated system or a small number of tasks
Computational power	Performs a variety of complex tasks, sometimes in parallel	Optimized for specific control and signal processing tasks
Architecture	General-purpose	Specialized for specific tasks
Connectivity	Extensive—RAM, graphics cards, storage devices, I/O devices	Reduced—analog and digital inputs/outputs, interfaces for sensors and actuators

Source Made by the author

- **Objective:** Microprocessors are designed to perform a wide range of tasks, including data processing, numerical analysis, and artificial intelligence. Microcontrollers are designed for more specialized tasks, such as controlling devices or monitoring systems.
- **Computational power:** Microprocessors have a higher computational power than microcontrollers. This is because they have a larger number of transistors and a faster clock speed.
- **Architecture:** Microprocessors have a general-purpose architecture, which means that they can be used for a variety of tasks. Microcontrollers have a specialized architecture, which is optimized for a specific task.
- **Connectivity:** Microprocessors have a wider range of connectivity options than microcontrollers. This is because they are often used in systems that require many peripherals. Microcontrollers have a more limited range of connectivity options, which is sufficient for their specialized tasks.

ESs are configurations designed to fulfill specific tasks while using a compact hardware infrastructure, all within a cost-effective budget. Having specialized hardware, dedicated OSs, and software with limited functionalities, these systems provide predefined functionality.

For instance, a kiosk allowing users to register for an event using a touchscreen and a printer represents an ES. It is designed to provide a specific functionality without requiring a general OS (such as Windows 10 Enterprise) or a wide range of applications. The system must facilitate user data input into a database and generate a unique number via an order receipt printed with the appropriate equipment.

Most such equipment is configured, updated, and programmed remotely through specialized tools, as will be demonstrated in the next sections.

A development board is a hardware platform incorporating a microcontroller or microprocessor, I/O ports, power supply, connectors, and additional elements to facilitate the development process. These boards are used to create prototypes, tests, and experiments in various fields such as IoT, robotics, automation, and electronics.

There are a variety of development boards:

1. General-purpose development boards, classified into:

- Boards with limited functionalities, for instance, Arduino Uno, PICkit Development Board, ESP8266/ESP32, etc. These boards are employed in projects that do not involve general-purpose or professional I/O devices. Typically, they are dedicated to applications exclusively using sensors and execution elements. These boards often employ microcontrollers, such as those from the Atmel AVR family for Arduino Uno, or Microchip's PIC microcontrollers for PICkit boards. The ESP8266 and ESP32 boards typically integrate microcontrollers from Espressif Systems.
- Boards with extended functionalities, for example, Raspberry Pi, Asus Thinker Board, Olimex, etc. These boards can utilize advanced software, APIs enabling integration with other software services, and general-purpose or professional I/O devices. These boards generally use microprocessors. For

instance, Raspberry Pi models typically employ ARM-based processors, such as those from the Broadcom BCM series.

2. FPGA (Field-Programmable Gate Array) development boards, such as Xilinx or Intel® FPGAs, are used for programming digital circuits in various applications, including signal processing. In this case, the processors involved may be soft processors configured within the FPGA or external processors used in conjunction with the FPGA for specific tasks.
3. Machine Learning development boards, for example, NVIDIA Jetson Nano or Google Coral Dev Board, are dedicated to projects involving AI elements and parallel computing. NVIDIA Jetson Nano employs an ARM Cortex-A57 quad-core processor with integrated NVIDIA Maxwell GPU cores, whereas Google Coral Dev Board incorporates a Google Edge TPU coprocessor for accelerated machine learning tasks.

The prevalence of ARM processors is noteworthy, indicating a widespread adoption of this architecture in the design and development of ESs. ARM, which stands for Advanced RISC Machines, offers a diverse range of processor designs characterized by reduced instruction set computing (RISC) principles. This architecture is well-suited for embedded applications due to its energy efficiency, scalability, and versatility. The use of different versions of ARM processors underscores the adaptability of this architecture to meet the specific requirements of diverse ESs.

The author of this paper, together with two other specialists in the field, developed an IoT system that monitors the stress level of adopted children. For the implementation of this prototype, an ESP32 microcontroller was used [16]. The paper [16] presents the proposed software product that aims to address key challenges in the child adoption process, including compatibility issues between children and adoptive parents, lack of accessibility, high costs, and monitoring stress levels during trial adoptions.

Two significant examples will be presented to highlight situations where the use of an ES with a microcontroller is appropriate, as well as scenarios where the deployment of an ES with a microprocessor is necessary.

In a home automation system, a microcontroller-based ES may be employed for simple tasks such as controlling individual smart light bulbs. A microcontroller, with its compact size and dedicated functionality, can efficiently manage the on/off states. The microcontroller is well-suited for this application due to its low power consumption, cost-effectiveness, and ability to execute predefined tasks without the need for a sophisticated OS.

Secondly, a more complex IoT application, such as a smart home hub, could benefit from a microprocessor-based ES. The microprocessor's higher computational power and capability to handle multiple concurrent tasks make it suitable for managing a diverse range of smart devices in a centralized manner. Tasks such as processing voice commands, coordinating various sensors, and running a more sophisticated OS for enhanced user interaction require the capabilities of a microprocessor-based system.

Consider an industrial monitoring system aimed at overseeing various sensors and actuators in a manufacturing plant. A microcontroller-based ES may be employed for a specific task, such as monitoring the temperature of individual machines. The microcontroller can efficiently read sensor data, make decisions based on predefined thresholds, and control the cooling systems as necessary. Its real-time processing capabilities and deterministic behavior make it suitable for such dedicated tasks.

In contrast, a more complex application, like a comprehensive industrial control system managing multiple processes, could leverage a microprocessor-based ES. The microprocessor's higher processing speed and ability to execute complex algorithms are advantageous when overseeing a network of interconnected sensors, coordinating multiple actuators, and running a sophisticated control algorithm. The flexibility of a microprocessor-based system allows for the implementation of more intricate control strategies, making it well-suited for applications with higher complexity in the realm of IoT.

The choice between microcontroller and microprocessor-based ESs depends on the specific requirements of the application. Microcontrollers excel in simpler, dedicated tasks with low power consumption, while microprocessors are more suitable for applications demanding higher computational power, multitasking capabilities, and the ability to run sophisticated OSs. The examples provided highlight the distinct advantages of each system in different IoT scenarios.

In embedded systems utilizing microprocessors, the OS is typically stored in non-volatile memory, such as Flash memory. This memory space is designed to retain information even in the absence of electrical power. Typically, upon system boot, the OS is loaded into Random Access Memory (RAM) to enable swift access to its functionalities and facilitate the execution of programs and system resource management. It is essential to note that the specific storage location may vary based on the architecture and configuration of the particular embedded system.

2 Compatibility Methodology

The evolution of AI, combined with the multitude of small-scale devices (such as sensors), has allowed the development of a distinct sector called the Internet of Things (IoT). This technology is based on the interconnection of different systems through local networks or the Internet, to share data. Subsequently, meaning will be attributed to the data, with the ultimate goal of making various decisions. Thus, this technology, previously known as the automation of different processes, has been expanded through significant advancements in software components, represented by AI services.

The constitutive elements of IoT technology are [17]:

- Automation technologies for processes through sensors and execution elements;
- Data sharing technologies locally or on the Internet, via Bluetooth, Wi-Fi, Zigbee, etc.;

- Data storage technologies through local or cloud servers, depending on the volume of data that is stored and subsequently analyzed;
- AI-specific technologies for data analysis. If the volume of data is large (big data), then dedicated tools can be used, and otherwise, tools based on classical algorithms can be employed;
- Security and confidentiality technologies by integrating encryption measures, controlled access, and intrusion detection, when sensitive data are managed.

Analyzing the multitude of technologies incorporated by the IoT field, it is evident that there is a need to understand and deepen a large number of disciplines in the computer science field. IoT applications are varied, including medical, industrial, agricultural, etc. However, the author of this chapter considers that the first stage in the development of IoT applications is represented by the hardware and software configuration of the system itself.

Designing the assembly by the application requirements is essential, beyond all these technologies that can create the illusion of a futuristic concept. The AI service that will be integrated is irrelevant if the hardware infrastructure does not physically allow the integration of such facilities.

To illustrate this idea more easily, let us assume that the system to be designed must transmit the ambient temperature value to a Cloud service every 5 min. To achieve this goal, the system must have an Internet connection. This connection can be:

- wired, so an Ethernet port is required;
- Wi-Fi connection, which requires the existence of a Wi-Fi module or adapter;
- via SIM card, so it requires an external GSM or 4G/LTE module.

The choice of one of the previous options will further dictate the type of application developed, which must also take into account the other hardware elements, as well as the restrictions of the OS used.

Therefore, the author considers that the central element of IoT technology is called **compatibility**. This concept aims at integrating a confluence of technologies capable of operating within the context of a specific hardware infrastructure, using a dedicated software configuration to achieve established objectives. Configuring a hardware-software infrastructure with incompatible equipment from the design stage results in high project costs. There are situations where integration may achieve project goals but with high costs; however, there are also cases where objectives cannot be met due to hardware or software limitations, especially when the compatibility of constituent elements is not analyzed from the design phase.

To achieve compatibility in the hardware-software infrastructure, the author proposes the following stages:

Stage 1: Analysis of the project's long-term requirements and objectives. The notion of "long-term" is deliberately emphasized because the design must take into account the possibility of extending the requirements. Failure to consider this possibility from the design stage may impede future development. Of course, this

flexibility may translate into higher project costs. Consequently, a balance must be struck between flexibility and costs.

Stage 2: Identification of the equipment to be used in the physical implementation of the project. The equipment, represented by input–output devices (encompassing both sensors and actuators), may impose certain restrictions on the overall hardware–software system. Most of the time, the restrictions of peripheral equipment come from the limitations of device drivers, given that the development of a custom driver can be truly cost-prohibitive. Device drivers are the ones that limit the use of certain OSs and a limited number of their versions. The reader should remember that input–output devices incorporate a wide range of equipment, which most of the time do not belong to the personal use class, being professional equipment;

Stage 3: Configuration of the hardware infrastructure. Limitations regarding the OS require choosing a specific hardware configuration that includes a certain processor architecture.

Stage 4: Analysis of software environment options. In most cases, projects additionally demand the development of a dedicated software component. This component is represented by a customized program that may, in turn, employ various pre-implemented services accessible through an API. The choice of programming language (PL) and template is imposed by the OS used.

Integration of the AI services requires careful consideration of the PL and development environment (DE). The compatibility between these elements and the chosen AI service is paramount for seamless collaboration. As an illustrative example, if a project involves utilizing Microsoft Azure services, opting for Visual Studio as the DE and a PL within the .NET framework becomes a normal choice. Microsoft Azure, a cloud computing platform, offers a range of AI services and the programmer has to align the development tools with the platform's requirements. Visual Studio, Microsoft's integrated development tool (IDE), stands out as a development tool for crafting applications across various domains, including AI. Its features and user-friendly interface make it a preferred choice for developers looking to integrate AI seamlessly. Choosing the appropriate PL within the .NET framework further enhances compatibility. The .NET framework, developed by Microsoft, supports multiple PLs such as C#, VB.NET, and F#. Visual Studio and .NET create a cohesive DE that streamlines the integration process with Microsoft Azure's AI services.

The successful integration of AI services hinges on the strategic alignment of the PL and DE with the chosen AI platform. However, the DE and PL are closely tied to the OS, making the compatibility process even more complex in the integration of IoT solutions. The seamless integration of IoT solutions is a multifaceted task that extends beyond selecting an appropriate AI service. The chosen DE and PL play an important role in this process, their compatibility is intricately linked with the underlying OS.

Furthermore, the OS acts as a bridge between the hardware and software layers, influencing how applications interact with the underlying device infrastructure. Ensuring compatibility among the DE, PL, and OS is imperative for a smooth and effective integration process. This interdependence becomes especially pronounced in the IoT landscape, where different devices with varying OSs need to seamlessly

communicate and collaborate. The careful consideration of the interplay among the DE, PL, and OS becomes paramount. The success of IoT integration relies on harmonizing these elements to create a cohesive ecosystem that facilitates communication and operation across the entire spectrum of connected devices.

Figure 1 illustrates the compatibility concept with the stages involved in developing an IoT solution within the context of AI. At the first level, the process begins with the project requirements and objectives, crucial for establishing the overarching framework. Simultaneously, considerations for input and output device restrictions set the stage for shaping the solution’s functionality. Moving to the second tier, the hardware considerations with a focus on the central processor architecture and board configuration are imposed by the first layer, concerning the third one. This stage lays the groundwork for the physical implementation of the IoT solution. The choice of the OS will influence how the solution interacts with the underlying infrastructure. The third level introduces the components of AI integration. API and AI service platforms become central, facilitating seamless communication between the IoT solution and AI functionalities. Concurrently, the choice of PL and DE needs to be compatible with the entire solution. This multi-tiered approach underscores the intricate interplay among hardware, software, and AI components in the development of an IoT solution. Each level represents a stage with decisions made at one stage cascading and influencing subsequent ones.

To highlight the concept of compatibility better, a case study will be carried out in the following section.

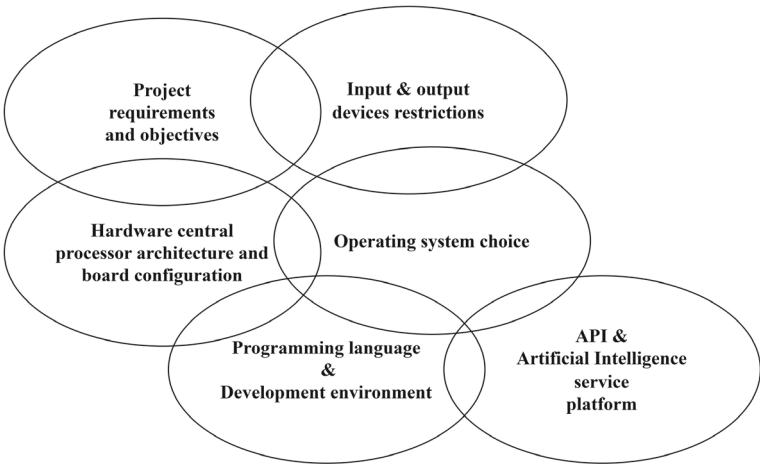


Fig. 1 The compatibility concept diagram. *Source* Made by the author

3 Results and Discussions

In this section, a case study focusing on the implementation of a kiosk will be proposed for discussion. The implementation of a kiosk (an autonomous electronic device often placed in a public space, providing users access to specific services or information) is requested. The kiosk's objective is to capture data entered by the user and print an order receipt. Given the compact size of this device, a professional printer dedicated to receipt printing will be used, specifically the Epson TM-T88V. Additionally, the kiosk needs to incorporate a small-sized computing system that represents a professional solution.

Current research focuses on utilizing SBCs for specific IoT solutions and AI functionalities. Article [18] proposes an IoT-based SBC prototype to automate hospital bed and stretcher management, addressing inefficiencies in resource allocation due to high demand. IoT technology streamlines processes, enhancing efficiency in medical institutions. Recent AI growth spans various domains, fueled by hardware accelerators from data centers to edge devices. Paper [19] compares FPGA and SBCs for deep neural network classification, emphasizing speed's importance.

For the case study implementation, three representative development boards in the IoT field will be considered, namely Raspberry Pi 3, considered the most popular device for enthusiasts [20], A20-OLinuXino-MICRO by Olimex, regarded as an experimental board for researchers [21], and Asus ThinkerBoard R2, being a professional board for the IoT domain [22].

Table 2 summarizes the technical specifications of the three development boards. These devices share the same release year and do not represent the latest versions to avoid increasing acquisition costs. The goal is to choose mid-range equipment with some limitations to analyze the solutions that arise in such choices.

This comparison of three single-board computers (SBCs)—Raspberry Pi 3b, Asus ThinkerBoard R2, and A20-OLinuXino-MICRO by Olimex (Table 2)—highlights their key specifications and suitability for different use cases.

The Raspberry Pi 3b, with its affordability (\$35), broad software support, and versatile I/O, remains a popular choice for beginners and hobbyists.

The Asus ThinkerBoard R2 offers a significant performance boost with its quad-core 1.8 GHz CPU, 4 GB RAM, and more robust I/O, making it suitable for more demanding applications like machine learning and robotics.

The A20-OLinuXino-MICRO stands out for its low price (\$50) and compatibility with a wide range of Linux distributions, making it a cost-effective option for basic SBC projects.

Ultimately, the choice of SBC depends on individual needs and budget. For those seeking a balance of affordability, versatility, and community support, the Raspberry Pi 3b remains a strong contender. For users requiring enhanced performance and expandability, the Asus ThinkerBoard R2 is a compelling choice. For those on a tight budget, the A20-OLinuXino-MICRO offers a reliable and affordable option.

Based on the previous comparison, the Asus ThinkerBoard R2 board will be used for the case study. The Asus ThinkerBoard R2 is presented in Fig. 2.

Table 2 Comparative analysis of the three proposed boards

Feature	Raspberry Pi 3b	Asus ThinkerBoard R2	A20-OLinuXino-MICRO bu Olimex	Favorite
Processor	Broadcom BCM2837B0	Rockchip RK3399	Allwinner H5	Asus ThinkerBoard R2
CPU cores	4	4	4	Tie
CPU frequency	1.4 GHz	1.8 GHz	1.2 GHz	Asus ThinkerBoard R2
GPU	Broadcom VideoCore IV	Mali-T760 MP2	Mali-400MP	Asus ThinkerBoard R2
RAM	1 GB	4 GB	1 GB	Asus ThinkerBoard R2
Storage	Memory Card	Memory Card	Memory Card	Tie
I/O	40 GPIO pins, 1 HDMI port, 1 Ethernet port, 1 USB 2.0 port	40 GPIO pins, 1 HDMI ports, 2 Ethernet ports, 4 USB 2.0 ports	40 GPIO pins, 1 HDMI port, 1 Ethernet port, 1 USB 2.0 port	Asus ThinkerBoard R2
OS	Raspbian, Ubuntu, and other Linux distributions	Android, Ubuntu, and other Linux distributions	Android, Ubuntu, and other Linux distributions	Tie
Price	\$35	\$107	\$50	Raspberry Pi 3b

Source Made by the author based on [20–22]

Asus ThinkerBoard represents a professional solution for an IoT environment. Choosing a professional development board emerges as a good decision, particularly in the integration of complex systems like those in the realm of IoT technologies. Professional development boards, tailored for specific applications, come equipped with enhanced capabilities that cater to the intricate requirements of IoT environments. These boards often boast advanced processing power, extensive connectivity options, and specialized hardware features designed to streamline the integration process within complex ecosystems. One key advantage of professional boards is their reliability. These boards undergo rigorous testing and quality assurance processes and are manufactured to industry standards, ensuring stability and performance under diverse conditions. In contrast, non-professional or community-developed boards, such as Raspberry or Olimex manufacturer, may lack the same level of scrutiny, potentially leading to issues in long-term reliability and compatibility within a complex IoT infrastructure.

Interoperability is another key consideration. Professional boards are often designed with industry standards in mind, facilitating seamless integration with a

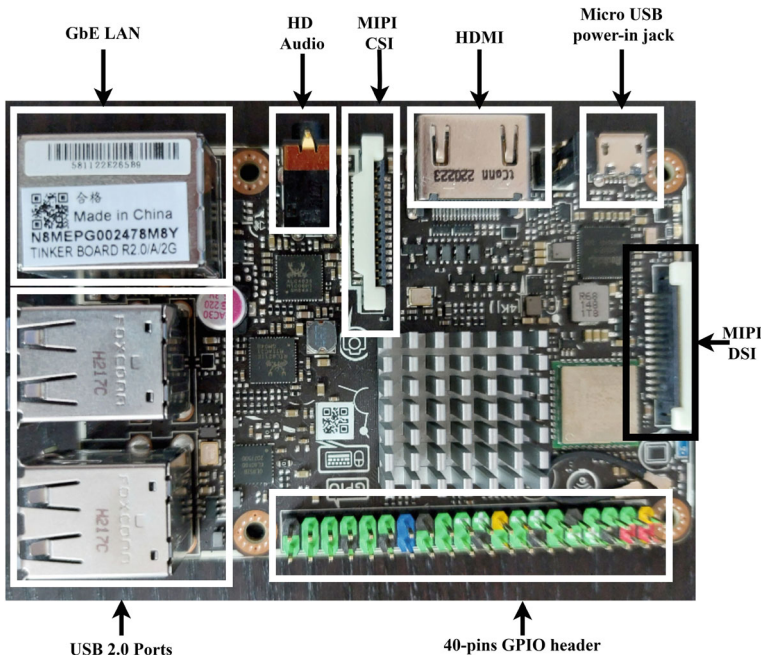


Fig. 2 Top view of the Asus ThinkerBoard R2. *Source* Made by the author based on [22]

variety of devices and technologies. This interoperability is crucial in IoT applications where diverse devices and protocols need to work cohesively within the same ecosystem.

Moreover, the long-term support and availability of professional boards are generally superior to their non-professional counterparts. As IoT applications evolve, having a board with ongoing support and compatibility updates becomes crucial to keeping the entire system up-to-date and secure.

In the following, we will analyze the possibilities and difficulties encountered in integrating such a solution.

Stage 1: Project Objectives Identification:

- The physical design of the kiosk, includes an Asus Thinker Board, a keyboard for data input, a mouse for cursor movement, a monitor for data display, and a printer for receipt printing;
- Implementation of the software application managing user-input data. The data is collected and stored in a database. Subsequently, other waiting users are identified, and an order receipt is generated.

Stage 2: Identification of Physical Equipment and, if Applicable, Technical Documentation for Each Piece of Equipment:

- Asus Thinker Board R2 development board [22];

- Epson TM-T88V Printer [23] presented in Fig. 3;
- Analyzing the technical documentation of the printer reveals that it has an exclusive device driver for Windows OSs. However, examining the technical documentation of the Asus board shows that it lacks a Windows OS. Considering this compatibility gap, a compromise must be sought. Opting for an alternative developer for the development board is not a viable solution since the printer does not provide an OS compatible with any device featuring such an ARM processor. Therefore, the only solution in this situation is to develop a driver that enables the use of the chosen OS.

Stage 3: Configuration of the hardware infrastructure. In this example, choosing a computing system with a processor that adheres to the printer's specifications is not feasible, as the problem's prerequisites would necessitate an Intel or AMD processor, which is not available for SBCs. The Epson TM-T88V printer is shown in action in Fig. 3.

Accessing the official reference at [24] reveals the available OSs for the selected board model:

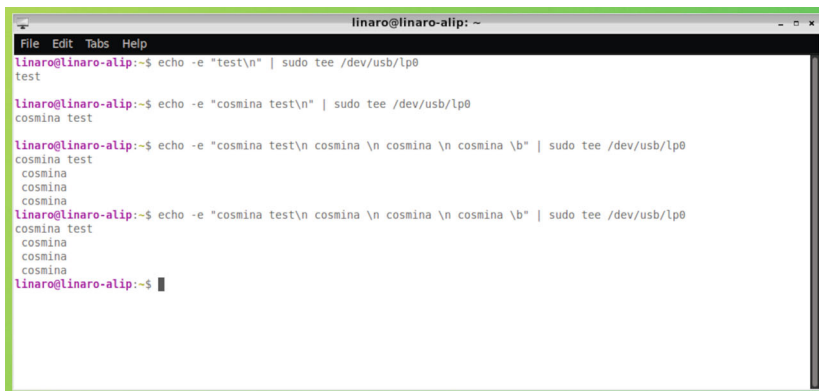
- Tinker OS, representing a Debian distribution;
- Debian for the ARM architecture;
- Android for the ARM architecture.

The following will consist of a comparative analysis of the three OSs, sequentially installed and tested on the Asus ThinkerBoard R3. The evaluation will encompass both user and programmer perspectives. In Table 3, the characteristics of the three OSs available for the Asus ThinkerBoard are presented. Some features are assessed on a scoring basis, with 1 representing poor, 2—good, and 3—excellent. The parameters analyzed in Table 3 are important considerations for choosing an OS for the SBCs. The values in the table are based on the author's experience with each OS.

Table 3 was compiled based on the author's experience, who tested the three working environments. Tinker OS and Debian are both excellent choices for programmers who want an OS with a wide range of software support and development tools. Tinker OS offers the additional benefit of customization options, while Debian offers the widest range of software support. Android is a good choice for users who want an easy-to-use OS with a wide range of features and applications. It is also the right decision for users who want to run mobile apps on their ThinkerBoard R2. However, processing speed is hindered in the case of the Android OS, particularly for the basic configuration of the SBC.

Analyzing Table 3 from the perspective of the primary purpose of SBCs, which is to host dedicated applications with limited functionalities, it is observed that Thinker OS represents the best solution for the Asus ThinkerBoard.

OS tailored for a specific SBC consistently outshines their community-developed or adapted counterparts by offering advanced features. This distinction arises from the meticulous customization and optimization carried out by dedicated developers to align the OS seamlessly with the unique hardware and capabilities of the particular board. Dedicated OS are crafted to harness the full potential of the underlying



```
linaro@linaro-alip: ~  
File Edit Tabs Help  
linaro@linaro-alip:~$ echo -e "test\n" | sudo tee /dev/usb/lp0  
test  
linaro@linaro-alip:~$ echo -e "cosmina test\n" | sudo tee /dev/usb/lp0  
cosmina test  
linaro@linaro-alip:~$ echo -e "cosmina test\n cosmina \n cosmina \n cosmina \b" | sudo tee /dev/usb/lp0  
cosmina test  
cosmina  
cosmina  
cosmina  
linaro@linaro-alip:~$ echo -e "cosmina test\n cosmina \n cosmina \n cosmina \b" | sudo tee /dev/usb/lp0  
cosmina test  
cosmina  
cosmina  
cosmina  
linaro@linaro-alip:~$
```

(a)



(b)

Fig. 3 The Epson TM-T88V printer with Thinker OS, running the CUPS printing service at the console level: **a** Screenshot with the terminal running the commands; **b** A picture of the EPSON device printing the messages transmitted from the console. *Source* Made by the author

Table 3 The characteristics of the three OSs

Feature	Tinker OS	Debian	Android	Winner
Programmer perspective				
Software support	3	2	2	Thinker OS
Development tools	3	2	1	Thinker OS
IDEs	Visual Studio Code, Eclipse, Atom, etc	Visual Studio Code, Eclipse, Atom, etc	Android Studio, Visual Studio Code, etc	–
Compilers	GCC, Clang, Rust, etc	GCC, Clang, Rust, etc	GCC, Clang, Rust, etc	–
Debuggers	GDB, LLDB, GDBTUI, etc	GDB, LLDB, GDBTUI, etc	GDB, LLDB, GDBTUI, etc	–
Other	Python, Node.js, Rust, etc	Python, Node.js, Rust, etc	Python, Node.js, Rust, etc	–
Signal processing speed in GUI	3	3	2	Thinker OS & Debian
Installation time	10 min	25 min	30 min	Thinker OS
User perspective				
Ease of use	2	2	3	Android
Features	2	3	2	Debian
Performance	2	3	2	Debian

Source Made by the author

hardware. By contrast, community-developed or adapted OS may lack this tailored optimization, leading to suboptimal utilization of the board’s capabilities. The dedicated OS is often equipped with specialized drivers, utilities, and tools specifically designed to complement the features of the board. This level of integration ensures the overall functionality of the development board.

Additionally, dedicated OS typically receives regular updates and support directly from the board’s manufacturer or a specialized development team. This ensures that the operating system stays current with the latest advancements, security patches, and optimizations. Conversely, community-developed systems may experience delays in updates or lack ongoing support, potentially leaving the board vulnerable to security risks or missing out on new IoT-specific features. Unnecessary components or functionalities that might be present in generic systems are often eliminated, optimizing the utilization of the IoT board’s resources and contributing to a more efficient operation within the broader IoT ecosystem.

Identifying and configuring I/O devices on an SBC, such as the case with the Asus Tinker Board development board, involves both general-purpose devices (such as the mouse, keyboard, and monitor), professional devices (such as printers, scanners, audio playback devices, etc.), or specialized devices (sensors and actuators).

Regarding general-purpose devices, these connect through USB ports that are automatically detected by the OS through plug-and-play functions. To connect a monitor to the Tinker Board, an HDMI cable is required.

Professional devices, such as printers, scanners, medical devices, or any other type of specialized equipment, require compatibility with both the OS and the system's processor to which they are connected. This means that specialized drivers must exist to enable communication between the OS and the equipment.

Typically, manufacturers of specialized devices provide drivers for popular OSs such as Windows, macOS, or Linux distributions. Users need to install these drivers to use the equipment. In addition to compatibility with the OS, the equipment must be compatible with the hardware architecture of the connected system. This implies that the device's hardware (such as the processor and communication interfaces) must be able to interact properly with the specialized device.

In the case of the Asus Thinker Board, the class of equipment with which the board can interact is limited. Limitations arise from the lack of drivers for the ThinkerOS OS and the 32-bit Rockchip processor architecture of the R2 version. Currently, most processors use 64-bit registers. Rockchip processors use a RISC-type ARM architecture, making them challenging to use with specialized equipment. Additionally, developing dedicated applications is challenging due to a lack of compatibility with modern frameworks.

Stage 4: Analysis of software environment options. Since the project requires the development of a script capable of communicating with the Epson printer in a ThikerOS configuration, derived from a Unix OS, the choice of PL must be one specific to scripting, such as Bash. Concerning the user application, the template used must have a Linux-style graphical interface, meaning that the Visual Studio environment is not an optimal choice, even in a Core-type selection, as it lacks templates with a portable graphical interface on a Linux system. A viable solution in this regard could be the QtCreator environment with the C++ programming language. On another note, the C++ programming language remains a reference for developing applications that need to run on a Linux kernel-based OS.

The compatibility process unfolded as follows:

- The requirement to create a small-sized device (kiosk) = > use of a development board (such as the Asus Thinker Board) = > Linux OS
- Printing equipment = > device driver incompatible with the Linux OS = > custom driver = > Bash script software component
- Linux graphical interface and OS = > compatible PL and DE.

Due to the lack of an Epson printer driver for ThinkerOS, a script was implemented that employs the CUPS2 service. CUPS2, or Common Unix Printing System 2, is a modular printing system designed for Unix-like OSs. It provides a framework for managing print jobs, handling printer queues, and supporting various printing protocols. CUPS2 facilitates the integration of printers with Unix-based systems by offering a standardized and extensible platform for print-related functionalities.

The code for the Bash script that allows the printer to display the message "Cosmina test" and on a new line "Cosmina" is:

```
#!/bin/sh  
echo -e "Cosmina test\n Cosmina\n" | sudo tee /dev/usb/lp0
```

The script will be interpreted using the shell and generate a text string that contains the specified message. It will pipe the generated string (the message) to the tee command, which, in turn, writes these data to the /dev/usb/lp0 device. The sudo command is used to obtain superuser privileges since writing directly to the printer device requires special permissions.

However, there are situations where the compatibility process must be carried out within an existing environment. Under these conditions, a solution must be found to harmonize new technologies with those already implemented.

4 Conclusions

This chapter focuses on various aspects related to how embedded systems relate to the IoT domain. Additionally, the hardware-software implementation challenges of embedded systems as part of the IoT domain have been analyzed. The concept of technology compatibility in embedded systems was introduced in this chapter, and two studies were conducted on the selection of the development board and operating system about the requirements of the embedded system as part of an IoT network.

Subsequently, a case study was examined, which required the use of a professional output device, and all the specific challenges of the problem were reported.

The author proposed a four-stage compatibility process, commencing with the analysis of the project's long-term requirements and objectives, followed by the identification of the equipment to be used in the physical realization of the project, the configuration of the hardware infrastructure, and concluding with the analysis of software environment options. Additionally, the case study underscored that opting for a professional development board is always a prudent choice when aiming for integration into a complex system, as is the case with IoT technologies. Furthermore, the case study demonstrated that operating systems dedicated to a specific board consistently offer more features than those developed either by the community or adapted to that particular board.

Future research can be centered on how these technologies need to be selected for the integration of artificial intelligence services at the application level hosted on such an embedded system, further increasing the integration constraints. A major issue identified by the author pertains to how research findings transition into production and the feasibility of this process.

References

1. Kannadhasan S, Venusamy K, Nagarajan R (2023) Recent trends in machine learning techniques, challenges and opportunities. In: Rani S, Bhambri P, Kataria A, Khang A, Sivaraman AK (eds) Big data, cloud computing and IoT, 1st edn. CRC Press, Boca Raton, pp 81–91. <https://doi.org/10.1201/9781003298335-6>
2. Poongodi T, Gopal R, Saini A (2021) IoT Architecture, communication technologies, and its applications. In: Kumar R, Wang Y, Poongodi T, Imoize AL (eds) Internet of things, artificial intelligence and blockchain technology. Springer, Cham, pp 1–24. https://doi.org/10.1007/978-3-030-74150-1_1
3. Maithani M, Meher D, Gupta S (2023) Multilingual text recognition system. In: Chakravarthy V, Bhateja V, Flores Fuentes W, Anguera J, Vasavi KP (eds) Advances in signal processing, embedded systems and IoT . Lecture notes in electrical engineering, vol 992. Springer, Singapore, pp 103–114. https://doi.org/10.1007/978-981-19-8865-3_9
4. Nuseir MT, Aljumah A, Urabi S, Kurdi BA, Alshurideh M (2023) Effects of social media marketing on consumer perception in liverpool, UK. In: Alshurideh M, Al Kurdi BH, Masa'deh R, Alzoubi HM, Salloum S (eds) The effect of information technology on business and marketing intelligence systems. Studies in computational intelligence, vol 1056. Springer, Cham, pp 45–63. https://doi.org/10.1007/978-3-031-12382-5_3
5. Veeman D, Varsha Shree M, Swapna Sai M, Kancharla A, Hussain CM (2022) AI in acquisition, analysis and processing of medical signal data collected by wearable devices. In: Mishra S, González-Briones A, Bhoi AK, Mallick PK, Corchado JM (eds) Connected e-Health. Studies in computational intelligence, vol 1021. Springer, Cham, pp 125–145. https://doi.org/10.1007/978-3-030-97929-4_6
6. Biji G (2021) White blood cell components separation for hematological disease detection. In: Komanapalli VLN, Sivakumaran N, Hampannavar S (eds) Advances in automation, signal processing, instrumentation, and control. i-CASIC 2020. Lecture notes in electrical engineering, vol 700. Springer, Singapore, pp 25–34. https://doi.org/10.1007/978-981-15-8221-9_3
7. Khoumeri EH, Cheggou R, Ferhah K (2020) An IoT based e-health platform using Raspberry Pi. In: Arai K, Kapoor S, Bhatia R (eds) Advances in information and communication. FICC 2020. advances in intelligent systems and computing, vol 1129. Springer, Cham, pp 257–271. https://doi.org/10.1007/978-3-030-39445-5_20
8. Disha DN, Seema S, Shastri KA (2022) Prediction of Autism in children with down's syndrome using machine learning algorithms. In: Shetty DP, Shetty S (eds) Recent advances in artificial intelligence and data engineering. Advances in intelligent systems and computing, vol 1386. Springer, Singapore, pp 109–122. https://doi.org/10.1007/978-981-16-3342-3_9
9. Anusha Bai R, Sangeetha V (2023) A comparative study on brain intracerebral hemorrhage classification using head CT scan for stroke analysis. In: Ranganathan G, EL Alloui Y, Piramuthu S (eds) Soft computing for security applications. ICSCS 2023. Advances in intelligent systems and computing, vol 1449. Springer, Singapore, pp 633–649. https://doi.org/10.1007/978-981-99-3608-3_44
10. Milenkovic M (2020) In: Internet of things: concepts and system design. Springer, Cham. <https://doi.org/10.1007/978-3-030-41346-0>
11. Li J, Hu Y (2023). A proposed method for using edge computing to secure existing IoT devices. In: Dong J, Zhang L (eds) IoT CIT 2022: proceedings of the international conference on internet of things, communication and intelligent technology, Changsha, August, 2022. Lecture notes in electrical engineering, vol 1015. Springer, Singapore, pp 774–781. https://doi.org/10.1007/978-981-99-0416-7_79
12. Wolf M (2019) Embedded system interfacing, design for the internet-of-things (IoT) and cyber-physical systems (CPS), 1st edn. Morgan Kaufmann, Cambridge, United States

13. Sarmiento M, Nguyen K-D, Duran C, Serrano R, Hoang T-T, Ishibashi K, Pham C-K (2022) Systems on a chip With 8 and 32 bits processors in 0.18- μ m technology for IoT applications. *IEEE Trans Circuits and Syst II: Express Briefs* 69(5):2438–2442. <https://doi.org/10.1109/TCSII.2022.3161494>
14. Guerreiro M, Kharade S, Dos Santos P, Liu S (2022) Power electronics control with system-on-a-chip-based platforms. In: 2022 IEEE 20th international power electronics and motion control conference, Brasov, Romania, 25–28 September 2022, pp 353–359. <https://doi.org/10.1109/PEMC51159.2022.9962949>
15. Sitnikovski B, Stojcevska B (2022) Integrating a virtual machine on a system-on-a-chip. In: 2022 57th international scientific conference on information, communication and energy systems and technologies (ICEST), Ohrid, North Macedonia, 16–18 June 2022, pp 1–2. <https://doi.org/10.1109/ICEST55168.2022.9828788>
16. Rosca C-M, Stancu A, Ariciu A-V (2024) Algorithm for child adoption process using artificial intelligence and monitoring system for children. *Internet of Things* 26. <https://doi.org/10.1016/j.iot.2024.101170>
17. IBM (2023) What is the internet of things (IoT)?. <https://www.ibm.com/topics/internet-of-things>. Accessed 25 Jan 2024
18. Velasco EP, Solarte SAD, Burbano ÁHP, Pardo EO, Guerrero CDR (2022) IoT-based SBC prototype for automating and managing hospital beds or stretchers. In: 2022 17th Iberian conference on information systems and technologies, Madrid, Spain, 22–25 June 2022, pp 1–8. <https://doi.org/10.23919/CISTI54924.2022.9820538>
19. Tamburello M, Caruso G, Adami D, Giordano S (2023) Experimental comparison between SBC and FPGA for embedded neural network acceleration. In: ICC 2023—IEEE international conference on communications, Rome, Italy, 28 May—01 June 2023, pp 6078–6083. <https://doi.org/10.1109/ICC45041.2023.10279670>
20. Raspberry Pi Ltd (2023) Raspberry Pi 3 Model B+. <https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf>. Accessed 29 Jan 2024
21. Allwinner Technology Co., Ltd. (2013) A20 User Manual Revision 1.0. <https://github.com/OLIMEX/OLINUXINO/blob/master/DOCUMENTS/A20-PDFs/A20%20User%20Manual%202013-03-22.pdf>. Accessed 30 Jan 2024
22. ASUS (2021) TINKER BOARD R2.0 Series. <https://tinker-board.asus.com/documentation/tbr2.html#qsg>. Accessed 30 Jan 2024
23. EPSON (2021) EPSON Advanced Printer Driver for TM-T88V. https://www.epson-biz.com/modules/pos/index.php?page=single_soft&cid=6898&pcat=3&pid=36. Accessed 30 Jan 2024
24. ASUS (2023) Tinker OS for Tinker Board R2.0. <https://tinker-board.asus.com/download-list.html?product=tinker-board-r2>. Accessed 30 Jan 2024

Chapter 5

Investigation of the Effect of Lateral Displacement and Damping Limits of Optimum Designed Seismic Base Isolators with Flower Pollination Algorithm on Structural Control



Ayla Ocak, Sinan Melih Nigdeli, Gebrail Bekdaş, and Xin-She Yang

1 Introduction

Base isolator systems are control systems that are generally positioned between the ground and the base of the structure for the seismic control of structures. They absorb the energy of the earthquake by acting with certain flexibility. They are known as horizontally moving ductile systems for the effects created by the seismic waves of the earthquake, which have destructive effects, by going for vertical rigid designs. To provide this flexibility, the period and mass of the isolator are adjusted. Isolators, which are designed with a weight of approximately one floor, are less preferred in multi-story structures because they can move freely horizontally. Rubber-based, slippery, mixed, or small-slippery can be designed, as well as spring-type isolator systems are designed in cases where vertical earthquake effects need to be isolated [1]. The characteristics of the material used for the carrier of the isolator come to the fore. The material contained in rubber-based isolators directly affects the load-carrying capacity. In recent years, studies have been carried out for alternative less costly designs such as an alternative to classical rubber material, sand-tire mixture, the use of scrap tires, and recycled rubber [2–4]. Gravel, paraffin wax, asphalt roofing felt,

A. Ocak · S. M. Nigdeli · G. Bekdaş (✉)

Department of Civil Engineering, Istanbul University—Cerrahpaşa, 34320 Avcılar, Istanbul, Turkey

e-mail: bekdas@iuc.edu.tr

S. M. Nigdeli

e-mail: melihnig@iuc.edu.tr

X.-S. Yang

Department of Design Engineering and Maths, Middlesex University, London, BT, UK

e-mail: x.yang@mdx.ac.uk

graphite, geotextile, green marble, natural limestone sand, rubberized mortar with rubber aggregate, and heated asphalt, instead of talc and sand, which are the early materials used as an isolation layer, in sliding floor isolation systems, since the basis of isolation is based on friction. The use of materials such as asphalt sand cushions has been investigated [5–10].

In the design of isolator systems, vertical seismic effects are generally neglected by considering the horizontal waves of the earthquake. Xiaohang et al. investigated the possibilities of structural collapse of a seismically isolated transportation hub by considering horizontal and vertical seismic motion effects [11]. According to the seismicity of the structure region, the type of isolator is selected. Isolator ductility is determined by considering random earthquakes belonging to the region. On the other hand, an isolator designed more ductile than necessary will move too much in a very severe earthquake and threaten the safety of the structure [12, 13]. The decision of the isolator period is important for the correct determination of the randomly determined ductility level. The correct selection of the period and the sufficient damping ratio are the basic requirements to ensure the rigidity of the system. Optimizing these choices will also optimize the safety and seismic control of the isolated structure.

Metaheuristic algorithms are optimization method that facilitates the application due to their simple and understandable theory using instinctive behavior. As examples, Genetic algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Harmony Search Algorithm (HS), Artificial Bee Colony (ABC), Bat Algorithm (BA), Teaching–Learning Based Optimization (TLBO), Flower Pollination Algorithm (FPA), Gray Wolf Optimization (GWO), and Whale Optimization Algorithm (WOA) are some metaheuristic algorithms produced in nature based on living behavior [14–23]. Its use in the optimization of isolators has become widespread in last years [24–31].

The flower pollination algorithm (FPA) is generated from the process in which flowers are pollinated spontaneously or with the help of a pollinator such as a bird or a bee [21]. There are studies in which it is used in various engineering problem applications, especially in the optimization of structural control elements [32–36].

In this chapter, far-fault earthquake records listed in FEMA P-695 containing various magnitude earthquakes were exited on a single-degree-of-freedom (SDOF) structural model [37]. Via placing a seismic isolator on the base of the constructed structure model, the model was analyzed on MATLAB with Simulink [38]. The isolator system is optimized with the flower pollination algorithm under seismic excitations for 10, 20, 30, 40, and 50% damping ratios and displacement limits of 20, 25, 30, 40, and 50 cm. Optimum isolator parameters were optimized to minimize the objective function defined as total acceleration by considering the displacement limit for the base motion occurring in the structure.

2 Methodology

This section presents the motion equations governing the base isolators, the computation of fundamental isolator parameters, and the primary equations utilized in applying the flower pollination algorithm for optimization purposes.

2.1 Optimization of Seismic of Isolator System

In the seismic isolator design, the isolator mass is taken into account by taking approximately the mass of one floor of the superstructure. The total mass (m_{total}) of the isolator structure is used to calculate such as Eq. 1 by adding the mass of the isolator (m_b) to the mass of the structure ($m_{\text{structure}}$).

$$m_{\text{total}} = m_b + m_{\text{structure}}. \quad (1)$$

The basic parameters that determine the mobility of the base isolators are the isolator period. When the building model to be used is a SDOF system that moves with the isolator, parameters such as damping ratio period, damping coefficient, and stiffness of the isolator become common. Thus, with the addition of an isolator, the structural properties that are different at the beginning become a single value. The period (T_b), stiffness (k_b) and damping coefficient (c_b) calculations of the base-isolated system are given in Eqs. 2–4, respectively.

$$T_b = \frac{2\pi}{w_b}, \quad (2)$$

$$k_b = m_{\text{total}} \times w_b^2, \quad (3)$$

$$c_b = 2 \times \zeta_b \times m_{\text{total}} \times w_b. \quad (4)$$

Here, the natural angular frequency (w_b) of the isolator system is indicated by the damping ratio (ζ_b). Equation 5 displays the motion equation of a system featuring a single-degree-of-freedom isolator.

$$m_{\text{total}}\ddot{X} + c_b\dot{X} + k_bX = -m_{\text{total}}\ddot{X}_g. \quad (5)$$

X in Eq. 5 is the structure displacement, $\dot{X}$ velocity, $\ddot{X}$ acceleration and $\ddot{X}_g$, ground acceleration. In base isolator systems, deciding on the ductility of the isolator is important for structure safety and control efficiency. Optimizing the damping ratio and period of the isolator can be a good solution to bring the ductility to the desired level. An objective function must be given to the system for the optimization process. Its remarkable performance in damping the displacement of isolators, especially the

acceleration of the earthquake is known. In the parameters optimized to increase the control efficiency of the isolator system, the objective function is to minimize the displacement and acceleration of the system. In Eqs. 6 and 7, the objective functions used to minimize the total acceleration and the constraint used as base displacement of the system are given, respectively.

$$f(x) = \max(|\ddot{X}|), \quad (6)$$

$$g(x) = \max(|X|). \quad (7)$$

During the optimization process employing the flower pollination algorithm, initially, the fixed-value characteristics and design constraints of both the structure and the algorithm are outlined, akin to other optimization techniques. In the solution generation phase, the system incorporates lower and upper limits, as illustrated in Eq. 8.

$$X_{\min} < X_{i,j} < X_{\max} \quad \begin{matrix} i = 1, 2, \dots, nf \\ j = 1, 2, \dots, n \end{matrix} \quad (8)$$

$X_{i,j}$ Eq. 8 indicates the variable, $X_{\min}$ is the lower limit of the variable, $X_{\max}$ is the upper limit of the variable, nf is the population number, and n is the number of variables.

After the population and the number of iterations are determined for the optimization, the design factors of the algorithm are also introduced to the system. For the pollination algorithm, this value is the switch probability (sp). Then, the process of generating a solution begins. The optimized variable between the lower and upper limits is randomly assigned as in Eq. 9, and each value is saved in the solution matrix.

$$X_{i,j} = X_{\min} + \text{rand} (X_{\max} - X_{\min}). \quad (9)$$

After obtaining the first solutions, two methods are used to generate new solutions. Which of these to use depends on the switch probability value chosen initially. This value is a design factor of the pollination process, local or global search, in other words, pollination by itself or with the help of a pollinator. First, a random value is taken and compared with the key probability value. If this random value is greater than the chosen switch probability, the local optimization is started. Generating new solutions in local search is done through the equation as in Eq. 10.

$$X_{\text{new}} = X_{i,j} + \text{rand}(0, 1)(X_k - X_l). \quad (10)$$

Here, X_{new} refers to the matrix where new solutions are recorded, $X_{i,j}$ represents the old solution, X_k and X_l are two values randomly selected from the old solutions.

If the randomly selected number is less than the chosen switch probability, the global optimization starts. This situation shows that pollination takes place with the

help of a pollinator. New solutions can be produced by using the equation called Lévy flight made by pollinators such as birds and bees during pollination. The Lévy flight equation is shown in Eq. 11. The new solution is generated according to Eq. 12.

$$L = \frac{1}{\sqrt{2\pi}} \varepsilon^{-1.5} e^{\frac{1}{2\varepsilon}}, \quad (11)$$

$$X_{\text{new}} = X_{i,j} + L(X_{\text{best}} - X_{i,j}). \quad (12)$$

L in the equations indicates the Lévy flight. The solution with the best objective function value in the solution matrix is X_{best} .

After the new solutions are produced, they are compared with existing solutions and solutions with higher acceleration values are replaced with new ones in each iteration. The optimum solution is reached by repeating these operations as many iterations.

3 Numerical Examples

In this numerical example, a seismic base isolator is positioned at the bottom of a ten-story and it is assumed as a SDOF structure model. A simulation was conducted using the FEMA P-695 earthquake records in MATLAB, as outlined in references [37, 38]. Figure 1 illustrates the ten-story SDOF structure model. The mass, stiffness, and damping coefficients of the structure are set to 360 tons, 650 MN/m, and 6.2 MNs/m, respectively, according to reference [39].

The base-isolated SDOF structure model underwent optimization using the flower pollination algorithm (FPA) under earthquake excitations. During optimization, the period and damping ratio of the structural system, coupled with the isolator, were optimized. The period of the isolated system ranged from 1 to 5 s, while the damping ratio was optimized within the range of 1% to a maximum of 10, 20, 30, 40, and 50%. Displacement constraints in the design were set at 20, 25, 30, 40, and 50 cm. Table 1 showcases the design limits and algorithm-specific factors employed in the optimization process. Table 2 presents the optimal values for the isolator system.

The critical earthquake record within the FEMA dataset for the structure without an isolator was identified as DUZCE/BOL090. Critical earthquake analyses for the DUZCE/BOL090 earthquake were conducted utilizing the optimal system parameters obtained at the damping limits of 10, 20, 30, 40, and 50%, and displacement limits of 20, 25, 30, 40, and 50 cm, adding an isolator into the structure. The resulting structure displacement and total acceleration values are presented in Table 3.

For the critical earthquake analysis under the 10% damping limit, identical optimal results were attained across all displacement limits. Figure 2 illustrates the time-dependent graphs showing the structure displacement and total accelerations obtained for the 10% damping limit, with displacement limits of 20, 25, 30, 40, and 50 cm.

Fig. 1 a 2D model, b 3D model

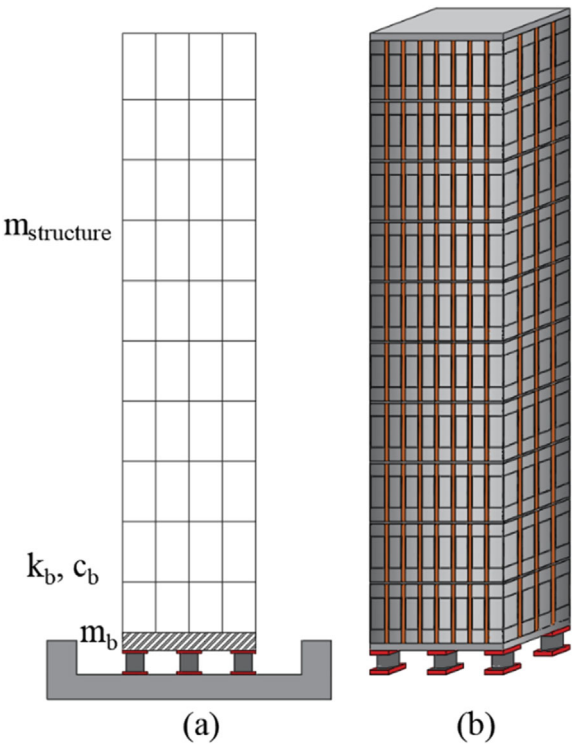


Table 1 Optimization design ranges and limits

Explanation	Design parameter
Isolator period	1–5 s
Isolator damping ratio	1–50%
Base displacement limit	20, 25, 30, 40, 50 cm
Population	10
Maximum iteration	200
Switch probability	0.5

The displacement and acceleration graphs resulting from the critical earthquake analysis at a 50% damping limit are given as follows: Fig. 3 illustrates the results for a displacement limit of 20 cm, Fig. 4 for 25 cm, Fig. 5 for 30 cm, Fig. 6 for 40 cm, and Fig. 7 for 50 cm, respectively.

Table 2 Optimum results of different case limitations

Variables		Damping ratio											
		For 10%		For 20%		For 30%		For 40%		For 50%			
		T_b (s)	ζ_b (%)	T_b (s)	ζ_b (%)	T_b (s)	ζ_b (%)	T_b (s)	ζ_b (%)	T_b (s)	ζ_b (%)	T_b (s)	ζ_b (%)
Displacement limit (m)	0.20	1.8240	10.00	1.4794	20.00	1.9202	30.00	2.1051	40.00	2.3157	50.00		
	0.25	1.8240	10.00	1.8927	20.00	1.9797	30.00	2.3929	40.00	2.6623	50.00		
	0.30	1.8240	10.00	1.8927	20.00	1.9797	30.00	2.6866	40.00	3.0414	50.00		
	0.40	1.8240	10.00	1.8927	20.00	2.8975	30.00	3.3026	40.00	3.6724	50.00		
	0.50	1.8240	10.00	2.9796	20.00	3.3952	30.00	3.8024	40.00	4.2966	50.00		

Table 3 Maximum responses for critical earthquake record

Damping ratio		Displacement limit (m)	Displacement (m)	Acceleration (m/s ²)
With isolator	10%	0.20	0.1635305	1.9718045
		0.25	0.1635305	1.9718045
		0.30	0.1635305	1.9718045
		0.40	0.1635305	1.9718045
		0.50	0.1635305	1.9718045
	20%	0.20	0.1602794	3.3648701
		0.25	0.1492046	2.0553210
		0.30	0.1492046	2.0553210
		0.40	0.1492046	2.0553210
		0.50	0.1372094	0.9474953
	30%	0.20	0.1385762	2.1991520
		0.25	0.1386310	2.1016463
		0.30	0.1386310	2.1016463
		0.40	0.1295347	1.1946032
		0.50	0.1273559	0.9340001
	40%	0.20	0.1281725	2.1209266
		0.25	0.1272473	1.7988978
		0.30	0.1245290	1.5483557
		0.40	0.1190686	1.1769276
		0.50	0.1316215	0.9676783
	50%	0.20	0.1188434	2.0815790
		0.25	0.1171452	1.7699054
		0.30	0.1129198	1.5112335
		0.40	0.1214041	1.2008699
		0.50	0.1325077	0.9861218
Without isolator			0.4101091	19.2833064

4 Discussion and Conclusions

In this study, the optimum isolator parameters were determined by using flower pollination algorithm optimization for five different damping and five different displacement limits by adding an isolator to a ten-story rigid SDOF structure under various earthquake excitations. Critical earthquake analyses were conducted to assess the impact of the isolator's damping ratio and mobility on the lateral displacement induced by the earthquake. The displacement and total acceleration reduction percentages of the isolator system in the critical earthquake analysis are presented in Table 4.

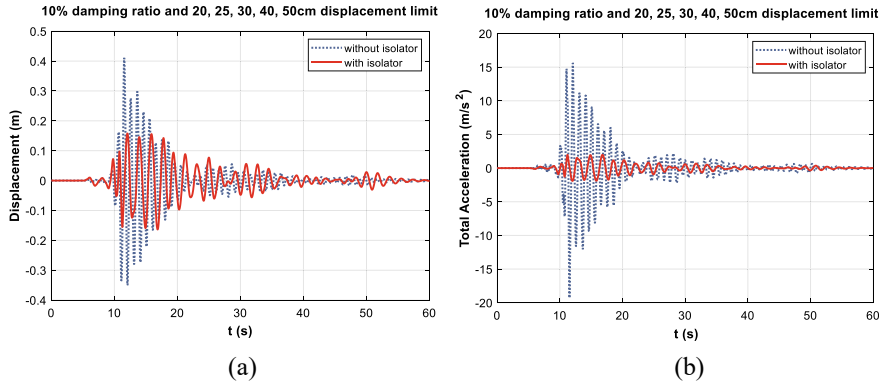


Fig. 2 Critical earthquake record plots for 10% damping and 20, 25, 30, 40, and 50 cm displacement limit: **a** displacement, **b** acceleration

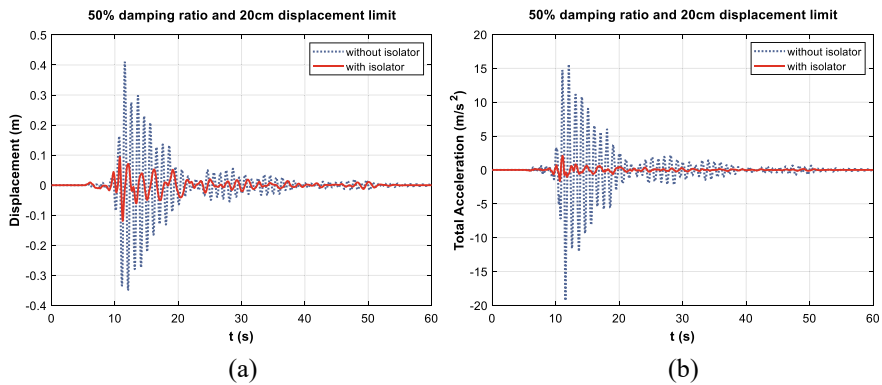


Fig. 3 Critical earthquake record plots for 50% damping and 20 cm displacement limit: **a** displacement, **b** acceleration

When Table 4 is examined, it is seen that the same control performance is observed for all displacement constraints at the damping limit as low as 10%. For the 20% damping limit, the same drop ratio was observed at 25, 30, and 40 cm, while the control efficiency increased as the displacement limit increased. As the displacement limit increases, the control performance increase is also seen at the 30% damping limit. While the displacement increase from 10, 20, and 30% damping limits showed little effect, the increase in the damping limit highlighted the effect of the change in displacement on the control. Optimum displacement reductions at 40% and 50% damping limits were observed at 40 cm and 30 cm, respectively. The optimum value for the 10, 20, and 30% damped system with lower damping was 50 cm. In this case, while maximum isolator mobility is desired for low-damping systems, it is understood that as the damping ratio increases, the need for isolator mobility decreases.

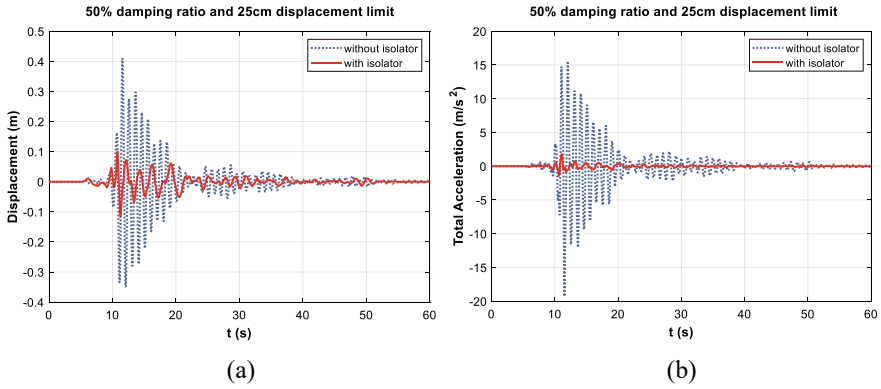


Fig. 4 Critical earthquake record plots for 50% damping and 25 cm displacement limit: **a** displacement, **b** acceleration

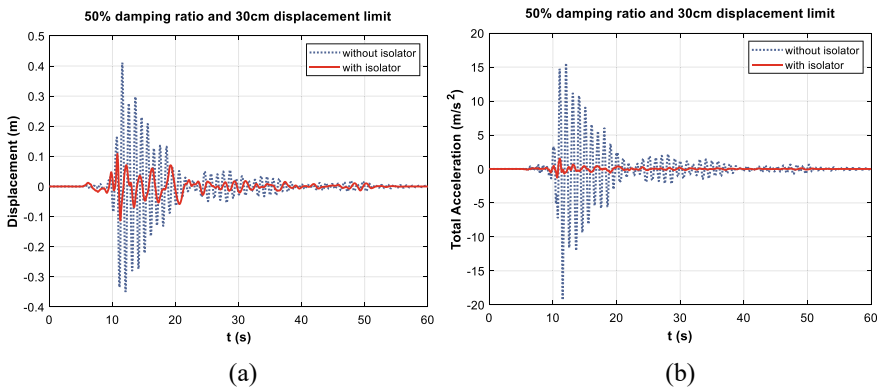


Fig. 5 Critical earthquake record plots for 50% damping and 30 cm displacement limit: **a** displacement, **b** acceleration

When all cases were examined, it was determined that the optimum total acceleration drops were at the maximum displacement limit of 50 cm. In this case, it can be said that in the displacement control of the structure, the mobility for low-damped isolators does not make much difference in control, while high-damped isolators can give better results at lower displacements.

Considering the variation of the optimum parameters according to the damping and displacement limit, it is seen that the increase in damping ratio extends the period of the isolator for all displacements, and the defined damping limits are reached to the limit values.

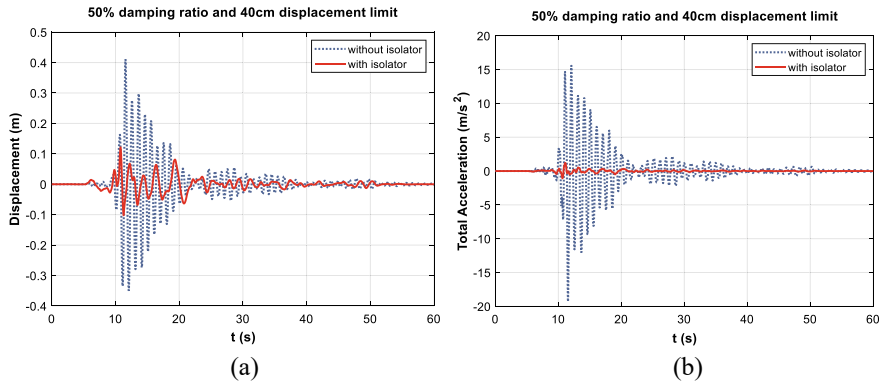


Fig. 6 Critical earthquake record plots for 50% damping and 40 cm displacement limit: **a** displacement, **b** acceleration

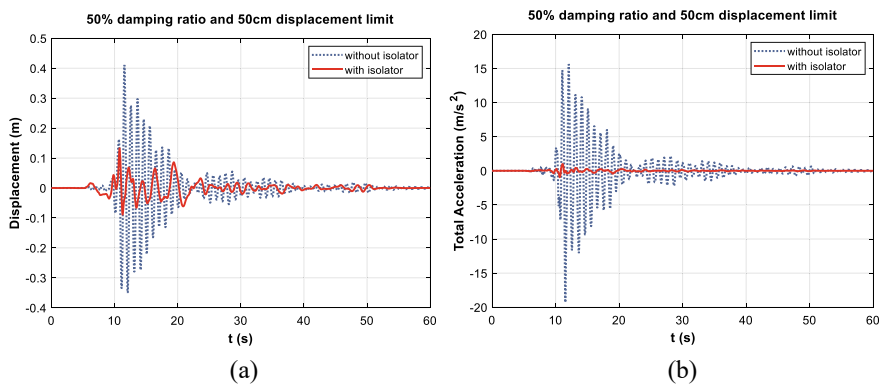


Fig. 7 Critical earthquake record plots for 50% damping and 50 cm displacement limit: **a** displacement, **b** acceleration

While the DUZCE/BOL090 earthquake stood as the critical earthquake record for the uncontrolled structure, the inclusion of an isolator altered the critical earthquake record of the system. Figure 8 displays the displacements in all FEMA P-695 earthquake records at the 10% damping limit. The variation of the structure displacement concerning the earthquake record is illustrated in Fig. 9 for 20% damping, Fig. 10 for 30% damping, Fig. 11 for 40% damping, and Fig. 12 for 50% damping, respectively.

Examining Figs. 8, 9, 10, 11, and 12, it is seen that in several earthquake records such as 8, 11, 17, 28, 30, and 38, the structure with an isolator exhibits higher displacement than the structure without an isolator across various damping ratios and displacement limits. This observation indicates that adding an isolator can indeed alter the critical earthquake for the structure. Specifically, when considering the 20% damping, some records surpass the structure's displacement, notably at the

Table 4 Structure responses reduction percentages

Damping ratio		Displacement limit (m) (%)	Displacement (%)	Acceleration (%)
With isolator	10	0.20	60.13	89.77
		0.25	60.13	89.77
		0.30	60.13	89.77
		0.40	60.13	89.77
		0.50	60.13	89.77
	20	0.20	60.92	82.55
		0.25	63.62	89.34
		0.30	63.62	89.34
		0.40	63.62	89.34
		0.50	66.54	95.09
	30	0.20	66.21	88.60
		0.25	66.20	89.10
		0.30	66.20	89.10
		0.40	68.41	93.81
		0.50	68.95	95.16
	40	0.20	68.75	89.00
		0.25	68.97	90.67
		0.30	69.64	91.97
		0.40	70.97	93.90
		0.50	67.91	94.98
	50	0.20	71.02	89.21
		0.25	71.44	90.82
		0.30	72.47	92.16
		0.40	70.40	93.77
		0.50	67.69	94.89

50 cm displacement limit. At the 30, 40, and 50% damping limits, isolator displacement exceeding the structure's displacement was predominantly observed at 40 and 50 cm displacement limits. Hence, it's apparent that enhancing the mobility of highly damped structures might result in greater displacement compared to the uncontrolled structure. The variation of the total acceleration in the structure concerning the earthquake record is depicted in Fig. 13 for the lowest damping limit of 10% and in Fig. 14 for the highest damping limit of 50%.

Upon examination of Figs. 13 and 14, it's apparent that the total acceleration value of the structure with the isolator does not exceed that of the uncontrolled structure, despite the alteration in critical earthquake caused by the addition of the isolator.

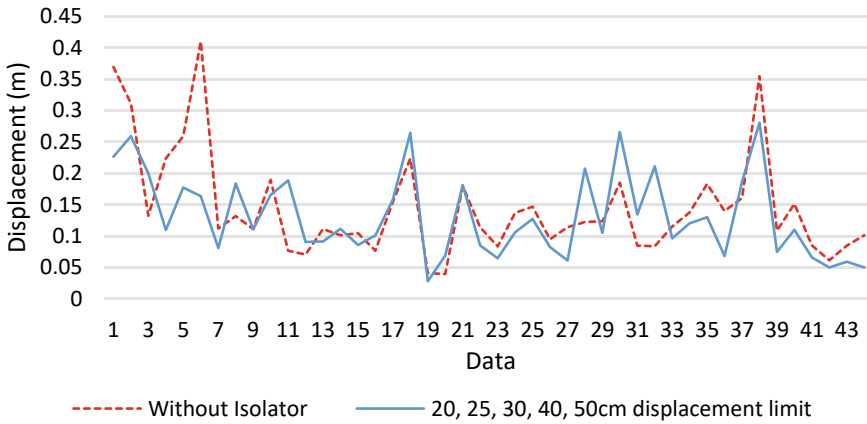


Fig. 8 Maximum base displacement for the 10% damping ratio limit

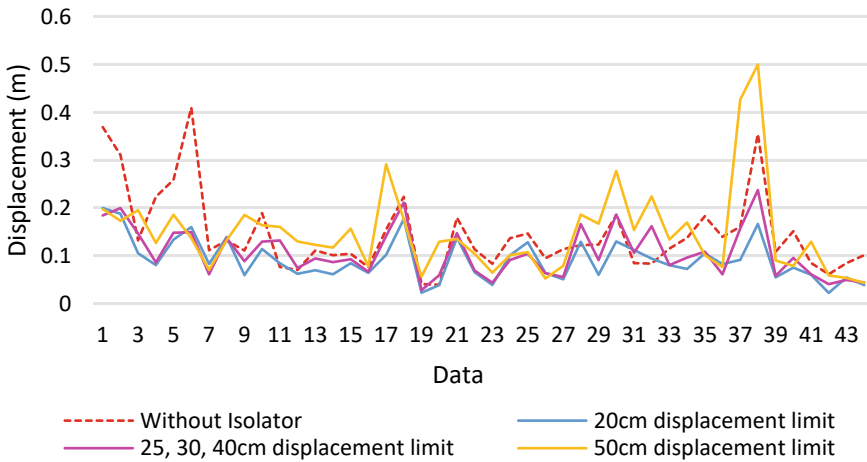


Fig. 9 Maximum base displacement for the 20% damping ratio limit

This suggests that while isolator systems may change the critical earthquake, their impact on total acceleration reduction is more significant than on displacement.

When considering all the results, the following conclusions can be drawn:

- Increasing the damping ratio of a system with isolators extends the system's period, and the system with isolator's experiences greater force due to reaching the limit values of the damping ratio.
- While augmenting the allowed isolator mobility in low-damping systems like 10 and 20% doesn't significantly affect control performance, it's observed that increasing the damping ratio in high damping systems reduces the isolator displacement requirement.

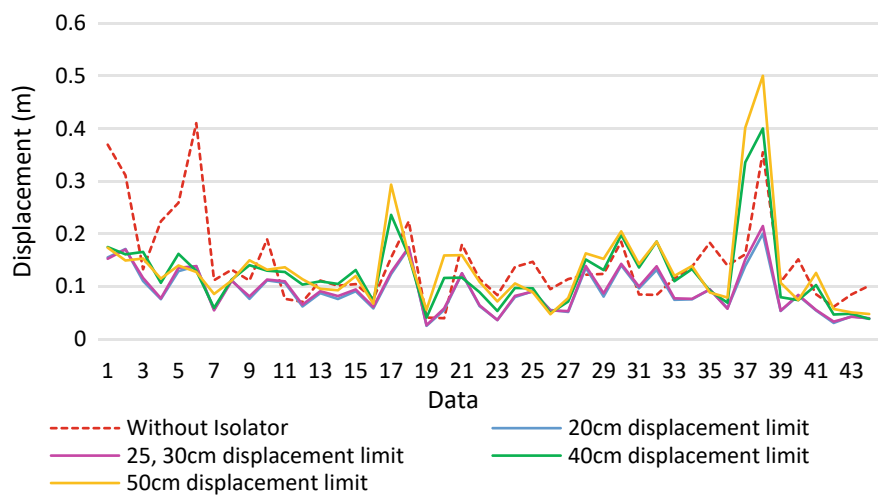


Fig. 10 Maximum base displacement for the 30% damping ratio limit

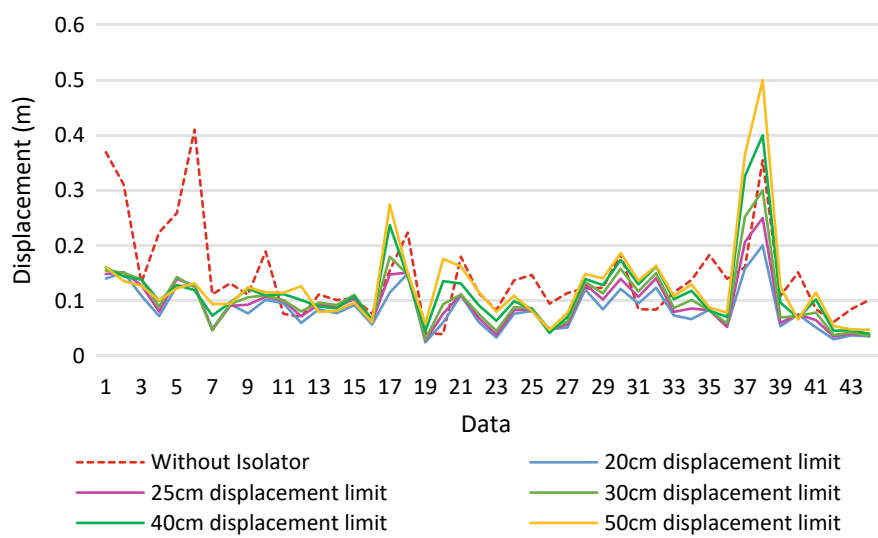


Fig. 11 Maximum base displacement for the 40% damping ratio limit

- Some systems with the same damping ratio exhibited similar total acceleration reductions. Optimal overall acceleration drops were observed when isolator mobility was at its highest. This indicates that the reduction of the total acceleration of the system depends more on displacement than on the damping ratio.
- Adding an isolator to the structure alters the critical earthquake for the structure. Although the total acceleration of the structure with isolators is scarcely affected

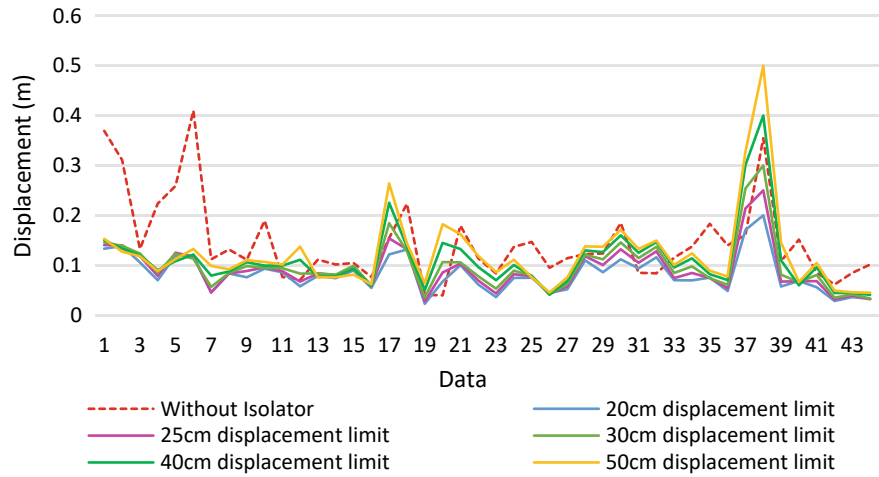


Fig. 12 Maximum base displacement for the 50% damping ratio limit

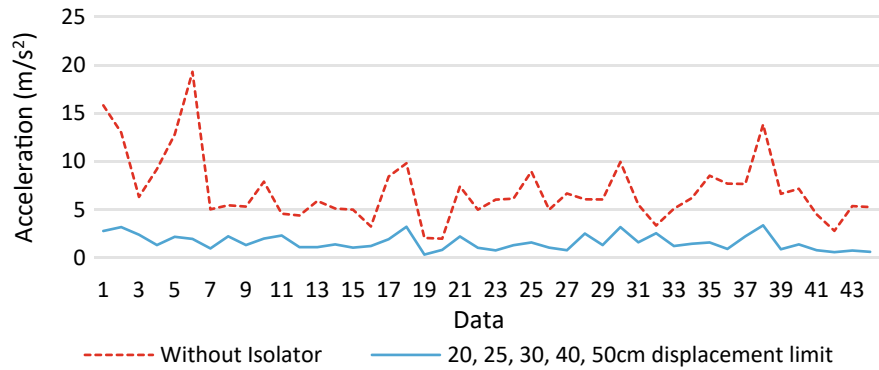


Fig. 13 Maximum total acceleration for the 10% damping ratio limit

by the change in the critical earthquake, this situation significantly impacts the displacement of the structure with the isolator, causing it to exceed that of the uncontrolled structure. Consequently, adding an isolator to the structure proves to be a more effective method in controlling total acceleration than displacement.

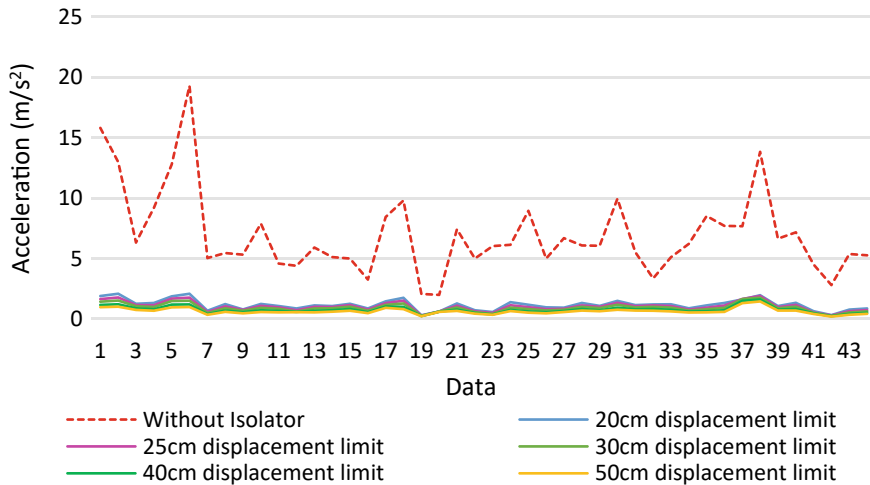


Fig. 14 Maximum total acceleration for the 50% damping ratio limit

References

1. Bakkaloğlu E (2018) Seismic isolator systems in hospital buildings the effects of the use decision on the building manufacturing process. Master's thesis, Istanbul Technical University, Institute of Science, Istanbul, Turkey
2. Tan KC, Hejazi F (2022) Rubber-bearing isolator with granular and polymer filler core and application on a building. In: Structures, vol 42. Elsevier, pp 309–332
3. Raj JCJ, Kumar MV (2022) Performance evaluation of eco-friendly scrap tire base isolation technology in distinct construction quality RC framed buildings located in a seismic risk zone. Sustain Energy Technol Assess 53:102511
4. Habieb AB, Formisano A, Milani G, Pianese G (2022) Seismic performance of unbonded fiber-reinforced elastomeric isolators (UFREI) made by recycled rubber. Influence of suboptimal crosslinking. Eng Struct 256:114038
5. Naeim F, Kelly JM (1999) Design of seismic isolated structures: from theory to practice. Wiley
6. Li Z, Rossow EC, Shah SP (1989) Sinusoidal forced vibration of sliding masonry system. J Struct Eng 115(7):1741–1755
7. Lou Y, Wang M, Su J (1992) Research of sliding shock absorbing multi-story brick building. In: Proceedings of 10th world conference on earthquake engineering, Barcelona, Spain
8. Nanda RP, Agarwal P, Shrikhande M (2012) Base isolation by geosynthetic for brick masonry buildings. J Vib Control 18(6):903–910
9. Banović I, Radnić J, Grgić N, Matešan D (2018) The use of limestone sand for the seismic base isolation of structures. Adv Civ Eng
10. Ali A, Zhang C, Bibi T, Zhu L, Cao L, Li C, Hsiao PC (2022) Investigation of five different low-cost locally available isolation layer materials used in sliding base isolation systems. Soil Dyn Earthq Eng 154:107127
11. Lou X, Huang Y, Lv Y, Huang Q (2022) Seismic performance of the series seismic isolation systems designed by the procedures of GB50011-2010 and ASCE/SEI 7-16. Case Stud Constr Mater e01184
12. Sheikh H, Van Engelen NC, Ruparathna R (2022) A review of base isolation systems with adaptive characteristics. In: Structures, vol 38. Elsevier, pp 1542–1555

13. Bao Y, Becker TC (2018) Inelastic response of base-isolated structures subjected to impact. *Eng Struct* 171:86–93
14. Holland JH (1975) *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI
15. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: *Proceedings of IEEE international conference on neural networks No. IV, 27 November–1 December Perth, IEEE Service Center, Piscataway, NJ, 1942–1948*
16. Dorigo M, Maniezzo V, Colormi A (1996) The ant system: an autocatalytic optimizing process. *IEEE Trans Syst Man Cybernet B* 26:29–41
17. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. *SIMULATION* 76(2):60–68
18. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization, vol 200, pp 1–10. Technical report-tr06, Erciyes University, engineering faculty, computer engineering department
19. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: *Nature-inspired cooperative strategies for optimization (NICSO 2010)*. Springer, Berlin, Heidelberg, pp 65–74
20. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput Aided Des* 43:303–315
21. Yang XS (2012) Flower pollination algorithm for global optimization. In: Durand-Lose J, Jonoska N (eds) *Lecture notes in computer science*, vol 27. Springer, London, 7445, 240–249
22. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
23. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
24. Zou XK, Chan CM (2001) Optimal drift performance design of base-isolated buildings subject to earthquake loads. *WIT Trans Built Environ* 54
25. Iemura H, Taghikhany T, Jain SK (2007) Optimum design of resilient sliding isolation system for seismic protection of equipment. *Bull Earthq Eng* 5(1):85–103
26. Jangid RS (2008) Equivalent linear stochastic seismic response of isolated bridges. *J Sound Vib* 309(3–5):805–822
27. Bucher C (2009) Probability-based optimal design of friction-based seismic isolation devices. *Struct Saf* 31(6):500–507
28. Dicleli M, Karalar M (2011) Optimum characteristic properties of isolators with bilinear force-displacement hysteresis for seismic protection of bridges built on various site soils. *Soil Dyn Earthq Eng* 31(7):982–995
29. Weber F, Distl H, Braun C (2016) Isolation performance of optimized triple friction pendulum. *Int Ref J Eng Sci* 5:55–69
30. Charnpis DC, Komodromos P, Phocas MC (2012) Optimized earthquake response of multi-story buildings with seismic isolation at various elevations. *Earthquake Eng Struct Dynam* 41(15):2289–2310
31. Ocak A, Nigdeli SM, Bekdaş G, Kim S, Geem ZW (2022) Optimization of seismic base isolation system using adaptive harmony search algorithm. *Sustainability* 14(12):7456
32. Nigdeli SM, Bekdaş G, Yang XS (2016) Application of the flower pollination algorithm in structural engineering. In: *Metaheuristics and optimization in civil engineering*. Springer, Cham, pp 25–42
33. Mergos PE (2021) Optimum design of 3D reinforced concrete building frames with the flower pollination algorithm. *J Build Eng* 44:102935
34. Yücel M, Bekdaş G, Nigdeli SM (2022) Metaheuristics-based optimization of TMD parameters in time history domain. In: *Optimization of tuned mass dampers*. Springer, Cham, pp 55–66
35. Bekdaş G, Kayabekir AE, Nigdeli SM, Toklu YC (2019) Transfer function amplitude minimization for structures with tuned mass dampers considering soil-structure interaction. *Soil Dyn Earthq Eng* 116:552–562
36. Ocak A, Bekdaş G, Nigdeli SM (2022) A metaheuristic-based optimum tuning approach for tuned liquid dampers for structures. *Struct Design Tall Spec Build* 31(3):e1907

37. FEMA P-695. Quantification of building seismic performance factors. Washington
38. The MathWorks, Matlab R2018a. Natick, MA (2018)
39. Singh MP, Singh S, Moreschi LM (2002) Tuned mass dampers for response control of torsional buildings. *Earthq Eng Struct Dynam* 31(4):749–769

Chapter 6

Review of Recent Advances on AI Applications in Civil Engineering



Yaren Aydın, Gebrail Bekdaş, and Sinan Melih Nigdeli

1 Introduction

Artificial intelligence (AI) is an automated system developed for jobs that require intelligence. Artificial intelligence is based on the principle that machines autonomously make inferences by gaining experience with learning methods and algorithms. In recent years, with the transition to the digital world, recording and collecting data has gained importance. Using artificial intelligence techniques to process and analyse the large amount of data collected, valuable information to be obtained from the data can be used more effectively and more accurate results can be obtained in decision-making mechanisms. Artificial intelligence imitates the thinking and understanding abilities of human intelligence [1].

Artificial intelligence has shown great development in time. There are important stages in the development process of artificial intelligence. These are:

In 1943, the first ANN model was generated by Warren McCulloch and Walter Pitts [2]. The model called “Boolean Circuit Model of the Brain” is simply based on a theory that mathematically explains how neurons work in the brain using certain assumptions. In the next step, in 1948, Shannon [3] suggested that computers could be used to prove complex mathematical theorems and play chess within the framework of certain algorithms. In 1950, the British mathematician Alan Turing [4] asked the question “can machines think?” and introduced the concept of the “Turing test”, a method for measuring the ability of artificial intelligence to mimic human-like behaviour. The term “artificial intelligence” was first used in 1956 by Marvin Minsky

Y. Aydın · G. Bekdaş (✉) · S. M. Nigdeli
Department of Civil Engineering, Istanbul University-Cerrahpaşa, 34320 Avcılar, Istanbul, Turkey
e-mail: bekdas@iuc.edu.tr

Y. Aydın
e-mail: yaren.aydin1@ogr.iuc.edu.tr

S. M. Nigdeli
e-mail: melihnig@iuc.edu.tr

and John McCarthy, who moderated a project on artificial intelligence at Dartmouth College [5]. In the two decades following this, there have been significant developments in artificial intelligence. For example, one of the first programmes to pass the Turing test, ELIZA, developed by Joseph Weizenbaum at MIT, was able to simulate a human conversation [6]. Between 1974 and 1980, during the period called “AI winter”, government support and interest in the field of artificial intelligence decreased after the publication of many reports criticising the process of developing artificial intelligence. In the 1980s, the UK revitalised artificial intelligence by re-funding this field in order to compete against Japan in artificial intelligence studies [7]. In the 1990s, great progress was made in the use of artificial intelligence. In 1997, the programme named “Deep Blue” produced by IBM defeated the world chess champion Garry Kasparov in a chess match. As a result, it was shown that computers can be better than humans in some subjects and increased the popularity of artificial intelligence [8]. In the 2000s, artificial intelligence applications have been used in many electronic devices. Robot toys were produced using artificial intelligence. Big companies such as Facebook, Netflix and Twitter started to use artificial intelligence [9].

The most widely used artificial intelligence techniques in civil engineering are machine learning, deep learning, fuzzy logic, image processing and swarm intelligence. Within the scope of the study, these techniques will be explained and especially the studies conducted in the last 6 years in the field of civil engineering using these techniques will be included.

1.1 Machine Learning

Machine learning (ML) was first proposed by Arthur Samuel in 1959 as the ability of computers to learn without explicit programming [10]. Machine learning is a subfield of artificial intelligence that enables the development of various techniques and algorithms to enable computers to learn in a similar way to humans [11].

ML algorithms differ in their approaches, data types and problem-solving techniques. There are two different learning methods in machine learning models: supervised learning and unsupervised learning. In unsupervised learning, the data is not labelled and the data to be predicted is not shown to the machine. In supervised learning, the data are labelled and both input and output data are given to the machine. The areas of use of machine learning models are very wide. They are frequently used in problems such as regression, classification and clustering. Figure 1 shows the learning methods and problem scopes used in machine learning.

ML is a subset of artificial intelligence and focuses on teaching computers how to make predictions and trying to get a computer to do the kind of things the human mind can do. Figure 2 represents the machine learning flowchart.

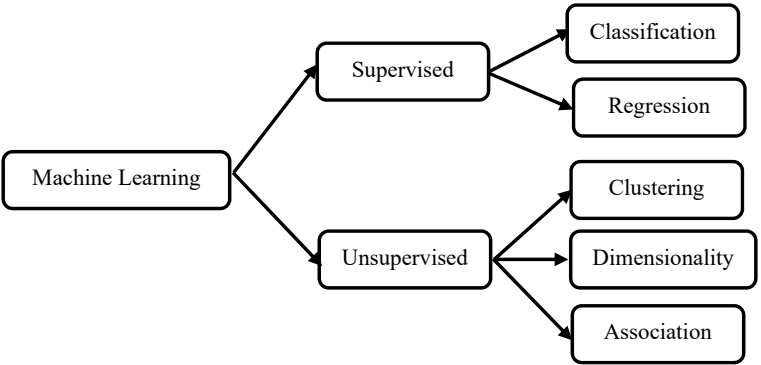
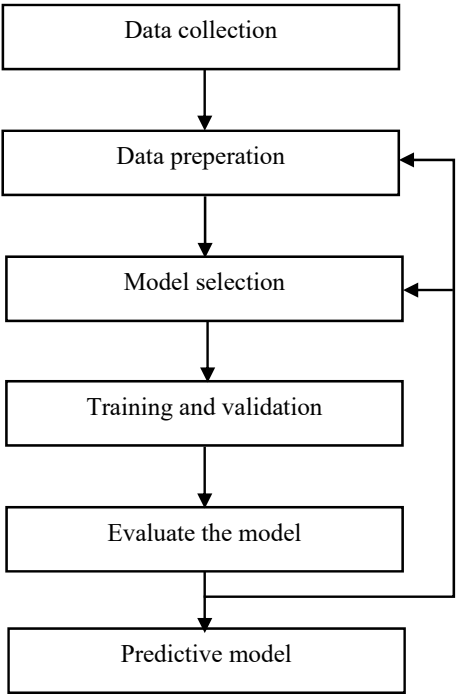


Fig. 1 ML methods

Fig. 2 Machine learning flowchart [12]



1.2 Deep Learning

While ML methods are performed in a single layer, deep learning consists of multi-layer neural networks. It explores a complex structure in large datasets by finding how the system parameters should be changed to calculate the output in each layer from the output in the previous layer of a machine using the backpropagation algorithm

[13]. Unlike machine learning, deep learning can automatically learn the functions that need to be fulfilled from the symbols of image, video, audio data and increases the prediction accuracy depending on the data size thanks to its flexible structure [14]. The feature that distinguishes deep learning from machine learning is the use of artificial neural networks (ANNs). In deep learning, a lot of data is needed for the relationship between labelled data, rather than a previously human-defined rule. Deep learning uses multiple layers of artificial neural network algorithms to establish this relationship, each of which is fed from the output of the previous layer and then different interpretations are made [15]. One of the most important features of deep learning is that it learns automatically through a self-learning procedure, although it is not taught or designed by humans [16].

In the functional structure of deep learning, the activation function is a mathematical formulation that calculates the output value of a neuron. The input value is multiplied by the neuron weight and the output is obtained by summing it with a value called bias. The concept of “bias” here is the name given to the value that allows the activation function to move to the right or left on the graph. Neuron weight is the name given to the parameter that transforms the input data in the hidden layers of the network.

Deep learning is the most general form of ANN structure with multiple layers. The output of each layer is the input of the previous layer. Thus, multiple representation levels are learned in response to different operations [17]. Figure 3 is the general architecture of deep neural networks.

In deep learning, the same ANN model is not used for the solution of every problem. Artificial neural networks differ according to their place of use and purpose. The model to be used varies according to the structure of the problem and the type of data. Recurrent neural network (RNN), convolutional neural network (CNN), generative adversarial networks (GANs), deep belief network (DBN) and long short-term memory (LSTM) are among the most widely used algorithms [19]. Figure 4 shows deep learning approaches as a scheme.

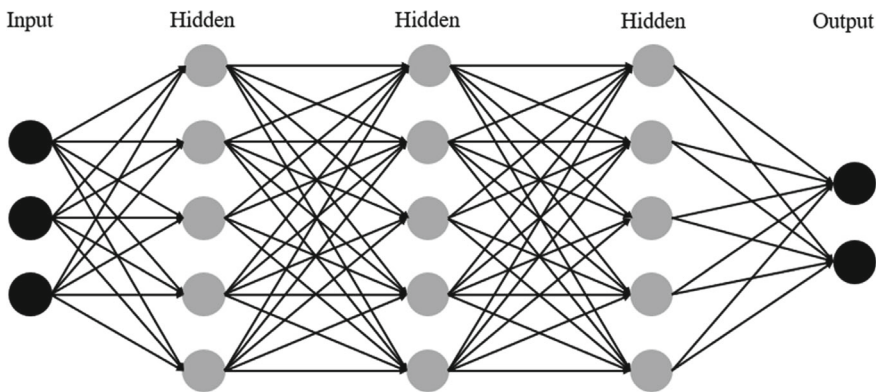


Fig. 3 Architecture of deep neural networks [18]

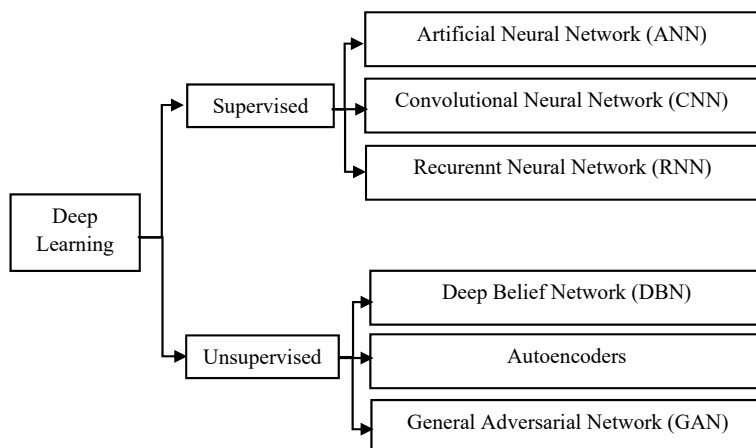


Fig. 4 Deep learning approaches [20]

1.3 Fuzzy Logic

In 1965, Lotfi Zadeh [21] developed fuzzy set theory at the University of California and introduced the word “fuzzy” in his paper Fuzzy Sets [22]. According to Zadeh, fuzzy sets have a multivariate structure and the two-valued logic system expressed as classical logic ($\{0,1\}$) is insufficient to explain human thoughts. Unlike classical logic, fuzzy logic is defined as systems used to analyse continuous values between 0 and 1 by making use of experienced expert knowledge [23]. Fuzzy logic realises the ability of machines to work by taking advantage of human experience and intuition by using symbolic expressions instead of numerical expressions. The mathematical basis for transferring symbolic expressions to machines is fuzzy logic set theory and fuzzy logic based on it. Fuzzy logic has better performance than classical methods and is particularly suitable for use in situations where it is not possible to analyse the system with classical methods and the nature of the information is uncertain or imprecise [24].

In fuzzy logic, there are fuzzification, creation, evaluation and collection of rule bases, and clarification processes. Fuzzy inference system is shown in Fig. 5 [25].

In Fig. 5, fuzzification is the process of converting some verbal expressions used in daily life (e.g. some, few, etc.) into membership functions. The input data are converted into numerical values between 0 and 1 by the fuzzification process [26]. Logical operators (and, or, not) are used to create and use the rule base needed for the inference process. The data and input values stored in the rule base are processed by the inference unit and the result is obtained. A rinse operation is used to convert the obtained results into real-world values [27].

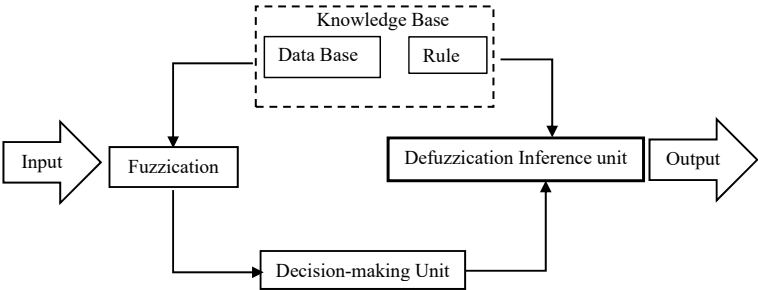
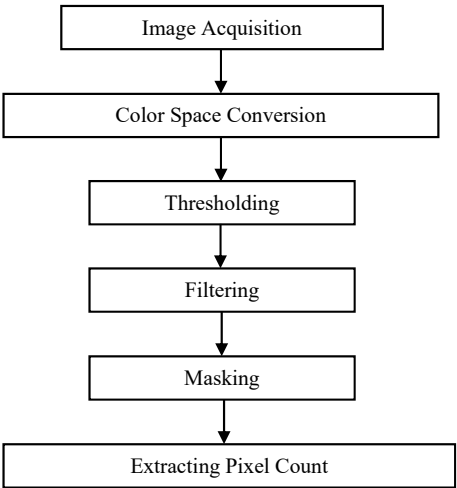


Fig. 5 Fuzzy inference system [25]

Fig. 6 Image processing diagram [30]



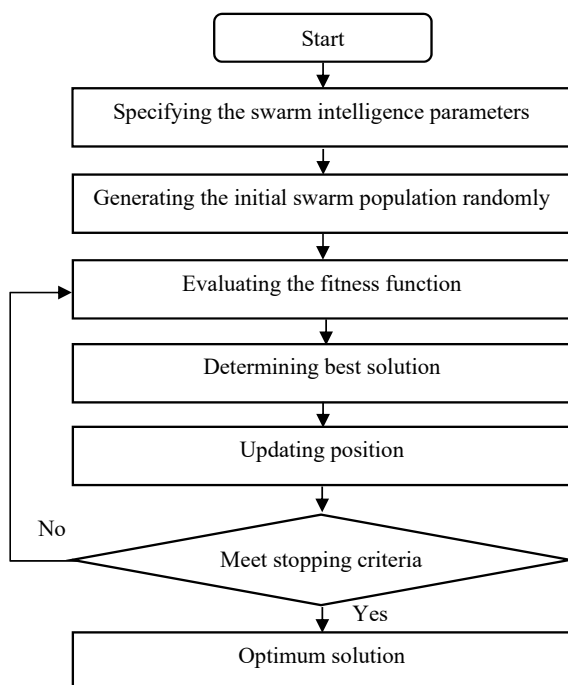
1.4 Image Processing

Image processing is the modification of digital image elements in accordance with the desired situation using computer software [28]. Image processing technique is an artificial intelligence model that covers the processes of increasing the clarity of the image, obtaining another image for the purpose from the existing image, creating a digital image and making analyses using these images [29]. Diagram of image processing is shown in Fig. 6.

1.5 Swarm Intelligence

Long ago, people were intrigued by the amazing patterns that existed in nature. Animals, insects and birds living in swarms exhibit interesting and regular behaviour.

Fig. 7 General flowchart of swarm intelligence algorithms [45]



Researchers have been intrigued by the foraging behaviour of ants and the echoing behaviour of bats in nature. The intelligent search strategy achieved by creating a better search strategy than random search, thanks to the cooperative behaviour of individuals in the swarm, is often referred to as swarm intelligence [31].

Swarm intelligence is the most developed branch among metaheuristic optimization algorithms [32]. Examples of these algorithms include particle swarm optimization [33], ant colony optimization [34], artificial bee colony algorithm [35], cuckoo algorithm [36], firefly algorithm [37], bat algorithm [38], flower pollination algorithm [39], grey wolf optimization [40], owl search algorithm [41], mouse swarm optimization [42], reptile search algorithm [43] and honey badger algorithm [44]. Figure 7 represents the general flowchart of swarm intelligence algorithms.

2 AI Applications in Civil Engineering

2.1 Machine Learning and Applications in Civil Engineering

Saravanan et al. applied advanced ML techniques to predict the overall construction quality of motorways, perform quality control and prevent premature failures [46]. Yucel et al. estimated the period and damping ratio parameters of tuned mass dampers

using machine learning methods. Optimum tuned mass damper parameters were estimated with the Artificial Neural Network (ANN) model used [47]. Seker and Ocak used six different machine learning algorithms and ensemble techniques to predict the performance of the roadheader using the collected dataset. Among the different methods used, random forest (RF) showed the best performance [48]. Noori Hoshyar et al. developed ML algorithms more successful than traditional SVM in detecting cracks [49]. Vakharia and Gujar predicted the compressive strength using isotonic regression, artificial neural network, support vector machine and RF machine learning models. As a result, low error was obtained with the models used [50]. Li et al. used extreme gradient boosting (XGBoost) and comprehensive learning particle swarm optimiser (CLPSO) for concrete strength estimation. As a result, the proposed model outperformed other machine learning algorithms [51].

Kurian and Liyanapathirana used machine learning (ML) algorithms that learn by experience to predict structural damage in concrete structures. Among the ML algorithms used, the random forest classifier (RFC) algorithm showed the highest performance [52]. Di Girolamo et al. used regression trees for the control of seismic vibrations. They used a benchmarking framework for the process they used [53]. Yucel et al. presented that the prediction of different structural design problems in civil engineering can be successfully achieved with the use of ANN [54]. Zhang et al. combined seven ML models using voting methods to predict compaction conditions in rock tunnels. As a result, the proposed ensemble was more successful (accuracy = 65%) than the individual classifiers [55]. Sathyan et al. evaluated the behaviour of superplasticised cement paste with the machine learning model XGBoost. As a result, the model developed with XGBoost had effective success [56]. Rusek et al. used support vector classifier (SVC) for damage risk assessment of reinforced concrete prefabricated building structures. The model, whose efficiency was verified, provided the basis for the reliability of existing reinforced concrete prefabricated building structures [57]. Singh et al. used machine learning models to predict the normalised bond strength of cement concrete substituted with sustainability products. The model was then estimated by regression analysis [58].

Soize and Orcesi used unsupervised machine learning methods to detect structural stiffness changes in a box girder bridge [59]. Lee et al. developed a ML-based prediction model using random forest (RF) for quantitative rock damage degree prediction considering various acoustic emission (AE) parameters. The results showed that RF was highly successful [60]. Fischer et al. used different ML models' methods to detect equipment activities on the construction site. Their study demonstrated the suitability of machine learning for simulating data [61]. Zhang et al. used a weighted voting method combining different ML models for the assessment of earthquake-induced liquefaction. As a result, the proposed ensemble model outperformed the other models [62]. Vu et al. used gradient tree boosting (GTB) models to predict the strength of concrete-filled steel tube (CFST) columns. Their model also verified by equations [63]. Chou et al. combined the firefly algorithm (FA), the one-to-one (OAO) method and the least squares support vector machine (LSSVM) for prediction in civil engineering. With the proposed method, the defects of the system in steel plates were predicted with high accuracy [64]. Asteris et al. used a hybrid ensemble

ML technique for the prediction of concrete compressive strength (CS). The machine learning technique, which they trained using ANN, has a high success rate [65].

Cakiroglu et al. predicted the properties of self-compacting concrete by using four ML models. These models predicted both outputs with high coefficient of determination [66]. Shahani et al. used ML and ANN to estimate Young's modulus. These models were compared with XGBoost and achieved effective results in Young's modulus prediction [67]. Cakiroglu et al. predicted the axial load carrying capacity of fibre RC columns using different ML models [68]. Rao et al. used different ML models to classify and forecast the shear strength, flexural strength and damage modes of shear walls. Among the models they used, random forest showed the highest performance (accuracy = 85%) [69]. Yücel et al. used a combination of optimization and machine learning to minimise structural damage in the frequency domain of tuned mass dampers (TMDs). With this approach, fast predictions are made instead of long time-consuming analyses [70]. Imran et al. used different ensemble models to predict the compressive strength of concrete produced by recycled concrete aggregate and ground granulated blast furnace slag. The generated model using the hybrid of convolutional neural networks (CNN) and long short-term memory (LSTM) showed the highest success [71]. Cakiroglu et al. used ML models to predict the axial compression capacity of columns. CatBoost showed the highest success for this prediction [72]. Shanmugasundaram et al. used ANN to determine the compressive strength of treated cementitious composite combined with fly ash and ground granular high slag pozzolans. This approach showed high success with a high R^2 coefficient [73]. Kidega et al. used gradient boosting algorithms to predict rock bursting. The model used showed the highest performance with $R^2 = 0.98$ [74]. Anjum et al. used different ML models (DT-Gradient Boosting, DT-XGBoost, DT-AdaBoost, DT-Bagging) for flexural strength prediction of steel-fibre reinforced concrete. As a result, DT-Bagging showed the highest performance ($R^2 = 0.95$) [75]. Binh Thai et al. used adaptive neuro-fuzzy inference system (ANFIS), support vector machine (SVM) and ANN models to predict the bearing capacity of bored piles. SVM showed the best performance ($R^2 = 0.98$) among the models [76]. Yin et al. used different machine learning models to predict the compressive strength of shotcrete coating. RF-QPSO, a hybrid of random forest and QPSO, showed the best performance ($R^2 = 0.93$) [77]. Yucel et al. performed the prediction of the optimum design of beams covered with carbon fibre reinforced polymers using different ML algorithms. They combined their ML algorithms with bagging and achieved success [78]. Zhang et al. used support vector machines (SVMs) and Bayesian optimization (BO) for the prediction of soil liquefaction potential. The hybrid-developed BOSVM had better prediction performance (accuracy = 96%) than the other models [79]. Truong and Chou developed a model combining jellyfish search (JS), fuzzy adaptive (FA), logic controller and stacking ML for scheduling. Their proposed method showed better performance than other models [80].

Cakiroglu et al. used different ML models to predict the properties of concrete with fibre reinforced rubber. The highest performance in compressive and tensile strength belongs to CatBoost [81]. Fathipour-Azar used ML models for cutting force prediction of rock. Among the models used, the most successful model ($R^2 = 0.812$)

is the DE-XGBoost model, which is a hybrid of differential evolution and extreme gradient boosting [82]. Yaghoubi et al. used ML models to predict the equivalent damping capacity in RC shear walls. The model developed using Gaussian process regression showed high success ($R^2 = 0.89$) [83]. Aydın et al. discussed the use of ML in civil engineering [84]. Bekdaş and Nigdeli emphasised the importance of optimization, metaheuristics and ML in civil engineering [85]. Mori et al. used Gaussian process regression model to predict the peak roof drift rate. This model is an important step for earthquake simulation [86]. Bekdaş et al. used ML algorithms to predict the cooling load in buildings. Among the regression techniques used, the models with the highest prediction success were histogram gradient boosting and stacking models [87]. Dinesh et al. showed that it is possible and effective to determine the compressive and tensile strength of concrete with XGBoost [88]. Kumar and Singh used ML models to predict the infiltration characteristics of a rain garden used for urban flood mitigation. Among the models used, the multivariate adaptive regression line (MARS) gave the best performance (RMSE = 0.129) [89]. Cakiroglu predicted the properties of concrete with aramid fibre reinforced polymer using several ML models. The tree regressor, XGBoost and K-nearest neighbour models achieved the highest success [90]. Alkhalwaldeh used light gradient boosting machine (LGBM) for the prediction of punching shear strength. Their hybridised approach with optimization achieved higher success than the stand-alone LGBM model [91]. Chen et al. used extreme gradient boosting and random forest for the prediction of soil shear strength. The XGBoost used showed higher success than RF [92]. Nasir Amin et al. used different ML models to predict the compressive strength of rice husk ash used in concrete. Among these models, bagging had the highest success ($R^2 = 0.93$) [93]. Mai et al. presented a hybrid ML to predict the permanent transverse displacement of steel members with circular hollow section using XGB with different optimization techniques. The hybrid of grey wolf optimiser and XGB achieved the highest success with $R^2 = 0.98$ [94]. Aydın et al. performed soil classification using ML models. The dataset they used was subjected to different processes, and they achieved the highest success with XGBoost [95]. Yücel et al. used machine learning models to determine the optimum parameters of different structural models including a three-bar truss and a simply supported reinforced concrete (RC) beam. They comprehensively compared the different models they used and determined the effective method [96]. Aydın et al. used machine learning models to investigate the least damage to the environment during the production of cement in concrete. Among the models used, multilayer perceptron (MLP) showed the highest performance ($R^2 = 0.999$) [97]. Alabduljabbar et al. used different ML models to predict the self-healing capacity of cementitious composites. With bagging, $R^2 = 0.97$ was achieved and the suitability of ML models for prediction was proved [98]. Khan et al. employed XGBoost to predict the characteristic compressive strength of ultrahigh-performance concrete (UHPC) containing recycled glass powder. High success was achieved with the ML model used [99]. Chi et al. used different ML models to predict the compressive strength of cement-based materials [100]. The models used showed high agreement in prediction. Aydın et al. used ML models to minimise environmental damage in concrete production. Among the models used, random forest showed the highest prediction success [101]. Yang

et al. presented a machine learning model for predicting the compressive strength of reinforced cement composites. Among the models, AutoGluon-Tabular had the highest performance [102]. Perfetto et al. used machine learning algorithms to determine the location and extent of damage in the structure. The model obtained by optimising the ensemble algorithm showed high success [103]. Aydın et al. used advanced ML to evaluate the carbon emission in the production of alkaline activated binders, which are used as alternative binders to minimise the environmental damage of ordinary Portland cement production. The highest accuracy was achieved with a genetic algorithm using hyperparameter optimization [104]. Cakiroglu et al. used ML models for the estimation of the tensile strength of basalt fibre reinforced concrete. The highest success in prediction ($R^2 > 0.9$) was achieved using XGBoost [105].

Kumar and Pratap evaluated the parameters in the model predicting concrete compressive strength and obtained effective prediction results [106]. Asgarkhani et al. developed a model based on ML to evaluate the seismic behaviour of anti-torsion braced frames. Their proposed model is promising for evaluation [107]. Shahab et al. utilised ML models to predict the compressive strength of concrete containing graphene nanoplatelets. As a result, one of the models used, gene expression programming, had the highest prediction performance [108]. Liu et al. [109] predicted the punching shear strength in different performances of fibre reinforced with XGBoost with high success.

2.2 *Deep Learning and Applications in Civil Engineering*

Gao and Mosalam used transfer learning (TL) to recognise structural damage from images and achieved success in identifying high damage in civil engineering structures [110]. Yang et al. used the full convolutional network (FCN) technique, which is one of the DL techniques in crack assessment. As a result of the technique used, 97.96% accuracy was achieved in crack assessment [111]. Liu et al. tested the statistics of excavation quantities using convolutional neural network (CNN) and transfer learning (TL) forms. It was concluded that these methods were effective in the test phase [112].

Tanyildizi et al. used a deep learning method called long short-term memory (LSTM) networks to predict the mechanical properties of concrete containing silica fume under high temperatures. As a result, LSTM was more successful than other coders [113]. Shin et al. proposed a convolution-based concrete multiple damage recognition neural network (CMDnet) for damage detection in concrete. As a result, the trained concrete multiple damage recognition neural network achieved high accuracy (98.9%) [114]. Weerasinghe and Jayasena used deep learning approaches (VGG16, VGG19, Densenet169, MobileNet) in flood management. These models were compared, and they achieved the highest success with MobileNet [115].

Li et al. used YOLOv3, one of the deep learning methods, for the automatization of steel bar counting. This method was found to be suitable for counting steel bars

in construction [116]. Cui et al. proposed the YOLO-v3 model for the identification of damage to concrete caused by wind erosion. This method showed high success in detection [117]. Tan et al. used the You Only Look Once version 3 (YOLOv3) method based on a convolutional neural network (CNN) for the inspection of sewer pipes. The model developed in the study showed high performance [118]. Hong et al. employed a convolutional neural network (CNN) for seismic ground response analysis. The model they proposed was successful [119].

Liu et al. evaluated the stability of retaining walls using a convolutional neural network (CNN). High accuracy was achieved as a result of the classifications performed with the method [120]. Wang et al. developed deep structural nonlinear analysis (DeepSNA) to predict the mechanical response of arched structures. With this method, the prediction efficiency is increased compared to classical methods [121].

Luleci et al. used generative adversarial networks (GAN) based on DL methods for structural health monitoring (SHM). The developed method was found promising for future studies [122]. Qayyum et al. automatically classified cracks using convolutional neural networks (CNNs) architectures. The deep learning methods used showed high success in crack classification [123]. Philip et al. used the YOLO v3 model to create an automatic system capable of detecting cracks in concrete walls. Their proposed method had high accuracy [124]. Matinfar et al. produced an artificial spectrum-matched earthquake acceleration record using deep convolutional GAN (DCGAN) using several ground motion records. The results showed that the method was successful [125]. Inam et al. used YOLOv5 to identify cracks in bridges. This method showed superior success in crack assessment [126]. Zhou et al. used YOLOv4 based on deep learning for the detection of tunnel pavement cracks. As a result, the YOLOv4 model accurately detects tunnel pavement cracks [127]. Bai et al. applied deep learning (DL) approaches for health monitoring and damage tracking of reinforced concrete structures. In this study using EfficientNet-V2, component damage recognition was successfully achieved [128]. Dinh and Ahn used convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for structural damage assessment. Their proposed method achieved effective results [129]. Zhang et al. presented physics-informed DL for the identification of vibration. Their model provided accuracy [130].

2.3 Fuzzy Logic and Applications in Civil Engineering

Hasheminejad et al. used adaptive neural fuzzy inference system (ANFIS), which is one of the fuzzy logic methods, for the prediction of soil collapsibility potential. With the hybrid optimization algorithms, they used while training ANFIS, they achieved effective results in predicting the collapsibility potential of the soil [131]. Siva Krishna and Ranga Rao used fuzzy logic to investigate the strength variation with molar concentration of geopolymers concrete. High prediction accuracy was achieved with the fuzzy logic model used [132]. Godfrey et al. tested the applicability of a

modified version of the build, operate and transfer method for urban water supply service using fuzzy logic statistical analysis. As a result of the study, it was concluded that this method is effective [133]. Hashemi and Zarabiha presented a fuzzy logic-based approach to determine the ground response in an earthquake. The approach they used was useful for determining the ground response to earthquakes [134]. Hormozabad and Soto used a hybrid of semi-active magneto-rheological dampers and passive base isolation bearings to reduce the vibration of highway bridge structures. Their approach had advantages in achieving vibration control [135]. Chakraborty and Goswami used fuzzy logic and regression analysis to predict the safety factor of slopes. Of these two models, fuzzy logic was more successful [136]. Rogulj et al. used fuzzy logic to prioritise the renovation of the roofs of public buildings [137]. Hakim et al. generated a model to predict the compressive strength of high-strength concrete via neuro-fuzzy systems. The fuzzy logic model used predicted the compressive strength with high success [138].

2.4 Image Processing and Applications in Civil Engineering

Damadipour et al. determined the particle size distribution using image processing. The result showed high agreement between the predicted and actual values [139]. Tiwari et al. used silica fume and polypropylene fibre to stabilise the soil base and used image processing technique to evaluate the shrinkage limit. The image processing technique used is an important step for soil improvement [140]. Riaz et al. estimated the weight and damage load of the modelled bridge using image processing technique. In the process carried out using the genetic algorithm, the weight and damage load of the bridge were successfully estimated [141]. Carrasco et al. used digital image processing techniques to detect the width of cracks. The results obtained showed that the prediction success was high [142]. Xu and Liu performed reconstruction based on monocular vision for buildings using digital image processing technology. As a result, success has been achieved in this field with this image processing [143]. Zhang et al. used thermal image processing for the detection of underground defects in civil infrastructure. The image processing techniques used were compared, and a high correlation coefficient was achieved [144]. Shi et al. proposed an optimization model using digital image processing techniques for coarse aggregate contact analysis in asphalt mix [145]. Ma et al. automatically realised the rebar rebuilding using digital imagery [146]. Jiang et al. performed 3D reconstruction of coarse aggregate using optical laser triangulation and image processing. Promising results were obtained [147]. Jayanthi et al. successfully detected concrete capillary cracks using image processing [148]. Ya'acob et al. used advanced image processing and YOLO for crack detection in pavements. The model has high accuracy [149].

2.5 *Swarm Intelligence and Applications in Civil Engineering*

Saleem and Jo used a model combining genetic algorithm (GA) with another algorithm to accurately determine the impact force on structures [150]. Bekdaş et al. used a new optimization algorithm for the optimal design of TMD for the safety of buildings. The algorithm used was compared with the GA [151]. Nigdeli et al. used modified particle swarm optimization (PSO) to adjust the contribution of the current values of the velocity in the inertia function [152]. Borodin and Zhangabay automatically and statistically determined the optimal composition of slag-filled concrete using a swarm intelligence algorithm [153]. Mishra et al. presented swarm intelligence methods used in slope stability assessment and compared them with antlion optimiser [154]. Yücel et al. found the optimum section values of the cantilever beam in order to minimise its total weight with swarm intelligence algorithms [155]. Kayabekir et al. performed the cost minimisation of L-shaped RC retaining wall with different metaheuristic algorithms including FPA. In the study, FPA outperformed the other algorithms [156]. Ulusoy et al. used algorithms including flower pollination algorithm (FPA) to improve the efficiency of active control with metaheuristics [157]. Hussien et al. introduced the crow search algorithm (CSA), which is one of the swarm intelligence algorithms and can be used in different engineering branches, with different aspects and different types [158]. Ulusoy et al. used different metaheuristic algorithms, including the bat algorithm, for the cost minimisation of RC beams. Finally, they concluded that an optimal design cannot be found with a single metaheuristic algorithm [159]. Kayabekir et al. used a flower pollination algorithm to minimise CO₂ emissions from reinforced concrete T-beams [160]. Bekdaş et al. modified the Lévy flight to three algorithms using swarm intelligence in various optimization to minimise the structural weight of a space truss. As a result, this modification had the most positive effect on the grey wolf algorithm [161]. Ulusoy et al. optimised the parameters of active control systems in the structure with FPA [162]. Naderpour et al. generated an estimation model for the moment capacity of spiral-reinforced concrete columns by using two different models combining neural network with PSO. The neural network using the PSO algorithm was more successful [163]. Yücel et al. used ANN to generate a prediction model for optimum dimensions of reinforced concrete cantilever retaining walls and employed flower pollination algorithm to obtain learning data [164]. Bekdaş et al. employed swarm-based optimization algorithms in the optimum design of span lengths of frame structures and optimization of control systems and obtained effective results [165]. Bardhan and Asteris combined ANN and grey wolf optimization to predict the optimum moisture content and maximum dry density of the soil. Their proposed model provided a good prediction success [166]. Yücel et al. stated that swarm intelligence algorithms are generally used for optimization in the process of data generation for machine learning in structural engineering and performed their applications in structural problems [167]. Aydın et al. used the FPA to minimise the cost of reinforced concrete beams. They reached the most economical solution with random switching probability [168].

Günay et al. successfully found the optimum sections of reinforced concrete column for different fire durations by hybridising the FPA and Jaya algorithm [169]. Huang and Wang used a hybrid RF and PSO for the prediction and control of bridge pier displacement in subway tunnel construction. The developed and optimised method is effective in prediction and control [170].

References

1. Bhushan S (2021) The impact of artificial intelligence and machine learning on the global economy and its implications for the hospitality sector in India. *Worldwide Hospitality Tourism Themes* 13(2):252–259
2. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5(4):115–133. <https://doi.org/10.1007/BF02478259>
3. Shannon C (1948) Claude shannon. *Inf Theory* 3:224
4. Turing A (1950) Turing. Computing machinery and intelligence. *Mind* 59(236):433–60
5. McCarthy J, Minsky ML, Rochester N, Shannon CE (2006) A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI Mag* 27(4):12–12. <https://doi.org/10.1609/aimag.v27i4.1904>
6. Haenlein M, Kaplan A (2019) A Brief History of Artificial Intelligence: on the past, present, and future of artificial intelligence. *Calif Manage Rev* 61(4):5–14. <https://doi.org/10.1177/0008125619864925>
7. Durakcan YC (2015) History of machine learning. <http://bilimfili.com/yapayzekanin-kisa-tarihcesi-history-of-machine-learning-algorithms/>
8. Gonsalves T (2019) The summers and winters of artificial intelligence. In: Khosrow-Pour M (ed) *Advanced methodologies and technologies in artificial intelligence, computer simulation, and human-computer interaction*. IGI Global, pp 168–179. <https://doi.org/10.4018/978-1-5225-7368-5.ch014>
9. Özgeldi M (2019) Artificial intelligence and human resources. In: Telli G (ed) *Artificial intelligence and the future*. Doğu Bookstore Publishing House, Istanbul, pp 198–222
10. Ongsulee P (2017) Artificial intelligence, machine learning and deep learning. In: 2017 15th international conference on ICT and knowledge engineering (ICT&KE), Bangkok, Thailand, pp 1–6. <https://doi.org/10.1109/ICTKE.2017.8259629>
11. Das S, Dey A, Pal A, Roy N (2015) Applications of artificial intelligence in machine learning: review and prospect. *Int J Comput Appl* 115(9)
12. Ng F, Jiang R, Chow JC (2020) Predicting radiation treatment planning evaluation parameter using artificial intelligence and machine learning. *IOP SciNotes* 1(1):014003
13. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521:436–444. <https://doi.org/10.1038/nature14539>
14. Kayaalp K, Süzen AA (2018) *Deep learning and its applications in Turkey*, ISBN 978-605-7510-53-2, Iksad Printing House, Ankara, 89 p
15. Alzubaidi L, Zhang J, Humaidi AJ et al (2021) Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* 8:53. <https://doi.org/10.1186/s40537-021-00444-8>
16. Bengio Y (2009) Learning deep architectures for AI. *Foundations Trends® Mach Learn* 2(1):1–127
17. Visvikis D, Cheze Le Rest C, Jaouen V et al (2019) Artificial intelligence, machine (deep) learning and radio(geno)mics: definitions and nuclear medicine imaging applications. *Eur J Nucl Med Mol Imaging* 46:2630–2637. <https://doi.org/10.1007/s00259-019-04373-w>
18. Isaksson M (2018) Create a drawing of a feed-forward neural network. <https://github.com/martisak/dotnets>

19. Doğan F, Türkoğlu I (2019) A review on deep learning models and application areas. *Dicle Univ Faculty Eng J Eng* 10(2):409–445. <https://doi.org/10.24012/dumf.411130>
20. Kaya B, Önal M (2021) Segmentation of lung CT images for COVID-19 detection. *Eur J Sci Technol*, Ejosat Special Issue 2021 (ICAENS), 1296–1303. <https://doi.org/10.31590/ejosat.1015061>
21. Zadeh LA (1965) Fuzzy sets. *Inf Control* 8(3):338–353
22. Klir GJ (2001) Foundations of fuzzy set theory and fuzzy logic: a historical overview. *Int J General Syst* 30(2):91–132. <https://doi.org/10.1080/03081070108960701>
23. Ganguly P, Kalam A, Zayegh A (2019) Modelling of an optimum fuzzy logic controller using genetic algorithm. In: Chattopadhyay S, Roy T, Sengupta S, Berger-Vachon C (eds) *Modelling and simulation in science, technology and engineering mathematics*. MS-17 2017. *Advances in intelligent systems and computing*, vol 749. Springer, Cham. https://doi.org/10.1007/978-3-319-74808-5_28
24. Elmas Ç (2003) Fuzzy logic controllers. Seçkin Publishing, Ankara
25. Jung CH, Ham CS, Lee KI (1995) A real-time self-tuning fuzzy controller through scaling factor adjustment for the steam generator of NPP. *Fuzzy Sets Syst* 74(1):53–60. [https://doi.org/10.1016/0165-0114\(95\)00035-J](https://doi.org/10.1016/0165-0114(95)00035-J)
26. Alavala CR (2008) Fuzzy logic and neural networks: basic concepts and application. *New Age Int*
27. Dernoncourt F (2013) Introduction to fuzzy logic. *Massachusetts Inst Technol* 21:50–56
28. Gonzalez RC, Woods RE (2007) *Digital image processing*, 3th edn, A.B.D. Prentice Hall, New Jersey
29. Sonka M, Hlavac V, Boyle R (2013) *Image processing, analysis and machine vision*. Springer
30. Calangian XARP, Gonzales JYC, Hilario CAN, Lopez JMM, Rulona BLE, Valencia IJC, Dadios EJP (2019) Vision-based canopy area measurements. <https://doi.org/10.1109/HNICEM.2018.8666251>
31. Bansal JC, Pal NR (2019) Swarm and evolutionary computation. In: Bansal J, Singh, P, Pal N (eds) *Evolutionary and swarm intelligence algorithms*. *Studies in computational intelligence*, vol 779. Springer, Cham. https://doi.org/10.1007/978-3-319-91341-4_1
32. Osaba E, Yang XS (2021) Applied optimization and swarm intelligence: a systematic review and prospect opportunities. In: Osaba E, Yang XS (eds) *Applied optimization and swarm intelligence*. *Springer tracts in nature-inspired computing*. Springer, Singapore. https://doi.org/10.1007/978-981-16-0662-5_1
33. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: *Proceedings of the 6th international symposium on micro machine and human science*, vol 1, October, Nagoya, Japan, pp 39–43. <https://doi.org/10.1109/MHS.1995.494215>
34. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. *IEEE Comput Intell Mag* 1(4):28–39. <https://doi.org/10.1109/MCI.2006.329691>
35. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Glob Optim* 39:459–471. <https://doi.org/10.1007/s10898-007-9149-x>
36. Yang XS, Deb S (2009) Cuckoo search via Lévy flights. In: 2009 world congress on nature and biologically inspired computing (NaBIC), December, Coimbatore, India, pp 210–214. <https://doi.org/10.1109/NABIC.2009.5393690>
37. Yang XS (2009) Firefly algorithms for multimodal optimization. In: Watanabe O, Zeugmann T (eds) *Stochastic algorithms: foundations and applications*. SAGA 2009. *Lecture notes in computer science*, vol 5792. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04944-6_14
38. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: González JR, Pelta DA, Cruz C, Terrazas G, Krasnogor N (eds) *Nature inspired cooperative strategies for optimization (NISCO 2010)*. *Studies in computational intelligence*, vol 284. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-12538-6_6
39. Yang XS (2012) Flower pollination algorithm for global optimization. In: Durand-Lose J, Jonoska N (eds) *Unconventional computation and natural computation*. UCNC 2012. *Lecture*

- notes in computer science, vol 7445. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-32894-7_27
40. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
 41. Jain M, Maurya S, Rani A, Singh V (2018) Owl search algorithm: a novel nature-inspired heuristic paradigm for global optimization. *J Intell Fuzzy Syst* 34(3):1573–1582. <https://doi.org/10.3233/JIFS-169452>
 42. Dhiman G, Garg M, Nagar A et al (2021) A novel algorithm for global optimization: rat swarm optimizer. *J Ambient Intell Human Comput* 12:8457–8482. <https://doi.org/10.1007/s12652-020-02580-0>
 43. Abualigah L, Abd Elaziz M, Sumari P, Geem ZW, Gandomi AH (2022) Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer. *Expert Syst Appl* 191:116158. <https://doi.org/10.1016/j.eswa.2021.116158>
 44. Hashim FA, Houssein EH, Hussain K, Mabrouk MS, Al-Atabany W (2022) Honey Badger algorithm: new metaheuristic algorithm for solving optimization problems. *Math Comput Simul* 192:84–110. <https://doi.org/10.1016/j.matcom.2021.08.013>
 45. Jumani TA, Mustafa MW, Alghamdi AS, Rasid MM, Alamgir A, Awan AB (2020) Swarm intelligence-based optimization techniques for dynamic response and power quality enhancement of AC microgrids: a comprehensive review. *IEEE Access* 8:75986–76001. <https://doi.org/10.1109/ACCESS.2020.2989133>
 46. Saravanan V, Pourhomayoun M, Mazari M (2018) A proposed method to improve highway construction quality using machine learning. In: 2018 international conference on computational science and computational intelligence (CSCI). IEEE, pp 1218–1221. <https://doi.org/10.1109/CSCI46756.2018.00234>
 47. Yucel M, Bekdaş G, Nigdeli SM, Sevgen S (2019) Estimation of optimum tuned mass damper parameters via machine learning. *J Build Eng* 26:100847. <https://doi.org/10.1016/j.jobe.2019.100847>
 48. Seker SE, Ocak I (2019) Performance prediction of roadheaders using ensemble machine learning techniques. *Neural Comput Appl* 31(4):1103–1116. <https://doi.org/10.1007/s00521-017-3141-2>
 49. Noori Hoshyar A, Rashidi M, Liyanapathirana R, Samali B (2019) Algorithm development for the non-destructive testing of structural damage. *Appl Sci* 9(14):2810. <https://doi.org/10.3390/app9142810>
 50. Vakharia V, Gujar R (2019) Prediction of compressive strength and portland cement composition using cross-validation and feature ranking techniques. *Constr Build Mater* 225:292–301. <https://doi.org/10.1016/j.conbuildmat.2019.07.224>
 51. Li Y, Gou J, Fan Z (2019) Particle swarm optimization-based extreme gradient boosting for concrete strength prediction. In: 2019 IEEE 4th advanced information technology, electronic and automation control conference (IAEAC), vol 1, pp 982–986. IEEE. <https://doi.org/10.1109/IAEAC47372.2019.8997825>
 52. Kurian B, Liyanapathirana R (2020) Machine learning techniques for structural health monitoring. In: Wahab M (eds) *Proceedings of the 13th international conference on damage assessment of structures. Lecture notes in mechanical engineering*. Springer, Singapore. https://doi.org/10.1007/978-981-13-8331-1_1
 53. Di Girolamo GD, Smarra F, Gattulli V, Potenza F, Graziosi F, D’Innocenzo A (2020) Data-driven optimal predictive control of seismic induced vibrations in frame structures. *Struct Control Health Monit* 27(4):e2514. <https://doi.org/10.1002/stc.2514>
 54. Yucel M, Nigdeli SM, Bekdaş G (2020) Artificial neural networks (anns) and solution of civil engineering problems: anns and prediction applications. In: *Artificial intelligence and machine learning applications in civil, mechanical, and industrial engineering*. IGI Global, pp 13–38. <https://doi.org/10.4018/978-1-7998-0301-0.ch002>
 55. Zhang J, Li D, Wang Y (2020) Predicting tunnel squeezing using a hybrid classifier ensemble with incomplete data. *Bull Eng Geol Environ* 79:3245–3256. <https://doi.org/10.1007/s10064-020-01747-5>

56. Sathyan D, Govind D, Rajesh CB, Gopikrishnan K, Kannan GA, Mahadevan J (2020) Modelling the shear flow behaviour of cement paste using machine learning–XGBoost. *J Phys Conf Ser* 1451(1):012026. IOP Publishing. <https://doi.org/10.1088/1742-6596/1451/1/012026>
57. Rusek J, Tajduś K, Firek K, Jędrzejczyk A (2020) Bayesian networks and Support Vector Classifier in damage risk assessment of RC prefabricated building structures in mining areas. In: 2020 5th international conference on smart and sustainable technologies (SpliTech). IEEE, pp 1–8. <https://doi.org/10.23919/SpliTech49282.2020.9243718>
58. Singh P, Garg C, Namdeo A, Agarwal KM, Rai RK (2020) Development of prediction models for bond strength of steel fiber reinforced concrete by computational machine learning. In: E3S web of conferences, vol 220. EDP Sciences, p 01097. <https://doi.org/10.1051/e3sconf/202022001097>
59. Soize C, Orcesi A (2021) Machine learning for detecting structural changes from dynamic monitoring using the probabilistic learning on manifolds. *Struct Infrastruct Eng* 17(10):1418–1430. <https://doi.org/10.1080/15732479.2020.1811991>
60. Lee HL, Kim JS, Hong CH, Cho DK (2021) Ensemble learning approach for the prediction of quantitative rock damage using various acoustic emission parameters. *Appl Sci* 11(9):4008. <https://doi.org/10.3390/app111094008>
61. Fischer A, Liang M, Orschlet V, Bi H, Kessler S, Fottner J (2021) Detecting equipment activities by using machine learning algorithms. *IFAC-PapersOnLine* 54(1):799–804. ISSN 2405-8963. <https://doi.org/10.1016/j.ifacol.2021.08.094>
62. Zhang J, Wang Y (2021) An ensemble method to improve prediction of earthquake-induced soil liquefaction: a multi-dataset study. *Neural Comput Appl* 33:1533–1546. <https://doi.org/10.1007/s00521-020-05084-2>
63. Vu QV, Truong VH, Thai HT (2021) Machine learning-based prediction of CFST columns using gradient tree boosting algorithm. *Compos Struct* 259:113505. <https://doi.org/10.1016/j.compstruct.2020.113505>
64. Chou JS, Pham TTP, Ho CC (2021) Metaheuristic optimized multi-level classification learning system for engineering management. *Appl Sci* 11(12):5533. <https://doi.org/10.3390/app11125533>
65. Asteris PG, Skentou AD, Bardhan A, Samui P, Pilakoutas K (2021) Predicting concrete compressive strength using hybrid ensembling of surrogate machine learning models. *Cem Concr Res* 145:106449. <https://doi.org/10.1016/j.cemconres.2021.106449>
66. Cakiroglu C, Bekdaş G, Kim S, Geem ZW (2022) Explainable ensemble learning models for the rheological properties of self-compacting concrete. *Sustainability* 14(21):14640. <https://doi.org/10.3390/su142114640>
67. Shahani NM, Zheng X, Liu C, Hassan FU, Li P (2021) Developing an XGBoost regression model for predicting young's modulus of intact sedimentary rocks for the stability of surface and subsurface structures. *Front Earth Sci* 9:761990. <https://doi.org/10.3389/feart.2021.761990>
68. Cakiroglu C, Islam K, Bekdaş G, Kim S, Geem ZW (2022) Interpretable machine learning algorithms to predict the axial capacity of FRP-reinforced concrete columns. *Materials* 15(8):2742. <https://doi.org/10.3390/ma15082742>
69. Rao P, Pollayi H, Rao M (2022) Machine learning based design of reinforced concrete shear walls subjected to earthquakes. *J Phys Conf Ser* 2327(1):012068. IOP Publishing. <https://doi.org/10.1088/1742-6596/2327/1/012068>
70. Yücel M, Nigdeli SM, Bekdaş G (2022) Machine learning-based model for optimum design of TMDs by using artificial neural networks. In: Bekdaş G, Nigdeli SM (eds) Optimization of tuned mass dampers. studies in systems, decision and control, vol 432. Springer, Cham. https://doi.org/10.1007/978-3-030-98343-7_10
71. Imran H, Ibrahim M, Al-Shoukry S, Rustam F, Ashraf I (2022) Latest concrete materials dataset and ensemble prediction model for concrete compressive strength containing RCA and GGBFS materials. *Constr Build Mater* 325:126525. <https://doi.org/10.1016/j.conbuildmat.2022.126525>

72. Cakiroglu C, Islam K, Bekdaş G, Isikdag U, Mangalathu S (2022) Explainable machine learning models for predicting the axial compression capacity of concrete filled steel tubular columns. *Constr Build Mater* 356:129227. <https://doi.org/10.1016/j.conbuildmat.2022.129227>
73. Shanmugasundaram N, Praveenkumar S, Gayathiri K, Divya S (2022) Prediction on compressive strength of engineered cementitious composites using machine learning approach. *Constr Build Mater* 342:127933. <https://doi.org/10.1016/j.conbuildmat.2022.127933>
74. Kidega R, Ondiaka MN, Maina D, Jonah KAT, Kamran M (2022) Decision based uncertainty model to predict rockburst in underground engineering structures using gradient boosting algorithms. *Geomech Eng* 30(3):259. <https://doi.org/10.12989/gae.2022.30.3.259>
75. Anjum M, Khan K, Ahmad W, Ahmad A, Amin MN, Nafees A (2022) New SHapley Additive ExPlanations (SHAP) approach to evaluate the raw materials interactions of steel-fiber-reinforced concrete. *Materials* 15(18):6261. <https://doi.org/10.3390/ma15186261>
76. Binh Thai P, Duc Nguyen D, Bui Thi QA, Duc Nguyen M, Tien Vu T, Prakash I (2022) Estimation of load-bearing capacity of bored piles using machine learning models. *Vietnam J Earth Sci* 44(4):470–480. <https://doi.org/10.15625/2615-9783/17177>
77. Yin X, Gao F, Wu J, Huang X, Pan Y, Liu Q (2022) Compressive strength prediction of sprayed concrete lining in tunnel engineering using hybrid machine learning techniques. *Underground Space* 7(5):928–943. <https://doi.org/10.1016/j.undsp.2022.01.003>
78. Yucel M, Kayabekir AE, Nigdeli SM, Bekdaş G (2022) Optimum design of carbon fiber-reinforced polymer (CFRP) beams for shear capacity via machine learning methods: optimum prediction methods on advance ensemble algorithms—bagging combinations. In: Management Association I (ed) *Research anthology on machine learning techniques, methods, and applications*. IGI Global, pp 308–326. <https://doi.org/10.4018/978-1-6684-6291-1.ch018>
79. Zhang X, He B, Sabri MMS, Al-Bahrani M, Ulrikh DV (2022) Soil liquefaction prediction based on Bayesian optimization and support vector machines. *Sustainability* 14(19):11944. <https://doi.org/10.3390/su141911944>
80. Truong DN, Chou JS (2022) Fuzzy adaptive jellyfish search-optimized stacking machine learning for engineering planning and design. *Autom Constr* 143:104579. <https://doi.org/10.1016/j.autcon.2022.104579>
81. Cakiroglu C, Shahjalal M, Islam K, Mahmood SF, Billah AM, Nehdi ML (2023) Explainable ensemble learning data-driven modeling of mechanical properties of fiber-reinforced rubberized recycled aggregate concrete. *J Build Eng* 76:107279. <https://doi.org/10.1016/j.jobte.2023.107279>
82. Fathipour-Azar H (2023) Mean cutting force prediction of conical picks using ensemble learning paradigm. *Rock Mech Rock Eng* 56:221–236. <https://doi.org/10.1007/s00603-022-03095-0>
83. Yaghoubi ST, Deger ZT, Taskin G et al (2023) Machine learning-based predictive models for equivalent damping ratio of RC shear walls. *Bull Earthquake Eng* 21:293–318. <https://doi.org/10.1007/s10518-022-01533-6>
84. Aydin Y, Bekdaş G, Işıkdag Ü, Nigdeli SM (2023) The state of art in machine learning applications in civil engineering. In: Bekdaş G, Nigdeli SM (eds) *Hybrid metaheuristics in structural engineering. studies in systems, decision and control*, vol 480. Springer, Cham. https://doi.org/10.1007/978-3-031-34728-3_9
85. Bekdaş G, Nigdeli SM (2023) Introduction and overview: hybrid metaheuristics in structural engineering—including machine learning applications. In: Bekdaş G, Nigdeli SM (eds) *Hybrid metaheuristics in structural engineering. Studies in systems, decision and control*, vol 480. Springer, Cham. https://doi.org/10.1007/978-3-031-34728-3_1
86. Mori F, Spina D, Bocchi F, Mendicelli A, Naso G, Moscatelli M (2023) Machine learning model for building seismic peak roof drift ratio assessment. *Geomat Nat Haz Risk* 14(1):2182658. <https://doi.org/10.1080/19475705.2023.2182658>
87. Bekdaş G, Aydın Y, Işıkdag Ü, Sadeghifam AN, Kim S, Geem ZW (2023) Prediction of cooling load of tropical buildings with machine learning. *Sustainability* 15(11):9061. <https://doi.org/10.3390/su15119061>

88. Dinesh A, Selvasofia SA, Datcheen KS, Varshan DR (2023) Machine learning for strength evaluation of concrete structures—critical review. *Mater Today Proc.* <https://doi.org/10.1016/j.matpr.2023.04.090>
89. Kumar S, Singh KK (2023) Infiltration rate assessment of experimental rain garden using conventional models and machine learning algorithms. In: *World environmental and water resources congress 2023*, pp 697–710
90. Cakiroglu C (2023) Explainable data-driven ensemble learning models for the mechanical properties prediction of concrete confined by aramid fiber-reinforced polymer wraps using generative adversarial networks. *Appl Sci* 13(21):11991. <https://doi.org/10.3390/app132111991>
91. Alkhawaldeh SMA (2024) Enhancing flat slab design: machine learning and metaheuristic approaches to predict punching shear strength. *Asian J Civ Eng* 25:2459–2469. <https://doi.org/10.1007/s42107-023-00919-4>
92. Chen W, Qu S, Lin L, Luo Q, Wang T (2023) Ensemble learning methods for shear strength prediction of fly ash-amended soils with lignin reinforcement. *J Mater Civ Eng* 35(4):04023022. [https://doi.org/10.1061/\(ASCE\)MT.1943-5533.0004676](https://doi.org/10.1061/(ASCE)MT.1943-5533.0004676)
93. Amin MN, Iftikhar B, Khan K, Javed MF, AbuArab AM, Rehman MF (2023) Prediction model for rice husk ash concrete using AI approach: Boosting and bagging algorithms. In: *Structures*, vol 50. Elsevier, pp 745–757. <https://doi.org/10.1016/j.istruc.2023.02.080>
94. Mai SH, Nguyen DH, Tran VL, Thai DK (2023) Development of hybrid machine learning models for predicting permanent transverse displacement of circular hollow section steel members under impact loads. *Buildings* 13(6):1384. <https://doi.org/10.3390/buildings13061384>
95. Aydın Y, Işıkdag Ü, Bekdaş G, Nigdeli SM, Geem ZW (2023) Use of machine learning techniques in soil classification. *Sustainability* 15(3):2374. <https://doi.org/10.3390/su15032374>
96. Yücel M, Bekdaş G, Nigdeli SM (2023) Evaluation of performance of different machine learning techniques for structural models. In: Vasant P et al (eds) *intelligent computing and optimization. ICO 2023. Lecture notes in networks and systems*, vol 729. Springer, Cham. https://doi.org/10.1007/978-3-031-36246-0_23
97. Aydın Y, Bekdaş G, Nigdeli SM, Işıkdag Ü, Geem ZW (2023) Comparison of multilayer perceptron and other methods for prediction of sustainable optimum design of reinforced concrete columns. In: Bekdaş G, Nigdeli SM (eds) *Hybrid metaheuristics in structural engineering. studies in systems, decision and control*, vol 480. Springer, Cham. https://doi.org/10.1007/978-3-031-34728-3_12
98. Alabduljabbar H, Khan K, Awan HH, Alyousef R, Mohamed AM, Eldin SM (2023) Modeling the capacity of engineered cementitious composites for self-healing using AI-based ensemble techniques. *Case Stud Constr Mater* 18:e01805. <https://doi.org/10.1016/j.cscm.2022.e01805>
99. Khan MI, Abbas YM, Fares G, Alqahtani FK (2023) Strength prediction and optimization for ultrahigh-performance concrete with low-carbon cementitious materials—XG boost model and experimental validation. *Constr Build Mater* 387:131606. <https://doi.org/10.1016/j.conbuildmat.2023.131606>
100. Chi L, Wang M, Liu K, Lu S, Kan L, Xia X, Huang C (2023) Machine learning prediction of compressive strength of concrete with resistivity modification. *Mater Today Commun* 36:106470. <https://doi.org/10.1016/j.mtcomm.2023.106470>
101. Aydın Y, Bekdaş G, Nigdeli SM, Işıkdag Ü, Kim S, Geem ZW (2023) Machine learning models for ecofriendly optimum design of reinforced concrete columns. *Appl Sci* 13(7):4117. <https://doi.org/10.3390/app13074117>
102. Perfetto D, Rezazadeh N, Aversano A, De Luca A, Lamanna G (2023) Composite panel damage classification based on guided waves and machine learning: an experimental approach. *Appl Sci* 13(18):10017. <https://doi.org/10.3390/app131810017>
103. Yang J, Zeng B, Hang Z et al (2023) Multimodal machine learning approach for exploring the 28-day compressive strength of nanomaterials-reinforced cement composites. *Archiv Civ Mech Eng* 23:202. <https://doi.org/10.1007/s43452-023-00738-z>

104. Aydın Y, Cakiroglu C, Bekdaş G, Işıkdag Ü, Kim S, Hong J, Geem ZW (2023) Neural Network predictive models for alkali-activated concrete carbon emission using metaheuristic optimization algorithms. *Sustainability* 16(1):142. <https://doi.org/10.3390/su16010142>
105. Cakiroglu C, Aydın Y, Bekdaş G, Geem ZW (2023) Interpretable predictive modelling of basalt fiber reinforced concrete splitting tensile strength using ensemble machine learning methods and SHAP approach. *Materials* 16(13):4578. <https://doi.org/10.3390/ma16134578>
106. Kumar P, Pratap B (2024) Feature engineering for predicting compressive strength of high-strength concrete with machine learning models. *Asian J Civ Eng* 25:723–736. <https://doi.org/10.1007/s42107-023-00807-x>
107. Asgarkhani N, Kazemi F, Jakubczyk-Gańczyńska A, Mohebi B, Jankowski R (2024) Seismic response and performance prediction of steel buckling-restrained braced frames using machine-learning methods. *Eng Appl Artif Intell* 128:107388. <https://doi.org/10.1016/j.engappai.2023.107388>
108. Shahab Z, Anwar W, Alyami M, Hammad AW, Alabduljabbar H, Nawaz R, Javed MF (2024) Experimental investigation and predictive modeling of compressive strength and electrical resistivity of graphene nanoplatelets modified concrete. *Mater Today Commun* 38:107639. <https://doi.org/10.1016/j.mtcomm.2023.107639>
109. Liu T, Cakiroglu C, Islam K, Wang Z, Nehdi ML (2024) Explainable machine learning model for predicting punching shear strength of FRC flat slabs. *Eng Struct* 301:117276. <https://doi.org/10.1016/j.engstruct.2023.117276>
110. Gao Y, Mosalam KM (2018) Deep transfer learning for image-based structural damage recognition. *Comput Aided Civ Infrastruct Eng* 33(9):748–768. <https://doi.org/10.1111/mice.12363>
111. Yang X, Li H, Yu Y, Luo X, Huang T, Yang X (2018) Automatic pixel-level crack detection and measurement using fully convolutional network. *Comput Aided Civ Infrastruct Eng* 33(12):1090–1109. <https://doi.org/10.1111/mice.12412>
112. Liu Q, Feng C, Song Z, Louis J, Zhou J (2019) Deep learning model comparison for vision-based classification of full/empty-load trucks in earthmoving operations. *Appl Sci* 9(22):4871. <https://doi.org/10.3390/app9224871>
113. Tanyildizi H, Şengür A, Akbulut Y et al (2020) Deep learning model for estimating the mechanical properties of concrete containing silica fume exposed to high temperatures. *Front Struct Civ Eng* 14:1316–1330. <https://doi.org/10.1007/s11709-020-0646-z>
114. Shin HK, Ahn YH, Lee SH, Kim HY (2020) Automatic concrete damage recognition using multi-level attention convolutional neural network. *Materials* 13(23):5549. <https://doi.org/10.3390/ma13235549>
115. Weerasinghe IDTT, Jayasena KPN (2020) Multimedia big data platform with a deep learning approach for flood emergency management. In: 2020 5th international conference on information technology research (ICITR). IEEE, pp 1–6. <https://doi.org/10.1109/ICITR51448.2020.9310903>
116. Li Y, Lu Y, Chen J (2021) A deep learning approach for real-time rebar counting on the construction site based on YOLOv3 detector. *Autom Constr* 124:103602. <https://doi.org/10.1016/j.autcon.2021.103602>
117. Cui X, Wang Q, Dai J, Zhang R, Li S (2021) Intelligent recognition of erosion damage to concrete based on improved YOLO-v3. *Mater Lett* 302:130363. <https://doi.org/10.1016/j.matlet.2021.130363>
118. Tan Y, Cai R, Li J, Chen P, Wang M (2021) Automatic detection of sewer defects based on improved you only look once algorithm. *Autom Constr* 131:103912. <https://doi.org/10.1016/j.autcon.2021.103912>
119. Hong S, Nguyen HT, Jung J, Ahn J (2021) Seismic ground response estimation based on convolutional neural networks (Cnn). *Appl Sci* 11(22):10760. <https://doi.org/10.3390/app112210760>
120. Liu Z, Hu S, Sun Y et al (2022) An exploratory investigation into image-data-driven deep learning for stability analysis of geosystems. *Geotech Geol Eng* 40:735–750. <https://doi.org/10.1007/s10706-021-01921-w>

121. Wang C, Song LH, Fan JS (2022) End-to-End Structural analysis in civil engineering based on deep learning. *Autom Constr* 138:104255. <https://doi.org/10.1016/j.autcon.2022.104255>
122. Luleci F, Catbas FN, Avci O (2023) Generative adversarial networks for labelled vibration data generation. In: Allen M, Davaria S, Davis RB (eds) *Special topics in structural dynamics and experimental techniques*, vol 5. Conference proceedings of the society for experimental mechanics series. Springer, Cham. https://doi.org/10.1007/978-3-031-05405-1_5
123. Qayyum W, Ehtisham R, Bahrami A, Camp C, Mir J, Ahmad A (2023) Assessment of convolutional neural network pre-trained models for detection and orientation of cracks. *Materials* 16(2):826. <https://doi.org/10.3390/ma16020826>
124. Philip RE, Andrushia AD, Mathews ME, Anand N (2023) YOLO v3-based concrete wall damage detection with single class classification. In: 2023 9th international conference on smart computing and communications (ICSCC). IEEE, pp 439–442. <https://doi.org/10.1109/ICSCC59169.2023.10334995>
125. Matinfar M, Khaji N, Ahmadi G (2023) Deep convolutional generative adversarial networks for the generation of numerous artificial spectrum-compatible earthquake accelerograms using a limited number of ground motion records. *Comput Aided Civ Infrastruct Eng* 38(2):225–240. <https://doi.org/10.1111/mice.12852>
126. Inam H, Islam NU, Akram MU, Ullah F (2023) Smart and automated infrastructure management: a deep learning approach for crack detection in bridge images. *Sustainability* 15(3):1866. <https://doi.org/10.3390/su15031866>
127. Zhou Z, Zhang J, Gong C, Wu W (2023) Automatic tunnel lining crack detection via deep learning with generative adversarial network-based data augmentation. *Underground Space* 9:140–154. <https://doi.org/10.1016/j.undsp.2022.07.003>
128. Bai Z, Liu T, Zou D, Zhang M, Zhou A, Li Y (2023) Image-based reinforced concrete component mechanical damage recognition and structural safety rapid assessment using deep learning with frequency information. *Autom Constr* 150:104839. <https://doi.org/10.1016/j.autcon.2023.104839>
129. Dinh NNH, Ahn YH (2023) A deep learning-based image captioning for automated description of structural components condition. In: *The international conference on sustainable civil engineering and architecture*. Springer Nature Singapore, Singapore, pp 213–220
130. Zhang M, Guo T, Zhang G, Liu Z, Xu W (2024) Physics-informed deep learning for structural vibration identification and its application on a benchmark structure. *Phil Trans R Soc A* 382(2264):20220400. <https://doi.org/10.1098/rsta.2022.0400>
131. Hasheminejad MM, Sohankar N, Hajiannia A (2018) Predicting the collapsibility potential of unsaturated soils using adaptive neural fuzzy inference system and particle swarm optimization. *Scientia Iranica* 25(6):2980–2996. <https://doi.org/10.24200/sci.2018.20176>
132. Siva Krishna A, Ranga Rao V (2019) Strength prediction of geopolymer concrete using FUZZY (Open Access). *Int J Recent Technol Eng* 7(6C2):668–671. <https://www.ijrte.org/wp-content/uploads/papers/v7i6c2/F11230476C219.pdf>
133. Godfrey S, Asmare G, Gossa T, Paba M (2019) Fuzzy logic analysis of the build, capacity build and transfer (B-CB-T) modality for urban water supply service delivery in Ethiopia. *Water* 11(5):979. <https://doi.org/10.3390/w11050979>
134. Hashemi H, Zarabiha H (2020) Application of Gustafson-Kessel algorithm in finding seismically co-pattern zones, a case study on Meybod city. *راهبردهای مهندسی ژرفا* 14(4):67–77. <https://doi.org/10.30499/IJG.2020.227165.1263>
135. Hormozabad SJ, Soto MG (2021) Load balancing and neural dynamic model to optimize replicator dynamics controllers for vibration reduction of highway bridge structures. *Eng Appl Artif Intell* 99:104138. <https://doi.org/10.1016/j.engappai.2020.104138>
136. Chakraborty A, Goswami A (2022) The application of statistics and fuzzy logic in predicting slope stability. In: Kumar PG, Subramaniam KVL, Santhakumar SM, Satyam DN (eds) *Recent advances in civil engineering*. Lecture notes in civil engineering, vol 233. Springer, Singapore. https://doi.org/10.1007/978-981-19-0189-8_12
137. Rogulj K, Jajac N, Batinić K (2023) Flat roofs renovation planning on public buildings using fuzzy multi-criteria analysis. *Sustainability* 15(7):6280. <https://doi.org/10.3390/su15076280>

138. Hakim S, Jamaluddin N, Boon K, Mokhtatar S, Khalifa A, Jamellodin Z (2024) Development of adaptive neuro-fuzzy inference system to predict concrete compressive strength. In: Sabtu N (eds) Proceedings of AWAM international conference on civil engineering 2022—volume 2. AICCE 2022. Lecture notes in civil engineering, vol 385. Springer, Singapore. https://doi.org/10.1007/978-981-99-6018-7_24
139. Damadipour M, Nazarpour M, Alami MT (2019) Evaluation of particle size distribution using an efficient approach based on image processing techniques. Iran J Sci Technol Trans Civ Eng 43(Suppl 1):429–441. <https://doi.org/10.1007/s40996-018-0175-3>
140. Tiwari N, Satyam N, Patva J (2020) Engineering characteristics and performance of polypropylene fibre and silica fume treated expansive soil subgrade. Int J Geosynth Ground Eng 6:18. <https://doi.org/10.1007/s40891-020-00199-x>
141. Riazi A, Karmo D, Shikh Ibrahim MA, Amadou S (2020) Estimating the weight and the failure load of a spaghetti bridge: a deep learning approach. J Exp Theor Artif Intell 32(5):875–884. <https://doi.org/10.1080/0952813X.2019.1694590>
142. Carrasco M, Araya-Letelier G, Velázquez R, Visconti P (2021) Image-based automated width measurement of surface cracking. Sensors 21(22):7534. <https://doi.org/10.3390/s21227534>
143. Xu B, Liu C (2022) A 3D reconstruction method for buildings based on monocular vision. Comput Aided Civil Infrastruct Eng 37(3):354–369. <https://doi.org/10.1111/mice.12715>
144. Zhang T, Rahman MA, Peterson A, Lu Y (2022) Novel damage index-based rapid evaluation of civil infrastructure subsurface defects using thermography analytics. Infrastructures 7(4):55. <https://doi.org/10.3390/infrastructures7040055>
145. Shi L, Liang H, Wang D, Liu T (2022) Mesogenetic evaluation and design of coarse aggregate contact within asphalt mixture. J Mater Civ Eng 34(8):04022187. [https://doi.org/10.1061/\(ASCE\)MT.1943-5533.000434](https://doi.org/10.1061/(ASCE)MT.1943-5533.000434)
146. Ma Z, Zhao Q, Zhu Y, Cang T, Hei X (2022) An automatic extraction method of rebar processing information based on digital image. Mathematics 10(16):2974. <https://doi.org/10.3390/math10162974>
147. Jiang R, Zhou X, Ran M (2023) 3D Reconstruction and morphology analysis of coarse aggregate using optical laser triangulation and image processing technology. Road Mater Pavement Design 1–30. <https://doi.org/10.1080/14680629.2023.2225637>
148. Jayanthi N, Ghosh T, Meena RK, Verma M (2023) Length and width of low-light, concrete hairline crack detection and measurement using image processing method. Asian J Civ Eng 1–10. Jayanthi N, Ghosh T, Meena RK et al (2023) Length and width of low-light, concrete hairline crack detection and measurement using image processing method. Asian J Civ Eng (2023). <https://doi.org/10.1007/s42107-023-00939-0>
149. Ya'acob N, Zuraimi MDI, Rahman AAA, Yusof AL, Ali DM (2024) Real-time pavement crack detection based on artificial intelligence. J Adv Appl Sci Eng Technol 38(2):71–82. <https://doi.org/10.37934/araset.38.2.7182>
150. Saleem MM, Jo H (2018) Heterogeneous data fusion for impact force identification in truss structures. In: Sensors and smart structures technologies for civil, mechanical, and aerospace systems, vol 10598. SPIE, p 494. <https://doi.org/10.1117/12.2296763>
151. Bekdaş G, Nigdeli SM, Yang XS (2018) A novel bat algorithm based optimum tuning of mass dampers for improving the seismic safety of structures. Eng Struct 159:89–98. <https://doi.org/10.1016/j.engstruct.2017.12.037>
152. Nigdeli SM, Bekdaş G, Sayin B (2018) The effect of usage of inertia function in particle swarm optimization. AIP Conf Proc 1978(1). AIP Publishing. <https://doi.org/10.1063/1.5043892>
153. Borodin K, Zhangabayuly Zhangabay N (2019) Mechanical characteristics, as well as physical-and-chemical properties of the slag-filled concretes, and investigation of the predictive power of the metaheuristic approach. Curved Layered Struct 6(1):236–244. <https://doi.org/10.1515/cls-2019-0020>
154. Mishra M, Gunturi VR, Miranda TFD (2019) Slope stability analysis using recent meta-heuristic techniques: a comprehensive survey. SN Appl Sci 1:1674. <https://doi.org/10.1007/s42452-019-1707-6>

155. Yücel M, Bekdaş G, Nigdeli SM (2020) Minimizing the weight of cantilever beam via metaheuristic methods by using different population-iteration combinations. *WSEAS Trans Comput* 19:69–77. <https://doi.org/10.37394/23205.2020.19.10>
156. Kayabekir AE, Yücel M, Bekdaş G, Nigdeli SM (2020) Comparative study of optimum cost design of reinforced concrete retaining wall via metaheuristics. *Chall J Concr Res Lett* 11:75–81. <https://doi.org/10.20528/cjcr.2020.03.004>
157. Ulusoy S, Bekdaş G, Nigdeli SM (2020) Active structural control via metaheuristic algorithms considering soil-structure interaction. *Struct Eng Mech Int J* 75(2):175–191
158. Hussien AG, Amin M, Wang M, Liang G, Alsanad A, Gumaei A, Chen H (2020) Crow search algorithm: theory, recent advances, and applications. *IEEE Access* 8:173548–173565. <https://doi.org/10.1109/ACCESS.2020.3024108>
159. Ulusoy S, Kayabekir AE, Bekdaş G, Nigdeli S (2020) Metaheuristic algorithms in optimum design of reinforced concrete beam by investigating strength of concrete. *Chall J Concr Res Lett* 11:33–37. <https://doi.org/10.20528/cjcr.2020.02.001>
160. Kayabekir AE, Bekdaş G, Nigdeli SM (2021) Optimum design of reinforced concrete T-beam considering environmental factors via flower pollination algorithm. *Int J Eng Appl Sci* 13(4):166–178. <https://doi.org/10.24107/ijeas.1037908>
161. Bekdaş G, Yücel M, Nigdeli SM (2021) Evaluation of metaheuristic-based methods for optimization of truss structures via various algorithms and Lévy flight modification. *Buildings* 11(2):49. <https://doi.org/10.3390/buildings11020049>
162. Ulusoy S, Bekdaş G, Nigdeli SM, Kim S, Geem ZW (2021) Performance of optimum tuned PID controller with different feedback strategies on active-controlled structures. *Appl Sci* 11(4):1682. <https://doi.org/10.3390/app11041682>
163. Naderpour H, Parsa P, Mirrashid M (2021) Innovative approach for moment capacity estimation of spirally reinforced concrete columns using swarm intelligence-based algorithms and neural network. *Pract Period Struct Des Constr* 26(4):04021043. [https://doi.org/10.1061/\(ASCE\)SC.1943-5576.0000612](https://doi.org/10.1061/(ASCE)SC.1943-5576.0000612)
164. Yücel M, Bekdaş G, Nigdeli SM, Kayabekir AE (2021) An artificial intelligence-based prediction model for optimum design variables of reinforced concrete retaining walls. *Int J Geomech* 21(12):04021244. [https://doi.org/10.1061/\(ASCE\)GM.1943-5622.0002234](https://doi.org/10.1061/(ASCE)GM.1943-5622.0002234)
165. Bekdaş G, Nigdeli SM, Kayabekir AE (2023) Optimum design and tuning applications in structural engineering via swarm intelligence. In: Biswas A, Kalayci CB, Mirjalili S (eds) *Advances in swarm intelligence. studies in computational intelligence*, vol 1054. Springer, Cham. https://doi.org/10.1007/978-3-031-09835-2_6
166. Bardhan A, Asteris PG (2023) Application of hybrid ANN paradigms built with nature inspired meta-heuristics for modelling soil compaction parameters. *Transp Geotech* 41:100995. <https://doi.org/10.1016/j.trgeo.2023.100995>
167. Yücel M, Nigdeli SM, Bekdaş G (2023) Estimation models for optimum design of structural engineering problems via swarm-intelligence based algorithms and artificial neural networks. In: Biswas A, Kalayci CB, Mirjalili S (eds) *Advances in swarm intelligence. Studies in computational intelligence*, vol 1054. Springer, Cham. https://doi.org/10.1007/978-3-031-09835-2_14
168. Aydın Y, Bekdaş G, Nigdeli SM (2023) Reinforced concrete beam optimization via flower pollination algorithm by changing switch probability parameter. In: Vasant P et al (2023) *Intelligent computing and optimization. ICO 2023. Lecture notes in networks and systems*, vol 852. Springer, Cham. https://doi.org/10.1007/978-3-031-50330-6_7
169. Günay U, Ulusoy S, Bekdaş G, Nigdeli SM (2023) Optimum design of reinforced concrete columns in case of fire. In: Bekdaş G, Nigdeli SM (eds) *Hybrid metaheuristics in structural engineering. studies in systems, decision and control*, vol 480. Springer, Cham. https://doi.org/10.1007/978-3-031-34728-3_3
170. Huang Z, Wang J (2024) Bridge pier displacement prediction and control in subway tunnel construction. *J Eng Project Prod Manag* 14(1)

Chapter 7

Meta-heuristic Algorithms as an Optimizer: Prospects and Challenges (Part I)



Ata Jahangir Moshayedi, Seyed Taha Mousavi Nasab,
Zeashan Hameed Khan, and Amir Sohail Khan

1 Introduction

The search for optimal solutions extends across many disciplines, including engineering and science. Optimization efforts are driven by the common goals of reducing energy consumption and costs while increasing profit, efficiency, productivity, and performance. In addition to traditional and meta-hierarchical approaches, optimization algorithms based on nature-inspired models have emerged as highly effective tools. In group-based approaches, agents work together, and the collective actions of these agents inevitably lead to convergence. However, not all algorithms are suitable for the problem under consideration as each one has its own advantages and limitations [1]. Within the domain of hybrid optimization, heuristic and meta-heuristic strategies serve as approximation techniques. Heuristic algorithms, oriented toward pragmatic, near-optimal solutions, adeptly navigate the constraints of computational expediency but lack the assurance of global optimality [2]. To ameliorate these imperfections and rectify the limitations inherent in heuristic methodologies, the advent of meta-heuristic algorithms has been instrumental [3]. In the realm of engineering applications, the ascendancy of meta-heuristic optimization methodologies is appreciable. Their intrinsic reliance on rudimentary principles, rendering gradient

A. J. Moshayedi (✉) · A. S. Khan
School of Information Engineering, Jiangxi University of Science and Technology,
Ganzhou, Jiangxi 341000, China
e-mail: ajm@jxust.edu.cn

S. T. M. Nasab
Department of Mathematics, Iran University of Science and Technology, Tehran, Iran
e-mail: stbeit@alumni.iust.ac.ir

Z. H. Khan
Interdisciplinary Research Center for Intelligent Manufacturing & Robotics (IRC-IMR),
King Fahd University of Petroleum & Minerals (KFUPM), Dhahran 31261, Saudi Arabia
e-mail: zeashan.Khan@kfupm.edu.sa

information superfluous, renders them facile in their implementation [4, 5]. Furthermore, their robustness to avoid trapping into the local optima enables them for applications in diverse domains [6]. Meta-heuristic algorithms typically include exit mechanisms designed to overcome regional optimization bottlenecks. The flexibility of the algorithm, which can be adjusted to address a wide range of optimization challenges, significantly enhances the effectiveness of these techniques [7]. Conversely, a meta-heuristic algorithm epitomizes a state-of-the-art approach for traversing solution spaces in pursuit of quality outcomes.

Algorithms characterized by a singular solution are exemplified by the refrigeration simulation algorithm, the initiation of the search process commences with an initial candidate solution, subsequently undergoing refinement over successive algorithmic iterations. In contrast, population-based algorithms utilize a large number of randomly chosen population of solutions as the starting point for optimization. As the algorithm progresses, the population is improved. The advantages of using population meta-heuristics, when compared to those of single-solution algorithms, are universally applicable as follows:

1. The search space is rich in information-sharing capabilities, allowing for fast transfers to promising locations within the search environment.
2. The abundance of solutions eliminates the risk of falling prey to local best-fit solutions.

The advantages of enhanced exploration of algorithms based on single solutions can be seen in Swarm Intelligence (SI) which is a fascinating subdomain within population-oriented meta-heuristics algorithms. The fundamental principles of SI were first formulated in 1993, which encapsulates the *emerging collective intelligence of clusters of elementary agents*. Swarm Intelligence algorithms offer a powerful and flexible approach to optimization and problem-solving. Their ability to harness the collective intelligence of simple agents, adapt to changing conditions, and explore a wide solution space makes them a perfect choice for a range of applications.

SI algorithms provide a decentralized, adaptable, and parallel approach to optimization by global exploration of solution spaces, making them robust to noisy and dynamic environments. SI algorithms are computationally efficient, versatile, and intuitive, making them applicable across various domains, from engineering to finance. Their collective intelligence and adaptability offers advantages, enabling efficient problem-solving in complex scenarios. Within the realm of meta-heuristic algorithms, a common characteristic of these algorithms is the division of the search process into two distinct phases, i.e., exploration and exploitation. The exploration phase entails a systematic examination of promising regions within the search space, encompassing a wide array of potential solutions. This phase necessitates the inclusion of stochastic operators in the algorithm, allowing for global and random traversal of the search space. Conversely, the exploitation phase involves the ability to conduct localized searches within the regions identified during the preceding exploration phase. Given the inherent randomness intrinsic to meta-heuristic algorithms, creating an appropriate balance between these two phases presents a formidable challenge.

Over the past two decades, optimization algorithms inspired by natural phenomena have achieved remarkable success as sophisticated optimization techniques, running in parallel with traditional methodologies.

This chapter explores algorithms inspired by mammalian behavior, subjecting them to rigorous investigation and comparative analysis, elucidating the respective pros and cons associated with each. The structure of this chapter is as follows: Sect. 2 provides an overview of various single-solution methods, while the subsequent Sect. 3 delves into population-based and nature-inspired algorithms. In the Sect. 4, a comprehensive discussion and comparison of meta-heuristic algorithms is discussed, shedding light on the benefits and drawbacks inherent in each approach. The final section encompasses the presentation of results, culminating in a comprehensive summary of these comparative analyses.

2 Single-Solution-Based Meta-heuristic Algorithms

This category of meta-heuristic algorithm is denominated as the “single-solution” approach, characterized by its capacity to refine and optimize a sole solution iteratively during implementation to attain the optimal outcome. In essence, these algorithms are dedicated to the amelioration of solutions while actively exploring the local neighborhood of the solution space. Numerous examples of single-solution-based algorithms exist, including but not limited to hill climbing, Simulated Annealing (SA), Tabu Search (TS), and Variable Neighborhood Search (VNS) [8]. Within this category, our debate will provide a concise overview of the SA and TS variants in the ensuing discussion.

2.1 *Simulated Annealing (SA)*

The Simulated Annealing algorithm (SA), conceived during the period spanning 1983 to 1985 by S. Kirkpatrick and colleagues [9], introduced a progressive annealing strategy tailored for addressing intricate optimization challenges. This innovative approach draws its inspiration from the principles of statistical mechanics, wherein the annealing process involves the gradual heating and subsequent cooling of a material, ultimately culminating in the formation of a stable crystalline structure. The robustness of this resulting structure is intricately linked to the rate at which the material undergoes cooling. If the initial temperature is too low or if a rapid cooling process is used, the result is imperfections in a metastable state, resulting in an incomplete thermal equilibrium. Conversely, a slow and steady cooling process results in the formation of robust crystalline structures.

The Simulated Annealing technique adeptly emulates the dynamic energy transformations intrinsic to cooling processes until the eventual attainment of thermal

Table 1 Simulated annealing algorithm (SA)

Start		
1	Input	Setting algorithm parameters and initial temperature
2	Initial solution	Generate an initial solution randomly
3	Starting temperature	Calculation of the degree of merit of the initial solution
4	Repeat	Save the initial solution as the best solution found so far
		Do steps 5–10 until the stop condition is met
5		Create a neighborhood of the current solution and consider it as the new solution
6		Calculate the merit of the new solution
7		If the merit of the new solution is better than the best solution, then consider the new solution as the best solution
8		Apply the SA selection rule
9		a. If the merit of the new solution is better than the current solution, replace it with the current solution
		b. If the merit of the new solution is not better than the current solution, then calculate the value of delta, which is equal to the difference between the merit of the new solution and the current solution
		c. Calculate the probability of selection according to the following formula
		$P = e^{\frac{-(E(s) - E(s'))}{T}} = e^{\frac{-\Delta f}{T}}$
		d. Replace the previous solution with the new solution with probability P
10		Reduce the temperature and go to step 5
11	Output	the best answer was discovered
End		

equilibrium is achieved, signifying a stationary frozen state. A description of core steps in SA algorithm are furnished in Table 1, while the algorithm’s workflow is visually encapsulated in Fig. 1.

Simulated Annealing exhibits degraded solutions under certain conditions, with the goal of preventing them from converging too soon. SA is different from other algorithms because it doesn’t rely on accumulated data during the search. The algorithm goes through many iterations, starting from an original solution. After each iteration, it creates a new random neighbor. It accepts any changes that make the cost function bigger, but it keeps a flexible approach to finding solutions. Depending on the temperature and how fast the objective function deteriorates, the neighboring solution is chosen based on a pre-determined probability called “acceptance probability”. This represents the disparity between the newly created nearby solution with

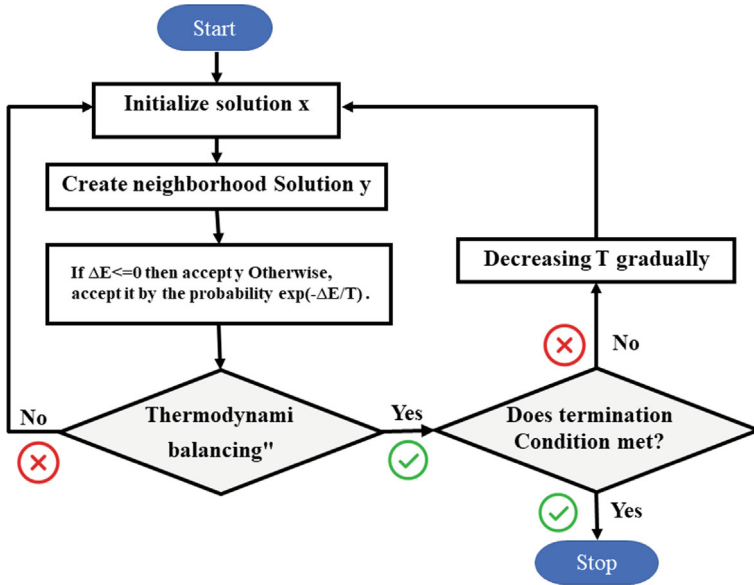


Fig. 1 Flowchart of simulated annealing algorithm

the current solution for the objective, often called energy. As the algorithm grows, the probability decreases over time. In a bigger picture, this probability follows the principles of Boltzmann's distribution.

$$P = e^{\frac{-(E(s) - E(s))}{T}} = e^{\frac{-\Delta f}{T}} \quad (1)$$

This method utilizes a pivotal control parameter known as “temperature” to compute the probability of accepting solutions that do not yield improvements. Once a state of equilibrium is attained, the temperature undergoes a gradual reduction in conformity with a prescribed cooling schedule, leading to the acceptance of a limited subset of solutions as the search process draws to a close.

2.2 Tabu Search (TS) Algorithm

In 1986, Glover [10] pioneered the development of the Tabu Search Algorithm (TS), a seminal meta-heuristic optimization algorithm which starts iterations from an initial solution, driven by the pursuit of the optimal solution to a given optimization problem. It delineates a neighborhood around the current solution at each iteration. Within this neighborhood, the algorithm identifies the most favorable neighboring solution provided this solution does not appear on the *Forbidden List*, the algorithm

Table 2 Tabu search algorithm (TS)

Start		
1	Initial Solution:	$S = S_0$
2		Initialize the tabu list, the medium-term memory, and the long-term memory
3	Repeat:	Locate the most acceptable neighbor s' . /* The non-tabu or aspirational standard is met*/
4		$s = s'$
5		Refresh your short-, medium-, and long-term memories as well as your tabu list and aspiration conditions
6		If the criteria for intensification are satisfied, then intensity
7		diversification if the diversification requirement is true
8		Till the Stopping criteria are satisfied
9	Output	The top response was found
End		

proceeds to transition to this next proximate solution (referred to as TL). If, however, the neighbor’s solution appears on the forbidden list, the algorithm uses a key “aspiration” factor that considers the possibility that if the neighbor’s solution shows better performance than the best solution in hand, the algorithm will move to it even if it’s on the forbidden list.

The forbidden list undergoes continual updates as the algorithm advances to neighboring solutions. Each prior move, which precipitated in the transition to a neighbor’s solution, is appended to this list to prevent the algorithm from reverting to the same solution in subsequent iterations. This mechanism, encapsulated within the forbidden list, plays a pivotal role in averting premature convergence to locally optimal solutions. The Tabu Tenure parameter assumes a critical role in determining the duration for which movements remain on the Forbidden List. The iterative process, transitioning from the current solution to neighboring solutions, persists until a predefined termination condition is met. Various termination conditions may be considered, including limits on the number of moves based on neighbor solutions.

The algorithm’s operational details are summarized in the pseudo-code Table 2, while Fig. 2 provides a graphical representation of the Tabu Search algorithm’s flowchart within the context of the *Prohibited Search Algorithm*.

In the realm of designing a Simple Tabu Search (TS) algorithm, several specific design considerations emerge, supplementing the customary challenges encountered in designing search-based Meta-heuristics (S-meta-heuristics), such as defining the neighborhood and initializing the solution. Notably, these include:

- **Tabu List:** The incorporation of a short-term memory mechanism serves the purpose of averting revisits to previously explored solutions. Yet, maintaining an exhaustive record of all accessed solutions for efficiency reasons proves impractical.

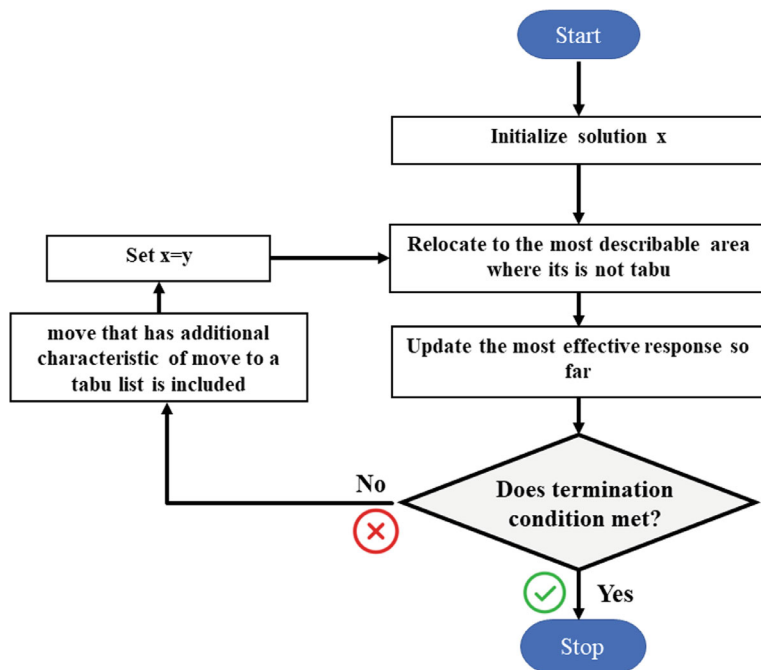


Fig. 2 Flowchart of Tabu Search (TS) algorithm

- Aspiration Criterion:** A prevalent aspiration criterion involves the selection of a tabu move exclusively when it yields a solution superior to the best one already ascertained. Consequently, the identification of the best solution within a subset of solutions possessing specific attributes can be influenced by a tabu move, thus constituting an additional dimension of the aspiration criterion.

In order to address the intricacy and diversity inherent in the search process, a suite of cutting-edge techniques is frequently incorporated into the framework of Tabu search:

- Medium-Term Memory:** Within this cognitive reservoir resides a repository housing the paramount, i.e., the most optimal, solutions unearthed during the course of the search. The objective here is to accord preferential treatment to the attributes inherent in this assemblage of superior solutions, often leveraging weighted probability methodologies. These attributes serve to bias and steer the outcomes of the search process, reinforcing the exploration of promising areas within the search space.
- Long-Term Memory:** The long-term memory compartment serves as a depository for information pertaining to solutions previously accessed and examined. It plays a pivotal role in guiding the search toward uncharted territories within the solution space. For instance, it serves as a bulwark against the inadvertent

generation of solutions mirroring the characteristics of elite solutions, thereby facilitating the expansion of the search into previously unexplored regions of the solution space.

3 Population-Based and Nature-Inspired Meta-heuristic Algorithms

In the pursuit of optimal solutions, population-based algorithms engender dynamic transformations within the set of candidate solutions. These algorithms encompass a diverse array of methodologies, including Genetic Algorithms (GA) [11], Ant Colony Optimization (ACO) [12], Particle Swarm Optimization (PSO) [13], as well as bio-inspired approaches such as bee and Firefly algorithms. In recent years, the landscape of meta-heuristic algorithms has witnessed the emergence of innovative and notable contenders inspired by the natural world. Among these algorithms, the Grey Wolf Optimization algorithm (GWO) [14], Dragonfly Optimization Algorithm (DA) [15], Flower Pollination Optimization algorithm (FPO) [16], Whale Optimization Algorithm (WOA) [17], Grasshopper Optimization Algorithm (GOA) [18], and Harris Hawks Optimization (HHO) [19] stand out prominently. This section proceeds to expound upon select algorithms from this distinguished cohort.

3.1 *The Genetic Algorithms (GA)*

John Holland, a researcher at the University of Michigan, came up with the idea of using hereditary computations in engineering optimizations in the 1970s. The exchange of acquired traits for qualities is at the heart of this computation. The task of machine intelligence is frequently accomplished by examining the problem area to identify the best response, albeit not necessarily optimal. The genetic algorithm is a typical approach that mimics the laws of cooperative organic progress. It employs Darwin's characteristic determination standards to find the best equation for anticipating or coordinating designs. For regression-based expectation techniques, the genetic algorithm is typically an effective choice.

Within the answer space, genetic algorithms execute the most of the solution search which begins by generating an era that may be relied on to generate a group of vital look focuses known as "primary population," which can be chosen deliberately or accidentally. In handling that relies on normal determination, the current population is chosen in proportion to the appropriateness of its generation for the next era, because GA uses factual strategies to directly look operations guaranteeing the ideal conclusion. Then hereditary administrators are connected, including various combinations, change, and other potential administrators, and a new population is created. Following that, an unused population replaces the previous population, and the cycle repeats.

Table 3 Genetic algorithm (GA) pseudo-code

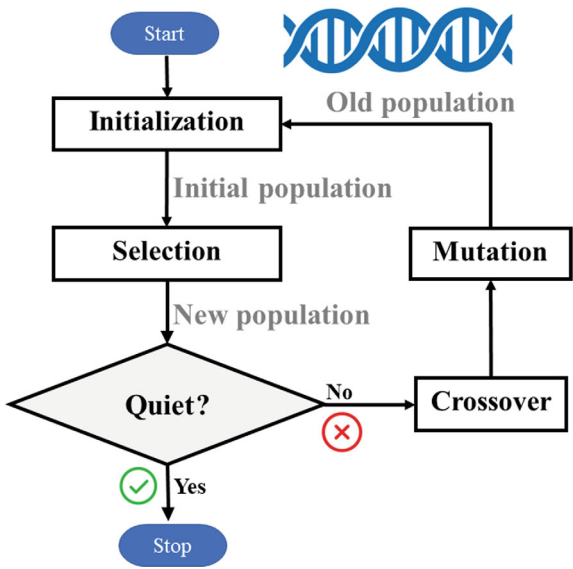
Start		
1	Beginning	Generate a population of n chromosomes randomly (proper solutions to the problem); /* initialization */
2	Rate	Evaluate the fitness $f(x)$ of each X chromosome in the population
3	New population	Form a new population. Repeat the following steps to complete the new population
4	Selection	Select two chromosomes (parents) from the population according to their fitness (the higher the fitness, The better the chance of selection)
5	Combination	Combine the parents to form new children according to the possibility of combining
6	Mutation	According to the probability of mutation, mutate the children in each locus (position in the chromosome)
7	Adoption	Incorporate new offspring into the new population
8	Replacement	Apply the newly created population to the algorithm process
9	Return	the most fittest individual from $f(x)$
End		

The emergent population typically exhibits an enhanced degree of fitness, signifying a progressive refinement over time. Attaining the zenith of conceivable iterations, either through convergence or the fulfillment of pre-determined termination criteria, culminates in a successful search. A succinct representation of the Genetic Algorithm is encapsulated within the pseudo-code provided in Table 3, while Fig. 3 intricately delineates the algorithm's procedural flowchart for the restricted search paradigm.

The key stages of a genetic algorithm:

1. **Coding:** In this initial and challenging step, the algorithm represents the problem's parameters in a coded format, often employing strategies like parallel coding, where the goal is to convert the solution into binary numbers.
2. **Assessment:** The fitness function is derived by transforming the target function and optimizing it. It assigns a numeric value to each solution, indicating its quality. Higher-quality solutions have a greater chance of contributing to the next generation.
3. **Combination:** The central operator in genetic algorithms is the combination operator. It merges chromosomes from previous generations to create a new generation. Parent sets exchange their traits in this process, promoting the sharing of favorable qualities and reducing genetic diversity.
4. **Mutation:** Mutation introduces variability by altering certain qualities within the new population. It may remove or introduce traits, fostering diversity and potentially leading to novel solutions.

Fig. 3 Flowchart of the genetic algorithm



5. **Decryption:** After the algorithm produces the best solution, decryption reverses the encoding process to reveal the correct version of the answer. This step ensures the final result is interpretable and actionable.

3.2 The Particle Swarm Optimization (PSO)

The Particle Swarm Optimization (PSO) approach, introduced in 1995 by James Kennedy and Russell Eberhart, constitutes an innovative population-based stochastic search algorithm tailored for addressing intricate nonlinear optimization problems replete with discontinuous solutions. Inspired by the collective behaviors observed in nature, such as bird and fish migration, PSO relies on the coordination of group communication patterns which facilitates the exchange of personalized information, similar to the navigational signals seen in migratory birds. Although, not all particles are aware of the global optimum, the internal social interactions of each particle help to identify promising solutions through convergence.

In this paradigm, particles are strategically dispersed throughout the search space corresponding to the target optimization function. At each particle’s present location, an evaluation of the associated function value guides its trajectory within the search domain. Subsequently, a particle selects one or more top-performing peers, alongside a direction for movement, estimated by a fusion of its current state and past achievements. This iterative process recurs multiple times to achieve the desired optimization objective as shown in Algorithm 1.

Algorithm 1 The pseudocode of PSO1: **procedure** START

$$V_{ij}^{t+1} = w V_{ij}^t + c_1 r_{1j}^t (P_{\text{best},i}^t - x_{ij}^t) + c_2 r_{2j}^t (G_{\text{best}} - x_{ij}^t) \quad (2)$$

$$x_{ij}^{t+1} = x_{ij}^t + V_{ij}^{t+1} \quad (3)$$

2: Here in Eq 2 and 3;

3: V_{ij}^{t+1} = The velocity of a particle i in the direction j at a time $t+1$;4: w = Particle inertia coefficient $[0,5,1]$; c_1 = Personal Acceleration Learning Factor $[0,2]$;5: c_2 = Coefficient of learning or collective acceleration $[0,2]$;6: r_1, r_2 = Random numbers with a uniform distribution between $[0,1]$;7: $P_{\text{best},i}^t$ = The best personal experience of a particle i at the time t ;8: G_{best} = The best particle collective experience;9: x_{ij}^{t+1} = Position of the particle i in the direction j at a time $t+1$ 10: **end procedure**

The PSO algorithm is not just the sum of its components. In isolation, individual particles lack the capacity to solve intricate problems. However, through interaction with other particles, a collective capacity for problem-solving is built as a social entity, resulting from the interactions and behaviors of individual particles. In the start, random initialization is performed on the particles to determine their initial positions and speeds.

Subsequently, the position and velocity of each particle evolves as the algorithm progresses, based on the observations from previous steps. The PSO algorithm is primarily concerned with the complex interactions and dynamics that control the collective consciousness of the swarm of particles, driving it toward the ultimate global optimization goal.

3.3 The Ant Colony Algorithm (ACO)

Figure 4 shows the inter-relationships within the particle swarm algorithm through a concise schematic representation where the velocity attributes of particles navigating the search space within the PSO algorithm are evident. A comprehensive overview of the algorithm is detailed in the pseudo-code within Table 4, while Fig. 5 reflects the algorithm's procedural flowchart, delineating its restricted search framework.

The procedure unfolds as follows: initially, the particle resides at its present location. This process involves the addition of the previous location parameter to the new velocity parameter, ultimately determining the particle's velocity, as described by Eq. 2. Subsequently, the particle progresses toward its best personal experience, $P_{\text{best},i}^t$, in a manner influenced by both random factor r_{1j}^t and an acceleration coefficient c_1 specific to that particle. It moves in alignment with this direction. In the next phase, the particle further advances along the trajectory guided by the best collective experience, G_{best} . This movement is influenced by the total learning coefficient c_2 and

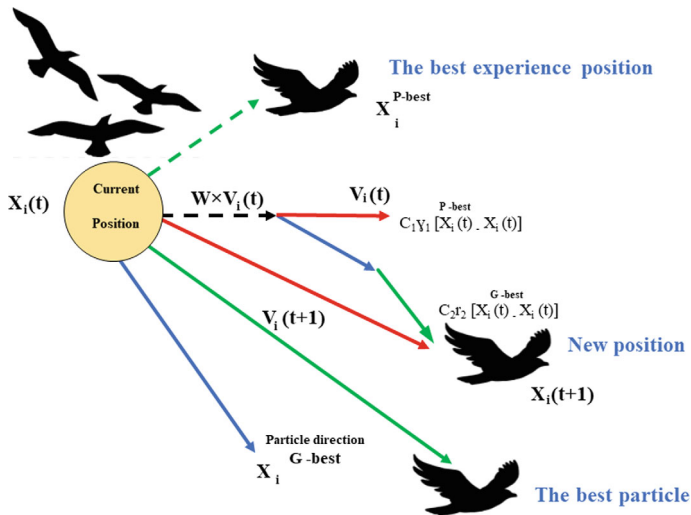


Fig. 4 Schematic representation of the particle swarm algorithm

Table 4 Particle swarm optimization algorithm (PSO)

Start		
1	Initialization	Initialization of algorithm parameters
2	Population	Creating a random population of particles
3	Evaluating	Evaluating each particle and calculating its merit
4		Considering the current position of each particle as its best experience
5	Repeat	Repeat steps 6–12 until the stop condition is met
6		Repeat steps 7–11 for each particle
7		Update the particle's velocity vector
8		Update the position of the particle according to its velocity
9		Calculate the merit of the new particle
10		If the competence level is better than the best personal experience, consider the new position as the best experience
11		Compare the new particle with the best overall particle and if it is better, replace it with the best overall
12	Output	If the stop condition is not met, go to step 5, otherwise end
End		

random factor r_{2j}^t . These three distinct motion vectors are then combined. Specifically, the vector connecting the current position of particle x_{ij}^t to the new position x_{ij}^{t+1} is constructed in accordance with the aforementioned steps. The resulting particle's velocity vector is denoted as the new step V_{ij}^{t+1} , and it is derived from the formula

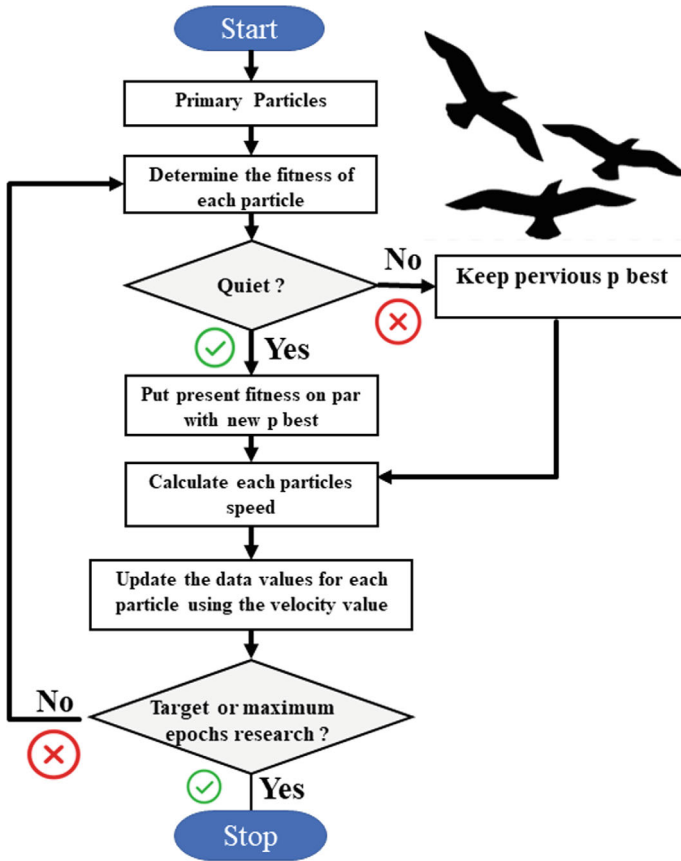


Fig. 5 The flowchart of PSO algorithm

in Eq. (3), which, when applied to the previous position of particle x_{ij}^t , yields the new position x_{ij}^{t+1} within the search space. The flowchart of the PSO algorithm is depicted in Fig. 5.

Dorigo's seminal work [12] introduced the concept of ant colony optimization (ACO) as a meta-heuristic approach for tackling challenging optimization problems. This innovative method draws its inspiration from the natural behavior of real ant populations. In the nature, ants search for food near their colonies through a random search. When they find food, they assess its nutritional value and transport a portion of it back to their nest. In the process, they deposit a chemical substance on the ground known as a "pheromone" trail which serves as a guide for subsequent journeys. Other ants are able to recognize and follow this trail, allowing them to collectively identify the most effective route from food source to colony which serves to discover optimal paths. This has inspired the development of artificial ant colonies capable of solving complex optimization problems, including the well-known traveling salesman prob-

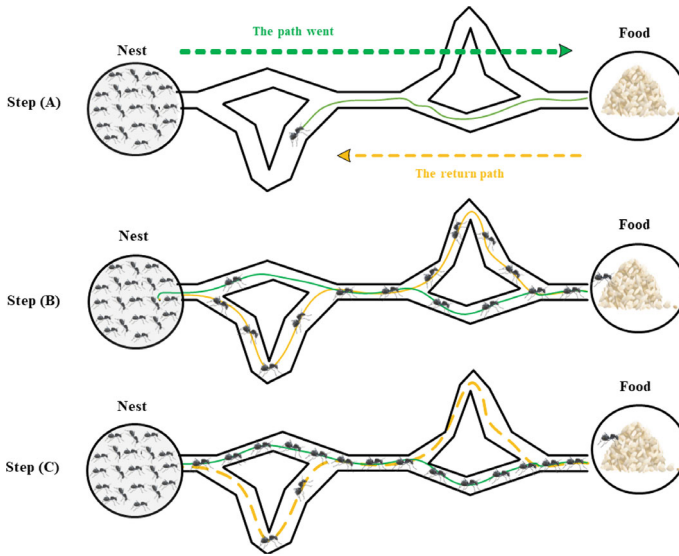


Fig. 6 The behavior of ants for food preparation

lem. Notably, since ants lack vision, their communication and interaction primarily rely on pheromone signals that mark their path. The more pheromone deposited on a particular route, the more frequently other ants utilize that path.

Consequently, the quantity of pheromone on a route signifies its optimality and efficiency. A concrete example in Fig. 6 illustrates the objective of finding the optimal route from the starting point (nest) to the destination point (food source). It may be assumed that the probability of an ant selecting the correct path equals the probability of selecting an incorrect one. However, the key insight lies in the fact that the first ant to reach the destination point and return to the nest effectively traces the shortest path, resulting in a higher concentration of pheromone on this route. Therefore, pheromone levels precisely indicate the preferred path for subsequent ants, ultimately leading to the discovery of the shortest and most optimal route. Pheromone evaporation and probability-based decision-making mechanisms endow ants with the ability to uncover the shortest path, thereby enhancing their adaptability in solving a wide array of optimization problems.

A crucial consideration in the ACO algorithm is the representation of the ant colony. In this context, a vector of n design variables, each corresponding to an individual ant, is employed to portray a colony of m ants. For continuous variables, the colony is represented as a matrix, denoted as $X = [x_1 \ x_2 \ \dots \ x_n]$. This representation framework forms the foundation upon which the ACO algorithm operates. The second aspect to consider pertains to the modeling of the pheromone communication network. In their work, Socha and Dorigo recommend the utilization of a normal distribution when devising a continuous model for this purpose. The standard deviation serves as a metric to gauge the dispersion of ants around the current optimum,

denoted as $x_{\min}$, within the design space. It quantifies the extent to which ants are distributed in proximity to this optimal point. To ensure consistent exploration of the design space by the ants, the algorithm selects the value of $x_{\min}$ randomly within the design space. This selection follows a uniform distribution that spans an area at least three times larger than the dimensions of the design space. Within the colony, each ant independently determines the values for the trial solutions at each iteration, guided by the distribution characteristics within the colony. The ACO algorithm continually updates the values of each design variable at each iteration, as described by Eq. 4.

Ultimately, the distribution of pheromones across the design space undergoes adjustment based on the information gathered during the optimization process. Given that Eq. 3 governs the representation of pheromone levels, the only requirement for this update is the tracking of $x_{\min}$. The `std(std("colony"))` function within the ant colony algorithm plays a pivotal role in maintaining the standard deviation for each individual plan parameter, thereby governing the dispersion of ants' trajectories across the solution space. The accumulation of pheromones is a critical endeavor within this algorithm, serving as a means to guide the discovery of optimal solutions within the candidate's territory. Conversely, strategies to mitigate the adverse effects of premature convergence are imperative. In this regard, a straight-forward approach involves the dissipation of pheromone levels. This dissipation mechanism operates by adjusting the prevailing standard deviation for each variable, introducing a dissolving rate denoted by $\gamma > 1$. It is worth noting that these artificial counterparts differ fundamentally from their real ant counterparts, as delineated in Table 4. Distinguishing real ants from artificial ants is based on the following:

- **Memory:** In the realm of artificial ants, the concept of memory assumes significance, enabling the retention of traversal paths and past movements for informed decision-making.
- **Artificial Obstacles:** Artificial ants introduce the notion of simulating obstacles within problem scenarios, altering problem intricacies to validate algorithmic robustness and explore diverse solution spaces.
- **Life in Discrete Environment:** Unlike their natural counterparts, genuine ants are inherently dependent on their colony for survival, unable to thrive in isolation.
- **Artificial Pheromone:** Artificial ants employ a representation of pheromone concentrations as a functional component, and guiding their exploration within the problem space.

The summary of the Ant Colony Optimization (ACO) algorithm is detailed in Table 5 in pseudo-code format. Additionally, Fig. 7 presents an illustrative flowchart delineating the algorithm's execution for the constrained search scenario.

Within this algorithm, the predominant components are Solution Construction and Pheromone Updating, both of which are iterative processes that constitute the core of the algorithm's operation.

- **Construction of Solutions:** The process of constructing solutions is governed by a probabilistic state transition rule. Artificial ants, functioning as stochastic greedy

Table 5 Ant colony optimization (ACO)

Start		
1	Initialization	Set the amount of pheromone
2	Repeat	For each ant Do
3		The creation of solutions utilizing the amount of pheromone
4		Amount of pheromone updated
5		Vaporization
6		Strengthening
7		Till stopping requirements
8	Output	The best solution
End		

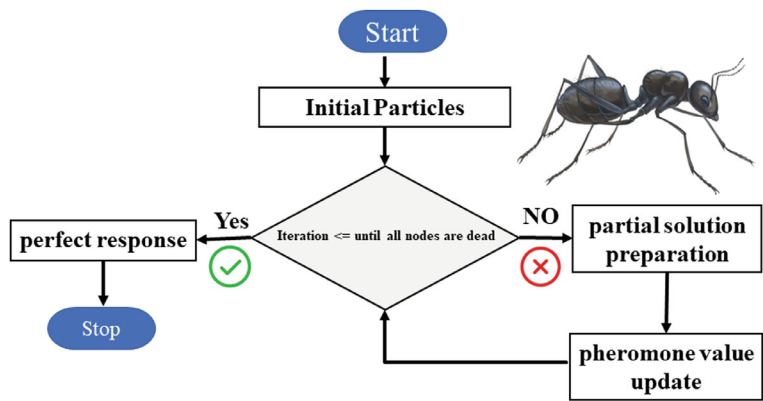


Fig. 7 The flowchart of the ACO algorithm

algorithms, employ probabilistic methods to assemble solutions. They combine partial solutions with complete ones iteratively until a comprehensive solution is formed. In the context of goal-oriented optimization problems, each ant constructs a path within a decision graph (or construction graph), adhering to this iterative procedure.

- **Pheromone Trails:** These trails serve as a repository of “optimal” solution characteristics, guiding ants in the construction of future solutions. These trails dynamically adapt throughout the search process to incorporate new insights. They effectively encapsulate the collective memory of the entire ant search operation.
- **Problem-Specific Heuristic Information:** Context-specific information provides ants with additional guidance as they make decisions regarding solution construction. This information equips ants with valuable cues for optimizing their solution-building process.

- **Pheromone Update:** The generation of solutions is aimed to update the pheromone levels. This involves two key procedures aimed at implementing a global pheromone update rule.
- **Evaporation Phase:** During this stage, the natural process of pheromone pathway reduction occurs. Each pheromone's value undergoes a specific percentage reduction, contributing to the overall adjustment of pheromone levels.

Algorithm 2 The Ant Colony Algorithms

1: **procedure** START

$$f_{\text{pheromone}}(x) = e^{-\frac{(x-x_{\min})^2}{2\sigma^2}} \quad (4)$$

$$\sigma = \text{std}(\text{colony}) \quad (5)$$

$$\tau_{ij} = (1 - \rho)\tau_{ij}, \quad \forall i, j \in [1, n] \quad (6)$$

2: **end procedure**

In Eq.6, the symbol $\rho \in [0, 1]$ denotes the rate at which pheromone levels decrease. This crucial parameter plays a pivotal role in preventing premature convergence of all ants toward “optimal” solutions. The evaporation process actively promotes diversity within the search area, fostering exploration. Another significant phase is the reinforcement step, which involves updating the pheromone trail based on the solutions generated. This update can be approached in three distinct ways, each offering a unique strategy for adjusting pheromone levels.

The applications of the ACO (Ant Colony Optimization) algorithm are diverse and extend to various domains where finding the shortest path is a critical objective. Notable examples include:

- **Solving Project Scheduling Problems with Limited Resources:** The ACO algorithm has been effectively employed to address project scheduling problems in scenarios with resource constraints, as demonstrated in reference [20].
- **Train Movement Scheduling:** An innovative model based on the concept of an ant community system has been applied to tackle the complex challenge of train movement scheduling, as highlighted in reference [21].
- **AntNet Algorithm Parameter Examination:** The ACO algorithm, particularly the AntNet variant, has been subject to in-depth examination and analysis of its algorithmic parameters, as discussed in reference [22].

These applications showcase the versatility and effectiveness of the ACO algorithm in addressing optimization problems across various domains.

3.4 The Artificial Bee Colony Algorithm (ABC)

Several swarm intelligence algorithms draw inspiration from the natural behavior of bees in the wild. These algorithms can be categorized into two groups: Exploration Behavior and Mating Behavior. The Artificial Bee Colony (ABC) algorithm, introduced by Karaboga and Basturk in 2008 [23], simulates the search behavior of bees. In this algorithm, individual entities, akin to individual bees within a colony, represent simple sets of behavioral policies such as migration, defense, and mortality. However, when these entities form a collective, mirroring a bee colony, they exhibit complex emergent behaviors characterized by valuable attributes like scalability and adaptability. The organizational system of bees is governed by a set of basic external rules common to insects. Despite the diverse species of insects on Earth, leading to variations in their behavior patterns, social insects like bees have demonstrated the ability to solve complex problems. A prime example is the nectar collection process, characterized by its sophisticated organization. Within a bee colony, bees prefer to follow established paths to flowers, rather than searching for new ones. Each beehive includes a designated dance floor where bees perform specific dances to communicate the location of flower patches. When a bee decides to forage for nectar, it selects a dancing bee from its hive, following the established path to the nectar source. After collecting nectar, each bee can perform one of the following actions based on measured probabilities:

- Drop the food source and seek another by following a dancing bee.
- Search for new food sources independently.
- Return to the hive and engage in dancing to attract other bees.

Algorithm 3 The Artificial Bee Colony Algorithms

1: **procedure** START

$$p_i = \frac{\text{Fitness}(S_i)}{\sum_{n=1}^{S_n} \text{Fitness}(S_n)} \quad (7)$$

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (8)$$

$$x_j^j = x_{\min}^j + \text{rand}[0, 1] \cdot (x_{\max}^j - x_{\min}^j) \quad (9)$$

$$(\mathbf{x}_i)(0) = \text{Init}(i, S), \quad 1 \leq i \leq N \quad (10)$$

$$\tau_i(t) = (1 - \rho)\tau_i(t - 1) + \frac{1}{\text{fit}(x_i) + 1} \quad (11)$$

2: **end procedure**

The decision-making process is influenced by probabilities, and bees suggest locations for nectar collection on the dance floor. While the mechanism of bee selection by other bees is not fully understood, it is clear that the quality of nectar found by the dancing bee plays a crucial role. The algorithm comprises worker bees, observer bees, and scout bees. Observer bees wait to make decisions about food sources, while

worker bees oversee their own food sources. Scout bees embark on random quests to discover new food sources. Each bee's position represents a potential solution to an optimization problem, with the quality of nectar (food quality) contributing to the solution's value. The algorithm initializes a virtual swarm of bees, and the search begins randomly. Bees communicate with one another until they locate the best nectar source, and the strength of these interactions determines the solution to the optimization problem. Worker bees assess the nectar quality of food sources and generate solutions based on local knowledge, primarily visual information. Bees remember the new positions and discard the previous ones because the newly discovered path have better quality than the previous ones. Once all worker bees complete their search, they inform the scout bees about the nectar sources and their locations which then choose a food source based on the probability (p_i), calculated using Eq. 7.

In the context of the algorithm, p_i represents the number of food sources, which is equivalent to the count of worker or observer bees. $Fitness(S_i)$ denotes the merit or quality value of the solution at position i , corresponding to the quantity of nectar obtained from the food source in that specific position. The ACO algorithm utilizes Eq. 8 to generate a new candidate food position based on the position stored in memory. Thus, the variables k (selected randomly from $k \in \{1, 2, \dots, S_n\}$) and j (chosen randomly from $\{1, 2, \dots, D\}$) represent individuals. It's worth noting that the value of k must be distinct from i . The parameter ϕ_{ij} assumes a random value within the range of $[-1, 1]$. ϕ_{ij} plays a crucial role in controlling the generation of neighboring food sources around x_{ij} . From this relationship, it becomes evident that reducing the difference between x_{ij} and x_{jk} leads to a decreased deviation from the position x_{ij} . Consequently, as the search process approaches the optimal solution, the step lengths also decrease. If the parameter's value exceeds a certain limit, it is reset to an acceptable value.

In the algorithm, food sources whose nectar is depleted by bees are replaced by new sources discovered by pioneering bees. This replacement involves randomly creating a new position and substituting it for an abandoned source. An artificial bee colony achieves this by randomly generating a new position and using Eq. 9 to identify a new food source if the situation does not improve after a specified number of iterations, which is referred to as "abandoned food source." The number of these iterations is a critical control parameter in the artificial bee algorithm and is commonly known as a constraint. Assuming the source that was abandoned is represented as $leftSleft$, a bee can discover a new food source by applying Eq. 9.

Following the creation and evaluation of a candidate source location ($v_i j$) by the artificial bee, its effectiveness is gauged in comparison to the previous location. If the new food source proves to be equal or superior in quality to the previous one, it replaces the old one in memory. Conversely, if the new source is inferior, the old one remains in memory. In essence, a decision is made through a greedy selection process between the prior location and the newly proposed location. The Artificial Bee Cloning Algorithm employs four distinct selection processes:

- 1. **Global Selection by Observer Bees:** Observer bees employ a global selection method to calculate the likelihood of discovering more promising regions at the current stage.
- 2. **Visual-Based Selection by Worker Bees:** Worker bees select locations within an area, and observer bees memorize food sources in proximity based on visual cues such as color, size, and scent of the blossoms (sources).
- 3. **Greedy Selection:** Greedy selection is conducted to identify a superior food source.
- 4. **Random Selection by Pioneer Bees:** Pioneer bees make random selections without following specific criteria.

These selection processes contribute to the algorithm’s ability to explore and exploit different food source options effectively, ultimately leading to improved solutions. The summary of the Artificial Bee Colony algorithm is presented in pseudo-code in Table 6, and the algorithm’s flowchart is illustrated in Fig. 8, providing a comprehensive overview of the algorithm’s operation and structure.

Initially, a designated area is selected for the search of nectar. Once the assessment of nectar availability within this defined zone is confirmed, the worker bee focus the food sources in close proximity. A re-evaluation of the nectar’s abundance is performed, leading to the selection of the most suitable nectar source among the options. In the event that the chosen nectar source fails to meet the desired quality criteria, the vigilant monitoring honey bee selects a neighboring food source for reassessment, thoroughly examining both its quantity and quality of nectar. Should the nectar align with the desired quality standards, the newly identified position is cataloged as the optimal food source and retained in memory for reference. The foraging bees then turn their attention to the task of identifying and regenerating abandoned food sources within their vicinity. If these newly identified food sources ultimately meet the quality criteria, the iterative cycle concludes; otherwise, the monitoring honey bee continues to select fresh proximate food sources for further evaluation.

Table 6 Artificial Bee Colony algorithm (ABC)

Start	
1	Create (n) populations or scout bees as random numbers
2	Finding the functional value of each of the bees in the first population
3	According to the value of the function, m places are selected as better points, out of which e points are chosen as the best points, and m-e places are selected as other better points among the places found by the bees
4	Specifying the neighborhood of these places to send new points
5	Send bees to selected points, and select more bees to be chosen
6	Determining the functional value of bees in each neighborhood or Golestan and its surroundings, to identify new scout bees, (m) new scout bees
7	Send new bees to (n-m) remaining points
8	Go to step 2 and repeat the steps until the stopping condition is met
End	

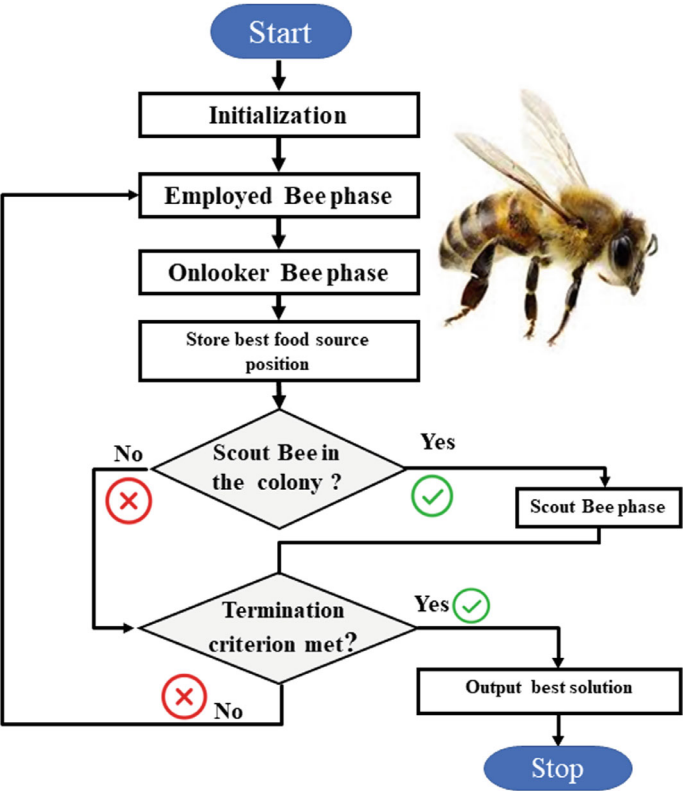


Fig. 8 The flowchart of the Artificial Bee Colony algorithm

4 Conclusion

In this chapter, we presented a comprehensive study of single-solution-based meta-heuristic algorithms as well as some of the population and nature-inspired meta-heuristic algorithms. The remaining nature based algorithms are discussed in the second part. The objective is to discuss the basic mechanism of action of each algorithm, their pseudo-codes, flowcharts, and key features that provides an insight into pros and cons of these optimization algorithms so that it becomes easy to choose one for a particular application.

References

- 1. Yang XS (2010) Nature-inspired metaheuristic algorithms. Luniver Press, Bristol
- 2. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: From natural to artificial systems, 1st edn. Oxford University Press, Oxford

3. Beheshti Z, Shamsuddin SMH (2013) A review of population-based meta-heuristic algorithms. *Int J Adv Soft Comput Appl* 5(1):1–35
4. Khan AH, Zhang W, Khan ZH, Shi J (2012) Evolutionary computing based modular control design for aircraft with redundant effectors. *Procedia Engineering* 29:110–117
5. Khan AH, Zhang W, Shi J, Khan ZH (2011) Optimized reconfigurable modular flight control design using swarm intelligence. *Procedia Eng* 24:621–628
6. Abdel-Basset M, Abdel-Fatah L, Sangaiah AK (2018) Metaheuristic algorithms: a comprehensive review. *Computational intelligence for multimedia big data on the cloud with engineering applications*. Elsevier, Amsterdam, pp 185–231
7. Dorigo M (1992) Optimization, learning and natural algorithms. Ph.D. Thesis, Politecnico di Milano
8. Talbi E-G (2009) Metaheuristics: from design to implementation. Wiley, Hoboken
9. Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
10. Glover F (1989) Tabu search-part I. *ORSA J Comput* 1(3):190–206
11. Brindle A (1980) Genetic algorithms for function optimization
12. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. *IEEE Comput Intell Mag* 1(4):28–39
13. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization: an overview. *Swarm Intell* 1:33–57
14. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
15. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Appl* 27:1053–1073
16. Yang X-S (2012) Flower pollination algorithm for global optimization. *International Conference on Unconventional Computing and Natural Computation*. Springer, Berlin, pp 240–249
17. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
18. Meraihi Y, Bennessaoud Gabis A, Mirjalili S, Ramdane-Cherif A (2021) Grasshopper optimization algorithm: theory, variants, and applications. *IEEE Access* 9:50001–50024
19. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Fut Gen Comput Sys* 97:849–872
20. Liu J, Liu Y, Shi Y, Li J (2020) Solving resource-constrained project scheduling problem via genetic algorithm. *J Comput Civil Eng* 34(2):04019055
21. Ghoseiri K, Morshedsolouk F (2006) ACS-TS: train scheduling using ant colony system. *J Appl Math Decis Sci*
22. Salman FM, Lehmoud AAM, Joda FA (2023) Adaptation of the ant colony algorithm to avoid congestion in wireless Mesh networks. *J Cyber Secur Mob* 785–812
23. Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. *Appl Soft Comput* 8(1):687–697
24. Yang X-S, He X (2013) Firefly algorithm: recent advances and applications. *Int J Swarm Intell* 1(1):36–50
25. Sayadi M, Ramezani R, Ghaffari-Nasab N (2010) A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. *Int J Ind Eng Comput* 1(1):1–10
26. dos Santos Coelho L, de Andrade Bernert DL, Mariani VC (2011) A chaotic firefly algorithm applied to reliability-redundancy optimization. In: 2011 IEEE congress of evolutionary computation (CEC). IEEE, pp 517–521
27. dos Santos Coelho L, Mariani VC (2012) Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning. *Comput Math Appl* 64(8):2371–2382
28. Dutta R, Ganguli R, Mani V (2011) Exploring isospectral spring-mass systems with firefly algorithm. *Proc R Soc A Math Phys Eng Sci* 467(2135):3222–3240
29. Yang X-S (2013) Chaos-enhanced firefly algorithm with automatic parameter tuning. In: *Recent algorithms and applications in swarm intelligence research*. IGI Global, pp 125–136
30. Rampriya B, Mahadevan K, Kannan S (2010) Unit commitment in deregulated power system using Lagrangian firefly algorithm. In: 2010 international conference on communication control and computing technologies. IEEE, pp 389–393

31. Fister Jr I, Yang X-S, Fister I, Brest J (2012) Memetic firefly algorithm for combinatorial optimization. arXiv preprint [arXiv:1204.5165](https://arxiv.org/abs/1204.5165)
32. Abdullah A, Deris S, Anwar S, Arjunan SN (2013) An evolutionary firefly algorithm for the estimation of nonlinear biological model parameters. *PLoS ONE* 8(3):e56310
33. Salman FM, Lehmoud AAM, Joda FA (2023) Adaptation of the ant colony algorithm to avoid congestion in wireless mesh networks. *J Cyb Secur Mob* 785–812
34. Horng M-H (2012) Vector quantization using the firefly algorithm for image compression. *Expert Syst Appl* 39(1):1078–1091
35. Zhang Y, Wu L (2012) A novel method for rigid image registration based on firefly algorithm. *Int J Res Rev Soft Intell Comput (IJRRSIC)* 2(2):141–146
36. Banati H, Bajaj M (2011) Firefly-based feature selection approach. *Int J Comput Sci Iss (IJCSI)* 8(4):473
37. Gandomi AH, Yang X-S, Talatahari S, Alavi AH (2013) Firefly algorithm with chaos. *Commun Nonlinear Sci Numer Simul* 18(1):89–98
38. Swarnkar KK (2012) Economic load dispatch problem with reduced power losses using firefly algorithm. *J Adv Comput Sci Technol* 1(2):42–56
39. Khan K, Sahai A (2012) A comparison of BA, GA, PSO, BP, and LM for training feed forward neural networks in e-learning context. *Int J Intell Syst Appl* 4(7):23
40. Yang X-S (2011) Bat algorithm for multi-objective optimization. *Int J Bio-Insp Comput* 3(5):267–274
41. Komarasamy G, Wahi A (2012) An optimized K-means clustering technique using bat algorithm. *Eur J Sci Res* 84(2):263–273
42. Lin J-H, Chou C-W, Yang C-H, Tsai H-L et al (2012) A chaotic Levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems. CIT, Citeseer
43. Gandomi AH, Yang X-S (2014) Chaotic bat algorithm. *J Comput Sci* 5(2):224–232
44. Nakamura RYM, Pereira LAM, Costa KA, Rodrigues D, Papa JP, Yang X-S (2012) BBA: a binary bat algorithm for feature selection. In: 2012 25th SIBGRAPI conference on graphics, patterns and images. IEEE, pp 291–297
45. Vu-Huu T, Pham-Van S, Pham Q-H, Cuong-Le T (2023) An improved bat algorithm for optimization design of truss structures. *Structures* 47:2240–2258
46. González JR, Sancho-Royo A, Pelta DA, Cruz C (2008) Nature-inspired cooperative strategies for optimization. In: Encyclopedia of networked and virtual organizations. IGI Global, pp 982–989
47. Yang X-S, Hossein Gandomi A (2012) Bat algorithm: a novel approach for global engineering optimization. *Eng Comput* 29(5):464–483
48. Yang X-S, Karamanoglu M, Fong S (2012) Bat algorithm for topology optimization in micro-electronic applications. In: The first international conference on future generation communication technologies. IEEE, pp 150–155
49. Komarasamy G, Wahi A (2012) An optimized K-means clustering technique using bat algorithm. *Eur J Sci Res* 84(2):263–273
50. Zhang JW, Wang GG (2012) Image matching using a bat algorithm with mutation. *Appl Mech Mater* 203:88–93
51. Reddy PDP, Reddy VU, Manohar TG (2012) Capacitor Placement for loss reduction in radial distribution networks: a two-stage approach. *J Electr Eng* 12(2):6–6
52. Chu S-C, Tsai P-W, Pan J-S (2006) Cat swarm optimization. In: PRICAI 2006: trends in artificial intelligence: 9th Pacific Rim international conference on artificial intelligence. Springer, pp 854–858
53. Tsai P-W, Pan J-S, Chen S-M, Liao B-Y, Hao S-P (2008) Parallel cat swarm optimization. In: 2008 international conference on machine learning and cybernetics. IEEE, vol 6, pp 3328–3333
54. Tsai P-W, Pan J-S, Chen S-M, Liao B-Y (2012) Enhanced parallel cat swarm optimization based on the Taguchi method. *Expert Syst Appl* 39(7):6309–6319
55. Santosa B, Ningrum MK (2009) Cat swarm optimization for clustering. In: 2009 international conference of soft computing and pattern recognition. IEEE, pp 54–59

56. Panda G, Pradhan PM, Majhi B (2011) IIR system identification using cat swarm optimization. *Expert Syst Appl* 38(10):12671–12683
57. Wang Z-H, Chang C-C, Li M-C (2012) Optimizing least-significant-bit substitution using cat swarm optimization strategy. *Inf Sci* 192:98–108
58. Bahrami M, Bozorg-Haddad O, Chu X (2018) Application of cat swarm optimization algorithm for optimal reservoir operation. *J Irrig Drain Eng* 144(1):04017057

Chapter 8

Meta-heuristic Algorithms as an Optimizer: Prospects and Challenges (Part II)



Ata Jahangir Moshayedi, Seyed Taha Mousavi Nasab,
Zeashan Hameed Khan, and Amir Sohail Khan

1 Population-Based and Nature-Inspired Meta-heuristic Algorithms (contd.)

The remaining algorithms in this section are described as follows:

1.1 The Termite Clone Optimization (TCO) Algorithm

In 1959, Paul-Pierre Grassé provided insights into the nest-building process of termites, revealing three distinct activities integral to their construction endeavors. Initially, a multitude of termites engages in seemingly random movements, with each termite commencing saliva secretion upon reaching a point slightly above the ground, thereby saturating the soil with saliva. Consequently, they shape small soil pellets through this salivary action. Remarkably, despite the apparent randomness of this behavior, the outcome exhibits a degree of regularity. At this stage's culmination, miniature mounds emerge in a confined area, fashioned from these saliva-infused soil pellets. These diminutive hills then trigger a shift in termite behavior, serving as signals for the termites. Upon reaching these hills, each termite elevates its saliva-

A. J. Moshayedi (✉) · A. S. Khan
School of Information Engineering, Jiangxi University of Science and Technology,
Ganzhou, Jiangxi 341000, China
e-mail: ajm@jxust.edu.cn

S. T. M. Nasab
Department of Mathematics, Iran University of Science and Technology, Tehran, Iran
e-mail: stbeit@alumni.iust.ac.ir

Z. H. Khan
Interdisciplinary Research Center for Intelligent Manufacturing and Robotics (IRC-IMR),
King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, Saudi Arabia
e-mail: zeashan.Khan@kfupm.edu.sa

fueled soil pellet production to a highly energetic level, effectively transforming the small hills into pillars. This conduct persists until each column reaches a predetermined height. Subsequently, termites exhibit a third behavior: if no other columns are proximate, they promptly abandon the current column; however, if numerous different columns surround the present one, termites embark on joining these columns, initiating nest construction. As the columns converge, termites communicate their need for assistance through scent and commence ceiling construction. Once a roof is complete, they diligently work on the remaining tops.

Grassé coined the term “signaling” to describe this behavior, denoting actions fostering coordination among organisms solely through environmental cues. Signaling presently stands as a pivotal concept in swarm intelligence. The termite clone optimization (TCO) algorithm, introduced by Hedayatzaheh et al. in 2010, draws inspiration from the intelligent actions of termites during movement and nest-building. Termites, akin to ants, navigate the search space randomly and gravitate toward areas rich in pheromones.

A termite colony embodies a decentralized organization capable of executing complex tasks by adhering to specific behavioral norms for each individual insect. Four primary factors govern termites’ self-organizational prowess: positive feedback, negative feedback, unpredictability, and extensive interactions. Table 1 provides a detailed breakdown of the source code for the termite colony optimization algorithm.

The termite clone optimization (TCO) method deploys a population of termites to navigate a D -dimensional search space denoted as $S \subset \mathbb{R}$ in pursuit of the optimal solution. Let us assume that there exist N termites within this region. Each termite, denoted as I within the population, is associated with a position vector $\mathbf{x}_i$, which furnishes a practical solution within the D -dimensional search space $S \subset \mathbb{R}$. In the context of TCO, each location vector $\mathbf{x}_i$ represents a hill, and the quality of this hill is quantified by the amount of pheromones, symbolized by the fitness value, adorning its slope.

The TCO algorithm leverages this approach to optimize a numerical function, wherein the initial steps involve the determination of termite count and Itermax, which signifies the maximum number of iterations. Subsequently, at the algorithm’s inception, each termite is randomly positioned within the search space. If i is a point in the search space S , and $\text{Init}(i, S)$ is the initialization function that places it somewhere randomly. Following starting, the termites alter their positions using the following method across iterations until the termination condition is met. After each iteration, the fitness value of each termite is evaluated. The fitness value determines the pheromone concentration at each termite site. The pheromone content at the j -th site is determined using the following equation.

In the context of the TCO algorithm, ρ represents the evaporation rate, which falls within the range of $[0, 1]$. Furthermore, τ_i and τ_{i-1} respectively denote the pheromone levels at the current and previous locations of the i -th termite. Based on localized perception, each termite undergoes adjustments to its trajectory and selects a new location after evaluating pheromone concentrations. The migration of termites

Table 1 Termite colony optimization algorithm (TCO)

Start:	
1	Initialize <i>Itermax</i> , Determine <i>N</i> , and τ
2	For <i>i</i> = 1 to <i>N</i>
3	Run equation (10). /Run equation (10). /every termite in the search space is placed at random when the algorithm starts/
4	Next <i>i</i>
5	Do
6	Calculate the termites' fitness
7	For <i>i</i> = 1 to <i>N</i>
8	Run equation (11)
9	Next <i>i</i>
10	For <i>i</i> = 1 to <i>N</i>
11	Find the termite's nearest neighbors in position <i>i</i>
12	Choose position <i>b_i</i> next-best.
13	If the termite is next to his neighbors, Then
14	If $\tau_i(t - 1) < \tau_{ii}(t - 1)$ Then
15	Run equation (13)
16	End If
17	Else
18	Run equation (12)
19	End If
20	Next <i>i</i>
21	Adjust radius τ and step size <i>s</i>
22	Until iter < <i>Itermax</i>
End	

is influenced by several factors, including the distance between a termite's current location and previously visited sites, as well as the pheromone concentration at those visited locations. These considerations have led to the emergence of two distinct termite movement patterns within the algorithm (Fig. 1).

When a termite finds itself within an area devoid of visited positions in its immediate vicinity, it resorts to random movement within that local region. Conversely, if a termite's nearby locations have been visited, it may choose to relocate to a position with the potential for higher returns. To explore more potentially lucrative areas, certain termites employ a random movement pattern. Within a radius of τ , these termites embark on a random walk, with their current location serving as the center of the search area. To update a termite's location, the following equation is employed:

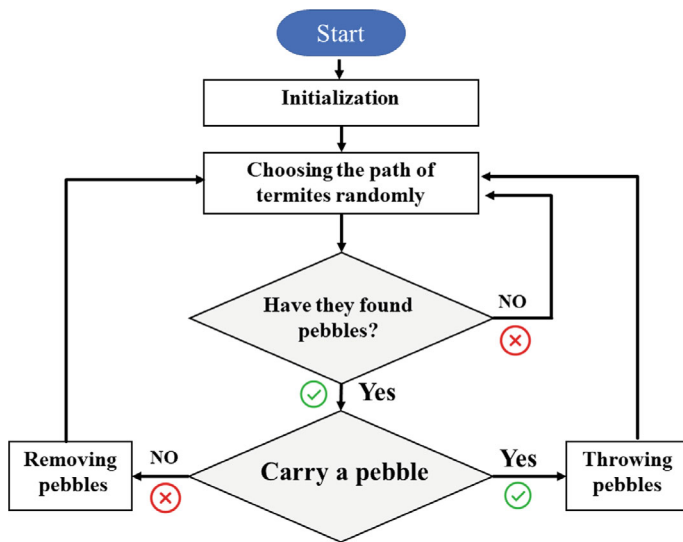


Fig. 1 Flowchart of the termite colony optimization algorithm

Algorithm 1 The Artificial Bee Colony Algorithms

1: **procedure** START

$$\mathbf{x}_i(t) = \mathbf{x}_i(t-1) + Rw(\tau, \mathbf{x}_i(t-1)) \quad (1)$$

$$\text{If } \tau_i(t-1) < \tau_{bi}(t-1), \quad \mathbf{x}_i(t) = \mathbf{x}_i(t-1) + \omega_b r_b(\mathbf{b}_i - \mathbf{x}_i(t-1)) \quad (2)$$

2: **end procedure**

In Eq. 1:

- $\mathbf{x}_i(t)$ = denotes the termite's former location, which is replaced with its new position (i.e., $\mathbf{x}_i(t)$).
- Rw = a random walk function that depends on the termite's current position and the radius of the search τ .
- $X_{\max}$ and $X_{\min}$ = indicate the maximum and minimum values of the search space along a dimension, respectively, are used to establish the starting value of radius as a percentage of $|X_{\max} - X_{\min}|$.

Over the course of repetitions, the value of the search space is progressively dropped from max to min based on how the termites modify their track. Termites can explore enormous areas of the search space because of the large value of τ during the initial rounds while the scout termites are encouraged to move more accurately in constrained areas by the modest values of τ in recent iterations.

A termite can utilize a selection process to modify its trajectory toward one of these pheromone gradients using the locally detected pheromones. In other words, the

termite assesses the given pheromone information and modifies its course to a spot where the concentration of pheromone is highest. If the amount of pheromone in the termite's present position is lower than that of the best local position (designated as b_i), it will travel toward the best local position. The equation below is used to regulate the termite's trajectory of movement. The termite's attraction to the local optimal place is probabilistically controlled by the factors $1 < w_b < 2$ and $0 < r_b < 1$.

The TCO algorithm encompasses two distinct movement patterns for its termite colony, fostering an efficient population-based approach for numerical function optimization through their heterogeneity. One subset of termites resorts to the random movement pattern, primarily consisting of individuals unable to harness the social information provided by their peers. This random pattern within TCO serves as a mechanism to introduce diversity and stochasticity into the behavior of the termite colony. The deliberate increase in population diversity is often employed to combat stagnation issues inherent in optimization. The contingent of termites engaging in random walks effectively regulates the colony's diversity within the TCO algorithm. This strategy serves as a perpetual stimulus, pushing the population out of states of stagnation by utilizing random termites to explore neighboring regions. This approach proves particularly effective when gradients do not align with the optimal solution or when numerous local optima point toward the global optimum.

Conversely, the other group of termites adopts the second movement style, leveraging social information to adjust their trajectory toward more promising areas. This pattern encourages the swarm to manage its exploration effectively and steer toward convergence on the global optimum. Moreover, it acts as a preventive measure against premature convergence and stagnation, which can arise from excessively constraining an individual's mobility trajectory within the colony.

1.2 The Firefly Algorithm (FA)

Despite its name, the firefly is a type of beetle and a subspecies of bugs. They possess unique sensory abilities to perceive their environment and possess the capacity to influence their surroundings. They emit a cool light devoid of infrared or ultraviolet wavelengths, displaying colors ranging from yellow to green or pale red, with wavelengths spanning 510 to 670 nm. Traditionally, scientists believed that the captivating displays of these insects' bio-luminescence served the purpose of attracting mates or luring prey. Approximately 2,000 species of fireflies have been documented in temperate and tropical climates and often found inhabiting marshes and forests, where their larvae find sustenance. Most species exhibit flight capability in both males and females though there are exceptions where females are flightless.

The firefly algorithm, developed by X. S. Yang [24], draws inspiration from the luminous qualities of fireflies, particularly the optical interactions among them. This algorithm exemplifies swarm intelligence, wherein the cooperation (and possibly competition) among individuals with simple and low-level intelligence results in a collective intelligence surpassing that of any single member. The characteristic

flashing light of fireflies serves as a pivotal aspect of this algorithm, and it serves two primary functions: attracting mates and warning potential threats.

The pattern of these flashes adheres to physical laws, with the flashing behavior of fireflies governed by criteria of attractiveness, brightness, and distance, as expressed by $I \propto \frac{1}{r^2}$. As distance (r) increases, light intensity (I) diminishes. This phenomenon has inspired Yang to model the firefly algorithm. Depending on the objective function, this flashing light can be represented in various ways, offering insights into the development of novel optimization techniques. The firefly algorithm does, however, exhibit two notable limitations: variations in light intensity and the formulation of attractiveness. For simplicity, one can assume that the inherent radiance of base-level fireflies corresponds to their luminosity and is further dictated by the objective function.

In the development of firefly-inspired algorithms, the following fundamental principles have been applied: All fireflies, regardless of gender, exhibit attraction to one another. Light intensity influences their attractiveness, with each pair of fireflies experiencing a reduction in attraction for dimmer individuals. The attraction strength correlates with their luminosity, which diminishes with increasing distance between two fireflies. If their light intensities are similar, their movement becomes random. The value of the target function determines the luminance of fireflies. These principles underscore the foundation of firefly-inspired optimization methods.

Types of Firefly algorithms While the fundamental firefly algorithm (FA) exhibits a considerable degree of efficacy, ongoing research has uncovered opportunities for enhancement. Numerous researchers have explored avenues to improve the algorithm's effectiveness and convergence rate. These research efforts have yielded various techniques, some of which are enumerated in Table 2.

The FA optimization algorithm stands out as one of the most effective techniques for addressing complex optimization problems. Leveraging concepts from the accelerated worm algorithm, it has spawned a family of multi-agent systems-based algorithms. Both real and artificial fireflies exhibit similar behaviors in this algorithm. It serves as a prime example of collective intelligence, demonstrating how relatively weaker agents can collaborate and work together to achieve outstanding results. The algorithm's applicability extends to renowned optimization challenges such as the "traveling salesman" problem and addressing routing challenges in long-distance communication networks, serving as a valuable tool for research and problem-solving in these domains.

Applications of the Firefly algorithm

The firefly algorithm has garnered significant attention in recent times and finds applications in various domains. A brief overview of some of its applications includes:

1. **Digital Image Compression:** Efficiently compressing digital images while minimizing computation time [33].
2. **Picture Registration Verification:** Employed for picture registration and verification tasks [34].
3. **Feature Selection:** Outperforming other feature selection algorithms in terms of consistency, efficiency, and optimality [35].

Table 2 Firefly algorithm (FA) variants

1	DFA: Discrete firefly algorithm	Sayadi et al. [25] have developed a potent discrete FA modification by extending the firefly method to handle NP-hard scheduling issues. Their findings suggested that DFA may outperform current algorithms like ACO
2	CFA: Chaotic firefly algorithm	Coelho et al. [26–28] presented a disordered FA in 2011, and this CFA outperformed previous methods. Yang [29] evaluated the primary FA irregularity features to maximize efficiency by altering, under various factors
3	LFA: Lagrangian firefly algorithm	Another fascinating form is the Lagrangian firefly technique, which Rampriya et al. [30] proposed to address the number requirement problem for a power system
4	MFA: Memetic firefly algorithm	With promising results, Fister Jr et al. [31] developed the discrete memetic firefly method, a variation of FA, to address combinatorial network coloring problems
5	HFA: Hybrid firefly algorithm	By fusing existing algorithms with FA, several hybrid algorithms have been created. For instance, Abdullah et al. [32] utilized FA with variable coefficients and ant colony optimization to estimate the parameters of the nonlinear biological model. They demonstrated how this combination may be a potent tool for numerous applications with the least amount of processing effort

- 4. **Engineering Design:** Demonstrated effectiveness in addressing nonlinear, multi-quality engineering design challenges [36].
- 5. **NP-Hard Scheduling Problems:** Successfully resolving NP-Hard scheduling problems through a discrete Firefly algorithm variant [37].
- 6. **Economic Loading:** Reducing power dissipation and solving economic loading problems [38].
- 7. **Scheduling and Traveling Salesperson Problems:** Offering viable solutions to scheduling and traveling salesperson problems.

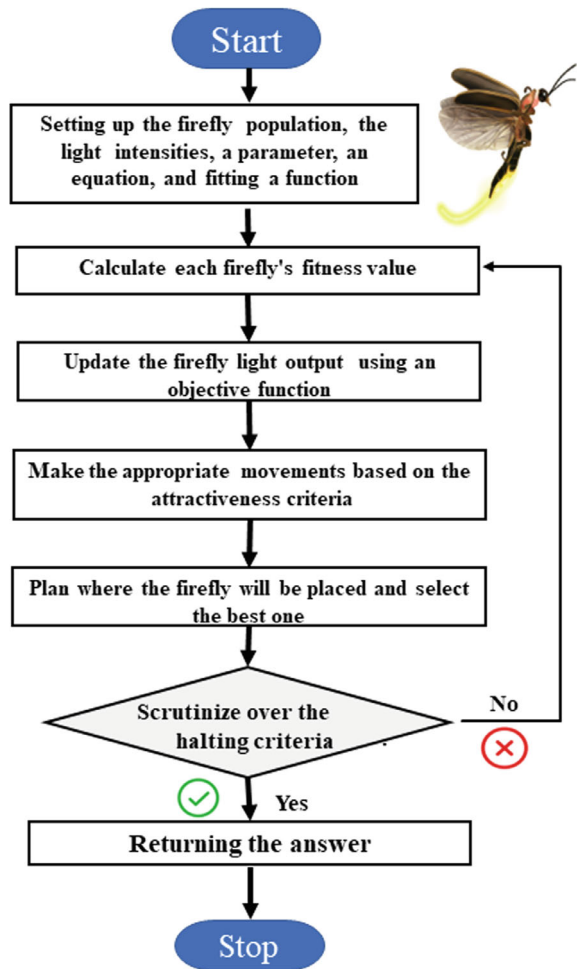
Furthermore, firefly algorithm finds applications in various other domains, including grid computing, queuing systems, combinatorial optimization, classification, clustering, optimization in dynamic environments, and independent test sequence generation in software testing. A summary of firefly algorithms is detailed as a pseudocode in Table 3, and Fig. 2 illustrates the flowchart of the prohibited search algorithm.

The firefly algorithm (FA) initiates by creating a population of fireflies, followed by an evaluation of their compatibility with the objective function. Subsequently, the light intensity of each firefly is adjusted based on this compatibility assessment. The fireflies are then ranked, and their positions are updated accordingly. If the desired objective has not been achieved, the process iterates, reevaluating all fireflies for compatibility with the goal function until the desired outcome is attained. This iterative process ensures that the algorithm converges toward the optimal solution.

Table 3 Firefly optimization algorithm (FA)

Start:	
1	Create a solution set $x_1, x_2, \dots, x_k$ at random
2	Calculate the intensity for each $I_1, I_2, \dots, I_k$ member solution
3	Move each firefly X toward other brighter fireflies, and If there aren't any other brighter fireflies nearby, relocate each firefly at random
4	Update the solution set
5	If a termination requirement is met, stop; if not, continue back to step 2
End	

Fig. 2 Flowchart of the firefly algorithm



Algorithm 2 The BAT Algorithms1: **procedure** START

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (3)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x^*) \cdot f_i \quad (4)$$

$$v_i^t = x_i^{t-1} + v_i^{t-1} \quad (5)$$

$$A_i^{t+1} = \alpha A_i^t, \quad r_i^{t+1} = r_i^0 (1 - \exp(-\gamma t)) \quad (6)$$

Where α and γ are fixed numbers. In essence, here is similar to the cooling coefficient in the Simulated Annealing algorithm. For every $0 < \alpha < 1$ and $\gamma > 0$ we have the following equation:

$$A_i^t \rightarrow 0, r_i^t \rightarrow r_i^0, t \rightarrow \infty \quad (7)$$

2: **end procedure****1.3 The Bat Algorithm (BA)**

Yang introduced the bat algorithm (BA) in 2010, inspired by the characteristics of echolocation used by bats to locate prey. The BA also incorporates a frequency tuning mechanism to expand the population's exploration capabilities. It simulates changes in pulse emission rate and sound characteristics of bats during their search for prey to strike a balance between exploration and exploitation.

Bats encompass a vast range of species, varying in size from small bee bats weighing merely 1.5 to 2 g to large bats with a wingspan of approximately 2 m and weighing up to 1 kg. Echolocation is used to varying degrees by different bat species. Microbats, the smaller varieties, rely extensively on echolocation for locating their nests in the dark, finding prey, and avoiding obstacles. They emit loud pulses of sound and perceive echoes from surrounding objects, with pulse characteristics differing among species and being associated with hunting behaviors. The focus here is on the characteristics of bat echolocation, and how they relate to optimizing an objective function, enabling the development of an intelligent BA algorithm. Yang devised the BA based on three fundamental principles:

1. **Universal Echolocation:** All bats utilize echoes to gauge distances and differentiate between food sources, obstacles, and background signals.
2. **Random Flight and Pulse Emission:** Bats fly at a velocity v_i in position x_i with a minimum frequency $f_{\min}$, variable wavelength, and loudness A_0 during their prey search. They autonomously adjust the wavelength (or frequency) of their emitted pulses and modulate the pulse emission rate r based on the proximity to their target.
3. **Variable Volume:** While loudness can vary in several ways, it is assumed to range from a high (positive) A_0 to a minimum constant value $A_{\min}$.

For simplicity, BA does not employ beam tracking, although it could be considered for future enhancements. Beam tracking is computationally intensive but finds utility in computational geometry and various applications. Additionally, frequency and wavelength are intrinsically linked.

For example, a frequency range of [20 kHz, 500 kHz] corresponds to air wavelengths ranging from 0.7 mm to 17 mm. Consequently, the adaptation of frequency or wavelength can be tailored for diverse applications, taking into account ease of implementation and other considerations. Each bat is characterized by a velocity v_i^t and a location x_i^t in iteration t within a d -dimensional search space or solution domain. The best current solution among all bats is denoted as x^* . Accordingly, the three aforementioned principles can be formulated as equations governing the evolution of x_i^t and v_i^t .

In Eq. 14, $\beta \in [0, 1]$ is a random vector derived from a uniform distribution. As mentioned before, we can use wavelength or frequency for implementation; depending on the amplitude of the problem, we will use $f_{\min} = 0$, and $f_{\max} = 0(1)$. Each bat is first given a frequency that is evenly drawn from $[f_{\min}, f_{\max}]$ and assigned at random. The bat algorithm may be viewed as a resonant frequency method as a result, offering a perfect blend of exploitation and exploration.

The volume and pulse emission in the area with prospective solutions offer a way for automated control and auto magnification. To provide an efficient mechanism in the bat algorithm for controlling exploration and exploitation and switching to the exploitation phase if necessary, we must modify the height and emission rate of the pulse during repetitions. The volume decreases as the pulse propagation speed rises when a bat locates its prey. As a result, the loudness may be set to any level of comfort between $A_{\min}$ presuming the bat has just caught its victim and has momentarily ceased making noise. With these presumptions, we arrive at the equation shown below:

$$A_i^t \rightarrow 0, \quad r_i^t \rightarrow r_i^0, \quad t \rightarrow \infty \quad (8)$$

In the simplest case, we use $\alpha = \gamma = 0.9$ to 0.98 in our simulations. The summary of the bat algorithm is written in the pseudocode Table 4, and Fig. 3 depicts the flowchart of the prohibited search algorithm.

First, the initial parameters are set, and after defining the frequency, beat rate, and volume, they are set. The procedure ends if the maximum number of iterations is more than t , but if it is less than t , fresh solutions and speed updates are created along with the solution. If $\text{Rand} < r_i$, the best solutions are chosen, and regional solutions are created; otherwise, this problem is not taken into account. Now, if the relationship between $\text{Rand} < A_i$ and $f(x_i) < f(x^*)$ is true, new solutions are allowed; if not, it rates the bats and chooses the best option that is now feasible.

Types of bat algorithms

One of the key advantages of bat algorithm is fast convergence in iteration by switching from exploration, which has made it an effective algorithm for applications such as classification. The types of BA are shown in Fig. 4, which include:

Table 4 Bat algorithm (BA)

Start:	
1	Set up the bat populations x_i^0 (for $i = 1, 2, \dots, n$) and v_i^0 from scratch
2	Initialize the loudness, A_i^0 , the pulse rates r_i^0 , and the pulse frequency f_i^0
3	While (the halt condition is not satisfied)
4	Create fresh approaches by varying the frequency
5	Changing locations and speeds (Eqs. (15, 16))
6	rand > r_i^t
7	choose the best option from the available options
8	Generate a local solution $x_{\text{new}} = x_{\text{old}} + \varepsilon A_k$
9	End if
10	Create a fresh answer by flying at random
11	if rand < A_i^k and $f(x_i^k) < f(x_{\text{bsf}}^k)$
12	Accept the fresh approaches
13	Reduce A_i^{k+1} and increase r_i^{k+1} (eq. (17))
14	End if
15	Sort bats according to current best x_{bsf}^t
16	End While
End	

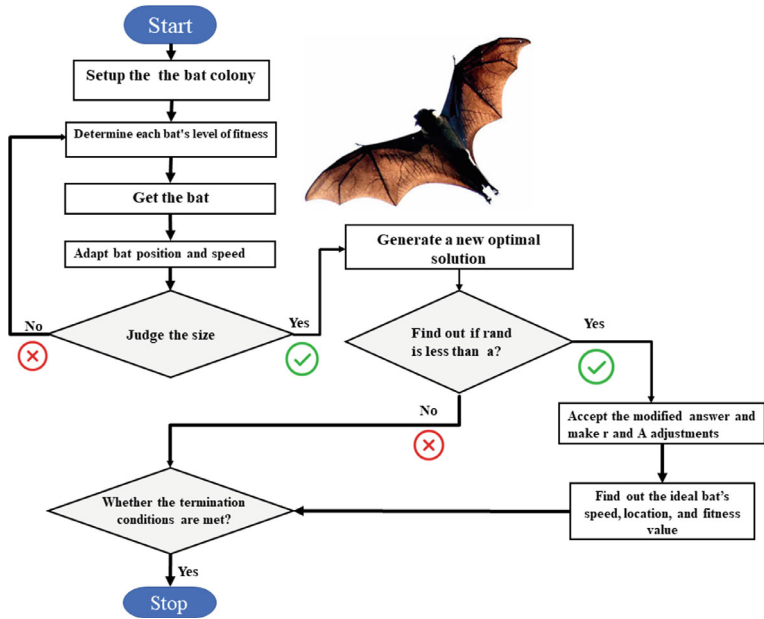


Fig. 3 Flowchart of the bat algorithm

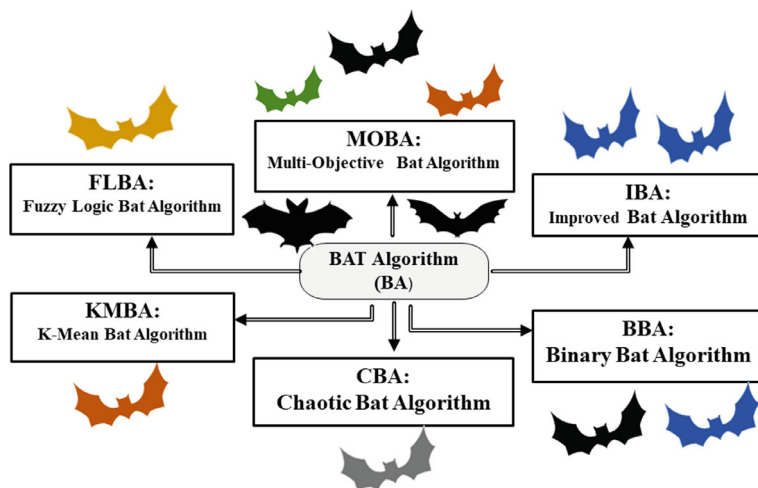


Fig. 4 Types of bat algorithms

1. **Fuzzy Logic Bat Algorithm (FLBA):** Sahai and Khan presented this type of algorithm by introducing fuzzy logic in BA [39].
2. **Multi-Objective Bat Algorithm (MOBA):** Yang added distributed multi-objective optimization to the bat algorithm, demonstrating its efficacy for resolving a variety of engineering design benchmarks [40].
3. **K-Mean Bat Algorithm (KMBA):** For efficient clustering, Komarasay and Wahi introduced a KMBA that combines K-means and bat algorithms [41].
4. **The Chaotic Bat Algorithm (CBA):** by Lin et al. They suggested this approach for parameter estimation in dynamic biological systems using Levy flight and disorder maps [42]. Another CBA utilizing various recurrent mappings was also presented by Gandomi and Yang [43].
5. **The Binary Bat Algorithm (BBA):** was developed by Nakamura et al. to address classification and feature selection issues [44].
6. **The Improved Bat Algorithm (IBA):** was presented by Jamil et al. [45] and effectively combined Levy flight with a precise estimate of changes in loudness and pulse rate.

Applications of Bat Algorithms

This algorithm has been used in the fields of optimization, classification, image processing, feature selection, scheduling, data mining, and other issues, which are explained in Table 5.

Table 5 Applications of bat algorithm

1	Continuous optimization	It is one of the earliest applications of BA that has received extensive study in optimization design engineering. It has been shown that BA can handle extremely nonlinear issues and find the best solutions [46, 47]. Case studies cover tall building design, batch systems, and pressure pipe design. BA is also proficient at resolving issues with many objectives
2	Combined scheduling and optimization	Continuous optimization issues can be readily addressed from the perspective of computational complexity, however, they may still be quite difficult
3	The estimate of parameters and inverse issues	By using BA, Yang et al. were able to perform parameter estimation as an inverse issue and examine topological shape optimization in microelectronics applications [48]. BA can deliver superior answers to least squares and sorting techniques if an inverse problem is appropriately phrased
4	Classification, clustering, and data mining	When Komarasamy and Wahi used BA to study clustering, they discovered that combining the two techniques might improve performance [49]. Other researchers have studied BA and found it advantageous in several domains, including electronic learning, clustering problems, using BA and its extension as a kind of Bayesian optimization, using BA to categorize data Subarray, etc
5	Image processing	A modified BA for image matching was reported by Zhang and Wang, who demonstrated that their model is more efficient and adaptable than previous models like genetic algorithms [50]
6	Fuzzy logic	Reddy and Manoj presented research that integrated fuzzy logic with BA to discover the capacitor values that would minimize overall losses in distribution networks [51]

1.4 Cat Swarm Optimization Algorithm (CSO)

Chu and his colleagues originally introduced the cat swarm optimization algorithm (CSO) in 2006 to address optimization issues[52]. This method is based on studying how cats behave. The studies' findings demonstrated that the cat swarm algorithm outperforms the particle swarm technique. Tsai and his associates introduced the parallel architecture of the cat crowding method in 2008 to enhance the program's convergence and search-ability [53]. Following that, adjustments to the parallel optimization technique of the cat swarm optimization were made by Tsai and his colleagues in 2011 to enhance the approach [54]. Cat swarming was a clustering technique used by Santosa and Ningrum in 2009 [55]. The clustering algorithm's performance is enhanced by changing the formulation. They enhanced the cat swarm optimization's categorization. This approach may also be used to solve optimization

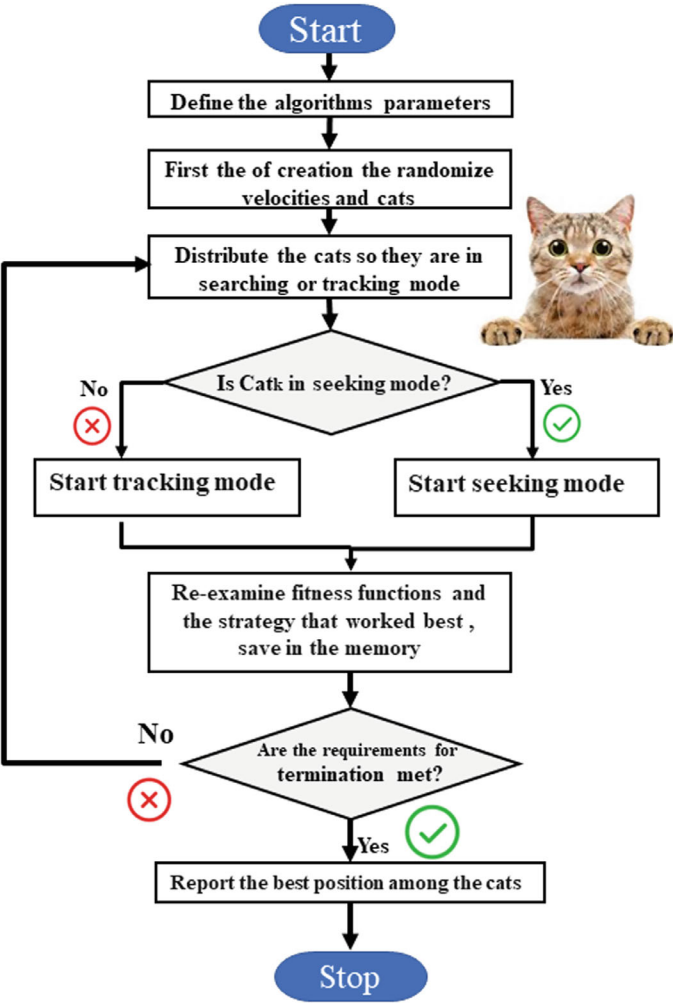


Fig. 5 Flowchart of the CSO algorithm

issues including system identification IIR [56], least-significant-bit replacement [57], and water resource systems [58].

The CSO algorithm’s precise computing processes are shown in Fig. 5, and the summary of the algorithm’s steps is included in Table 6 while the computational methods are summarized.

Seeking Mode (Resting)

The cat is sleeping in this manner while keeping an eye on its surroundings. The cat chooses its next course of action when it detects prey or danger. The cat moves slowly and cautiously if it chooses to do so and watches into the M-dimensional solution

Table 6 Cat swarm optimization algorithm (CSO)

Start:	
1	Make a population of cats as a starting point, disperse them over the M-dimensional solution space $X_{i,d}$, and select a maximum velocity value at random for each cat's velocity $V_{i,d}$
2	Give each cat a flag and classify them into the searching or tracing mode operation based on the MR value
3	Save the cat with the highest fitness level after considering each cat's fitness level. The finest answer to date is represented by the location of the best cat X_{best}
4	Utilize the cats in the tracing or seeking modes in accordance with the following instructions based on their flags
5	Terminate the procedure if the termination requirements are met. Alternatively, perform steps 2 through 5 again
End	

space while it is in the seeking mode, much as while asleep, to choose its next course of action. The cat is cognizant of its surroundings, its surroundings, and its options for mobility in this scenario. Four parameters—searching memory pool (SMP), seeking range of the selected dimension (SRD), counts of dimension to change (CDC), and self-position consideration—are used in the CSO algorithm to describe these (SPC).

SMP is the total number of copies made of each cat during the search procedure. SRD is the largest difference between the new and old values in the dimension selected for mutation. The CDC predicts how many dimensions will change. The algorithm's searching process is defined by each of these parameters. The Boolean variable SPC represents the cat's present location as a potential migration location. SMP's value cannot be impacted by SPC. Below is a description of the searching mode's operation.

- **Step1.** Create SMP versions of each cat. The cat's present location is kept as one of the copies if the SPC value is true, which causes SMP-1 copies to be produced.
- **Step2.** Use the equation below to get the new location in relation to CDC for each copy.

Where X_{cn} is the new position, R is a random number ranging from 0 to 1, and X_c is the current position.

- **Step3.** The fitness scores (FS) for the new positions should be calculated. The probability of each candidate's points being selected as one if all FS values are identical, Otherwise, use the following computation to determine the likelihood of choosing each candidate point.

Where P_i is the probability that the current candidate, cat_i , will be chosen, FS_i is the fitness value of cat_i , FS_{max} is the maximum value of the fitness function, FS_b is equal to FS_{max} for minimization problems, FS_{min} is the minimum value of the fitness function, and FS_b is equal to FS_{min} for maximization problems.

Algorithm 3 Cat Swarm Optimization Algorithm

 1: **procedure** START

$$X_{cn} = (1 \pm SRD \cdot R) \cdot X_c \quad (9)$$

$$P_i = \frac{|FS_i - FS_b|}{|FS_{\max} - FS_{\min}|}, \quad \text{where } 0 < i < j \quad (10)$$

$$v_{k,d} = v_{k,d} + r_1 \cdot c_1 (X_{\text{best},d} - X_{k,d}) \quad (11)$$

$$X_{k,d,\text{new}} = X_{k,d,\text{old}} + v_{k,d} \quad (12)$$

 2: **end procedure**

Monitoring Mode (Movement)

The tracking mode mimics a cat pursuing its prey. The cat chooses its speed and direction depending on the position and speed of the prey after locating it while in resting (search mode). The CSO provides the following as the cat k speed in dimension d . Where in Eq. 11, $v_{k,d}$ is the location of the cat with the best solution, $X_{k,d}$ is the position of the cat_k , c_1 is a constant, and r_1 is a random number between $[0,1]$. The cat traverses the M -dimensional decision space at this speed, reporting each new location it assumes. The cat's speed will be set to the maximum speed if it surpasses the maximum speed. Each cat's new location is determined using the formula below. While in Eq. 12, $X_{k,d,\text{new}}$ is the cat's new location in dimension d and $X_{k,d,\text{old}}$ is the cat's existing position in dimension d .

Termination Requirements

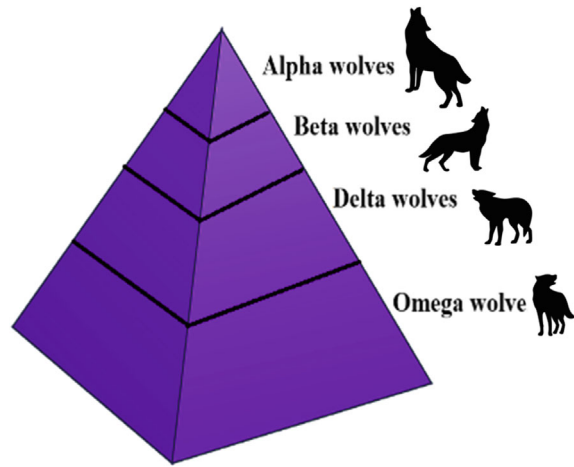
The algorithm is terminated based on the termination criterion. A good termination condition must be chosen in order for the method to converge correctly. Common CSO termination criteria include running duration, iteration count, and degree of improvement.

1.5 The Grey Wolf Optimization Algorithm (GWO)

In 2014, Seyed Ali Mirjalili published the grey wolf optimization algorithm. Grey wolves' hierarchical structure and social behavior while hunting inspired this meta-heuristic algorithm. This technique is population-based, has a straightforward procedure, and can be easily scaled up to large-scale challenges. Grey wolves are top predators, meaning they are at the very top of the food chain. Grey wolves tend to dwell in packs of 5 to 12 individuals. This group's members have a clear social structure and particular responsibilities. Each wolf herd has four degrees to hunt, as shown in Fig. 6.

- Alpha wolves (α), who can be either male or female, are the pack leaders. The herd is dominated by these wolves.
- Beta wolves (β), assist alpha wolves in making decisions and are often chosen to place them.

Fig. 6 Pyramidal structure of the wolf population hierarchy



- Lower than beta wolves, delta wolves (δ) comprise senior wolves, predators, and wolves that care for the young.
- Omega wolves (ω), the position at the bottom of the hierarchy has the fewest privileges over the other members. And anyway, they only consume food and do not take part in decision-making.

There are three basic steps in this algorithm:

1. Observing, tracking, and pursuing the hunt.
2. Approaching the prey, encircling (looping) it, and deceiving it until it stops moving
3. Hunting is under attack. Optimization is done using alpha, beta, and delta wolves.

One wolf is assumed to be the alpha wolf of the algorithm, the wolf beta, and delta are involved, and the other wolves are considered to be their followers. Grey wolves have the ability to estimate the hunting position. To model this process, see the following steps: In the initial search, we have no idea about the hunting position. α , β and δ wolves are thought to have better first-hand knowledge of hunting position (optimal answer point).

The alpha wolf commands the attack when the prey is encircled by wolves and stops moving. The reduction of vector a is used in the modeling of this operation. As a lower, the coefficient vector A similarly reduces since A is a random vector in the interval $[-2a, 2a]$. If $|A| < 1$, the wolf's alpha (and the rest of the wolves) will go after its prey and if $|A| > 1$ the wolf will stay away from the prey (and the rest of the wolves).

All wolves must adjust their locations by alpha, beta, and delta positions by the grey wolf algorithm. Grey wolves circle the victim as they hunt. The following equations give a mathematical model of siege behavior. Coefficient vectors, the prey position vector, and the grey wolf position vector are all included below the current iteration, t .

Algorithm 4 The Grey Wolf Optimization Algorithm

1: procedure START

$$\mathbf{D} = |\mathbf{C}(\mathbf{X}_p)(t) - \mathbf{X}(t)| \quad (13)$$

$$\mathbf{X}(t+1) = \mathbf{X}_p(t) - \mathbf{A} \cdot \mathbf{D} \quad (14)$$

Vectors A and B are calculated as follows:

$$\mathbf{A} = 2\mathbf{a} \cdot \mathbf{r}_1 - \mathbf{a} \quad (15)$$

$$\mathbf{C} = 2\mathbf{r}_2 \quad (16)$$

$$\mathbf{D}_\alpha = |\mathbf{C}_1 \cdot \mathbf{X}_\alpha - \mathbf{X}|,$$

$$\mathbf{D}_\beta = |\mathbf{C}_2 \cdot \mathbf{X}_\beta - \mathbf{X}|, \quad (17)$$

$$\mathbf{D}_\delta = |\mathbf{C}_3 \cdot \mathbf{X}_\delta - \mathbf{X}|.$$

$$\mathbf{D}_\alpha = |\mathbf{C}_1 \cdot \mathbf{X}_\alpha - \mathbf{X}|,$$

$$\mathbf{D}_\beta = |\mathbf{C}_2 \cdot \mathbf{X}_\beta - \mathbf{X}|, \quad (18)$$

$$\mathbf{D}_\delta = |\mathbf{C}_3 \cdot \mathbf{X}_\delta - \mathbf{X}|.$$

$$\mathbf{X}(t+1) = \frac{\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3}{3} \quad (19)$$

2: end procedure

In the relationships mentioned above, r_1 , r_2 are random numbers in the interval $[0,1]$ as well as the variable declines linearly from 2 to 0 over the iterations. Alpha is generally in charge of hunting operations. Hunting is possible for beta and delta wolves on occasion. In the grey wolf hunting behavior mathematical model, we hypothesized that α , β , and δ have superior knowledge of probable prey sites. The first three answers are the most useful, as given in the equations below, the other agents must update their positions based on the parts of the best search agents.

The summary of the grey wolf optimization is written in the pseudocode Table 7, and Fig. 7 depicts the flowchart of the prohibited search algorithm. The order of the GWO algorithm is as follows:

1. All of the solutions are assessed for acceptability, and the top three are chosen as α , β , and δ until the procedure is completed.
2. The top three answers (α , β , and δ wolves) can predict the hunting position using the following equation in each cycle.
3. The rest of the answers are updated by tracking the locations of the α , β , and δ wolves in each iteration.
4. The vectors $\mathbf{a}$ and (therefore) $\mathbf{A}$ are modified after each iteration.
5. The position of the alpha wolf is introduced as the optimal place at the end of the repetitions.

Table 7 Grey wolf optimization algorithm

Start:		
1	Begin	Initializing algorithm patterns
2	Generate	Creating a random population of wolves
3		Assessing the condition of each wolf and calculating its merit
4	Calculate	identifying the best wolf as Alpha wolf
5		Repeat steps 6 to 12 until the stop condition is stopped
6		Repeat steps 7 to 11 for each wolf
7		Calculate three new positions for the wolf
8		Find the position of the wolf according to three positions
9		Calculate the merit of the new wolf
10		If the merit of the new wolf is better than the alpha wolf, consider the new wolf as alpha
11		If the merit of the new wolf is not better than the alpha wolf, compare it with the beta and delta wolves
12		If the merit of the new wolf is not better than the alpha wolf, compare it with the beta and delta wolves
13	Output	Return alpha
End		

2 Investigating the Benefits and Drawbacks of Meta-heuristic Algorithms

A meta-heuristic technique is a framework for solving optimization problems that may be applied to a wide range of difficulties with modest modifications. Finding excellent solutions to problems involving hybrid optimization is greatly facilitated by using meta-heuristic methods. Or, to put it another way, a meta-heuristic algorithm is a creative way to explore the answer space for optimal solutions. Every meta-heuristic method attempts to solve a recognized, difficult optimization issue. Meta-heuristic techniques have the following characteristics:

1. These methods provide competitive solutions and make it possible to avoid the algorithm being trapped in the local minima.
2. These methods are usually presented to solve discrete problems but can also be used for continuous problems.
3. These techniques frequently draw inspiration from biology, animal behavior, and physics ideas.
4. The difficulty of setting and matching a number of parameters is the most typical drawback of these methods.

Each proposed meta-heuristic algorithm has advantages and disadvantages, and therefore, it is significant to examine some of the notable pros and cons of these algorithms (Table 8).

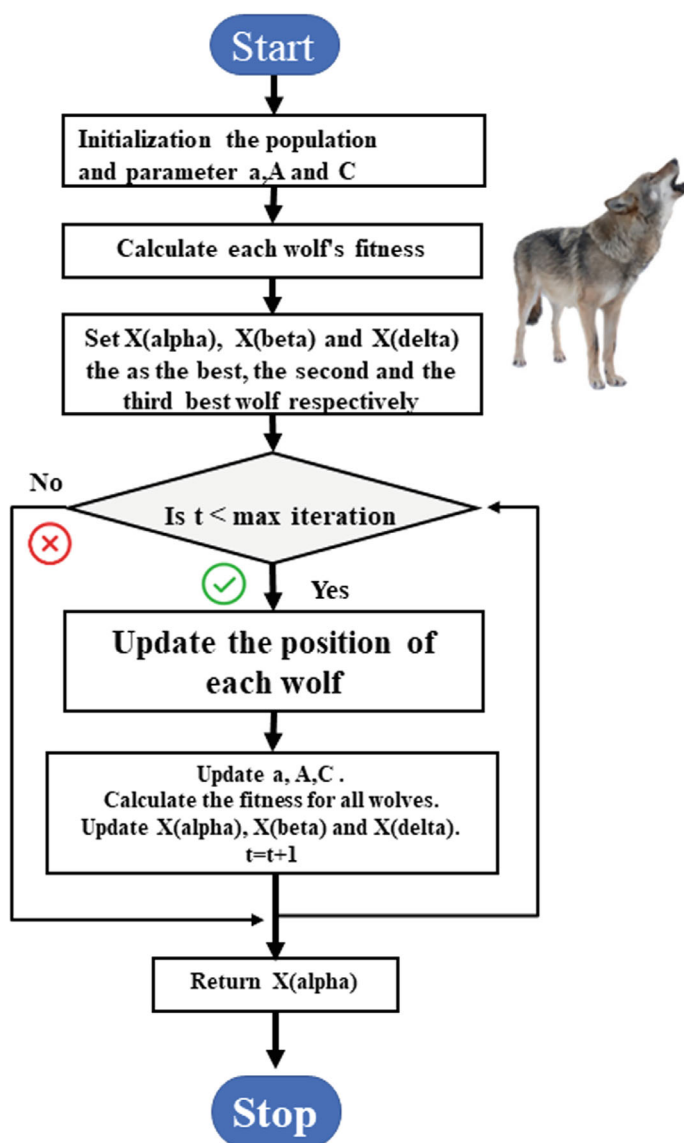


Fig. 7 Flowchart of the GWO algorithm

Table 8 Comparison of different optimization algorithms

Algorithm	Advantages	Disadvantages
SA	1. Very low memory consumption 2. Easy to implement 3. Provides acceptable answers due to the focus on local search 4. Due to the existence of a guided random process (low acceptance probability for non-optimal responses), it has the ability to surpass local optima 5. Converges to the optimal solution through searching for justified solutions to an optimization problem	1. Depends heavily on the initial value of the parameters 2. If you choose the wrong value for the initial temperature parameter, it will most likely get stuck in the local optimum 3. Predicting the appropriate initial value for the problem parameters is impossible without a benchmark
TS	1. Flexible memory structures 2. Long-term memory 3. An attitude based on periodicity 4. Focusing and diversifying 5. Strategic fluctuations 6. Reconnect the route 7. Creating a variety of lists of answers to avoid clutter in the responses and getting stuck in the local optima 8. The various list of answers in the previous step are updated in each step 9. Guide the search to achieve good answers in a complex answer space 10. Better use of search information 11. Variety in search	1. Following an unprofitable path and facing an unfavorable situation without hope of escape 2. Creating rounds and sequences in search 3. Long computation time and generating cheerful answers 4. Lack of proper diversification at high computation times
GA	1. Effective search in large spaces to find the overall optimal point 2. Using the value of the target function to guide the search 3. Utilizing prospective rules as a tool to direct space search activities 4. The generality of the algorithm and the independence of its components 5. A very simple and understandable search approach 6. Performing accurate calculations 7. Encoding continuous functions discretely	1. How to write a fitness operator 2. Algorithm immaturity; the distance between one chromosome and other chromosomes that leads to the problem's optimal local solution 3. High memory consumption 4. Achieving an acceptable balance between exploration and exploitation

(continued)

Table 8 (continued)

Algorithm	Advantages	Disadvantages
PSO	1. Memory benefit 2. Information sharing and collaboration among particles 3. Accelerated convergence 4. Greater adaptability in the face of an ideal local situation 5. Ease of implementation and execution	1. Convergence occurs early 2. Getting carried away by the optimism of the locals. 3. Dwindling population variety
ACO	1. Flexibility 2. Parallel, random, and directed search 3. Solving hybrid optimization problems 4. Optimization in problems that require finding the shortest path	1. Lack of intelligence of elements due to symbolic communication and lack of direct communication with each other 2. Lack of a predetermined plan in the behavior of ants due to the random nature of their behavior with each other 3. Life in a discrete environment; real ants cannot survive apart from colonies 4. Time-consuming
ABC	1. Very fast convergence 2. Collective intelligent data 3. Ability to perform global searches	1. Population variety is dwindling 2. Getting carried away by the optimism of the locals
FA	1. Automatic segmentation 2. The ability to deal with challenges involving multiple quality levels 3. Random control is set to repeat ratio 4. Fast convergence	1. Oscillation algorithm in search space 2. If there are more absorptions, the algorithm's computation time should be extended
BA	1. Simple, flexible, and easy to implement 2. Frequency regulation 3. Auto zoom 4. Parameter control	1. This technique's accuracy suffers at high speeds and may plateau after a few steps
CSO	1. Solve optimization issues 2. The cat swarm clustering algorithm may be regarded as a good enough clustering technique 3. When in tracking mode, they communicate the same details on the best option, in their opinion	1. Finding a workable solution takes a while, impacting the algorithm's efficiency and convergence 2. Requires a high-speed processor to obtain reasonable results
GWO	1. It only needs to initialize two parameters 2. It's simple to generalize to the following n spaces 3. High-speed convergence	1. The GWO is not appropriate for global optimization problems since it is prone to slipping into the trap of local optimization

3 Conclusion

Population-based random search is a type of evolutionary algorithm. A clear advantage is that they are intrinsically parallel, allowing them to look at the search space from diverse angles. A comparable study of sub-spaces leads to locations where the statistical average of the objective function is high, and the possibility of a final optimal point is higher. Because these approaches, unlike one-way methods, explore the entire answer space, there is less possibility of being stuck in a locally optimal spot. Evolutionary algorithms are procedures that are not reliant on a specific path. This distinguishes them from prior methods, allowing them to scan the whole search space effectively.

Furthermore, evolutionary algorithms are not reliant on the function's or derivative's continuity. It is only essential to determine the value of the goal function at different stages during the search process. These algorithms just demand data on the quality of the solutions produced by each set of variables, whereas accurate approaches necessitate a comprehensive grasp of the issue structure and variables.

The theory of creating a model for congestion intelligence introduces a new idea for developing artificial intelligence disciplines that create new possibilities for building hardware and designing computer software. Congestion intelligence provides new ways to control multi-factorial systems. The ability of these systems is due to their teamwork, and this makes them more valuable than when one of these factors works alone. Studying crowded intelligence helps us identify systems with agents having the same goals. The simple design of these systems helps to build better information technology systems that are as robust and appropriate as systems found in nature. Congestion intelligence is the key to managing a world that increasingly includes complex networks of computers.

This research examined a few population-based and single-solution meta-heuristic methods. Each of these algorithms was also given its own set of benefits and drawbacks. The quality and speed with which meta-heuristic algorithms work varies is also important. Depending on the nature of problems, these methods perform differently. In some situations, an algorithm may perform better than others in a given task, while others may perform worse. According to some recent studies, each algorithm can assist consumers in resolving various issues. Each method may answer numerous questions in various fields, including image processing, fuzzy basins, reservoir operation, etc., to name a few.

References

1. Yang XS (2010) Nature-inspired metaheuristic algorithms. Luniver Press
2. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems (1st ed.). Oxford University Press
3. Beheshti Z, Shamsuddin SMH (2013) A review of population-based meta-heuristic algorithms. *Int J Adv Soft Comput Appl* 5(1):1–35

4. Khan AH, Zhang W, Khan ZH, Shi J (2012) Evolutionary computing based modular control design for aircraft with redundant effectors. *Proc Eng* 29:110–117
5. Khan AH, Zhang W, Shi J, Khan ZH (2011) Optimized reconfigurable modular flight control design using swarm intelligence. *Proc Eng* 24:621–628
6. Abdel-Basset M, Abdel-Fatah L, Sangaiah AK (2018) Metaheuristic algorithms: a comprehensive review. In: *Computational intelligence for multimedia big data on the cloud with engineering applications* (pp. 185–231). Elsevier
7. Dorigo M (1992) *Optimization, learning and natural algorithms* (Ph.D. Thesis). Politecnico di Milano
8. Talbi E-G (2009) *Metaheuristics: from design to implementation*. John Wiley & Sons
9. Kirkpatrick S, Gelatt Jr CD, Vecchi MP (1983) Optimization by simulated annealing. *Science*, 220(4598), 671–680. American Association for the Advancement of Science
10. Glover F (1989) Tabu search-part I. *ORSA J Comput* 1(3):190–206 INFORMS
11. Brindle A (1980) Genetic algorithms for function optimization
12. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. *IEEE Comput Intell Mag* 1(4):28–39
13. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization: An overview. *Swarm Intell* 1:33–57
14. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
15. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Appl* 27:1053–1073
16. Yang X-S (2012) Flower pollination algorithm for global optimization. In: *international conference on unconventional computing and natural computation* (pp. 240–249). Springer
17. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
18. Meraihi Y, Benmessaoud Gabis A, Mirjalili S, Ramdane-Cherif A (2021) Grasshopper optimization algorithm: theory, variants, and applications. *IEEE Access* 9:50001–50024
19. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Futur Gener Comput Syst* 97:849–872
20. Liu J, Liu Y, Shi Y, Li J (2020) Solving resource-constrained project scheduling problem via genetic algorithm. *J Comput Civ Eng* 34(2):04019055 American Society of Civil Engineers
21. Ghoseiri K, Morshedsolouk F (2006) ACS-TS: train scheduling using ant colony system. *J Appl Math Decis Sci* 2006. Hindawi
22. Salman FM, Lehmoud AAM, Joda FA (2023) Adaptation of the ant colony algorithm to avoid congestion in wireless mesh networks. *J Cyber Secur Mobility* 785–812
23. Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. *Appl Soft Comput* 8(1):687–697
24. Yang X-S, He X (2013) *Firefly algorithm: recent advances and applications*. Int J Swarm Intell 1(1):36–50 Inderscience Publishers Ltd
25. Sayadi M, Ramezani R, Ghaffari-Nasab N (2010) A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. *Int J Ind Eng Comput* 1(1):1–10
26. dos Santos Coelho L, de Andrade Bernert DL, Mariani VC (2011) A chaotic firefly algorithm applied to reliability-redundancy optimization. In *2011 IEEE Congress of Evolutionary Computation (CEC)* (pp. 517–521). IEEE
27. dos Santos Coelho L, Mariani VC (2012) Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning. *Comput Math Appl* 64(8):2371–2382
28. Dutta R, Ganguli R, Mani V (2011) Exploring isospectral spring-mass systems with firefly algorithm. *Proc Royal Soc A: Math Phys Eng Sci* 467(2135):3222–3240 The Royal Society Publishing
29. Yang X-S (2013) Chaos-enhanced firefly algorithm with automatic parameter tuning. In: *Recent Algorithms and Applications in Swarm Intelligence Research* (pp. 125–136). IGI Global
30. Rampriya B, Mahadevan K, Kannan S (2010) Unit commitment in deregulated power system using Lagrangian firefly algorithm. In: *2010 international conference on communication control and computing technologies* (pp. 389–393). IEEE

31. Fister Jr I, Yang X-S, Fister I, & Brest J (2012) Memetic firefly algorithm for combinatorial optimization. arXiv preprint [arXiv:1204.5165](https://arxiv.org/abs/1204.5165)
32. Abdullah A, Deris S, Anwar S, Arjunan SN (2013) An evolutionary firefly algorithm for the estimation of nonlinear biological model parameters. PLoS ONE 8(3):e56310 Public Library of Science San Francisco, USA
33. Salman FM, Lehmoud AAM, Joda FA (2023) Adaptation of the ant colony algorithm to avoid congestion in wireless mesh networks. J Cyber Sec Mobility 785-812
34. Horng M-H (2012) Vector quantization using the firefly algorithm for image compression. Expert Syst Appl 39(1):1078–1091
35. Zhang Y, Wu L (2012) A novel method for rigid image registration based on firefly algorithm. Int J Res Rev Soft and Intell Comput (IJRRSIC) 2(2):141–146
36. Banati H, Bajaj M (2011) Firefly-based feature selection approach. Int J Comput Sci Issues (IJCSI) 8(4):473
37. Gandomi AH, Yang X-S, Talatahari S, Alavi AH (2013) Firefly algorithm with chaos. Commun Nonlinear Sci Numer Simul 18(1):89–98
38. Swarnkar KK (2012) Economic load dispatch problem with reduced power losses using firefly algorithm. J Adv Comput Sci Technol 1(2):42–56
39. Khan K, Sahai A (2012) A comparison of BA, GA, PSO, BP, and LM for training feed forward neural networks in e-learning context. Int J Intell Syst Appl 4(7):23 Modern Education and Computer Science Press
40. Yang X-S (2011) Bat algorithm for multi-objective optimization. Int J Bio-Inspired Comput 3(5):267–274 Inderscience Publishers
41. Komarasamy G, Wahi A (2012) An optimized K-means clustering technique using bat algorithm. Eur J Sci Res 84(2):263–273
42. Lin J-H, Chou C-W, Yang C-H, Tsai H-L et al (2012) A chaotic Levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems. CIT, Citeseer
43. Gandomi AH, Yang X-S (2014) Chaotic bat algorithm. J Comput Sci 5(2):224–232
44. Nakamura RYM, Pereira LAM, Costa KA, Rodrigues D, Papa JP, Yang X-S (2012) BBA: A binary bat algorithm for feature selection. In 2012 25th SIBGRAPI conference on graphics, patterns and images (pp. 291-297). IEEE
45. Vu-Huu T, Pham-Van S, Pham Q-H, Cuong-Le T (2023) An improved bat algorithm for optimization design of truss structures. In Structures 47:2240–2258
46. González JR, Sancho-Royo A, Pelta DA, Cruz C (2008) Nature-inspired cooperative strategies for optimization. In: encyclopedia of networked and virtual organizations (pp. 982-989). IGI Global
47. Yang X-S, Hossein Gandomi A (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput 29(5):464–483 Emerald Group Publishing Limited
48. Yang X-S, Karamanoglu M, Fong S (2012) Bat algorithm for topology optimization in micro-electronic applications. In: the first international conference on future generation communication technologies (pp. 150-155). IEEE
49. Komarasamy G, Wahi A (2012) An optimized K-means clustering technique using bat algorithm. Eur J Sci Res 84(2):263–273
50. Zhang JW, Wang GG (2012) Image matching using a bat algorithm with mutation. Appl Mech Mater 203:88–93 Trans Tech Publ
51. Reddy PDP, Reddy VU, Manohar TG (2012) Capacitor Placement for loss reduction in radial distribution networks: A two-stage approach. J Electr Eng 12(2):6–6
52. Chu S-C, Tsai P-W, Pan J-S (2006) Cat swarm optimization. In: PRICAI 2006: trends in artificial intelligence: 9th pacific rim international conference on artificial intelligence (pp. 854-858). Springer
53. Tsai P-W, Pan J-S, Chen S-M, Liao B-Y, Hao S-P (2008) Parallel cat swarm optimization. In: 2008 international conference on machine learning and cybernetics (Vol. 6, pp. 3328-3333). IEEE
54. Tsai P-W, Pan J-S, Chen S-M, Liao B-Y (2012) Enhanced parallel cat swarm optimization based on the Taguchi method. Expert Syst Appl 39(7):6309–6319 Elsevier

55. Santosa B, Ningrum MK (2009) Cat swarm optimization for clustering. In: 2009 international conference of soft computing and pattern recognition (pp. 54-59). IEEE
56. Panda G, Pradhan PM, Majhi B (2011) IIR system identification using cat swarm optimization. *Expert Syst Appl* 38(10):12671–12683
57. Wang Z-H, Chang C-C, Li M-C (2012) Optimizing least-significant-bit substitution using cat swarm optimization strategy. *Inf Sci* 192:98–108
58. Bahrami M, Bozorg-Haddad O, Chu X (2018) Application of cat swarm optimization algorithm for optimal reservoir operation. *J Irrig Drain Eng* 144(1):04017057 American Society of Civil Engineers

Chapter 9

Depth Perspective-Aware Multiple Object Tracking



Kha Gia Quach, Pha Nguyen, Chi Nhan Duong, Tien Dai Bui, and Khoa Luu

1 Introduction

Multiple Object Tracking (MOT) has long been considered one of the most critical problems in computer vision. It has recently received much attention due to its potential in many relevant real-world applications. Solving MOT efficiently and robustly is challenging due to missing or inaccurately detected objects, occlusions, and overlapping in crowded scenes. Among those factors, occlusions caused by object movements or detectors' failure have been shown to deteriorate the overall performance of existing MOT solvers greatly [17]. Our work tackles some crucial drawbacks in current MOT solvers, especially occlusions, by proposing a simple yet effective geometric-based approach that leverages perspective information from the scene.

To track multiple targets, e.g., pedestrians on the move, a standard paradigm is track-by-detection, which primarily has two main steps: (1) detecting objects in each frame and (2) assigning ID for each detection by linking them across multiple frames.

K. G. Quach · C. N. Duong
pdActive Inc., Longmont, USA
e-mail: kquach@ieee.org

C. N. Duong
e-mail: dcnhan@ieee.org

P. Nguyen (✉) · K. Luu
Electrical Engineering and Computer Science, University of Arkansas, Fayetteville, USA
e-mail: panguyen@uark.edu

K. Luu
e-mail: khoaluu@uark.edu

T. D. Bui
Computer Science and Software Engineering Department, Concordia University,
Montreal, QC, Canada
e-mail: bui@cse.concordia.ca

Our work focuses on the second stage, i.e., *data association and tracking*. Hence, our algorithm can be used on top of any existing object detectors. Most MOT methods [1, 6, 9, 21, 26, 36, 41, 44, 49, 56] that solely rely on the detection results, i.e., 2D bounding boxes and their associated features, could lead to ID switching between tracked objects, when an object is occluded in the scene, either by the other objects or scene backgrounds. This challenging occlusion problem can be technically handled with the support of perspective or depth information from the scene. Hence, by tracking 3D real-world coordinates of the objects, 3DMOT methods [15, 30, 37, 45] can provide better results. Unfortunately, using such 3D information usually comes with an expensive cost. In particular, a monocular depth estimation method is needed if depth sensors are unavailable. Thus, during training, such methods often require video frames and depth information whose ground truth labels sometimes need to be made available or affordable in practice. Moreover, their 3D-based algorithms are also computationally expensive, which could also affect the real-time performance of many algorithms.

Contributions of this Work: This work introduces Depth Perspective-aware Multiple Object Tracking (DP-MOT) to solve the occlusion problems in MOT. Our contributions are four-fold.

- A simple yet efficient *Subject-Ordered Depth Estimation* (SODE) method is derived from a camera model and geometry-based technique to approximate the depth positions of the detected subjects provided by any detector.
- A new *Active pseudo-3D Kalman filter* method with an *adaptive control variable* provides dynamic and stable tracking of object movements w.r.t depth ordering.
- A new *high-order association* matrix is computed by measuring *first-order* and *second-order* similarity between the detected objects and tracks to improve the association step in handling occlusions (as shown in Fig. 1c).
- The proposed DP-MOT method consistently *achieves the state-of-the-art* (SOTA) results compared to recent MOT approaches [1, 8, 9] on the standard MOT benchmarks.

SODE versus Traditional Calibration Methods: Although the presented SODE is simple and performs unsupervised, it can achieve high real-time accuracy. More important, it is quite different from traditional methods (Fig. 1a) where *manual calibration* is required to transform to a bird's eye view. Indeed, SODE can generally

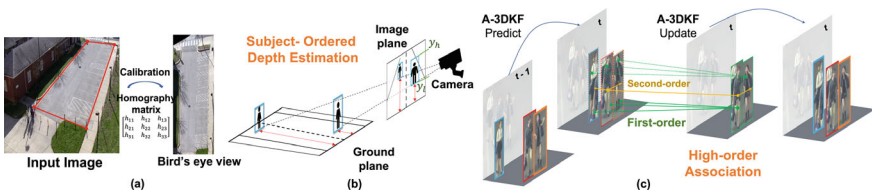


Fig. 1 Illustrations of **a** Traditional calibration method, **b** Subject-Ordered Depth Estimation, and **c** High-order Association

work for any new videos *without calibration* while the traditional methods require a manual calibration in every new setup between two views.

Assumptions in SODE: Given a set of 2D frames in a video captured with a *fixed camera height and rotation*, this work aims to tackle the occlusion problem in real-time using a new simple geometry-based depth perspective-aware approach (see Fig. 1b). Particularly, far apart from prior work, the proposed method *does not require* depth or 3D boxes as ground truth. It can be used as an “add-on” component to support other tracking methods with any detector to improve performance.

Human Detection and Appearance Extraction in SODE: It is noticed that SODE can work with any off-the-shelf deep learning methods for human detection and subject appearance deep feature extraction. The development of human detection and deep feature extraction is not in the scope of this work. In the experiments in this work, deep Feature Embedding Network (FEN) derived from [44] is used to extract appearance features for the detected boxes to obtain enriched objects.

2 Related Work

This section reviews recent works on track-by-detect MOT algorithms, summarized in Table 1. Generally speaking, the algorithms can be categorized as follows:

Tracking by solving the matching problem: The Hungarian algorithm has been used in several methods as a matcher to associate the detection by some kinds of appearance features [31]. Kim et al. [17] proposed to use colors as the appearance features. Some other methods try to improve the accuracy of the matcher by anticipating the locations of objects in the next frame. Ren et al. [36] and Chen et al. [6] introduced reinforcement learning to predict the movement of objects. Furthermore, Maksai et al. [26] applied a bi-directional Long Short-Term Memory (LSTM) to estimate the motion of trajectories. Quach et al. [33] proposed a dynamic graph to transform pre-computed Re-ID features to new context-aware ID features, hence it performs better matching and yields more accurate results. Similarly, Nguyen et al. [29] proposed a novel global association graph model with Link Prediction approach to predict existing tracklets location and link detections with tracklets via cross-attention motion modeling and appearance re-identification. Recently, Li et al. [22] proposed an end-to-end identity association network that consists of a geometry refinement network and an identity verification module to perform data association and reason the mapping between the detections and tracklets.

Tracking as an end-to-end framework: Sun et al. [41] and Xu et al. [49] presented an assignment algorithm that optimizes the tracker in an end-to-end framework. Wang et al. [44] introduced a shared weight model between the detection and the association stage with an improved triplet loss to push away the embedding of different identities in one batch. Bergmann et al. [1] (Tracktor) shows a simple approach by exploiting the bounding box regression of the object detector to guess the position of objects in the next frame in a high-frame-rate video sequence without camera motion. Chu and Ling [9] designed a model having an affinity estimation

and multi-dimensional assignment component for visual object tracking. Zhou et al. [55] employed a straightforward approach, which trained a network to predict the movement offset from the previous frame and then matched it with the nearest tracklet center point. Joint Detection Embedding (JDE) [44] is a fast, simple, yet practical approach that solves detection and embedding representation tasks in a single network structure and a fused loss function. Li and Gao et al. [21] introduced a graph network to extract appearance and motion similarity separately. Sun et al. [42] constructed three networks to compute three matrices representing the object's motion, type, and visibility for every matching step. Peng et al. [32] passed a couple of frames to a single network, which can generate the target's bounding box pair between two frames. Therefore, no association step is needed. Xu et al. [50] (Deep-MOT) formulated MOT metrics as a loss function to construct a complete trainable tracking procedure. Yin et al. [51] trained a Siamese neural network for the joint task of single object tracking and multiple object association simultaneously. Chan et al. [5] proposed an end-to-end network for detecting and tracking multiple objects simultaneously. Meinhardt et al. [27] introduced a new tracking-by-attention mechanism with data association via attention between the frames. Recently, Yu et al. [52] proposed using a visual-language model (VLM) to generate pseudo-textual descriptions for each tracking target, which are domain invariant and can enhance object matching across frames. They also introduce a Trackbook, a collection of pre-defined textual prompts that capture common object attributes in different domains.

Tracking with 3D information. Some tracking methods use additional data from multiple cameras or special GPS or IMU sensors to solve the occlusion problem. Hu et al. [15] relied on ego-motion provided by a sensor to predict a vehicle's future location and used a weighted bipartite matching algorithm on affinities between tracks and detections. Weng et al. [45] use LiDAR point cloud to obtain oriented 3D boxes from a 3D object detector, then use a primary 3D Kalman filter to estimate objects' state and the Hungarian algorithm to associate tracks with new detection. Scheidegger et al. [37] exploited the 3D information of surrounding pedestrians to achieve highly accurate predictions. Weng et al. [46] (GNN3DMOT) took full advantage of 2D and 3D motion and appearance features, then constructed a graph neural network to compute the affinity matrix. Rajasegaran et al. [34] lifted people's 3D information to represent the 3D pose of the person, their location in the 3D space, and the 3D appearance then computed the similarity between predicted states and observations in a probabilistic manner.

However, these physics-based approaches are also prone to failure with sensor noises and uncertainty in detecting occluded people, which often occur in real-world scenarios.

3 The Proposed DP-MOT Approach

This section will detail three main components in the proposed framework as shown in Fig. 2, including the Subject-Ordered Depth Estimation to estimate depth ordering of

Table 1 Comparison of the properties between our online MOT approach and others

Methods	Domain	Depth aware	3D label Req.	Association	Dyn. Mov. Upd.	RT
Tracktor [1]	2D	✗	N	FO	✗	✗
CenterTrack [55]		✗	N	FO	✗	✓
JDE [44]		✗	N	FO	✗	✓
DeepMOT [49]		✗	N	FO	✗	✗
FAMNet [9]		✗	N	PW	✗	✗
STRN [48]		✗	N	Graph	✗	—
MOTDT [7]		✗	N	FO	✗	✗
AB3DMOT [45]	3D	✓	Y	FO	✗	✓
3DT [15]		✓	Y	FO	✓(LSTM)	✗
3D-PMBM [37]		✓	Y	—	✗	✓
Ours	2D	✓	N	High-order (FO+SO)	✓(A-3DKF)	✓

Kalman Filter (KF), Requirement (Req.), Dynamic (Dyn.), Movement (Mov.), Update (Upd.), first-order (FO) and second-order (SO), Real-time (RT), Yes (Y), No (N). “—” denotes an unknown property

bounding boxes, the active pseudo-3D Kalman Filter component for tracking object positions, and the High-order Association for computing first-order and second-order perspective-aware association cost matrices.

3.1 Problem Definition

Assume that the video sequence contains N frames, and there are K objects that need to be tracked, we define the trajectory of the k -th object as $T_k = \{\mathbf{b}_k^0, \mathbf{b}_k^1, \dots, \mathbf{b}_k^t, \dots, \mathbf{b}_k^N\}$, and $\mathbf{b}_k^t = (x, y, w, h)$, where t denotes the frame number, the pair (x, y) specifies the top-left location of the bounding box, while w and h , respectively, are the width and height of the detected object. Let $\mathcal{B}^t = \{\mathbf{b}_1^t, \mathbf{b}_2^t, \dots, \mathbf{b}_k^t, \dots, \mathbf{b}_{N_O}^t\}$ denotes a set of N_O detected objects at the t -th frame. The pipeline in this work consists of three main steps:

Firstly, the depth estimation SODE module (Sect. 3.2) provides the estimated depth $\hat{z}$ based on 2D geometry information of the boxes. Then, FEN [44] is used to extract appearance features $\mathbf{e} \in \mathbb{R}^{512 \times 1}$ to obtain objects $\hat{\mathbf{b}}_k^t = \{x, y, \hat{z}, w, h, \mathbf{e}\}$. Secondly, the active pseudo-3D Kalman filter (Sect. 3.3) will help to keep track trajectories T_k up-to-date and predict new 3D-location of the tracked objects in the next frame. Finally, we compute a cost matrix $\mathbf{C} \in \mathbb{R}^{N_T \times N_O}$ (N_T and N_O are the number of tracked objects in frame $t - 1$ and the number of detected objects in frame t , respectively), by considering appearance similarity and 3D-spatial distance between each pair of predicted track T_k from frame $t - 1$ and detected object $\hat{\mathbf{b}}_i^t$ in frame t . The computing of this matrix $\mathbf{C}$ details in Sect. 3.4. Based on this cost matrix

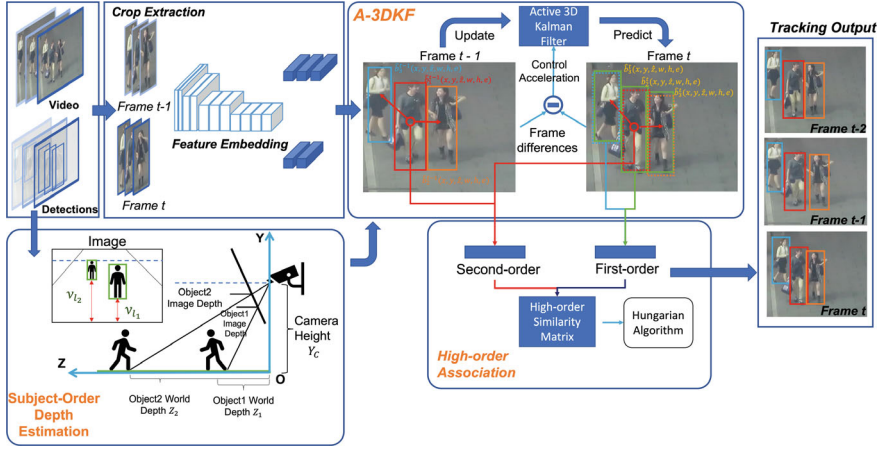


Fig. 2 Our proposed framework with three main components: the subject-order depth estimation, the active pseudo-3D Kalman filter, and the high-order association. The subject-order depth estimation approximates the ordering of objects in the scene according to their distance from the camera using only the provided bounding box from the detector. In the active pseudo-3D Kalman filter, solid color boxes, i.e., blue, red, and orange, represent track IDs, while dash color boxes represent the predicted location of tracked objects in the current frame t . Green color boxes are detected objects in the current frame t

C, the association problem is solved using the Hungarian algorithm to correctly assign detected objects to their corresponding existing tracks T_k or create new tracks for new objects. We also remove lost tracks when objects exit the scene as such tracks cannot match with any detected objects for several frames.

3.2 Subject-Ordered Depth Estimation

Given the shortcomings of existing learning-based monocular depth estimation approaches, we propose a novel mechanism to *mathematically* approximate depth ordering, which can be used on top of any existing MOT trackers. Motivated by the standard camera setup in most tracking applications such as video surveillance (see Fig. 1b), we revisit a classical geometry-based technique to estimate the depth ordering of the detected bounding boxes directly.

To utilize the setup as shown in Fig. 2, we choose the world coordinates of 3D world points (x, y, z) defined by X , Y , Z axes and origin O such that the ground plane (the plane where objects reside) coincides with the plane $y = 0$, i.e., an object's height is represented by its y coordinate. In contrast, z represents its depth, i.e., distance from the origin (see Fig. 2). This simple setup for the world coordinate allows the camera pose to be represented by only camera height h and pitch angle θ_x to form a camera coordinate frame with X_c , Y_c , Z_c axes, and origin C .

Let $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ and $\mathbf{P} = [\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{3 \times 4}$ denote the camera intrinsic and extrinsic matrices, respectively. Given the setup in Fig. 1b, the rotation matrix $\mathbf{R}$ contains components along the $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ axes as we allow small errors for the roll θ_Z and yaw θ_Y angles, while the translation vector $\mathbf{t} = [0, \mathbf{Y}_c, 0]$.

The projection of a point $(x, y, z, 1)$ in the homogeneous coordinates to the image coordinate $(u, v, 1)$ then reads:

$$\begin{aligned} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \frac{1}{\bar{z}} \mathbf{K} \mathbf{P} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \\ &= \frac{1}{\bar{z}} \begin{bmatrix} f & 0 & u_c \\ 0 & f & v_c \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} r_1 & r_2 & r_3 & 0 \\ r_4 & r_5 & r_6 & \mathbf{Y}_c \\ r_7 & r_8 & r_9 & 0 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \end{aligned} \quad (1)$$

where the rotation matrix is defined as

$$\begin{aligned} \mathbf{R} &= \mathbf{R}_z(\theta_Z) \mathbf{R}_y(\theta_Y) \mathbf{R}_x(\theta_X) = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta_Z & -\sin \theta_Z & 0 \\ \sin \theta_Z & \cos \theta_Z & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos \theta_Y & 0 & \sin \theta_Y \\ 0 & 1 & 0 \\ \sin \theta_Y & 0 & \cos \theta_Y \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_X & -\sin \theta_X \\ 0 & \sin \theta_X & \cos \theta_X \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta_Z \cos \theta_Y & \cos \theta_Z \sin \theta_Y \sin \theta_X - \sin \theta_Z \cos \theta_X & \cos \theta_Z \sin \theta_Y \cos \theta_X + \sin \theta_Z \sin \theta_X \\ \sin \theta_Z \cos \theta_Y & \sin \theta_Z \sin \theta_Y \sin \theta_X + \cos \theta_Z \cos \theta_X & \sin \theta_Z \sin \theta_Y \cos \theta_X - \cos \theta_Z \sin \theta_X \\ -\sin \theta_Y & \cos \theta_Y \sin \theta_X & \cos \theta_Y \cos \theta_X \end{bmatrix} \end{aligned} \quad (2)$$

Expanding the terms in (1) for the v coordinate and using some basic algebraic manipulations, we obtain the relation between the object's position y, z in the world coordinate and the image coordinate v as follows (note that we write y as a function of v):

$$y(v) = \frac{z((v - v_c)r_9 - fr_6) - f\mathbf{Y}_c + x((v - v_c)r_7 - fr_4)}{(v_c - v)r_8 + fr_5}. \quad (3)$$

Recall that most tracking applications; it is expected to track objects moving on the grounds. The grounds can be either flat, slope, or bumpy; however, to simplify the problem, we use a plane to approximate the grounds. Thus, the object's translation in the $\mathbf{Y}$ -axis is approximately equal to 0. Under this approximation, the bottom of the detected bounding boxes corresponds to the coordinates $y = 0$ in the real world. Thus, we have $y(v_l) = 0$, where v_l represents the v coordinate at the bottom of a detected bounding box. The estimated object depth can then be derived as in Eq. (4).

$$\bar{z} = \frac{f\mathbf{Y}_c + x(fr_4 + (v_c - v_l)r_7)}{(v_l - v_c)r_9 - fr_6}. \quad (4)$$

We choose the world coordinate of 3D world points such that θ_Y and θ_Z are equal to 0. Thus, r_4 and r_7 are equal to 0, and we can omit the second term in Eq. 4. It shows that the approximated depth $\bar{z}$ of a 3D point (x, y, z) and its corresponding image coordinate v_l appear co-linear as we consider other values to be constant and relatively small. As a result, we can directly use v_l , i.e., the bottom line of the objects' bounding boxes, to estimate the depth ordering between objects. We aim to relax the need to estimate absolute depth value z in meters. In other words, our estimated depth order $\hat{z}$ of an object box $\mathbf{b}_k^t$ is defined as a quantized depth value z as $\hat{z} = q(\frac{1}{v_l})$. Setting up the world coordinate to have zero yaw and roll angles in $\mathbf{P}$ should be relatively easy via standard camera calibration in practice, compared to the huge amount of labeled training data required by other learning-based methods. In the case of moving cameras, e.g., handheld cameras or cameras mounted on vehicles, the world coordinate is replaced by an ego-coordinate system with the origin $\mathbf{O}$ at the center bottom of the moving car/person holding the camera at the current time t . As a result, we can rewrite extrinsic matrix $\mathbf{P}^t$ as

$$\mathbf{P}^t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \mathbf{Y}_c \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (5)$$

Since we only consider objects' locations at time t and $t + 1$ while performing online tracking, we can approximate those coordinates with the same extrinsic matrix, i.e., $\mathbf{P}^t \approx \mathbf{P}^{t+1}$. Then, the estimated object depth can be rewritten as in Eq. (6).

$$\bar{z} = \frac{f\mathbf{Y}_c}{(v_l - v_c)}. \quad (6)$$

Thus, we can also obtain estimated depth order $\hat{z}$ of an object box $\mathbf{b}_k^t$ as $\hat{z} = q(\frac{1}{v_l})$. Therefore, our depth ordering estimation is applicable in the case of dynamic cameras. We conduct some experiments using this depth ordering computation. Surprisingly, it works well in different scenarios, including static and moving cameras, on sequences in both MOT challenge [20, 28] and nuScene dataset [3].

In the next Sect. 3.3, we will show how the estimated depth ordering information can be integrated into our active pseudo-3D Kalman filter to improve tracking performance significantly. Furthermore, the pseudo-3D Kalman filter can help correct missed objects caused by human jumping or inaccurately detected boxes.

3.3 Active Pseudo-3D Kalman Filter

Based on the refined objects with the estimated depth $\hat{z}$, we propose an extension of Kalman filter (KF), named active pseudo-3D Kalman Filter (A-p3DKF), to significantly improve the tracking accuracy. In particular, each object in the scene is

associated with a KF tracker, $\mathbf{K}_t[i]$, $i = 1 \dots N_T$, that actively maintains and updates its state. In the following, we describe the configurations of a particular tracker.

Given depth information for each detected bounding box, we formulate states for each tracker as $\mathbf{s}_{t,t}[i] = [c_x, c_y, \hat{z}, s, r, c'_x, c'_y, \hat{z}', s']$, where c_x, c_y specify the center of the box, $\hat{z}$ is the *estimated* depth of the object computed based on (4), and s, r are scale and ratio parameters, respectively. The remaining are the derivatives of box location and scale, i.e., the velocity of those measures. We use double subscript $\mathbf{s}_{t_1, t_2}$ to denote that $\mathbf{s}$ is predicted at time step t_1 and updated at t_2 , respectively. Unlike traditional pseudo-3D Kalman filters, we proposed to add a control variable, i.e., acceleration, defined as $\hat{\mathbf{u}}_{t,t} = [c''_x, c''_y, \hat{z}'']$ and a control matrix ($\mathbf{G}$). Given an initial state $\mathbf{s}_{1,0}$ and the co-variance (uncertainty) matrix $\mathbf{A}_{1,0}$, we can first predict the next state $\mathbf{s}_{t+1,t}$ and uncertainty $\mathbf{A}_{t+1,t}$ at time $t + 1$ taking into account sudden changes or movements as follows,

$$\mathbf{s}_{t+1,t} = \mathbf{F}\mathbf{s}_{t,t} + \mathbf{G}\hat{\mathbf{u}}_{t,t}, \mathbf{A}_{t+1,t} = \mathbf{F}\mathbf{A}_{t,t}\mathbf{F}^T + \mathbf{Q}, \quad (7)$$

where $\mathbf{F}$ and $\mathbf{G}$ are the state transition and the control matrix, and $\mathbf{Q}$ is the process noise uncertainty.

$$\mathbf{G} = \begin{bmatrix} \text{diag}_{3 \times 3}(0.5\Delta t^2) \\ \text{diag}_{3 \times 3}(\Delta t) \end{bmatrix}, \mathbf{Q} = \mathbf{G}\sigma^2\mathbf{G}^T, \Delta t = 1. \quad (8)$$

We can compute the control variable $\hat{\mathbf{u}}_{t,t}$ from the difference between three adjacent frames based on the optimized optical flow [18] instead of pre-defined fixed acceleration values, e.g., set as zero for constant velocity. To reduce the computational cost of optical flow, we only compute flow between two small sets of key points, i.e., 16 from our experiments, limited within an extended region of detected objects. This parameter $\hat{\mathbf{u}}_{t,t}$ can be estimated from the scene dynamics in the following steps.

- *Step 1.* Compute the local optical flow of each tracker around the location at (c_x, c_y) in three consecutive frames as $\delta^{t-1,t}$ and $\delta^{t,t+1}$.
- *Step 2.* Compute control variable as $\hat{\mathbf{u}}_{t,t} = [\delta_x^{t,t+1} - \delta_x^{t-1,t}, \delta_y^{t,t+1} - \delta_y^{t-1,t}, \gamma(\delta_y^{t,t+1} - \delta_y^{t-1,t})]$, where γ is the ratio obtained from Eq. (3).

Finally, we can correct the prediction of A-3DKF with a new predicted bounding box (measurement) at time t in the following steps.

- *Step 1.* Compute the Kalman Gain $\mathbf{K}_t = \mathbf{A}_{t,t-1}\mathbf{H}^T(\mathbf{H}\mathbf{A}_{t,t-1}\mathbf{H}^T + \mathbf{R}_t)^{-1}$, where $\mathbf{H}$ is the observation matrix and $\mathbf{R}_t$ is the measurement uncertainty.
- *Step 2.* Update state with measurement $\mathbf{s}_{t,t} = \mathbf{s}_{t,t-1} + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}\mathbf{s}_{t,t-1})$, where $\mathbf{z}_t$ is the detector prediction.
- *Step 3.* Update covariance matrix $\mathbf{A}_{t,t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{A}_{t,t-1}$, where $\mathbf{I}$ is the identity matrix.
- *Step 4.* Predict the new state $\mathbf{s}_{t+1,t} = \mathbf{F}\mathbf{s}_{t,t} + \mathbf{G}\hat{\mathbf{u}}_{t,t}$,

3.4 High-Order Association Matrix Computing

Given the enriched information obtained from our framework, i.e., the relative object depths, the velocity, and acceleration vectors predicted from the A-3DKF, this section describes a novel usage of such information to improve the association task. Unlike traditional similarity/cost matrix computation, our proposed cost matrix is computed based on two crucial pieces of information: *first-order* and *second-order* relationships between tracks and detections. Our cost matrix $\mathbf{C} \in \mathbb{R}^{N_{\mathcal{T}} \times N_{\mathcal{O}}}$ measures the similarity between $N_{\mathcal{T}}$ tracked objects predicted from A-3DKF from previous frame $t - 1$ and $N_{\mathcal{O}}$ detected objects at current frame t .

3.4.1 Matrix Computing Based on First-Order Relationship

The first-order cost matrix $\mathbf{C}_F \in \mathbb{R}^{N_{\mathcal{T}} \times N_{\mathcal{O}}}$ is directly computed based on appearance features and depth-aware intersection over Union (d-IoU) distances between tracks and detections. These distances are formulated as two matrices, $\mathbf{C}_a \in \mathbb{R}^{N_{\mathcal{T}} \times N_{\mathcal{O}}}$ and $\mathbf{C}_{d-IoU} \in \mathbb{R}^{N_{\mathcal{T}} \times N_{\mathcal{O}}}$. In the following, we describe the design of $\mathbf{C}_a$ and $\mathbf{C}_{d-IoU}$.

Appearance Matrix. Our appearance matrix follows closely with recent works on Multiple Object Tracking, where the score of associating the i -th track to the j -th detected object is assigned to the matrix element $\mathbf{C}_a(i, j)$ and is computed by the cosine similarity between two appearance vectors, i.e.,

$$\mathbf{C}_a(i, j) = \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i\| \cdot \|\mathbf{e}_j\|} \quad (9)$$

Note that here $\mathbf{e} \in \mathbb{R}^{512}$ is the appearance vector obtained from the FEN introduced in Sect. 3.1.

Depth-aware Intersection over Union (IoU) Matrix. Since we only estimate the depth ordering of the detected and tracked objects, we extend the traditional 2D overlapping cost matrix to IoU considered within an allowable range of distance in-depth, named d-IoU. The d-IoU matrix is defined as,

$$\mathbf{C}_{d-IoU}(i, j) = \frac{\mathbf{b}_i \cap \mathbf{b}_j}{\mathbf{b}_i \cup \mathbf{b}_j} \otimes (\|\hat{z}_i - \hat{z}_j\| < \tau_Z) \quad (10)$$

where τ_Z , which is experimentally set to 3, is the ranging threshold for depth steps to consider objects having proximity or overlapping.

3.4.2 Matrix Computing Based on Second-Order Relationship

The second-order cost matrix $\mathbf{C}_P \in \mathbb{R}^{N_{\mathcal{T}} \times N_{\mathcal{O}}}$ is computed based on appearance features and spatial features of each object/track concerning other objects/tracks within

a frame. This section describes our novel method to utilize the output of SODE and A-3DKF to compute *second-order spatial and appearance distances*.

Let us denote by $\mathbf{x}_k = \{c_x, c_y, \hat{z}\} \in \mathbb{R}^3$ and $\mathbf{v}_k = \{c'_x, c'_y, \hat{z}'\} \in \mathbb{R}^3$ the location and the velocity vector of the k -th tracked object ($k = 1 \dots N_T$) at time $t - 1$, respectively. Given $\mathbf{x}_k$ and $\mathbf{v}_k$ from our active pseudo-3D Kalman filters, we can obtain the predicted location $\tilde{\mathbf{x}}_k$ for the k -th tracked object. After obtaining the predicted 3D positions of the tracked objects, we denote by $\tilde{d}_{ij}$ the distance between $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$, i.e., $\tilde{d}_{ij} = \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|$. We also define a vector $\tilde{\mathbf{D}}_k = s([\tilde{d}_{k1}, \dots, \tilde{d}_{k(k-1)}, d_{k(k+1)}, \dots, \tilde{d}_{kN_T}]^T)$ where the function $s(\cdot)$ sorts all the provided elements in ascending order. Intuitively, the vector $\tilde{\mathbf{D}}_k$ associated with the k -th tracked object contains the sorted distances between itself to the remaining tracked objects.

Let $\{\mathbf{y}_l\}_{l=1}^{N_O}$ denote the bounding boxes and estimated depth values returned by the detector and SODE at time step t , where $\mathbf{y}_l = \{c_x, c_y, \hat{z}\} \in \mathbb{R}^3$. Similarly, let $d_{pq} = \|\mathbf{y}_p - \mathbf{y}_q\|$ be the distance between the predicted bounding boxes p and q , and $\mathbf{D}_l = s([d_{l1}, \dots, d_{l(l-1)}, d_{l(l+1)}, \dots, d_{lN_O}]^T)$ be the sorted distances between the l -th detected bounding box to the remaining boxes. We assume that the number of objects and their assigned IDs remain unchanged from frame $t - 1$ to frame t and that the detected bounding boxes in frame t match their predicted locations. Under such assumption, if tracked object k matches the bounding box l , it is expected that $\|\tilde{\mathbf{D}}_k - \mathbf{D}_l\| = 0$. In this case, $\tilde{\mathbf{D}}_k$ and $\mathbf{D}_l$ act as useful feature vectors for the association task.

However, in practice, it usually happens that $N_T \neq N_O$ since new objects may appear in frame t or some existing objects are no longer captured by the camera. In such scenarios, algorithms such as graph matching have proven valuable choices for robust matching. However, employing such algorithms is computationally expensive, which may degrade the real-time performance of our tracking pipeline. Therefore, we propose an approximation scheme to use the distance information while keeping the existing linear assignment framework. In particular, given two vectors $\tilde{\mathbf{D}} \in \mathbb{R}^{N_T}$ and $\mathbf{D} \in \mathbb{R}^{N_O}$ from a tracked object and a detected box, without loss of generality, let first consider $N_T \leq N_O$. We propose a new second-order function $f(\tilde{\mathbf{D}}, \mathbf{D}) : \mathbb{R}^{N_T} \times \mathbb{R}^{N_O} \mapsto \mathbb{R}$ to measure the distance between $\tilde{\mathbf{D}}$ and $\mathbf{D}$ as follows,

$$f(\tilde{\mathbf{D}}, \mathbf{D}) = \sum_{i=1}^{N_T} \|\tilde{\mathbf{D}}_{[i]} - \mathbf{D}_{[j-1+i]}\|, \quad (11)$$

$$\text{where } j = \arg \min_{j=1 \dots N_O} \|\tilde{\mathbf{D}}_{[1]} - \mathbf{D}_{[j]}\|$$

where the notation $\mathbf{D}_{[i]}$ denotes the i -th element of the vector $\mathbf{D}$. Note that the above problem can be solved efficiently, since $\tilde{\mathbf{D}}$ and $\mathbf{D}$ are sorted vectors, by first searching for j using binary search in $O(\log N_O)$, and then compute the value of f in $O(N_T)$. In the case of $N_T > N_O$, we will ignore all the other i -th elements where $i > N_O$. In summary, our *second-order spatial distance matrix* $\mathbf{C}_{P_d}$ is defined as $\mathbf{C}_{P_d}(i, j) = f(\tilde{\mathbf{D}}_i, \mathbf{D}_j)$. This way, we can incorporate a second-order spatial relationship between tracks and detections within a current frame and a constrained spatial relationship

between weighted bipartite graph matching between current and previous frames. Similarly, we can compute our *second-order appearance distance matrix* $\mathbf{C}_{P_a}$ as $\mathbf{C}_{P_a}(i, j) = f(\tilde{\mathbf{A}}_i, \mathbf{A}_j)$, where $\tilde{\mathbf{A}}_i$ and $\mathbf{A}_j$ are the sorted cosine distances between the appearance vector of tracked objects and detected bounding boxes to their remaining ones in frame t .

In summary, our cost matrix $\mathbf{C} \in \mathbb{R}^{N_T \times N_O}$ is the combination of the following

$$\mathbf{C} = \mathbf{C}_F + \mathbf{C}_P = \alpha (\mathbf{C}_a + \mathbf{C}_{d-IoU}) + \beta (\mathbf{C}_{P_a} + \mathbf{C}_{P_d}) \quad (12)$$

The values of hyper-parameters α and β , which control the weights for each component, are manually chosen through experiments in the next section.

4 Experiment Results

4.1 Databases and Evaluation Metrics

Datasets. To evaluate the effectiveness of the proposed DP-MOT, it experiments on MOT Challenge [20, 28] with the pedestrian tracking dataset. This dataset has two different versions, including MOT16 and MOT17. Each includes several real-world video sequences recorded on the streets with numerous occlusions and the challenging motion of people crossing and hiding each other. Videos have diverse views, illuminated conditions, and video frame rates. MOT16 and MOT17 share the same videos but have different annotation sets. nuScenes [3] is one of the large-scale datasets for Autonomous Driving with 3D object annotations. It contains 1,000 videos of 20-second shots in a setup of 6 cameras, i.e., three front and three rear ones, with a total of 1.4M images. It also provides 1.4M manually annotated 3D bounding boxes of 23 object classes based on LiDAR data. This dataset is an official split of 700, 150, and 150 videos for training, validation, and testing, respectively.

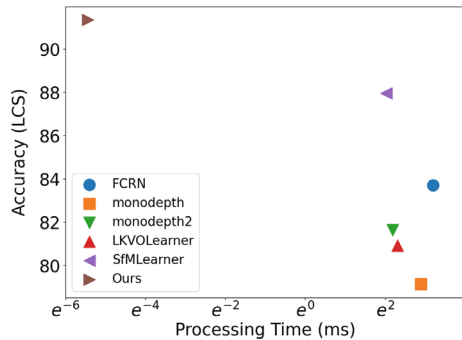
Metrics. MOT methods are commonly evaluated in multiple criteria. Multiple Object Tracking Accuracy (MOTA) and ID F1 score (IDF1) are the most critical MOT metrics, together with Mostly Tracked objects (MT), Mostly Loss objects (ML), and ID switches (ID Sw.).

4.2 Ablation Studies

In this section, we perform several ablation studies to show the role of each proposed module.

Comparisons of Subject Ordered Depth Estimation and SOTA Depth Estimation. This experiment aims to study the effectiveness of the proposed SODE method in estimating the orders of objects in the scene. Since we only need to approximate

Fig. 3 Comparisons of the depth estimation approaches, including FCRN [19], Monodepth [12], Monodepth2 [13], LKVOLEARNER [43], SfMLearner [54] and our proposed SODE on People RGB-D dataset. The higher and more left one is better. **Best viewed in color**



relative depth information, i.e., depth ordering between detected objects w.r.t. the camera, in this experiment, we propose to evaluate the accuracy of our estimator by comparing the estimated relative depth to the ground-truth depths provided by the RGB-D People dataset [24, 39].

In particular, for each frame t , based on the ground-truth depth information, we sort the bounding boxes by their depths in ascending order to obtain a set of indexes d_{truth} . With the same set of bounding boxes, we use our proposed method to estimate their relative depths and sort the bounding boxes according to their estimated depths to obtain the vector $d_{\text{estimated}}$. Then, the accuracy of depth orders is validated via the Longest Common Subsequence (LCS) metric between $d_{\text{estimated}}$ and d_{truth} as $\text{Accuracy} = \frac{\text{LCS}(d_{\text{truth}}, d_{\text{estimated}}) \times 100}{|d_{\text{truth}}|}$. Figure 3 illustrates the achieved performance in terms of speed and accuracy of our SODE compared to other depth estimation methods on 2,000 randomly selected database frames. Since all methods also used the same detected bounding boxes in their estimation process, we excluded the detection time and only reported the depth estimation step to obtain the subject-ordered list. As can be seen, our SODE method outperforms other approaches in computing the subject order w.r.t. depth property while maintaining real-time processing speed. The processing time of the SODE step is fast due to its simplicity in computation. The SODE method achieves relatively good accuracy thanks to re-using the learned “perspective” knowledge from the detector. Figure 4 presents the relative visualization of depth estimation methods. Notice that the outputs of other methods are the depth maps with exact depth values, so the depth value for each pedestrian is computed as the average depths of all pixels inside the bounding box of that subject and projected on the XZ plane. We found that such a way of getting depth orders could negatively impact many existing depth estimation approaches in our experiments due to overlapping with background and other foreground objects. Our SODE is less likely to be confused by other background/foreground information as the only source of information is the “perspective” knowledge from the pre-trained pedestrian detector.

To further evaluate our proposed SODE, we conduct another experiment on the MOT-17 validation set using different estimated depth values from other depth estimation approaches in our MOT framework. In such a way, we will demonstrate that

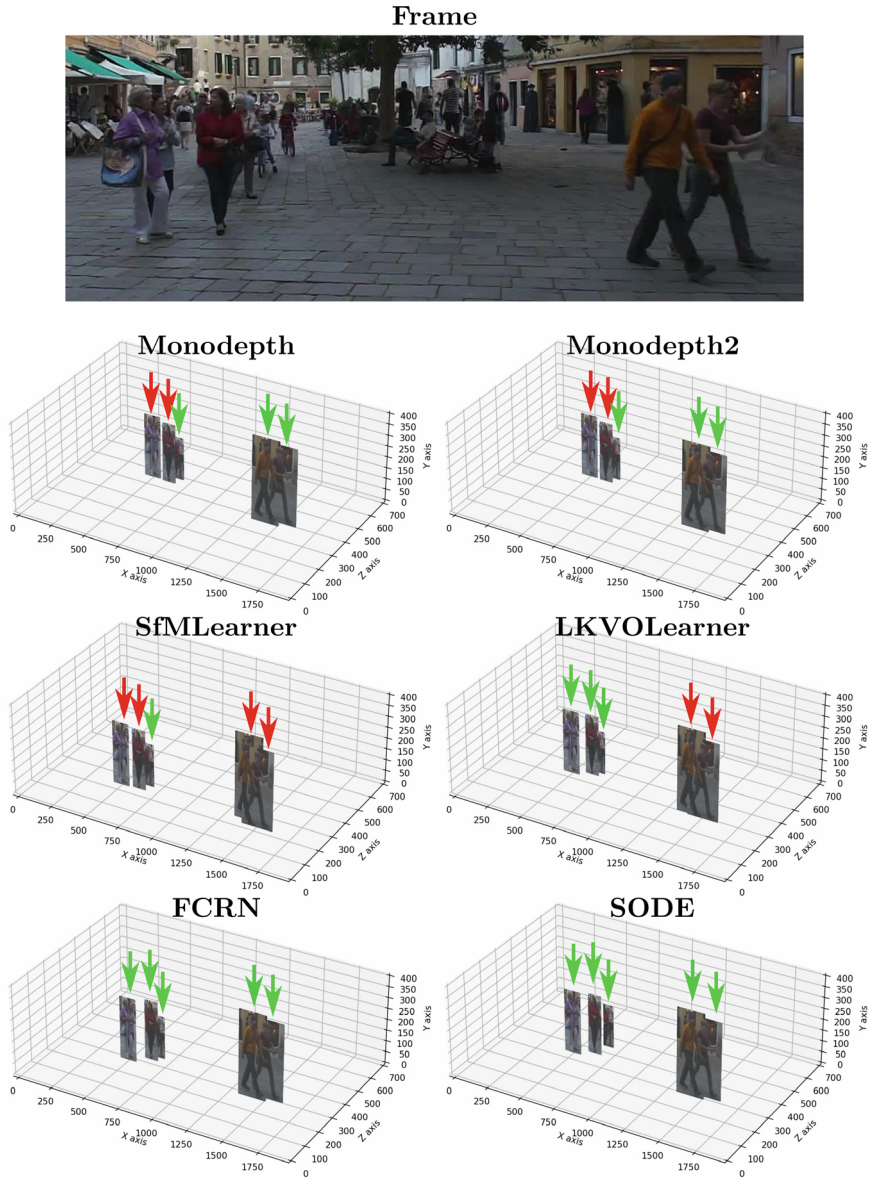


Fig. 4 Visualization of the depth estimation approaches, including FCRN [19], Monodepth [12], Monodepth2 [13], LKVO Learner [43], SfMLearner [54] and our proposed SODE on MOT17 dataset. Green arrows indicate right positional correlations and red arrows indicate wrong positional correlations on the XZ plane. **Best viewed in color and zoom in**

Table 2 Evaluation on MOT-17 (SDP detectors) validation set with provided boxes from SDP detector for different depth estimation approaches regarding MOT metrics and depth ordering accuracy

Depth models	MOTA $\uparrow$	IDF1 $\uparrow$	MT $\uparrow$	ML $\downarrow$	ID Sw. $\downarrow$	LCS Acc. $\uparrow$
Monodepth [12]	63.9	68.2	142	152	975	79
Monodepth2 [13]	64.1	70.2	145	154	862	81.5
SfMLearner [54]	64.7	70.7	151	155	537	88.2
LKVOLEARNER [43]	64.9	72.4	153	154	507	81.3
FCRN [19]	65.5	72.2	157	145	330	84.1
Our SODE	66.6	72.8	179	142	399	91.8

$\downarrow$ means the smaller the better; $\uparrow$ means the larger the better

using the proposed depth estimation method, we can observe similar or better performance than directly using the absolute depth, i.e., information in meters. The results are shown in Table 2. As shown in the last column of Table 2, the depth orders provided by other depth estimation methods are less accurate than our SODE method. Thus, it reduces the performance of our MOT framework since we are using the same tracking approach described in Sects. 3.3 and 3.4. Such a gap in the accuracy of depth orders is because of the depth orders estimated using the average depths of all pixels inside the bounding box of an object. The depth information within the bounding box can be redundant and noisy.

Improvements with Active 3D Kalman Filter and High-order Association.

Firstly, we target the use of object's 3D locations from our SODE in the motion model, i.e., the use of 3D information and the use of the acceleration in Kalman Filter as Active Kalman Filter and the use of high-order association approach, i.e., considering the second-order relationship between objects within a frame in addition to the first-order similarity.

Table 3 shows the evaluation results on the MOT-16 and MOT-17 validation set using the provided bounding box annotations. By using either the proposed A-3DKF motion model or the high-order association approach, we can achieve about 2–3% improvements on all MOT metrics, especially for MT and ML, which shows that it keeps tracking the pedestrian precisely.

Hyper-parameters in Cost Matrix. In this section, we conduct experiments on the impact of the choice of α and β in Eq. (12). Table 4 shows that slightly more weight on the first-order cost matrix than the second-order cost matrix gives the best overall results than other configurations for α and β . Note that $\alpha = 1.0$ and $\beta = 0.0$ are the same as only using first-order in the cost matrix for association.

Table 3 Evaluation on MOT-16 (FRCNN detector) and MOT-17 (SDP detectors) validation set with provided boxes from various detectors for improvements with KF using 3D information, acceleration, and high-order similarity matrix

Motion models	Association	Detector	MOTA $\uparrow$	IDF1 $\uparrow$	MT $\uparrow$	ML $\downarrow$	ID Sw. $\downarrow$
2DKF	First-order	FRCNN	54.7	55.3	25.7	26.8	1254
2DKF	High-order	FRCNN	57.7	57.3	30.6	23.6	1172
3DKF	First-order	FRCNN	57.6	56.2	28.8	30.7	955
3DKF	High-order	FRCNN	58.7	58.2	32.1	21.7	903
A-2DKF	First-order	FRCNN	57	55.5	28.63	25.1	1524
A-2DKF	High-order	FRCNN	57.2	57.6	29	24.95	819
A-3DKF	First-order	FRCNN	58.6	58.3	29.8	22.4	862
A-3DKF	High-order	FRCNN	58.9	59.6	31.33	21.6	821
2DKF	First-order	SDP	68.6	63.7	39.4	16.7	1275
2DKF	High-order	SDP	70.8	65.2	46.2	13.6	1023
3DKF	First-order	SDP	71.1	66.5	45.4	13.2	928
3DKF	High-order	SDP	72.4	66.8	48.2	12.5	773
A-2DKF	First-order	SDP	69	63.5	47.19	18.07	1205
A-2DKF	High-order	SDP	71.7	58.4	47.62	8.24	651
A-3DKF	First-order	SDP	71.9	65.1	49.6	13	1075
A-3DKF	High-order	SDP	72.4	68.3	51.5	13.4	524

$\downarrow$ means the smaller the better; $\uparrow$ means the larger the better

Table 4 Evaluation on MOT-17 (SDP detectors) validation set with various choice of hyperparameters in computing affinity matrix

α	β	MOTA $\uparrow$	IDF1 $\uparrow$	MT $\uparrow$	ML $\downarrow$	ID Sw. $\downarrow$
0.1	0.9	65.7	61.7	38.58	17.09	3137
0.3	0.7	66.0	61.7	41.36	16.48	3008
0.5	0.5	72.2	60.2	47.7	11.5	840
0.6	0.4	72.4	68.3	51.5	13.4	524
0.7	0.3	71.2	59.5	42.3	15.3	861
0.9	0.1	72.0	51.9	44.6	17.3	875
1.0	0.0	71.9	65.1	49.6	13	1075

$\downarrow$ means the smaller the better; $\uparrow$ means the larger the better

4.3 Benchmark Evaluation

Comparisons with the State-of-the-art MOT systems on MOTChallenge. We achieve state-of-the-art results on both MOT16 and MOT17 testing benchmarks using public detection and online settings. In this experiment, we adopt our best model setting with SODE and A-3DKF for motion model and high-order association. Table 5 shows the results achieved on these MOT challenges’ benchmark sets. These results

Table 5 Comparisons among MOT methods with public detections on MOTChallenge test sets

Set	Method	MOTA $\uparrow$	IDF1 $\uparrow$	MT $\uparrow$	ML $\downarrow$
MOT17	DP-MOT (ours)	65.0	67.1	28.8	27.5
	TMOH [40]	62.1	62.8	26.9	31.4
	Lif_T [14]	60.5	65.6	27.0	33.6
	LPC_MOT [10]	59.0	66.8	29.9	33.9
	MPNTrack [2]	58.8	61.7	28.8	33.5
	Tracktor++ V2 [1]	56.3	55.1	21.1	35.3
	FAMNet [9]	52.0	48.7	19.1	33.4
	eHAF [38]	51.8	54.7	23.4	37.9
	jCC [16]	51.2	54.5	20.9	37.0
MOT16	DP-MOT (ours)	63.7	66.2	25.8	28.2
	TMOH [40]	63.2	63.5	27.0	31.0
	Lif_T [14]	61.3	64.7	27.0	34.0
	LPC_MOT [10]	58.8	67.6	27.3	35.0
	MPNTrack [2]	58.6	61.7	27.3	34.0
	Tracktor++ [1]	54.4	52.5	19.0	36.9
	KCF [8]	48.8	47.2	15.8	38.1
	GCRA [25]	48.2	48.6	12.9	41.1
	MOTDT [7]	47.6	50.9	15.2	38.3

$\downarrow$ means the smaller the better; $\uparrow$ means the larger the better

further emphasize the advantages of our DP-MOT with significant MOTA score improvements of 8.7% and 9.3% on MOT17 and MOT16, respectively. Qualitative results on several videos of MOT challenges are illustrated in Figs. 5 and 7. Our proposed DP-MOT can effectively handle the occlusions during the tracking process. Using the depth orders from SODE helps the later steps, i.e., high-order association computation, correctly associate the detections with track IDs. For example, in Fig. 5, when two persons walk across each other, having the depth order of objects helps to avoid box switching from happening as in other approaches.

Comparisons with the State-of-the-art MOT systems on nuScene dataset. Our goal in this experiment is to demonstrate the generalization of our proposed method. It applies not only to static camera settings but also to dynamic camera settings, i.e., autonomous driving scenarios, on nuScene dataset [3]. We also compare our approach with other vision-based MOT approaches, including CenterTrack [55] and DEFT [4] on nuScene dataset. Our proposed approach achieves competitive results with other MOT approaches (Table 6).



Fig. 5 Tracking results in comparison with other methods (Tracktor [1], FAMNet [9] and eHAF [38]) on MOT challenge video MOT16-08. **Best viewed in color**

Table 6 Comparison of MOT performance on the nuScenes **validation set** for Vision Track challenge

Method	AMOTA	AMOTP	MOTAR	MOTA ↑	MOTP ↑	RECALL ↑	MT ↑	ML ↓	IDS ↓	FRAG ↓
CenterTrack [55]	0.068	1.543	0.349	0.061	0.778	0.222	524	4378	2673	1882
DEFT [4]	0.213	1.532	0.49	0.183	0.805	0.4	1591	2552	5560	2721
Ours	0.242	1.541	0.568	0.197	0.832	0.453	1643	2162	1362	1462

5 Conclusions

This work has presented a simple yet efficient subject-ordered depth estimation to tackle the MOT occlusion problem by ordering the detected subjects’ depth positions. Then, the new active pseudo-3D Kalman filter was proposed to keep updating the movement and order of objects dynamically. In addition, we have shown that introducing first-order and second-order relationships between the tracks and detected objects to the data association steps helps to consistently boost the performance to achieve state-of-the-art results on the MOT challenges. Our simple SODE will foster future work in studying what prior knowledge is available in off-the-shelf detectors. Due to the limitation of optical flow, people tracking in a crowd scene can be slower than in a normal scene. Our future work aims to study the impact of various 2D detectors on the accuracy of depth orders and how to extend and improve recent end-to-end 2D-MOT approaches using a similar pseudo-3D idea.

Acknowledgements This work is partly supported by NSF Data Science and Data Analytics that are Robust and Trusted (DART) grant. We also thank Huu Le for the invaluable discussions and acknowledge the Arkansas High-Performance Computing Center for providing GPUs.

Appendix

Method Details

This section provides more details on the proposed methods, i.e., A-3DKF and SODE, and network structure for feature embedding.

Feature Embedding Network

Network Structure. The Feature Embedding Network (FEN), a deep convolutional neural network, is used to extract appearance features $\mathbf{e}$ for each detected box. FEN is based on the network employed in [44] where we only use its feature embedding branch and modify its inputs to take detected boxes and provided features of the corresponding boxes. The backbone of FEN is DarkNet-53 [35] to balance accuracy and speed. Feature Pyramid Network [23] structure is applied to the backbone to handle objects at different scales. The feature embedding branch is connected with the backbone structure by a convolution layer with 512 channels to obtain identity embedding features at each location and scale. The embedding feature vector $\mathbf{e} \in \mathbb{R}^{512 \times 1}$ is extracted from the resulting feature map.

Comparison of Network Structure for Feature Embedding. In this section, we compare the choice of deep network structures, i.e., in terms of input sizes, in the feature sub-net (FEN). Each detected box will be cropped from an image and resized to fit the network input. We perform this experiment using three different input sizes, i.e., small (32×96), medium (45×135), and large (64×192). Table 7 shows the results of the MOT-16 training set to demonstrate the effects of the input sizes on the MOT results. The higher resolutions mean better appearance features are extracted, but the network is more complex. To balance accuracy and speed, we can choose an appropriate input size accordingly. The running speed of each configuration is 46 fps, 38 fps, and 30 fps for FEN-Small, FEN-Medium, and FEN-Large, respectively. We use the large input size in all other experiments for the best performance.

Occlusion Handling with A-3DKF

One observation from our experiments is that most methods fail because the detector mistakenly identifies two targeted subjects as one, i.e., a bounding box covers both

Table 7 Various configurations of Feature Embedding sub-net evaluated on MOT-16 training set. ↓ means the smaller the better; ↑ means the larger the better

Network structure	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	ID Sw. ↓	Speed (fps) ↑
FEN-Large	55.4	54.7	27	26	789	30
FEN-Medium	54	54	24.5	29.3	806	38
FEN-Small	52.1	55.0	23.6	31.5	1116	46



Fig. 6 An example of how our proposed approach (the last row) successfully handles a large overlapping case in MOT Challenge Video comparing with the state-of-the-art approaches Tracktor [1], eHAF [38], and FAMNet [9] (other rows). **Best viewed in color**

objects or partially when a large overlapping occurs. (See Fig. 6). Such cases can be detected by checking the column of matrix $\mathbf{C}$ containing the two smallest values where the difference is smaller than threshold τ_c . We will not use the detected box to update the state of A-3DKF. We will keep the two corresponding tracks activated.

Implementation Details

Parameters. The weights combining first-order and frame-wise cost matrix, i.e., α and β , are experimentally set to 0.45 and 0.55, respectively. The threshold for distance in depth τ_z is set to $\frac{1}{10} \times H$, where H is the height of the input image.

Dataset splits. FEN is trained on multiple large-scale datasets with pedestrian identity annotations described in [44]. These pedestrian datasets, which contain both

bounding box and identity annotations, are Caltech [11], MOT-16 training set [28], CUHK-SYSU [47], and PRW [53].

We use a dataset published by MOT challenges to evaluate the performance of pedestrians, including MOT 2016 and MOT 2017. We use the validation set of MOT16 (including *MOT16-02*, *MOT16-04*, *MOT16-05*, *MOT16-09*, *MOT16-10*, *MOT16-11* and *MOT16-13* videos), for ablation study and parameter configuration. The testing set of MOT16 (including *MOT16-01*, *MOT16-03*, *MOT16-06*, *MOT16-07*, *MOT16-08*, *MOT16-12* and *MOT16-14* videos) and MOT17 (including *MOT17-01*, *MOT17-03*, *MOT17-06*, *MOT17-07*, *MOT17-08*, *MOT17-12* and *MOT17-14* videos) are used to generate results and submit to challenge for Table 5.

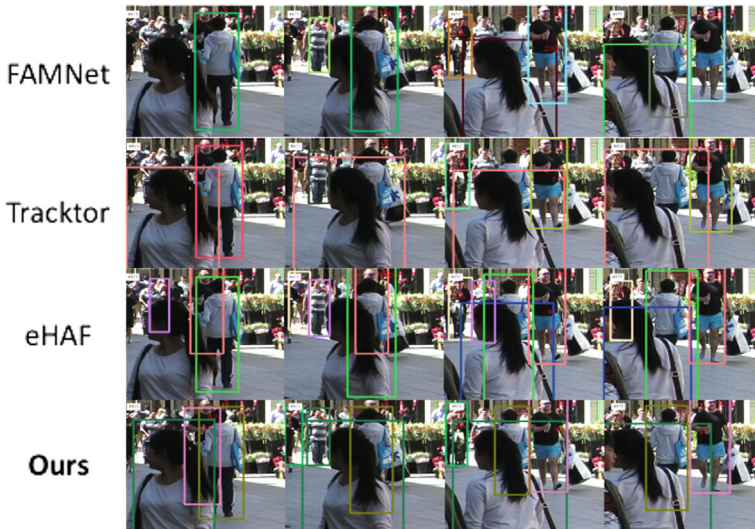


Fig. 7 Tracking results in comparison with other methods (Tracktor [1], FAMNet [9] and eHAF [38]) on MOT challenge video MOT16-08. **Best viewed in color**

Table 8 Results of individual sequences in MOTChallenge MOT-17 test sets. ↓ means the smaller the better; ↑ means the larger the better

Sequences	Camera setting	MOTA ↑	IDF1 ↑	MT ↑	ML ↓
MOT17-01	static	54.9	63.6	7	5
MOT17-03	static	81.9	78.6	106	11
MOT17-08	static	36.3	40.7	16	30
MOT17-06	handheld	59.8	60.0	69	59
MOT17-07	handheld	59.4	57.8	15	14
MOT17-12	handheld	53.8	59.3	20	35
MOT17-14	vehicle	45.3	55.2	17	51

Additional Results

This section illustrates more results than other SOTA methods in Fig. 7. We also show the results of individual sequences in the MOT-17 challenge in Table 8.

References

1. Bergmann P, Meinhardt T, Leal-Taixe L (2019) Tracking without bells and whistles. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 941–951
2. Brasó G, Leal-Taixé L (2020) Learning a neural solver for multiple object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 6247–6257
3. Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan, Y, Baldan G, Beijbom O (2020) nuscenes: a multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 11621–11631
4. Chaabane M, Zhang P, Beveridge R, O'Hara S (2021) Deft: detection embeddings for tracking. ArXiv preprint [arXiv:2102.02267](https://arxiv.org/abs/2102.02267)
5. Chan S, Jia Y, Zhou X, Bai C, Chen S, Zhang X (2022) Online multiple object tracking using joint detection and embedding network. Pattern Recognition 130:108793. <https://www.sciencedirect.com/science/article/pii/S0031320322002746>, <https://doi.org/10.1016/j.patcog.2022.108793>
6. Chen B, Wang D, Li P, Wang S, Lu H (2018a) Real-time 'actor-critic' tracking. In: Proceedings of the European conference on computer vision (ECCV), pp 318–334
7. Chen L, Ai H, Zhuang Z, Shang C (2018b) Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: IEEE international conference on multimedia and expo (ICME), IEEE, pp 1–6
8. Chu P, Fan H, Tan CC, Ling H (2019) Online multi-object tracking with instance-aware tracker and dynamic model refreshment. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV), pp 161–170
9. Chu P, Ling H (2019) Famnet: joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 6172–6181
10. Dai P, Weng R, Choi W, Zhang C, He Z, Ding W (2021) Learning a proposal classifier for multiple object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 2443–2452
11. Dollár P, Wojek C, Schiele B, Perona P (2009) Pedestrian detection: a benchmark. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 304–311
12. Godard C, Mac Aodha O, Brostow GJ (2017) Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 270–279
13. Godard C, Mac Aodha O, Firman M, Brostow GJ (2019) Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 3828–3838
14. Hornakova A, Henschel R, Rosenhahn B, Swoboda P (2020) Lifted disjoint paths with application in multiple object tracking. In: The 37th international conference on machine learning (ICML), pp 4364–4375
15. Hu HN, Cai QZ, Wang D, Lin J, Sun M, Krahenbuhl P, Darrell T, Yu F (2019) Joint monocular 3d vehicle detection and tracking. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 5390–5399

16. Keuper M, Tang S, Andres B, Brox T, Schiele B (2018) Motion segmentation and multiple object tracking by correlation co-clustering. *IEEE Trans Pattern Anal Mach Intell* 42:140–153
17. Kim C, Li F, Ciptadi A, Rehg JM (2015) Multiple hypothesis tracking revisited. In: *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp 4696–4704
18. Kroeger T, Timofte R, Dai D, Gool LV (2016) Fast optical flow using dense inverse search. In: *Proceedings of the European conference on computer vision (ECCV)*, pp 471–488
19. Laina I, Rupprecht C, Belagiannis V, Tombari F, Navab N (2016) Deeper depth prediction with fully convolutional residual networks. In: *Fourth international conference on 3D vision (3DV)*, IEEE, pp 239–248
20. Leal-Taixé L, Milan A, Reid I, Roth S, Schindler K (2015) MOTChallenge 2015: towards a benchmark for multi-target tracking. [arXiv:1504.01942](https://arxiv.org/abs/1504.01942) [cs] <http://arxiv.org/abs/1504.01942>
21. Li J, Gao X, Jiang T (2020) Graph networks for multiple object tracking. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV)*, pp 719–728
22. Li R, Zhang B, Teng Z, Fan J (2022) An end-to-end identity association network based on geometry refinement for multi-object tracking. *Pattern Recognition* 129:108738. <https://www.sciencedirect.com/science/article/pii/S0031320322002199>, <https://doi.org/10.1016/j.patcog.2022.108738>
23. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp 2117–2125
24. Luber M, Spinello L, Arras KO (2011) People tracking in rgb-d data with on-line boosted target models. In: *2011 IEEE/RSJ international conference on intelligent robots and systems, IEEE*, pp 3844–3849
25. Ma C, Yang C, Yang F, Zhuang Y, Zhang Z, Jia H, Xie X (2018) Trajectory factory: tracklet cleaving and re-connection by deep Siamese bi-gru for multiple object tracking. In: *IEEE international conference on multimedia and expo (ICME)*, IEEE, pp 1–6
26. Maksai A, Fua P (2019) Eliminating exposure bias and metric mismatch in multiple object tracking. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp 4639–4648
27. Meinhardt T, Kirillov A, Leal-Taixe L, Feichtenhofer C (2022) Trackformer: multi-object tracking with transformers. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 8844–8854
28. Milan A, Leal-Taixé L, Reid I, Roth S, Schindler K (2016) MOT16: a benchmark for multi-object tracking. [arXiv:1603.00831](https://arxiv.org/abs/1603.00831) [cs] <http://arxiv.org/abs/1603.00831>
29. Nguyen P, Quach KG, Duong CN, Le N, Nguyen XB, Luu K (2022) Multi-camera multiple 3d object tracking on the move for autonomous vehicles. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops*, pp 2569–2578
30. Nguyen P, Quach KG, Duong CN, Phung SL, Le N, Luu K (2024a) Multi-camera multi-object tracking on the move via single-stage global association approach. *Pattern Recognit* 110457
31. Nguyen P, Quach KG, Kitani K, Luu K (2024b) Type-to-track: retrieve any object via prompt-based tracking. *Adv Neu Inf Proc Syst* 36
32. Peng J, Wang C, Wan F, Wu Y, Wang Y, Tai Y, Wang C, Li J, Huang F, Fu Y (2020) Chained-tracker: chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In: *Proceedings of the European conference on computer vision (ECCV)*, Springer, pp 145–161
33. Quach KG, Nguyen P, Le H, Truong TD, Duong CN, Tran MT, Luu K (2021) Dyglip: a dynamic graph model with link prediction for accurate multi-camera multiple object tracking. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 13784–13793
34. Rajasegaran J, Pavlakos G, Kanazawa A, Malik J (2022) Tracking people by predicting 3d appearance, location and pose. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 2740–2749
35. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. *ArXiv preprint arXiv:1804.02767*

36. Ren L, Lu J, Wang Z, Tian Q, Zhou J (2018) Collaborative deep reinforcement learning for multi-object tracking. In: Proceedings of the european conference on computer vision (ECCV), pp 586–602
37. Scheidegger S, Benjaminsson J, Rosenberg E, Krishnan A, Granström K Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering. In: IEEE intelligent vehicles symposium (IV), IEEE, pp 433–440
38. Sheng H, Zhang Y, Chen J, Xiong Z, Zhang J (2018) Heterogeneous association graph fusion for target association in multiple object tracking. *IEEE Trans Circ Syst Video Technol* 29:3269–3280
39. Spinello L, Arras KO (2011) People detection in rgb-d data. In: IEEE/RSJ international conference on intelligent robots and systems, IEEE, pp 3838–3843
40. Stadler D, Beyerer J (2021) Improving multiple pedestrian tracking by track management and occlusion handling. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 10958–10967
41. Sun S, Akhtar N, Song H, Mian A, Shah M (2019) Deep affinity network for multiple object tracking. *IEEE Trans Patt Anal Mach Intell* 43:104–119
42. Sun S, Akhtar N, Song X, Song H, Mian A, Shah M (2020) Simultaneous detection and tracking with motion modelling for multiple object tracking. In: Proceedings of the European conference on computer vision (ECCV), pp 626–643
43. Wang C, Miguel Buenaposada J, Zhu R, Lucey S (2018) Learning depth from monocular videos using direct methods. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 2022–2030
44. Wang Z, Zheng L, Liu Y, Li Y, Wang S (2020) Towards real-time multi-object tracking. In: Proceedings of the European conference on computer vision (ECCV), Springer, pp 107–122
45. Weng X, Wang J, Held D, Kitani K (2020a) 3d multi-object tracking: a baseline and new evaluation metrics. In: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, pp 10359–10366
46. Weng X, Wang Y, Man Y, Kitani KM (2020b) Gnn3dmtot: graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 6499–6508
47. Xiao T, Li S, Wang B, Lin L, Wang X (2017) Joint detection and identification feature learning for person search. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 3415–3424
48. Xu J, Cao Y, Zhang Z, Hu H (2019) Spatial-temporal relation networks for multi-object tracking. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 3988–3998
49. Xu Y, Osep A, Ban Y, Horaud R, Leal-Taixé L, Alameda-Pineda X (2020a) How to train your deep multi-object tracker. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6787–6796
50. Xu Y, Osep A, Ban Y, Horaud R, Leal-Taixé L, Alameda-Pineda X (2020b) How to train your deep multi-object tracker. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 6787–6796
51. Yin J, Wang W, Meng Q, Yang R, Shen J (2020) A unified object motion and affinity model for online multi-object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 6768–6777
52. Yu E, Liu S, Li Z, Yang J, Han S, Tao W et al (2023) Generalizing multiple object tracking to unseen domains by introducing natural language representation. *AAAI*
53. Zheng L, Zhang H, Sun S, Chandraker M, Yang Y, Tian Q (2017) Person re-identification in the wild. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 1367–1376
54. Zhou T, Brown M, Snavely N, Lowe DG (2017) Unsupervised learning of depth and ego-motion from video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 1851–1858

55. Zhou X, Koltun V, Krähenbühl P (2020) Tracking objects as points. In: Proceedings of the European conference on computer vision (ECCV), pp 474–490
56. Zhu J, Yang H, Liu N, Kim M, Zhang W, Yang MH (2018) Online multi-object tracking with dual matching attention networks. In: Proceedings of the European conference on computer vision (ECCV), pp 366–382

Chapter 10

AI-Driven Predictive Maintenance



Ramazan Ünlü and İsmet Söylemez

1 Introduction

Industry 4.0 has greatly affected maintenance strategies in production and other sectors, as in many areas. In particular, developments in the field of artificial intelligence have caused classical maintenance strategies to change almost completely, making innovative and data-based approaches more popular. Although traditional approaches with a “fix it when it breaks” mentality may seem economically beneficial at first glance, unexpected downtimes, production line interruptions and rushed repairs make this strategy expensive and risky in the long run [1]. Preventive maintenance strategy, on the other hand, represents a more proactive maintenance strategy that is planned in advance to eliminate equipment malfunctions. This strategy, which often follows a planned schedule based on time or usage capacity, can increase operating costs by causing unnecessary downtime and resource usage. Moreover, since the real-time status of the equipment is not taken into account, it is not possible to detect unexpected malfunctions, which can further increase operating costs [2]. Beyond these approaches, with the emergence of Industry 4.0 and the development of artificial intelligence in almost every field, an AI-based predictive maintenance (PdM) strategy has overtaken traditional methods. This advanced strategy leverages AI and machine learning to predict equipment failures before they occur. By analyzing complex patterns in machine condition data, PdM not only improves the efficiency of maintenance strategies, but also offers an element of sustainability by extending the life of machine components [3]. AI algorithms play a pivotal role in this strategy, offering a refined and data-driven approach to monitoring critical machinery situations, such as tool wear and spindle motor functionality. This approach aims to

R. Ünlü (✉) · İ. Söylemez

Industrial Engineering Department, Abdullah Gul University, Kayseri, Turkey
e-mail: ramazan.unlu@agu.edu.tr

İ. Söylemez

e-mail: ismet.soylemez@agu.edu.tr

optimize maintenance scheduling, as opposed to the lack of maintenance or excess maintenance of traditional methods.

PdM aims to integrate the system developed using real-time data into operational decision-making processes to ensure an optimum balance between corrective and preventive maintenance strategies. This approach will contribute to the optimal realization of maintenance strategies in a way that traditional methods cannot. While traditional methods come into play either after a malfunction or after a pre-planned time, PdM enables malfunctions to be detected in advance by following a predictive mechanism [4]. Therefore, adopting artificial intelligence-driven maintenance strategies in production is also very important in terms of efficiency and strategic decision-making. It goes beyond traditional methodologies by incorporating advanced data analytics, ultimately increasing the operational life of production equipment and streamlining maintenance processes. Beyond all this, the successful implementation of PdM also means a transition toward more sustainable and smart production in line with the principles of Industry 4.0 [5]. On the other hand, the shift from reactive and preventative maintenance to AI-based maintenance strategies also represents a fundamental shift in the manufacturing industry. This shift represents not just a change in maintenance techniques, but a wholesale transformation toward a more data-driven, efficient and sustainable approach to maintaining equipment health. As the manufacturing industry continues to evolve under the influence of Industry 4.0, predictive maintenance powered by artificial intelligence and machine learning is expected to play an increasingly important role in improving operational efficiency and competitiveness.[7, 8]. In the literature there can be found aforementioned maintenance strategies (see Fig. 1) those are encapsulated and explained how those strategies are needed to revolutionized through this chapter.

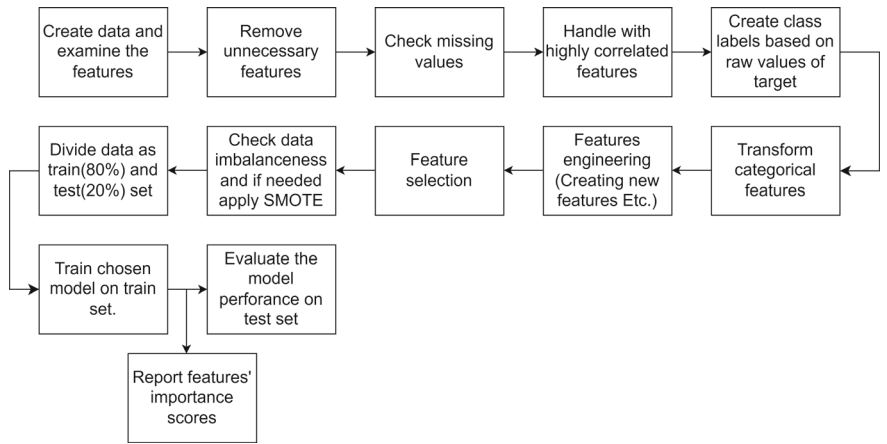


Fig. 1 Workflow of the case study

2 Traditional Maintenance Approaches

Reactive maintenance, which can be described as a “wait until it breaks” strategy, means taking action only after a failure occurs. While this approach is seemingly cost-effective due to the lack of upfront maintenance costs, it can often lead to higher overall costs. Unforeseen malfunctions of machines not only require urgent repair costs, but also cause production losses due to unexpected downtime. Failure to carry out regular maintenance checks in this method means that wear and deterioration in equipment goes unnoticed until a failure occurs, which can lead to more serious and costly damage. Additionally, reactive maintenance can negatively impact the lifespan of machines, as operating equipment to the point of failure often causes further damage. The unpredictability associated with reactive maintenance also creates challenges in resource allocation and inventory management, as emergency repairs often require urgent procurement of parts and labor, which can be significantly more expensive and less efficient than planned maintenance activities [9]. Preventive maintenance, on the other hand, is a more proactive approach and is structured around performing maintenance tasks based on specific usage metrics (such as machine usage hours or production cycles) at predetermined intervals. The primary purpose of preventive maintenance is to prevent malfunctions before they occur and extend the overall service life of the equipment. This approach leads to more efficient operations and a reduced risk of unexpected downtime, but it is difficult to optimally plan maintenance intervals. Maintenance may be performed too frequently, which may cause unnecessary machine downtime and increase labor and material costs. Conversely, if maintenance intervals are too long, the risk of equipment failure increases. On the other hand, preventive maintenance is often performed by following the same strategy for many different machines, which does not take into account the varying conditions and usage patterns of each machine, potentially leading to inefficient maintenance practices. It also does not use the advanced diagnostics and predictive analytics available in modern maintenance strategies that can more accurately predict equipment failure and optimize maintenance schedules [6].

2.1 *Limitations and Challenges of Traditional Maintenance Methods*

The transition from traditional strategies to more complex and artificial intelligence-based systems actually represents a radical paradigm shift. However, this radical change has now become necessary due to various limitations of traditional methods. The definitions and limitations of traditional methods are briefly summarized below:

1. Reactive Maintenance

- It operates on a “fix it when it breaks” philosophy.

- Often results in unplanned operational pauses and inflated maintenance costs due to emergency repairs.
- Potentially causes to more severe failures and safety hazards over time [1].

2. Preventive Maintenance

- A proactive approach involving scheduled maintenance activities.
- Despite its proactive nature, it can cause to inefficiencies due to unnecessary downtime and resource utilization.
- Often does not consider the real-time condition of equipment, leading to either over or under-maintenance [2].

3. Limitations of Traditional Methods

- Traditional methods often fail to prevent failure or predict future conditions of equipment accurately.
- Reactive and preventive approaches do not effectively leverage advancements in data analytics and IoT technologies.
- These methods are increasingly inadequate for modern, automated, and complex manufacturing environments, leading to higher operational costs and reduced equipment lifespan [10].

Hence, the shift from traditional maintenance approaches to predictive maintenance, underpinned by AI and machine learning, reflects the manufacturing industry's move toward more strategic, efficient, and sustainable practices. This transition is not merely a change in techniques but a transformation toward an approach that is aligned with the advanced, data-driven, and predictive capabilities required in contemporary manufacturing settings.

3 Fundamentals of AI-Enhanced Predictive Maintenance

Adopting AI in the field of Predictive Maintenance (PdM) is transforming conventional maintenance practices by leveraging the capabilities of sophisticated technologies like machine learning, data analytics, and the Internet of Things (IoT). The core principle of PdM revolves around the proactive use of insights gleaned from data to identify the most opportune times for maintenance actions. This method involves collecting and analyzing information from various sources, such as sensor outputs, patterns of equipment usage, and environmental conditions, to detect early signs of potential equipment failures. Consequently, PdM aims to improve the planning of maintenance activities, thereby decreasing unexpected equipment downtime and enhancing the longevity of the machinery [11].

In the context of AI-powered Predictive Maintenance (PdM), the amalgamation of state-of-the-art technologies is reshaping how industrial maintenance is performed. The quintessence of PdM is found in the smooth integration of Machine Learning

(ML), Data Analytics, and the Internet of Things (IoT), which collectively reform conventional maintenance methodologies and procedures. These integral technologies collaborate to transform large datasets into usable knowledge, paving the way for predictive functionalities previously deemed unachievable. An introductory description of each critical component follows:

Machine Learning in PdM: Machine learning algorithms enable PdM models to process and analyze data collected from industrial systems to operate effectively. These algorithms go beyond classical statistical methods and are quite successful in detecting complex and nonlinear relationships. Therefore, it is possible to achieve satisfactory prediction success even in very complex systems. It is possible to use a wide variety of algorithms depending on the structure of the problem and the dataset.

Deep Learning in PdM: Deep learning methods, a special branch of machine learning, are more advanced than machine learning and have the ability to capture very complex patterns. On the other hand, it can be used to process and analyze much more complex data such as images and sounds. It has become possible to design advanced PdM systems by detecting some patterns that relatively simpler Machine Learning cannot capture, thanks to Deep Learning methods [3].

Data Analytics in PdM: Data analytics, which covers various processes aimed at analyzing data from industrial machines and operations, generally includes the processes of preparing data by using many data pre-processing methods. Thanks to advanced analytics, including statistical analysis and trend analysis, it transforms raw data into valuable information that is crucial to detect equipment failures in advance, optimize maintenance schedules and increase operational efficiency [2].

Internet of Things (IoT) in PdM: IoT plays a key role in PdM by providing the infrastructure to collect data continuously and in real time through a network of sensors and devices embedded within industrial machines. This setup allows instant monitoring of many parameters such as temperature, vibration, etc., which are very important for early abnormality detection and therefore preventive maintenance action, and the collection of these data to create raw data [12].

To summarize, the use of machine learning, deep learning, data analytics and IoT systems have become a necessity for the creation of effective PdM systems. These systems, each connected to each other, constitute the main structure of PdM systems. Together, they form an integrated ecosystem that increases efficiency, minimizes downtime and promotes a proactive maintenance culture in the manufacturing industry.

3.1 Benefits Over Traditional Methods

In the ever-evolving landscape of the manufacturing sector, the integration of AI-driven predictive maintenance (PdM) marks a watershed moment, heralding a new era

in line with the visionary ethos of Industry 4.0. This shift from traditional maintenance methods to more advanced, predictive strategies is not just a change in techniques; it represents a fundamental transformation toward practices that are exponentially more efficient, anticipatory, and sustainable. But what exactly sets this new approach apart from its predecessors? As we delve into the multifaceted benefits of predictive maintenance, from its impact on cost reduction to its role in extending equipment life and enhancing safety, the profound implications of this technological leap become increasingly clear. This evolution in maintenance strategies is a testament to the power of foresight in technology, paving the way for a more robust and resilient manufacturing landscape.

AI-driven PdM significantly uplifts operational efficiency by aligning maintenance schedules with the actual needs of the equipment. This precise scheduling is based on real-time data and predictive analytics, ensuring that maintenance is only performed when truly necessary. Such an approach markedly reduces periods of unproductive downtime, which is often the consequence of unexpected equipment failures or unnecessarily frequent maintenance under traditional strategies. The efficiency gains from PdM are not just limited to reduced downtime. They also encompass improved utilization of maintenance resources, including manpower and spare parts, and better planning of maintenance activities. This optimized resource allocation directly translates into smoother production processes and heightened overall productivity in manufacturing settings. By ensuring equipment operates at its optimal condition, PdM contributes to the minimization of operational interruptions and maintains consistent production quality [13].

Improving the efficiency of a system will have a number of other benefits as well. For instance, by anticipating potential failures and addressing them proactively, PdM helps avoid the significant costs associated with unplanned downtime and emergency repairs. This predictive approach allows for the scheduling of maintenance tasks during non-peak hours, minimizing the impact on production. Furthermore, PdM extends the lifespan of equipment by preventing the excessive wear and tear that often results from operating faulty machinery. The extended equipment life and optimized maintenance interventions culminate in considerable savings over time, contributing to a more favorable financial standing for the organization. These cost savings are further augmented by the efficient use of maintenance resources and spare parts, reducing waste and inventory costs. In essence, PdM offers a robust return on investment, making it a financially prudent choice for modern manufacturing firms [14].

On the other hand, safety is a paramount concern in manufacturing, and PdM contributes significantly to creating safer work environments. By predicting and preventing equipment failures, PdM reduces the risk of accidents that can result from malfunctioning machinery. This proactive maintenance approach is particularly crucial in industries where equipment failure can lead to serious safety hazards. By ensuring that machinery is functioning correctly, PdM mitigates the risk of catastrophic failures that could endanger the lives of workers. Additionally, a well-maintained machine is less likely to contribute to workplace accidents, further enhancing the safety of the manufacturing environment. The implementation of PdM also fosters a

culture of safety and responsibility, as it demonstrates a commitment to maintaining equipment in top condition, thereby prioritizing the well-being of employees [15].

Besides all these benefits, AI-driven PdM has broader implications in Industry 4.0, where it is integral to the ethos of Industry 4.0 and represents a paradigm shift in maintenance strategies. Its reliance on AI, IoT, and data analytics dovetails with the broader goals of Industry 4.0, which emphasize automation, data exchange, and manufacturing technologies. PdM plays a critical role in realizing the vision of smart factories, where data-driven decision-making and automation lead to more efficient, sustainable, and competitive manufacturing operations. The proactive nature of PdM aligns with the sustainability objectives of modern industries by reducing waste, optimizing energy usage, and prolonging equipment life, thereby contributing to more environmentally friendly manufacturing practices [3].

Finally, the ongoing advancements in IoT and AI technologies have been pivotal in enabling the collection, analysis, and interpretation of large volumes of data, which are the cornerstone of effective PdM systems. These technologies facilitate the development of sophisticated predictive models that can accurately forecast equipment failures, allowing for timely and precise maintenance interventions. As more data becomes available and these models are refined, the predictions become increasingly accurate, leading to more effective and efficient maintenance strategies. This continuous improvement cycle is essential for the evolution and success of PdM in the context of Industry 4.0 [16], and those improvements help PdM to mark a significant advancement in the field of industrial maintenance. Its emphasis on efficiency, cost-effectiveness, and safety, coupled with its alignment with the principles of Industry 4.0, positions it as a critical component in the future of smart manufacturing. The integration of advanced technologies in PdM is not just enhancing operational capabilities but also paving the way for more sustainable, safe, and competitive industrial practices.

4 Technological Foundations of Predictive Maintenance

In the dynamic landscape of Industry 4.0, predictive maintenance (PdM) has emerged as a critical strategy, reshaping maintenance approaches in manufacturing through the integration of advanced technologies as aforementioned machine learning, data analytics, sensors, and the Internet of Things (IoT). The technological foundations of PdM represent a confluence of these elements, each playing a crucial role in transforming data into actionable insights for maintenance decision-making. Some of those technologies are briefly summarized as follows.

4.1 *Machine Learning and Data Analytics in Predictive Maintenance*

Machine learning and data analytics are at the heart of predictive maintenance, providing the computational power to process and analyze vast amounts of data collected from industrial equipment. These methodologies enable the identification of patterns and anomalies that indicate potential equipment failures. Predictive maintenance leverages a variety of machine learning algorithms, from basic regression models to sophisticated neural networks, to forecast equipment health and maintenance needs. The power of machine learning and data analytics allows for more accurate predictions of equipment failures, facilitating timely maintenance interventions that enhance operational efficiency and extend equipment lifespan [11, 17].

Neural Networks: Deep Neural Networks (DNNs), including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have become pivotal in PdM. DNNs, with their multiple layers of processing, are adept at modeling complex, nonlinear relationships in data. CNNs are particularly effective for processing data with spatial hierarchies, such as images or time-series data from sensors, making them suitable for tasks like anomaly detection. RNNs excel in handling sequential data, essential for analyzing time-series data and capturing temporal dependencies crucial for predicting equipment failures [18].

Regression Analysis: Regression models, both linear and nonlinear, are fundamental in PdM for establishing relationships between various equipment parameters and predicting future failures. For instance, logistic regression and polynomial regression are used to model complex, nonlinear relationships that simpler linear models cannot capture. These regression models are crucial for tasks like Remaining Useful Life (RUL) estimation, where predicting the time until equipment failure is essential for maintenance scheduling [19].

Ensemble Learning Methods: Techniques like Gradient Boosting Machines (GBM), Light GBM, and XGBoost are increasingly being used in PdM. These methods combine multiple learning algorithms to improve predictive accuracy. For example, GBM and XGBoost are particularly effective for handling unstructured and complex datasets often encountered in industrial settings.

Supervised Learning Algorithms: Algorithms like Support Vector Machines (SVMs) and K-Nearest Neighbors (K-NN) are widely used in PdM for classification and regression tasks. These algorithms help in identifying the likelihood of equipment failure and the classification of equipment health status, based on historical and real-time data [20].

Data Preprocessing and Feature Engineering: Efficient data processing and feature extraction are crucial in maximizing the predictive quality of machine learning models for PdM. This includes handling high-dimensional data, addressing class imbalance, and selecting relevant features that contribute most significantly to predicting failures[21].

Application in Various Industries: Machine learning-based PdM is not limited to a single industry but has widespread applicability across different sectors, includ-

ing aerospace, semiconductor manufacturing, and discrete parts manufacturing. This versatility showcases the robustness of machine learning algorithms in catering to diverse maintenance requirements. [22, 23]

Continual Learning in PdM: As industries evolve and equipment undergoes changes, continual learning approaches in machine learning are becoming vital. These methods enable predictive models to adapt and incorporate new knowledge after deployment, ensuring that they remain effective over time despite changes in data distribution and operational conditions [24].

To sum up, machine learning and data analytics are thus fundamental to the efficacy of predictive maintenance strategies. These methodologies not only enhance the accuracy of maintenance predictions but also significantly contribute to the overall efficiency and sustainability of manufacturing operations. The adoption and continual advancement of these techniques are pivotal in maintaining the competitiveness and operational excellence of modern industries.

4.2 Role of Sensors and IoT in Data Collection

The integration of sensors and the Internet of Things (IoT) into predictive maintenance (PdM) systems has brought about a revolutionary change in data collection within the manufacturing industry. This shift from traditional methodologies is underscored by the ability to continuously monitor and assess the health of equipment, leading to maintenance interventions that are both timely and precise. At the forefront of this transformation is the expanded role of sensors in data collection. These devices are strategically embedded in industrial equipment, monitoring various parameters such as temperature, vibration, pressure, and sound. This deployment of diverse sensors offers real-time insights into the operational status of machinery. Early detection of any anomalies that might indicate impending equipment failure becomes possible. Furthermore, modern sensors have evolved beyond mere data collection tools. Equipped with advanced features like wireless communication and self-diagnostic capabilities, they are adept at functioning in the demanding conditions of industrial environments, providing high-quality, reliable data. The granularity of sensor data is another crucial aspect. High-resolution data capture allows for a more nuanced analysis of equipment performance and health. This level of detail is instrumental in making accurate predictions and diagnoses.

IoT's central role in this paradigm extends to data collection and analysis. It provides the necessary infrastructure for connecting these myriad sensors to centralized systems. This interconnectedness not only facilitates the aggregation of sensor data but also its transmission to cloud-based platforms or on-premise data centers for advanced analysis, as detailed by [25]. The real-time processing of sensor data, a capability intrinsic to IoT technologies, is indispensable in PdM. This immediacy in information processing, highlighted by [26], is critical in preempting equipment failures and optimally scheduling maintenance activities. Additionally, IoT platforms are

instrumental in integrating sensor data with other relevant operational and environmental information. This comprehensive data amalgamation enhances the accuracy of predictive analysis, offering a holistic view of equipment health and performance. However, this IoT-driven data collection landscape is not without its challenges and opportunities. The overwhelming volume and speed at which data is generated pose significant challenges in terms of effective data management. Strategies must be devised to handle this influx, ensuring the relevance and accuracy of the data analyzed for maintenance predictions. Security and reliability also emerge as paramount concerns with the proliferation of IoT devices in industrial settings. Implementing robust security measures to safeguard against data breaches and ensure data integrity is essential, a concern echoed by [27]. Furthermore, the integration of IoT technologies with existing industrial systems demands careful planning and execution to ensure compatibility and minimize operational disruptions, a process explored by [28].

In conclusion, the role of sensors and IoT in enhancing data collection for predictive maintenance is both complex and transformative. These technologies bring forth significant advantages over traditional maintenance methods. The continuous, real-time monitoring capabilities they provide pave the way for a maintenance approach that is not only more proactive but also markedly more efficient. This enhancement in maintenance management is a critical factor in boosting the reliability and longevity of manufacturing equipment, ultimately propelling the industry toward a more innovative and resilient future.

4.3 Data Processing and Analysis for Prediction

The collected data undergoes extensive processing and analysis to extract meaningful insights for predictive maintenance. Data preprocessing techniques such as normalization and filtering are applied to clean the data and remove noise. Machine learning models are then trained on this processed data to detect patterns indicative of equipment failure. These models continuously evolve and improve as more data is collected, leading to increasingly accurate predictions. The integration of machine learning with data analytics enables the development of robust predictive models that inform maintenance decisions, thereby reducing downtime and maintenance costs [29, 30].

5 Case Study

In this section, we embark on an in-depth case study focused on the implementation of predictive maintenance for milling machines, a cornerstone in modern manufacturing. By integrating a sophisticated classification model, this study aims to not only identify but also categorize various potential failures, a step crucial for enhancing the

efficiency and longevity of these vital machines. This approach represents a significant leap from traditional maintenance practices, employing advanced data analytics to address issues before they escalate into costly downtime. Here, we will explore the methodology, challenges, and transformative potential of this model, providing valuable insights into its practical application in the field of industrial machinery maintenance. All implemented steps are summarized and illustrated in Fig. 1.

5.1 Dataset

This dataset is based on a milling machine model and includes 10,000 data points created by [31], each with 14 distinct features. The features are:

- **UID:** Unique identifier for each data point, ranging from 1 to 10,000.
- **Product ID:** Denoted by a letter (L, M, or H) representing the product quality (Low, Medium, High) and a serial number specific to each variant.
- **Type:** Product type derived from the Product ID.
- **Air Temperature [K]:** Generated through a random walk process, normalized around 300 K with a standard deviation of 2 K.
- **Process Temperature [K]:** Also generated via a random walk, normalized to a standard deviation of 1 K, and set higher than the air temperature by 10 K.
- **Rotational Speed [rpm]:** Derived from a power of 2860 W, with added normally distributed noise.
- **Torque [Nm]:** Normally distributed around 40 Nm with a standard deviation of 10 Nm.
- **Tool Wear [min]:** Varies based on the product quality variant, adding 2 to 5 min of wear per process.
The dataset also includes a “machine failure” labels, which is marked “1”, “2”, “3”, “4”, and “5” based on the following failure modes occur:
- **Tool Wear Failure (TWF):** Tool replacement or failure happens randomly between 200–240 minutes of wear.
- **Heat Dissipation Failure (HDF):** Occurs if the difference between air and process temperature is below 8.6 K, and the rotational speed is under 1380 rpm.
- **Power Failure (PWF):** If the calculated power (from torque and rotational speed) is outside the 3500 W to 9000 W range.
- **Overstrain Failure (OSF):** Triggered if the product of tool wear and torque exceeds certain limits based on the product variant.
- **Random Failures (RNF):** A 0.1% chance of failure regardless of process parameters.

The dataset is designed for predictive maintenance applications, where the objective is to predict machine failure without explicit knowledge of which specific failure mode triggered it. Therefore, this is an multi-class classification problem and, so failure types, define as target, are labeled based on given explanation above (Table 1).

Table 1 Illustration of the raw data

UDI	Product ID	Type	Air temperature (K)	Process temperature (K)	Rotational speed (rpm)	Torque (Nm)	Tool wear (min)	Machine failure	TWF	HDF	PWF	OSF	RNF
2817	L49996	L	300.1	309.4	1434	47.3	127	0	0	0	0	0	0
6260	M21119	M	301.2	310.8	1396	47.4	5	0	0	0	0	0	0
8569	L55748	L	297.9	308.6	1691	29.6	116	0	0	0	0	0	0
2314	M17173	M	299.1	308.8	1725	26.6	172	0	0	0	0	0	0
7066	L54245	L	300.8	310.6	1384	52.9	165	0	0	0	0	0	0
6899	M21758	M	301	311.6	1448	43.9	145	0	0	0	0	0	0
5702	M20561	M	302.5	311.9	1347	54.3	125	0	0	0	0	0	0
3942	L51121	L	302.2	311.4	1987	19.9	200	0	0	0	0	0	0
3160	H32573	H	300.4	309.8	1409	47.5	138	0	0	0	0	0	0
2034	H31447	H	298.8	308.5	1359	47.6	85	0	0	0	0	0	0
7431	M22290	M	300	311.4	1384	46.1	7	0	0	0	0	0	0
5953	L53132	L	300.7	310.8	1563	33.8	104	0	0	0	0	0	0
6075	M20934	M	301	310.8	1533	39.3	194	0	0	0	0	0	0
3286	L50465	L	301.2	309.9	1687	29.7	30	0	0	0	0	0	0
3627	M18486	M	302.2	311.5	1360	53	36	0	0	0	0	0	0
5116	M19975	M	304.2	313.7	2113	18.4	139	0	0	0	0	0	0
1903	L49082	L	298	307.6	1694	29.1	169	0	0	0	0	0	0
4822	L52001	L	303.5	311.9	1405	45.6	10	0	0	0	0	0	0
5329	M20188	M	303.7	312.9	1406	41.9	48	0	0	0	0	0	0
6061	L53240	L	301	311.1	1615	35.6	158	0	0	0	0	0	0
8345	L55524	L	298.8	309.7	1609	40.4	169	0	0	0	0	0	0

(continued)

Table 1 (continued)

UDI	Product ID	Type	Air temperature (K)	Process temperature (K)	Rotational speed (rpm)	Torque (Nm)	Tool wear (min)	Machine failure	TWF	HDF	PWF	OSF	RNF
4075	L51254	L	302	310.7	1453	42.7	92	0	0	0	0	0	0
3591	L50770	L	301.8	310.6	1403	46.8	151	0	0	0	0	0	0
944	M15803	M	295.6	306.2	1477	45	40	0	0	0	0	0	0
4832	L52011	L	303.5	312	1366	54.5	30	1	0	1	0	0	0
5946	M20805	M	300.8	310.8	1415	44.5	87	0	0	0	0	0	0
8260	L55439	L	299.1	310.5	1471	38.4	146	0	0	0	0	0	0
6880	H36293	H	300.8	311.2	2237	17	95	0	0	0	0	0	0
5622	M20481	M	302.9	312.2	1292	54.4	136	0	0	0	0	0	0
7555	M22414	M	300.3	311.4	1371	50.2	109	0	0	0	0	0	0
3652	L50831	L	302.1	311.5	1444	44.9	97	0	0	0	0	0	0
5090	M19949	M	303.9	313.1	1483	47.2	71	0	0	0	0	0	0
6816	L53995	L	300.9	310.7	1427	44.1	141	0	0	0	0	0	0
7614	L54793	L	300.5	311.3	1475	46.3	49	0	0	0	0	0	0
6992	L54171	L	300.8	311.1	1509	33.4	172	0	0	0	0	0	0
7428	L54607	L	300	311.4	1634	31.9	0	0	0	0	0	0	0
6099	M20958	M	300.8	310.7	1924	22.4	36	0	0	0	0	0	0
3574	L50753	L	301.8	310.6	1452	48.4	110	0	0	0	0	0	0
7227	L54406	L	299.7	309.6	1399	51.3	144	0	0	0	0	0	0
7830	L55009	L	299.8	311.2	1664	28.3	173	0	0	0	0	0	0
1972	L49151	L	297.9	307.6	1298	52.7	132	0	0	0	0	0	0
7448	M22307	M	300.3	311.9	1677	30.8	48	0	0	0	0	0	0

5.2 *Preprocessing the Data*

In the subsequent phase of our exploration, we delve into the crucial stage of data preprocessing, a foundational step in transforming raw data into a refined format suitable for effective analysis. This process is essential in ensuring the integrity and usability of the dataset, particularly in predictive maintenance where accuracy is paramount. We will navigate through various techniques employed to clean, normalize, and organize the data, addressing potential anomalies and inconsistencies. By meticulously preparing the data, we lay the groundwork for the sophisticated analysis that follows, setting the stage for accurate and reliable predictive modeling.

The initial step involves a preliminary review of the data. Upon examination, it's apparent that there are two indices present: the general index (UDI) and the ProductID which can be eliminated, they cannot give any useful information about the target values. Additionally, the "Type" is a categorical variable, while the rest of the data appears to be numerical. Categorical variables can be challenging, since some algorithms are not able to handle with them because of the mathematical foundation of them. Hence, we will need to also handle with that categorical variable during the preprocessing step.

To proceed step by step, it's essential to check for missing values in the dataset. Once identified, suitable strategies need to be implemented for handling these gaps. For instance, we could use imputation methods where missing values are replaced with statistical estimates like the mean or median of the respective column. Alternatively, if the missing data is not randomly distributed and could bias the analysis, we might consider using more sophisticated techniques such as k-nearest neighbors (KNN) imputation. Another approach could be to completely remove rows or columns with a high percentage of missing values, provided that such exclusion does not significantly impact the dataset's integrity or the study's outcome. Because this is a synthetic dataset, we do not face with any missing values.

In the intricate process of data analysis, the significance of checking for correlations between variables cannot be overstated. This step is crucial as it aids in uncovering underlying relationships that might not be immediately apparent, providing deeper insights into the behavior and characteristics of the dataset. Understanding these correlations is especially vital in predictive maintenance, as it helps in identifying redundant features that could skew the model's accuracy. For instance, highly correlated variables can lead to multicollinearity, potentially distorting the results of predictive models. Thus, by meticulously examining these relationships, we ensure that our analysis is not only more precise but also more reflective of the true dynamics within the data. This approach paves the way for more effective and efficient predictive models, crucial in the realm of machine maintenance. We observe that certain features, specifically process temperature and air temperature, exhibit a strong correlation. Similarly, a notable correlation is present between torque and rotational speed. In light of this, it might be prudent to eliminate one of the temperature variables to reduce redundancy. However, the distinct relationship between torque and rotational speed could serve as a valuable indicator of potential failures,

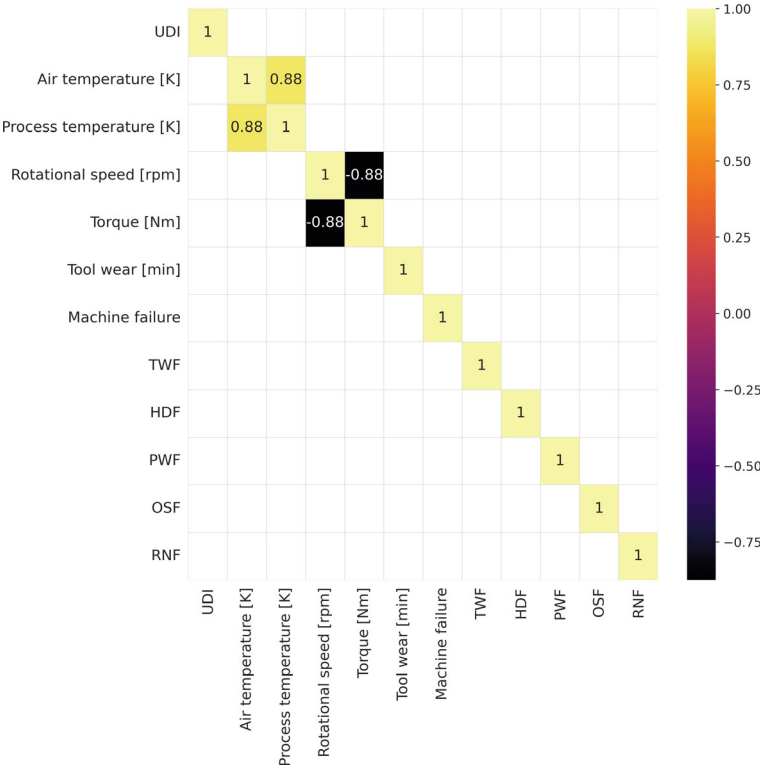


Fig. 2 Correlations between attributes

warranting the retention of both these measurements in our analysis. Figure 2 shows the correlation between variables. It should be noted that a threshold value of 0.8 has been established to display only features with high correlation.

Also, other variables, namely “UDI” and “Product ID”, are dropped as mentioned before, because they may not contribute significantly to our predictive analysis due to their lack of predictive power. After doing this, we streamline our dataset by consolidating the various failure modes into a singular feature, designed for multiclass classification. This consolidation not only simplifies our model but also enhances its analytical focus. Subsequently, after this integration, we remove the individual failure mode columns to avoid redundancy and maintain the clarity of our dataset. On the other hand, as we mentioned above, we need to handle with categorical variable, namely “Type”, in order to prepare dataset for the classification mode. We implemented this by using “get_dummies” function from “pandas” module and created two new columns called as “Type_L” , “Type_M”. “Type_H” is disregarded since highly dominated by others, and so might not able to insightful information about the target.

Unearthing hidden patterns in data often requires the astute derivation of new features from existing ones. This approach can be instrumental in revealing insights that are not immediately apparent. In our case study, for instance, we introduce the following new features to deepen our analysis:

- **Power:** Calculated as the product of rotational speed and torque, this feature aims to provide a more comprehensive understanding of the machine's operational dynamics.
- **Power and Wear:** By multiplying Power with Tool Wear, we create a feature that reflects the interaction between the machine's performance and its wear rate, offering a nuanced perspective on its overall efficiency and health.
- **Process Temperature per Power Output:** This ratio gives us a clearer picture of how the machine's process temperature correlates with its power output, potentially unveiling critical operational patterns.

These newly crafted features are not just additional columns in our dataset; they represent a strategic effort to uncover deeper, more meaningful relationships within the data, enhancing the predictive power of our model. After doing all these preprocessing steps, data looks like as shown in Table 2.

At this juncture, there's another critical step to undertake: assessing the dataset for any imbalance issues. Evaluating and rectifying imbalances is essential, as skewed data can lead to biased models that do not accurately reflect real-world scenarios. Imbalance often occurs when one class of data is significantly more prevalent than others, which can cause predictive models to favor the majority class, overlooking the nuances and patterns of less represented classes. This can especially impact the performance of the model in real-life applications where balanced prediction is crucial. By identifying and addressing these imbalances, we can enhance the robustness and fairness of our predictive model, ensuring it performs effectively across all data scenarios. Figure 3 shows how the data is imbalanced. As it can be seen the normal signals dominates the failure signals, so we need to handle with this fact while creating train and test set will be described in Sect. 5.4.

5.3 Feature Selection

In the next crucial phase of our analysis, we turn our attention to feature selection, a process pivotal in refining our predictive model for optimal performance. Feature selection involves meticulously identifying and retaining only those variables that contribute most significantly to the predictive power of our model. This not only enhances the model's efficiency but also aids in avoiding the pitfalls of overfitting and complexity. Among various techniques, within this framework, we have employed backward regression as our primary method for feature selection. This technique methodically evaluates the dataset, progressively removing the least significant features based on specific criteria until only the most impactful variables remain. By

Table 2 Illustration of the dataset after preprocessing

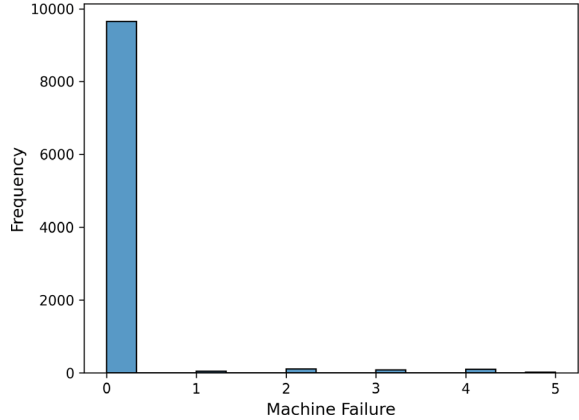
Air tem- perature (K)	Process tempera- ture (K)	Rotational speed (rpm)	Torque (Nm)	Tool wear (min)	Power	Power wear	Temperature difference	Temperature power	Machine failure	Type_L	Type_M
301.3	310	1593	30.8	210	49064.4	10303524	8.7	0.000177318	0	0	1
301.9	310.9	1309	48	124	62832	7791168	9	0.000143239	0	1	0
301.3	310.6	1426	40.7	54	58038.2	3134062.8	9.3	0.000160239	2	1	0
297.9	307.3	1824	26.3	5	47971.2	239856	9.4	0.000195951	0	1	0
297.9	309.5	1390	47.9	127	66581	8455787	11.6	0.000174224	0	1	0
297.3	308.7	1655	34.1	21	56435.5	1185145.5	11.4	0.000202001	0	0	1
299.6	309.4	1643	32.9	145	54054.7	7837931.5	9.8	0.000181298	0	1	0
297.9	307.8	1451	38.2	80	55428.2	4434256	9.9	0.000178609	0	1	0
303.2	311.2	1427	45.6	25	65071.2	1626780	8	0.000122942	0	1	0
296.8	308	1453	51	24	74103	1778472	11.2	0.000151141	0	0	0
301.2	310.9	1416	53.3	66	75472.8	4981204.8	9.7	0.000128523	1	1	0
300.6	310.7	1654	32	180	52928	9527040	10.1	0.000190825	0	0	1
298.5	309.4	1365	48.7	56	66475.5	3722628	10.9	0.00016397	0	0	1
297.6	308.3	1623	36.9	61	59888.7	3653210.7	10.7	0.000178665	0	1	0
300.9	311.3	1449	40.2	168	58249.8	9785966.4	10.4	0.000178541	0	1	0
301.9	310.8	1563	39.2	39	61269.6	2389514.4	8.9	0.00014526	0	0	1
300.8	312.3	1435	44.8	134	64288	8614592	11.5	0.000178883	0	1	0
298.4	309.2	1610	32.4	194	52164	10119816	10.8	0.000207039	3	1	0
298.2	309.4	1447	47.1	185	68153.7	12608434.5	11.2	0.000164334	0	1	0
298.3	309.1	1430	40.9	162	58487	9474894	10.8	0.000184656	0	0	0
297.4	308.2	1639	33.8	39	55398.2	2160529.8	10.8	0.000194952	0	1	0

(continued)

Table 2 (continued)

Air tem- perature (K)	Process tempera- ture (K)	Rotational speed (rpm)	Torque (Nm)	Tool wear (min)	Power	Power wear	Temperature difference	Temperature power	Machine failure	Type_L	Type_M
300	311.4	1927	24.6	132	47404.2	6257354.4	11.4	0.000240485	0	1	0
297.3	308.1	1566	35.7	156	55906.2	8721367.2	10.8	0.000193181	0	1	0
298.6	309.7	1429	45.8	113	65448.2	7395646.6	11.1	0.0001696	5	1	0
298.4	309.5	1519	35.4	133	53772.6	7151755.8	11.1	0.000206425	0	0	1
300.8	311.1	1583	32.1	116	50814.3	5894458.8	10.3	0.000202699	0	0	1
298	307.7	1387	50.8	204	70459.6	14373758.4	9.7	0.000137668	0	1	0
297.1	307.9	1456	50	166	72800	12084800	10.8	0.000148352	0	1	0
299.2	309	1881	23.9	137	44955.9	6158958.3	9.8	0.000217991	0	0	1
301.5	310.6	1403	47.4	34	66502.2	2261074.8	9.1	0.000136838	0	0	1
299.7	309.3	1779	25.5	129	45364.5	5852020.5	9.6	0.000211619	0	1	0
301.2	311.2	1635	29.6	211	48396	10211556	10	0.000206629	0	1	0
301.6	310.5	1397	49.4	104	69011.8	7177227.2	8.9	0.000128963	0	1	0
297.7	308.4	1510	38.6	72	58286	4196592	10.7	0.000183578	0	1	0
299.1	308.3	1431	52.3	38	74841.3	2843969.4	9.2	0.000122927	0	1	0
300.5	309.3	1475	46.8	94	69030	6488820	8.8	0.000127481	0	1	0
301.1	310.6	1782	24.9	93	44371.8	4126577.4	9.5	0.0002141	0	0	0
298.2	309.1	1467	44.5	80	65281.5	5222520	10.9	0.000166969	0	1	0
301.7	310.7	1605	33.3	220	53446.5	11758230	9	0.000168393	0	1	0
301.7	310.2	1471	51.1	8	75168.1	601344.8	8.5	0.00011308	0	1	0
298.6	308.2	1361	68.2	172	92820.2	15965074.4	9.6	0.000103426	4	1	0
298.9	309.8	1287	61.9	216	79665.3	17207704.8	10.9	0.000136822	4	1	0

Fig. 3 Total number of failure types



applying backward regression, we ensure that our final model is both streamlined and potent, equipped with the most relevant features that drive accurate predictions. In our analysis, we established a threshold for p -values at 0.05 to identify and eliminate features lacking significant predictive information. As a result of this stringent criterion, the newly created feature “Type_M” did not meet the threshold and was subsequently removed from the dataset. This decision was based on the principle that retaining only statistically significant features would lead to a more accurate and efficient predictive model. Therefore, the training stage will be started by remaining 11 features, namely “Air temperature (K)”, “Process temperature (K)”, “Rotational speed (rpm)”, “Torque (Nm)”, “Tool wear (min)”, “Power”, “Power wear”, “Temperature difference”, “Temperature power”, “Type_L”.

5.4 Modeling and Evaluation

At this stage of our analysis, we have selected XGBoost as our training model, drawn to its unique ability to map out decisions and their possible consequences. The XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted ensemble learning model. It provides parallel tree boosting and can be used for regression, classification, and ranking problems. To ensure the effectiveness of this stage, we first employed the Synthetic Minority Over-sampling Technique (SMOTE). This technique addresses the challenge of imbalanced datasets by creating synthetic samples from the minority class, thus balancing the dataset without losing crucial information. By doing so, SMOTE helps in enhancing the model’s ability to identify and learn from the underrepresented class, improving its overall predictive performance.

Next, we divided our dataset into training and testing subsets. Randomly selecting 20% of the data for testing ensures that our model is evaluated on a diverse and

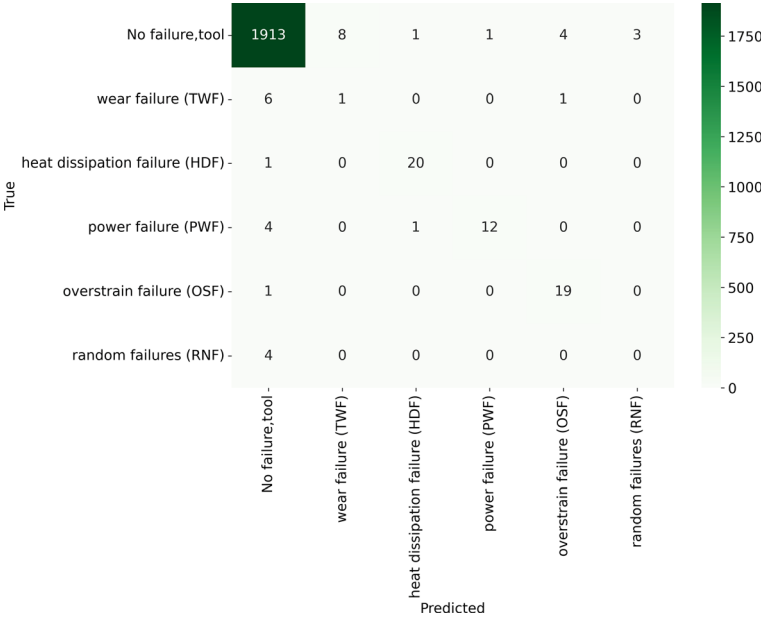


Fig. 4 Confusion matrix of the results of the test data

representative sample, providing us with a reliable measure of its performance. This approach helps in avoiding overfitting, ensuring that our model can generalize well to new, unseen data. Lastly, we normalized the training set using the MinMaxScaler from the scikit-learn.preprocessing module. Normalization is crucial as it scales the feature values to a uniform range, typically 0 to 1. This process is particularly beneficial for models like XGBoost, as it ensures that all features contribute equally to the decision-making process, preventing any single feature with a larger scale from dominating the model’s predictions.

Once the model training is completed, the next crucial step involves assessing its performance using the test dataset. The effectiveness of the model is quantitatively analyzed through a confusion matrix. A confusion matrix is not just a mere table; it’s a powerful tool that provides comprehensive insights into the accuracy of predictions. It helps in breaking down the number of correct and incorrect predictions, categorizing them into true positives, true negatives, false positives, and false negatives. This clarity is instrumental in understanding the model’s strengths and weaknesses, particularly in differentiating between types of errors. By using a confusion matrix, we can fine-tune our model’s parameters, aiming for a balance between sensitivity (true positive rate) and specificity (true negative rate), which is crucial for enhancing the overall predictive performance of the model. Figure 4 illustrates the confusion matrix of the results on the test data.

Drawing insights from the confusion matrix allows us to offer the following fundamental interpretations:

- True Positives for “No failure”: The model correctly predicted “No failure” 1913 times.
- False Positives for “No failure”: The model incorrectly predicted “No failure” when it was actually one of the failure types a total of 16 times (6 “wear failure (TWF)”, 1 “Heat dissipation failure (HDF)”, 4 “Power failure (PWF)”, 1 “Overstrain failure (OSF)”, 4 “random failures”).
- False Negatives for “No failure”: The model failed to predict “No failure” when it should have 17 times (8 “wear failure (TWF)”, 1 “Heat dissipation failure (HDF)”, “Overstrain failure (OSF)”, 1 “Power failure (PWF)”, 3 “Random failures (RNF)”).
- For “wear failure (TWF)”, the model correctly identified 2 case, but there were 6 cases where it predicted “No failure” instead (False Negatives).
- “Heat dissipation failure (HDF)” was correctly identified 20 times, but there was 1 instance where the model predicted “No failure” (False Negative).
- “Power failure (PWF)” was correctly identified 12 times, but there were 4 instances where “No failure” and 1 instance where “Heat dissipation failure (HDF)” was incorrectly predicted (False Negatives).
- “Overstrain failure (OSF)” was correctly identified 19 times, with 1 instance of a False Negative.
- “Random failures (RNF)” were not correctly identified at all; the model incorrectly predicted “No failure” in all 4 instances.

In summary, the model is highly accurate in predicting “No failure” but less so for “wear failure” and “random failures”. It performs reasonably well for “heat dissipation”, “power”, and “overstrain” failures. The higher numbers on the diagonal from top-left to bottom-right indicate correct predictions, and the model’s strength seems to lie in identifying “No failure”, “heat dissipation failure”, “power failure”, and “overstrain failure”. However, it struggles with “wear failure” and “random failures”, where it tends to predict “No failure” instead. Improvements in the model could aim to reduce the False Negatives in “wear failure” and “random failures” while maintaining the high True Positive rates for the other failure modes.

Another useful thing is examining importance of features on the model to see change on which feature highly affect the results. Figure 5 provides a visual representation of the various elements that contribute to the predictive power of a model, allowing us to discern which aspects carry the most weight in decision-making. At the foundation of this hierarchy are “Type_M” and “Type_L”, which are accorded minimal importance. Their influence on the model’s outcome is relatively marginal, suggesting that the product types, in this case, offer little in the way of predictive value.

Ascending the scale of importance, “Process Temperature” and “Air Temperature” emerge with a slight edge over the product types, indicating a modest contribution to predictive accuracy. “Temperature Power” takes a step further, offering a moderate enhancement to the model’s discernment. “Rotational Speed” and “Torque” closely follow, marking themselves as notable yet not dominant features within the predictive framework. The narrative changes as we consider “Temperature Difference”, which stands out with a higher degree of importance. This suggests that the

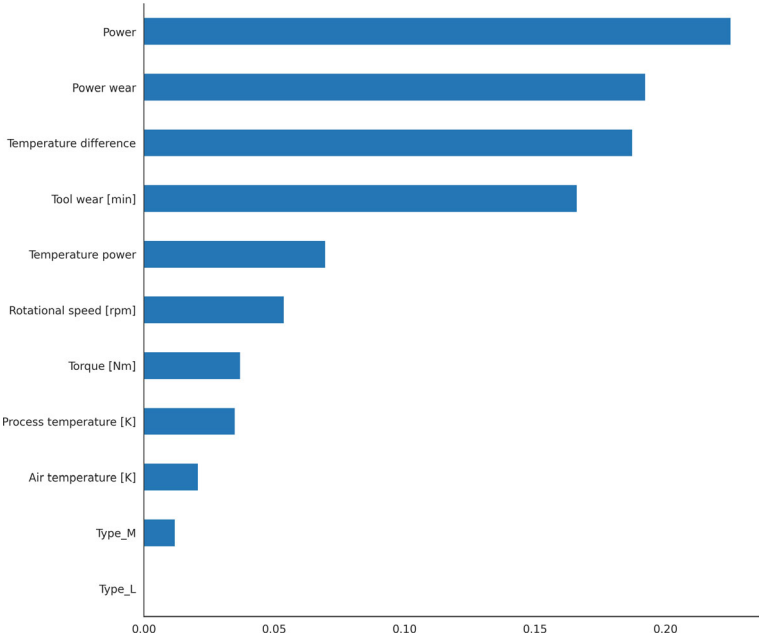


Fig. 5 Sorted importance of the features

model relies more heavily on the relative difference between temperatures rather than individual temperature readings. “Power” then assumes a pivotal role, substantially influencing the model’s predictions. It’s indicative of the model’s sensitivity to the energy dynamics of the system being studied. Above all, “Power Wear” commands the highest importance, signifying its role as the cornerstone of predictive insights. This feature, likely an interaction between the machine’s power and wear characteristics, is deemed most indicative of the outcomes the model aims to predict. “Tool Wear” similarly plays a significant role, just a notch below ‘Power Wear’, underscoring its relevance in assessing the machine’s condition. In essence, the figure tells a story of hierarchy and significance where “Power Wear” and “Tool Wear” are the protagonists, bearing the most substantial narrative weight in the machine’s operational tale. Conversely, the types of products linger in the background, contributing less to the plot of prediction. Through this lens, one gains not only an understanding of each feature’s role but also guidance on where to direct attention for maintenance strategies and further data analysis.

6 Benefits and Challenges

As we set the stage for an in-depth case study on predictive maintenance, it's crucial to understand the broader context in which these technologies operate within the manufacturing sector. The following analysis synthesizes the theoretical underpinnings with a practical exploration, offering insights into the tangible rewards and hurdles encountered when applying AI-driven PdM in a real manufacturing environment. In the current manufacturing environment, the incorporation of AI-driven predictive maintenance (PdM) is a significant development, bringing with it a host of benefits as well as a set of challenges that need careful navigation. This analysis aims to delve deeply into these aspects, offering insights into the rewards and obstacles associated with PdM, and discussing strategies to effectively manage them. A key advantage of AI-driven predictive maintenance is the enhancement of operational efficiency. PdM significantly reduces unplanned downtime and improves the availability of equipment. Predictive algorithms facilitate timely maintenance, ensuring that machinery operates at peak performance. This not only accelerates production rates but also improves the overall equipment effectiveness (OEE), a crucial metric in manufacturing, as noted in the research by Lee et al. [1]. Another significant benefit is cost reduction. By predicting equipment failures before they occur, PdM minimizes the expenses related to unplanned maintenance and emergency repairs. This proactive approach leads to more efficient use of resources, including labor and spare parts, and extends the lifespan of equipment, thereby reducing overall maintenance costs, a point highlighted in the study by Biggio and Kastanis [32]. Moreover, PdM contributes to safer work environments by anticipating equipment failures that could lead to accidents, ensuring the safety of the workforce and compliance with industry safety standards, thus reducing the risk of costly accidents and liability issues, as observed by Khan et al. [33].

However, the implementation of AI-driven predictive maintenance is not without its challenges. Technical complexities are a significant hurdle. Successful PdM deployment requires sophisticated data analytics capabilities and the integration of various technologies, including IoT and machine learning. Managing the quality, storage, and analysis of data can be particularly challenging, as discussed by Susto et al. [34]. Additionally, the initial financial investment for implementing PdM can be substantial. The costs associated with sensors, IoT infrastructure, data storage, and analytics software can be daunting, especially for small and medium-sized enterprises. Moreover, integrating PdM into existing operations demands significant organizational changes. There can be resistance to adopting new technologies, and training personnel to effectively use and interpret predictive maintenance data is a crucial yet challenging task, as indicated in the work by Darabnia and Demichela [35].

To overcome these challenges, a balanced approach is necessary. Strategic investment in PdM technologies, prioritizing high-value assets for early implementation, can help balance costs and benefits. Building technical expertise is also crucial. Collaborating with technology partners and investing in employee training are essential for overcoming technical challenges. Leveraging expertise from vendors and aca-

dem institutions can aid in the seamless integration of PdM technologies into existing systems. Furthermore, fostering a culture that embraces technological advancements is vital. Effective change management practices, clear communication of the benefits of PdM, and involving employees in the transition process can mitigate resistance to new technologies. Hence, AI-driven predictive maintenance offers clear benefits in enhancing operational efficiency, reducing costs, and improving safety in the manufacturing sector. However, the associated challenges, including technical, financial, and operational obstacles, require a balanced and strategic approach. Addressing these challenges through careful planning, skill development, and fostering an organizational culture receptive to technology will be key as industries advance toward more sophisticated manufacturing paradigms, ensuring their competitiveness and sustainability in the long run. Finally, this discussion on the potential and pitfalls of AI-driven PdM sets the foundation for a closer examination within our case study. The subsequent section will apply these concepts to an actual scenario, demonstrating how the theoretical benefits and challenges of PdM manifest in a practical setting. By observing PdM in action, we will see the direct impact of these strategies on operational efficiency, cost management, and safety compliance, and how the barriers to implementation can be navigated in a live manufacturing environment.

7 Future Trends and Developments

The future of AI in predictive maintenance (PdM) within the manufacturing sector is on the cusp of a transformative era, marked by the seamless integration of emerging technologies and a continuous evolution in maintenance practices. This exploration is not just a glimpse into the future but a deep dive into the predictions, emerging technologies, and the evolving landscape of manufacturing maintenance. At the forefront of this transformation is the advancement in machine learning and deep learning. These technologies are not just progressing; they are reshaping how we approach equipment failure predictions. This evolution is set to bring about a significant leap in the accuracy and nuance of predictive models [36]. Concurrently, the convergence with Industry 4.0 is deepening, paving the way for more efficient fault detection and diagnosis, particularly in vital components like induction motors. This synergy is not just an addition to PdM; it's a reinvention of its core capabilities.

Moreover, emerging technologies such as AI-powered anomalous sound detection are slated to play a pivotal role in smart maintenance. This is not just about detecting faults; it is about revolutionizing how these faults are identified - through automated detection that enhances the capabilities of factory automation [7]. Furthermore, the rise of digital twins is not just an advancement; it is a game-changer. By creating virtual replicas of physical systems, they offer a level of insight into equipment health and performance that was previously unattainable, making maintenance predictions more accurate and insightful, a transformation [37]. The impact of these emerging technologies is profound. Deep learning, for instance, is set to significantly

influence maintenance strategies. Its ability to handle complex datasets is not just an improvement; as noted in the literature it is a crucial development for advanced production systems [38]. Additionally, the incorporation of intelligent decision-making algorithms in response to failure predictions is expected to enhance the predictive maintenance process substantially [39]. Also, machine learning's role in fostering sustainable smart manufacturing is becoming central within the Industry 4.0 framework, automating fault detection and diagnosis, thereby minimizing downtime and increasing component utilization rates [3]. As we look at the transforming landscape of manufacturing maintenance, predictive maintenance in smart manufacturing is increasingly reliant on intelligent technologies like cyber-physical systems and big data analytics. The use of multi-agent systems and deep learning offers efficient solutions for PdM in smart industrial production systems, a forward-thinking approach underscored [36]. AI-based algorithms, particularly those utilizing machine learning and deep learning, are being extensively applied to manage the health status of industrial equipment, enhancing performance and reliability [1]. Additionally, the emergence of Smart and Intelligent Predictive Maintenance (SIPM) integrates intelligent sensors in smart factories, addressing contemporary challenges and future research directions which can be defines as a novel trend [11].

In conclusion, the trajectory of AI-driven predictive maintenance in the manufacturing sector is characterized by a journey toward technological sophistication and innovative application. The integration of advanced methodologies like machine learning, deep learning, digital twins, intelligent decision-making algorithms, and the concept of SIPM, will not only enhance predictive capabilities but also significantly contribute to the sustainability and efficiency of smart manufacturing practices.

References

1. Lee W, Wu H, Yun H, Kim H, Jun M, Sutherland J (2019) Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data. *Procedia CIRP*
2. Hrnjica B, Softic S (2020) Explainable AI in manufacturing: a predictive maintenance case study, pp 66–73
3. Cinar Z, Nuhu AA, Zeeshan Q, Korhan O, Asmael MBA, Safaei B (2020) Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability*
4. Liu Q, Dong M, Lv W, Ye C (2019) Manufacturing system maintenance based on dynamic programming model with prognostics information. *J Intell Manuf* 30:1155–1173
5. Resende C, Folgado D, Oliveira J, Franco B, Moreira W, Oliveira A, Cavaleiro A, Carvalho R (2021). *Tip4.0: Industrial internet of things platform for predictive maintenance*. Sensors (Basel, Switzerland) 21
6. Cooke FL (2003) Plant maintenance strategy: evidence from four British manufacturing firms. *J Qual Mainten Eng* 9(3):239–249
7. Mrazovac B, Ilian V, Hulea M (2021) Beyond industry 4.0: leveraging AI-powered anomalous sound detection for smart maintenance. In: 2021 Zooming innovation in consumer technologies conference (ZINC), pp 43–47

8. Kiangala KS, Wang Z (2020) An effective predictive maintenance framework for conveyor motors using dual time-series imaging and convolutional neural network in an industry 4.0 environment. *IEEE Access* 8:121033–121049
9. Friedrich C, Lechler A, Verl A (2014) Autonomous systems for maintenance tasks—requirements and design of a control architecture. *Procedia Technol.* 15:595–604. 2nd International conference on system-integrated intelligence: challenges for product and production engineering
10. Diaz-Elsayed N, Hernandez L, Rajamani R (2021) A framework for the health-ready characterization of manufacturing systems. *Manuf Lett*
11. Pecht M, Kang M (2019) Predictive maintenance in the IoT era, pp 589–612
12. Kaliyannan GV, Anbupalani MS, Kandasamy S, Sivaraj S, Gunasekaran R (2023) Role of IoT in industry predictive maintenance. In: *Integration of mechanical and manufacturing engineering with IoT: a digital transformation*, pp 191–213
13. Wang X, Liu M, Liu C, Ling L, Zhang X (2023) Data-driven and knowledge-based predictive maintenance method for industrial robots for the production stability of intelligent manufacturing. *Expert Syst Appl* 234:121136
14. Cho S, May G, Tourkogiorgis I, Pérez R, Lázaro Ó, de la Maza B, Kiritsis D (2018) A hybrid machine learning approach for predictive maintenance in smart factories of the future, pp 311–317
15. Zheng H, Paiva ARC, Gurciullo C (2020). Advancing from predictive maintenance to intelligent maintenance with AI and IIOT. [arXiv:abs/2009.00351](https://arxiv.org/abs/2009.00351)
16. Turner C, Emmanouilidis C, Tomiyama T, Tiwari A, Roy R (2019) Intelligent decision support for maintenance: an overview and future trends. *Int J Comput Integr Manuf* 32:936–959
17. Bose SK, Kar B, Roy M, Gopalakrishnan PK, Basu A (2018) Adepos: anomaly detection based power saving for predictive maintenance using edge computing. In: *Proceedings of the 24th Asia and South Pacific design automation conference*
18. Krishnamurthy V, Nezafati K, Stayton EL, Singh V. Explainable AI framework for imaging-based predictive maintenance for automotive applications and beyond. *Data-Enabled Disc Appl* 4:1–15
19. Lee J, Mitici M (2023) Deep reinforcement learning for predictive aircraft maintenance using probabilistic remaining-useful-life prognostics. *Reliab Eng Syst Saf* 230:108908
20. Gutschi C, Furian N, Suschnigg J, Neubacher D, Voessner S (2019) Log-based predictive maintenance in discrete parts manufacturing. *Procedia CIRP* 79:528–533
21. Rosati R, Romeo L, Cecchini G, Tonetto F, Viti P, Mancini A, Frontoni E (2023) From knowledge-based to big data analytic model: a novel IoT and machine learning based decision support system for predictive maintenance in industry 4.0. *J Intell Manuf* 34(1):107–121
22. Welte R, Estler M, Lucke D (2020) A method for implementation of machine learning solutions for predictive maintenance in small and medium sized enterprises. *Procedia CIRP* 93:909–914
23. Natanael D, Sutanto H (2022) Machine learning application using cost-effective components for predictive maintenance in industry: a tube filling machine case study. *J Manuf Mater Process* 6(5):108
24. Hurtado J, Salvati D, Semola R, Bosio M, Lomonaco V (2023) Continual learning for predictive maintenance: Overview and challenges. *Intell Syst Appl* 200251
25. Jayashree LS, Selvakumar G, Jayashree LS, Selvakumar G (2020) Enterprise IoT development platforms. *Getting started with enterprise Internet of Things: design approaches and software architecture models*, pp 129–142
26. Murugiah P, Muthuramalingam A, Anandamurugan S (2023) A design of predictive manufacturing system in IoT-assisted industry 4.0 using heuristic-derived deep learning. *Int J Commun Syst* 36(5):e5432
27. Nangia S, Makkar S, Hassan R (2020) Iot based predictive maintenance in manufacturing sector. In: *Proceedings of the International conference on innovative computing & communications (ICICC)*
28. Passlick J, Dreyer S, Olivotti D, Grützner L, Eilers D, Breitner MH (2021) Predictive maintenance as an internet of things enabled business model: a taxonomy. *Electron Mark* 31:67–87

29. Yeh C, Lin M-H, Lin C-H, Yu C-E, Chen M-J (2019) Machine learning for long cycle maintenance prediction of wind turbine. *Sensors* (Basel, Switzerland) 19
30. Traini E, Bruno G, D'Antonio G, Lombardi F (2019) Machine learning framework for predictive maintenance in milling. *IFAC-PapersOnLine*
31. Matzka S (2020) Explainable artificial intelligence for predictive maintenance applications. In: 2020 third international conference on artificial intelligence for industries (AI4I)
32. Biggio L, Kastanis I (2020) Prognostics and health management of industrial assets: current progress and road ahead. *Front Artif Intell* 3:578613
33. Khan M, Ahmad A, Sobieczky F, Pichler M, Moser BA, Bukovský I (2022) A systematic mapping study of predictive maintenance in SMES. *IEEE Access* 10:88738–88749
34. Susto GA, Schirru A, Pampuri S, Beghi A, De Nicolao G (2018) A hidden-gamma model-based filtering and prediction approach for monotonic health factors in manufacturing. *Control Eng Pract* 74:84–94
35. Darabnia B, Demichela M (2013) Data field for decision making in maintenance optimization: an opportunity for energy saving. *Chem Eng Trans* 33:367–372
36. Drakaki M, Karnavas YL, Tzifafettas IA, Linardos V, Tzionas P (2022) Machine learning and deep learning based methods toward industry 4.0 predictive maintenance in induction motors: state of the art survey. *J Ind Eng Manage (JIEM)* 15(1):31–57
37. Falekas G, Karlis A (2021) Digital twin in electrical machine control and predictive maintenance: state-of-the-art and future prospects. *Energies* 14(18):5933
38. Wang K, Wang Y, Strandhagen JO, Yu T (2018) *Advanced manufacturing and automation VII*, vol 451. Springer
39. Bousdekis A, Lepenioti K, Apostolou D, Mentzas G (2021) A review of data-driven decision-making methods for industry 4.0 maintenance applications. *Electronics* 10(7):828

Chapter 11

Artificial Intelligence in Production Systems and Integration with Lean Manufacturing



İsmet Söylemez and Ramazan Ünlü

1 Introduction

Lean is described as a set of management approaches that eliminate waste to improve efficiency and effectiveness. Lean's basic principle is to decrease and eliminate non-value-added operations and waste. According to the American Society for Quality [1], "Lean manufacturing, or lean production, is a system of techniques and activities for running a manufacturing or service operation. The techniques and activities differ according to the application at hand but they have the same underlying principle: the elimination of all non-value-adding activities and waste from the business". Value is the process that directly affects a product or service. Value-added activity can also be referred to as the part that the customer pays for. This is because when customers pay for the product, they pay for the activities that achieve the quality standards they expect in the product. However, some processes are unavoidable. These processes/activities may not directly change/affect the product (such as quality control). Processes that are necessary to complete the product but do not add value are called non-value-added activities. Some processes are neither value-adding nor non-value-adding. All activities that fall outside of these are called waste. Seven wastes are defects, overproduction, waiting, transportation, inventory, motion, over-processing [2]. Recently, an additional type of waste, "non-utilized talent/capability", has been considered as the eighth waste in lean production practices [3].

Quality defects refer to physical defects in the product. The customer does not accept these products. Additional time is spent fixing these defective products. In some cases, these defects cannot be corrected and are completely removed from production as scrap. Managers decide to overproduction more than demand to avoid

İ. Söylemez (✉) · R. Ünlü

Industrial Engineering Department, Abdullah Gul University, Kayseri, Turkey
e-mail: ismet.soylez@agu.edu.tr

R. Ünlü

e-mail: ramazan.unlu@agu.edu.tr

some variability or uncertainty. However, since no analytical solution method is used to determine how much more they will produce, there may be products in stock for a long time. There may be waiting for raw materials or finished products in enterprises. Semi-finished products are also waiting between processes. The product is transported between processes, or after completion, the finished product is transported to the finished product warehouse. Although this transport is considered, necessary transport is a major source of waste because it may be over-transported due to long routes. During the moving products may cause damaged, lost, broken, etc., products. All the money spent on raw materials or finished products are wasted by holding them in stock. The money tied up here could be used for other investments to be more competitive. There are also work-in processes (WIP) between machines/stations. For instance, the product has been cut and some of the cut parts have been processed in the next machine, while some of them are placed in an intermediate inventory area to be used later should also be considered waste. As compared to transportation, motion refers to the producer or worker. If there is more movement than the employee should do, this is also waste (i.e. if the employee bends twice when he/she could complete the process with a single bend). Activities that are performed for more quality operations than the customer expects are referred to as over-processing. In addition, it is also a waste not to utilize the talent/capabilities of an employee (engineer, worker, etc.) such as not allowing them to use them, or assigning them different and/or simpler tasks.

In the literature, it is seen that smart systems applied in manufacturing systems are named differently. In some studies, it is seen as smart manufacturing [4] and in others as advanced manufacturing [5]. Technologies connected to smart manufacturing were categorized into the past, present and future, along with recommendations for future development paths. A survey was carried out, primarily based on publications viewed from the standpoint of smart manufacturing or the eight associated technologies used in production [4]. The idea of intelligent manufacturing is a broad one that is always evolving. Three fundamental paradigms can be distinguished within intelligent manufacturing: new-generation intelligent manufacturing, digital-networked manufacturing and digital manufacturing. Advanced manufacturing technology and new-generation AI technologies are deeply integrated into new-generation intelligent manufacturing [5]. AI has the potential to significantly advance sustainable development. The expectations are predicated on the utilization of a few crucial AI-related skills. These are skills related to computer vision, such as image recognition and classification, or natural language processing, such as speech recognition and machine translation [6]. AI can be used in lean manufacturing applications where such information needs to be utilized.

ML algorithms can be classified in reinforcement learning, supervised, unsupervised and deep learning algorithms [7]. There are a lot of papers in the field of manufacturing [8–10]. Hybrid machine learning approach combining unsupervised learning and semi-supervised learning methods and novel predictive maintenance approach is provided [11]. For the clustering and classification, both optimization and machine learning techniques are presented as a systematic review by Azevedo

et al. [12]. Moreover, machine learning-based crash prediction models are searched as a systematic review by Yasir et al. [13].

At the first stage, it would be useful to first present the similarities and differences between traditional production and lean production. Traditional manufacturing systems have the following characteristics [14]:

1. During the production process, if a process is working, don't fix it. You should complete the process where necessary and intervene at the end.
2. Standardized work (people performing the same task the same way and duration) only exists in documents like standard operation procedures (SOPs), rarely in reality. You have to comply with how and how long it is defined in this SOP document.
3. Focuses on training and relies on people to not make mistakes. Trainings are planned and all staff are trained. They are also trusted that employees will no longer make mistakes.
4. Systems thinking (views the organization as a whole), often ignoring or unable to see the enormous opportunities for improvement. Managers consider whether there are prospects for whole system improvement.

In contrast to traditional manufacturing, lean manufacturing (LM) has the following characteristics:

1. Always look for ways to improve processes. Even if there is a product in production, the system is stopped and an on-site inspection (as called Gemba at LM literature) is carried out. As a team (responsible managers, supervisor, foreman and operator), the points to be improved are identified. If immediate improvement is possible, it is implemented; if a certain time is required, it is planned for later. In lean manufacturing, it is even recommended to organize a suggestion system for identifying improvements. In this way, improvements can be realized not only by managers but also by those responsible operators for production.
2. Everyone performs the same task the exact same way until a better way is discovered; then everyone performs the task the new and improved way. Before starting the standard operation procedure preparation process, all the points where the process can be improved are studied. Then the standard operation procedure is prepared for the improved system.
3. Focuses on building processes that are error proofed (a person cannot make a mistake or it would be difficult to do so). It is critical to develop employees through training. However, along with training in lean manufacturing, it argues that instead of relying on the employee, a system should be established to prevent him/her from making mistakes.
4. Views the organization as a series of interrelated processes that can and should be improved. From the most basic production stage (such as drilling a piece of wood on a CNC drilling machine) to the entire production system (such as all planning and/or purchasing processes), the focus is on improvement in every stage/area.

LM studies are generally in the form of case studies in the literature. Theoretical studies are almost non-existent. But different applications have been presented

for different systems. Some literature studies are summarized as follows. Manurung et al. [15] presented articles from 2017 to 2022 applying the lean manufacturing technique. Kumar et al. [16] focused on studies which are integration of LM and Industry 4.0. Authors presented a systematic literature review. Moreover, another article approaches the issues of LM and Industry 4.0 in a novel way, measuring the adoption of its technology, methodologies and tools and presenting it in network visualizations [17]. The following article critically reviews lean manufacturing and reverse engineering for joint propeller shafts, covering definitions, design, organization, materials and tools. It analyses LM concepts, wastes, strategies, obstacles and implementation, focusing on how lean tools and techniques contribute to market reputation and performance improvements [18].

In this chapter, the relationship between artificial intelligence (AI) and lean manufacturing (LM) aims to make manufacturing processes more efficient, flexible and customer-centring. AI offers the potential to optimize production processes and solve problems using features such as data analysis, machine learning algorithms and recursion capabilities. After this stage, some lean manufacturing techniques and their relationship with artificial intelligence will be detailed.

2 Methodology

At this stage of the study, some LM techniques and their definitions will be mentioned. Real-life sample applications are also available in this section. An overview of the relationship between AI and LM is given in Fig. 1.

2.1 *Kaizen*

There are many different definitions of Kaizen. Some of them are as follows. Looking back at early Japanese literature, Imai defines Kaizen as “a culture of ongoing improvement involving everyone-top management, managers and workers” [19]. Davis defines Kaizen as “a process aimed at making continuous improvement, by focusing on and eliminating wastes inherent to the manufacturing processes being employed” [20]. Coimbra defines Kaizen as “a long-term strategy that systematically seeks to achieve small, incremental changes in processes in order to improve efficiency and quality” [21]. Abdulmouti defines Kaizen as a “Productivity tool that emphasizes continuous improvement through regular, incremental improvement” [22]. In summary, Kaizen can be expressed as continuous improvement or change for the better. Improvements can sometimes be realized easily by taking action quickly. These are called small Kaizen. In some situations, a long time and a team of people need to work in a planned way to apply improvements. These kinds of Kaizens are called such as kobatsu Kaizen, Kaizen in 8 or 10 steps.

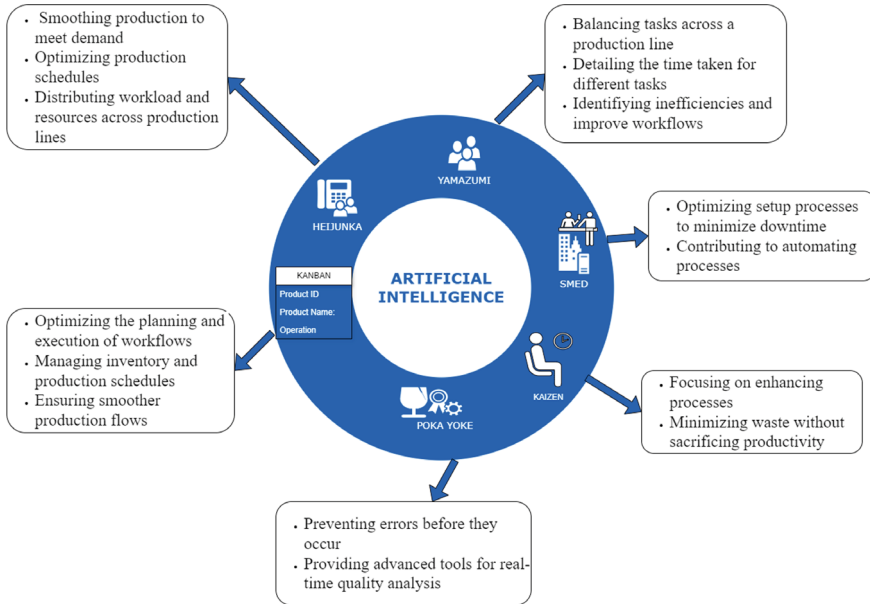


Fig. 1 An overview of the relationship between AI and LM

Kaizen practices are usually planned by white-collar employees. However, the observation of the points to be improved goes to the actual area where the production is carried out or the service is provided. This activity is called Gemba in LM literature. The whole team (white-collar engineers, managers and employees) goes to the real area, and the problem is identified. Then root cause of the problem is defined, and the most suitable improvements are decided. Improvements are made either immediately (small Kaizen) or later in a planned manner (large Kaizen).

In the manufacturing sector, it is possible to see more Kaizen applications. Improvements can be carried out to shorten the time spent in the process of placing raw materials or finished products in the warehouse. You can also see the usage of equipment used to assemble products on the assembly line with the help of an overhead suspension system. This way, the employee can quickly pick up and use the equipment that will be used throughout the day, instead of constantly leaving it on the belt and picking it back up. However, Kaizens can be applied not only in the production sector but also in the service sector. There are real-life applications that will shorten and facilitate the time spent by nurses in a hospital to prepare the syringe during the blood collection process for patients.

A sample system is given in Fig. 2. In a system with two operators (Opt 1 and Opt 2), the second operator leaves the equipment she/he uses on the line while assembling the products. There may be a loss of time constantly taking the equipment and returning it to the table throughout the day. However, thanks to the hanger system, the first operator leaves the equipment when the operation is completed, instead of

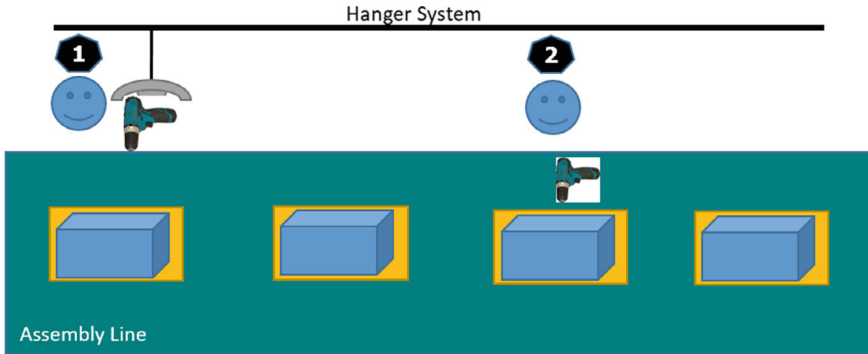


Fig. 2 Example of Kaizen

leaving it on the line, and the equipment remains suspended in the air. The operator eliminates the loss of time due to pick-up and drop-off. In addition, the weight of the equipment he/she will use throughout the day is loaded on the hanger system instead of the handle. Thanks to this Kaizen, the first operator can work more efficiently.

Kaizens can be identified in different departments such as quality control, maintenance and production planning. Following are some examples of potential Kaizens in different departments, and how machine learning techniques can be applied. In the production line, it is difficult to understand why there are deviations at a certain stage in the production process. AI algorithms can analyse historical production data and identify abnormal situations in processes. They can also predict future production performance based on this data. In quality control, quality issues are recurring frequently on the production line. AI can be used to optimize quality control processes on the production line. Visual recognition algorithms can be used to identify and reject defective products. Processes are interrupted due to unexpected machine breakdowns during the production process. Using AI algorithms, AI can be analysed machine data and predict potential failures of a specific machine. In this way, planned maintenance processes can be optimized in the maintenance department. Productivity in production processes may be low. By analysing data from production processes, AI can identify opportunities for improvement. This can be used to optimize process parameters, adjust speeds and reduce waste. Finally, an example for the human resources management department could be as follows. Staff are experiencing productivity losses due to specific skill deficiencies. By analysing staff performance data, the AI can identify specific training needs. In this way, customized training programmes can be created to improve staff skills.

2.2 *Heijunka*

Heijunka is another important concept in the principles of lean manufacturing. Heijunka aims to smooth out fluctuations in the production process and make production activities more regular. The word “Heijunka” is of Japanese origin and means “levelling” [23, 24]. Heijunka is used especially to meet fluctuations in customer demand and to stabilize variations in the production process. The concept is based on correcting irregularities in production and making processes more stable. Some of Heijunka’s main principles are as follows:

- **Production Smoothing:** Heijunka focuses on reducing fluctuations of production volume. This aims for a continuous and constant production rate and thus smooths out variations in the supply chain.
- **Steady Demand Fulfilment:** Heijunka adjusts the production process to meet customer demands. This can increase the ability to respond more flexibly to sudden changes in demand.
- **Kanban Utilization:** Heijunka incorporates the Kanban system. Kanban is a management system that organizes the production process and controls the flow of materials. This system is used to ensure a continuous flow of materials and to keep inventory levels low. This topic will be explained in detail in the Kanban section.
- **Balancing Workstations:** Heijunka aims to balance the workload between workstations. Thus, it minimizes variations in the production process and increases efficiency.

Heijunka’s aim is to reduce the effects of fluctuations in the production process, lower inventory levels and meet customer demands more consistently. This can help to create a production system that is in line with the principles of lean manufacturing in general.

Heijunka is levelling at planning systems. Actually, levelling is applied in planning department. It means planning in smaller batches in the production plan. It can be understood more clearly with the example given. Let’s assume that five different products are produced in a production company. These products have different demand quantities. Generally, set-up times are required before the production of products begins. Managers do not want to switch to the production of another product before completing the entire product order in order to minimize the loss of time that will occur due to setup time. That is, first the entire order of type A product is completed. Then we move on to other products. Unlike traditional production, lean manufacturing involves production in small batches. That is, instead of completing the entire order of product A, a part of it is completed and moved on to other products. In this way, production of different products can be achieved in short periods by making small production plans. In order to avoid more setup time, first of all, improvements are made to minimize the setup time. In Table 1a, there is a traditional production schedule of one day (540 min). After all the demands for the products are completed, we move on to the next product. However, Table 1b demonstrates the system with a production plan in smaller batches.

Table 1 Product Demand Distribution

Product Types		Total Demand	Time Periods									
			60	120	180	240	300	360	420	480	540	
A	20	10	10									
B	30			10	10	10						
C	10						10					
D	20							10	10			
E	10									10		

Table 1b

Product Types	Total Demand	Time Periods														
		30	60	90	120	150	180	210	240	270	300	330	360	390	420	450
A	20	5			5						5					
B	30		5		5			5				5				
C	10		5	5										5		
D	20			5			5		5				5			
E	10									5						

In order to realize Heijunka applications, demand forecasting must first be done appropriately and accurately. Some examples of Heijunka applications of machine learning are as follows. Demand fluctuations cause production disruptions and inventory problems. AI can generate future demand forecasts by analysing historical sales data and other relevant factors. This can be used to align production levels with demand. Fluctuations in production processes affect machine or worker capacities, reducing productivity. AI algorithms can predict fluctuations in the production flow and generate a more homogeneous production plan in a given period. This can be used to optimize capacity planning and regulate worker/machine utilization. By analysing stock levels and order history, AI can determine optimal stock levels. In this way, stock waste can be reduced and orders can be managed more accurately. Moreover, imbalances on the production line are causing overloads at certain workstations and capacity gaps at others. AI can analyse fluctuations on the production line and optimize workstations to use them in a more balanced way. This can make every stage of the process run more efficiently. AI can add value to Heijunka applications in data analysis, forecasting and optimization. However, for this technology to be applied effectively, it is important to focus on the right data sets, the selection of algorithms and the principles of continuous improvement.

2.3 *Yamazumi*

Yamazumi is a term used in lean manufacturing techniques and is used specifically for the organization of workstations and workload equalization. Yamazumi is a Japanese term meaning “load balancing” or “load equalization”. The Yamazumi is an approach for visualizing time data of specified actions in an investigated process [25]. Yamazumi analysis aims to balance the workload of workstations on the production line. This aims to optimize the amount of work at each workstation and thus minimize time losses on the production line. This technique is aimed at eliminating inequalities in the production process and increasing efficiency. Yamazumi analysis is used to identify and correct imbalances between workstations. Thus, the workload at all workstations is optimally distributed, reducing waiting times and waste in the production process. This allows the production process to be faster and more efficient [26]. Let’s take an automotive assembly line. Vehicle parts are assembled at different workstations of this line. However, some stations spend more time than others, leading to imbalances across the line. Yamazumi analysis focuses on balancing the workload on this assembly line. First, the time spent at each workstation is determined. Then, appropriate adjustments are made to equalize these times. For example, if one station has a longer processing time than the others, additional workers can be assigned to that station or unnecessary steps in the process can be eliminated to reduce the time. Thus, the workload is equalized and time losses on the production line are minimized. Through Yamazumi analysis, it is possible to eliminate imbalances on the production line, reduce time differences between work-

stations and increase productivity. This ultimately contributes to a faster and more cost-effective production process.

“How the artificial intelligence may be used in the Yamazumi application?” This question is gaining importance due to the increasing popularity of artificial intelligence. Let’s take an automotive production line where vehicle doors are assembled. Let’s consider the use of artificial intelligence to balance between workstations:

- **Data Analytics:** By analysing historical production data, AI can determine the average times of door assembly operations at each workstation. This analysis can reveal which stations are spending more time and the need for balancing.
- **Optimization Algorithms:** AI can use mathematical optimization algorithms to balance times between workstations. By optimizing the workload, these algorithms can distribute the work times at each station in the best possible way.
- **Continuous Improvement:** By monitoring production data in real time and evaluating performance metrics, AI can make recommendations for continuous improvements. For example, if a station is consistently spending more time on a particular process, it can report this to the operators and suggest correction strategies.
- **Learning Algorithms:** AI can make future workload predictions by analysing data changes and trends over time. Thus, it can better adapt to future changes to balance between workstations.

In this example, AI can be used in areas such as data analysis, optimization, continuous improvement and learning in the Yamazumi analysis process to more effectively balance the workload on the production line. In the given example (see in Table 2), the product is produced by three operators (Opt 1, Opt 2 and Opt 3). The operations shown in yellow are denoted by non-value-added activity (NVA), in green by value-added activity (VA) and in red by Muda (M). The total workloads of each operator are 49 s, 70 s and 42 s, respectively. In other words, a product can be produced in every 70 s. When we analyse the red-coloured operations, it can be said that there are inefficiencies. The first operator has 16.32% wastage, and the second operator has around 20% wastage. These wastes should be focused on first, and wastes should be tried to be eliminated. At the same time, improvements can be made even in the yellow-coloured processes, which are expressed as mandatory wastes. In this way, productivity can be increased even more. Even if only wastes are removed, a product can be produced in 56 s (i.e. productivity can be increased by 20%). Even a more balanced production can be achieved by shifting some tasks of the second operator to the previous or next operator. For example, if the task “Tighten the 4th wheel” is transferred to the following operator, productivity will be further increased. Because in the new line balance the total workloads are 41, 48 and 52, respectively. The productivity gain would be 25.71% (Fig. 3).

AI can be used for workload estimation or coordination between workers and also employee talent and performance analysis. For the Yamazumi applications, difficult to balance in the process due to workload fluctuations between workstations. AI algorithms can predict future workload by analysing historical workload data. This can help to plan workstations in a more balanced way over time.

Table 2 Detailed operator activities

Operator	Type	Verb	Detail	Time (s)
Opr 1	NVA	Pick up	Pick up the subframe	2
Opr 1	NVA	Pick up	Pick up 2 stamps and put them on	3
Opr 1	NVA	Put	Pick and place 2pcs wheels	2
Opr 1	NVA	Fix	Pick up 2 screws and fix it	7
Opr 1	VA	Tighten	Tighten the 1st wheel	12
Opr 1	VA	Tighten	Tighten the 2nd wheel	15
Opr 1	M	Push	Give subframe to the 2nd operator	8
Opr 2	NVA	Rotate	Rotate the subframe	8
Opr 2	NVA	Take	Pick up 2 stamps and put them on	4
Opr 2	NVA	Put	Pick and place 2pcs casters	5
Opr 2	NVA	Fix	Pick up 2 screws and fix it	2
Opr 2	VA	Tighten	Tighten the 3rd wheel	27
Opr 2	VA	Tighten	Tighten the 4th wheel	10
Opr 2	M	Drag	Drag the finished subframe aside	8
Opr 2	M	Turn back	Turn back for the new product	6
Opr 3	NVA	Take	Pick up the box	4
Opr 3	NVA	Take	Pick up the product	9
Opr 3	VA	Place	Place in the bin	22
Opr 3	NVA	Control	Quality control	2
Opr 3	NVA	Take	Take the bin and put on the pallet	5

Workload imbalance between workstations reduces productivity. AI algorithms can be used to optimize communication and collaboration between workstations. Workload forecasts can be communicated to workstations appropriately, enabling more effective coordination. Worker skill levels and performance may be uneven. By analysing worker skill and performance data, AI can assess workers' abilities and performance. AI can help to more appropriately assign workers to workstations and better balance the workload. By tracking and analysing instant workload changes, AI can provide solutions that can be reacted to quickly. This is especially important when there are sudden changes in demand or unexpected situations. In Yamazumi applications, AI can be used to optimize workload on workstations, increase efficiency and minimize imbalances.

2.4 Kanban

One of the basic principles of lean manufacturing is also Kanban. Kanban is a Japanese word meaning “visual card” or “visual sheet” [27, 28]. It is a management method used in lean manufacturing systems to make the production process

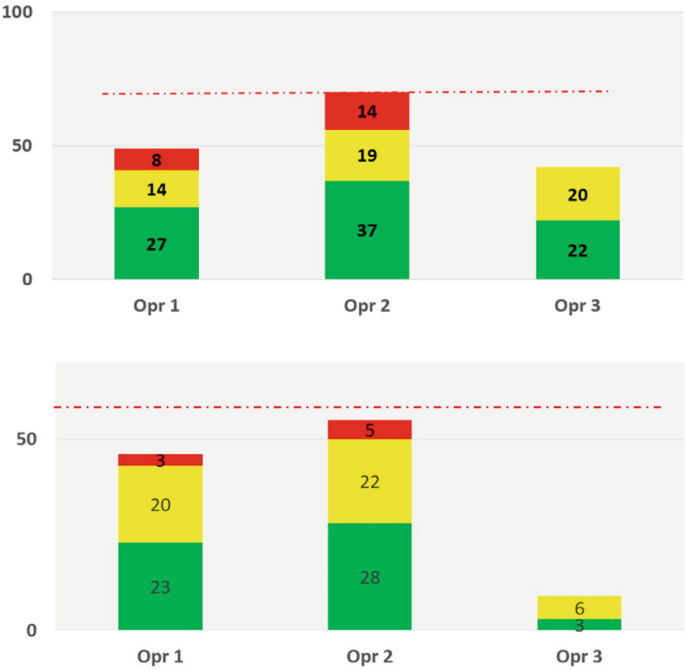


Fig. 3 Example of Yamazumi

more effective and efficient. The Kanban system aims to organize, control and optimize the flow of materials at a workstation or in a production process. Basically, thanks to this system, a workstation receives a signal before receiving materials or products from the previous station. A Kanban system usually includes the following steps:

- Demand Determination: Each workstation determines its need for materials or products and expresses this need with a Kanban card.
- Kanban Card: A Kanban card is a visual card containing information about the material or product. This card contains the quantity requested, predetermined specifications and other relevant information.
- Request Signal: When the first workstation has produced the requested material or product, it communicates this request to the next workstation with a Kanban card. This is a signal that the material or product has been requested.
- Material Flow: When the next workstation receives the Kanban card, it replaces it with the represented material or product and starts the process. In this way, the material flow and the production process continue in a continuous and balanced manner.

The Kanban system provides advantages such as keeping stock levels low, preventing imbalances between workstations, optimizing production processes and responding

to customer demands more quickly and flexibly [29]. Kanban is generally divided into two main types [27].

1. Signal Kanban (Production Flow Signal Kanban): System sends signal cards to the manufacturer of the materials a workstation needs, indicating how much of which material is required. If the material has run out or dropped to a certain level, these cards signal that new material must be produced or procured.
2. Production Kanban: System be used to organize the flow of material between workstations. A workstation receives a Kanban card showing the materials needed to start its production.

These classifications are used to adapt the basic principles of Kanban to specific use cases. Signal Kanban plays an important role in regulating material flow and controlling stock levels, while production Kanban is intended to regulate and optimize the workflow between workstations.

In the furniture industry, Kanban can provide a number of advantages and make business processes more effective. Here are some examples of the purposes of Kanban in this industry:

- Order Tracking and Forward Planning: The Kanban system allows to visually track each order in the furniture production process. This allows the business to see more clearly which orders it is working on at which stage. At the same time, it can be used to better plan future orders.
- Stock Control: Kanban cards contain information about the status of each material and product. In this way, stock levels can be visually monitored and unnecessary stock build-ups can be avoided.
- Improving Production Flow: Kanban can help to better organize the workflow between workstations. Since each workstation has a specific capacity, it prevents overloads and increases efficiency.
- Quick Response and Flexibility: The ability to respond quickly to changes in customer demands is another advantage of using Kanban in the furniture industry. A flexible production process makes it possible to adapt more quickly to customer needs.

Kanban applications in the furniture industry can be used to optimize business processes, respond better to customer demands, keep inventory levels under control and increase overall efficiency. This can bring significant benefits in terms of both production and customer satisfaction. The Kanban system manages material demands based on certain rules. However, stock levels can fluctuate due to unexpected demand fluctuations. AI algorithms can predict future demands by analysing historical demand data. This can be used to better plan stock levels and respond to demand fluctuations more effectively. Moreover, the Kanban system aims to minimize waste by keeping stock levels at a certain point. However, there can be difficulties in determining optimal stock levels. AI learning algorithms can optimize inventory levels by analysing supply chain data and demand patterns. This can reduce costs by reducing waste. The production rate can vary depending on demand fluctuations or capacity issues at certain workstations. AI learning algorithms can help to balance workstations more

effectively by analysing production rate data. This enables dynamic adjustment of production speed according to demand.

2.5 *Single Minute Exchange of Dies (SMED)*

SMED stands for “Single-Minute Exchange of Die”. This method is one of the other tools within lean manufacturing principles that are used to increase production speed, reduce inventories and improve productivity [30]. The main goal of SMED is to minimize changeover times (tool changeover times) between machines or workstations. This enables small batches to be produced more frequently in mass production and the adoption of flexible production processes. The main elements of SMED are [31]:

- **Internal and External Tool Change:** It divides tool change processes into two main categories, “internal” and “external”. Internal tool change refers to operations that can be performed while the machine is running, while external tool change refers to operations that require machines to be stopped.
- **Parallel Processes:** Intrinsic tool change uses parallel operations to perform part of the processes while the machine is running and the other part when the machine is stopped. This is a strategy to reduce the total tool change time.
- **Standardization:** Standardizing tool change processes ensures that the same steps and processes are applied for each change. This reduces inconsistencies and errors in the process.
- **Work Units and Layout:** Organized placement of tools and materials ensures that the necessary tools can be found quickly. This makes tools easily accessible and processes faster.

SMED is an important tool, especially in production processes that need to cope with short production runs and rapidly changing customer demands. This method is frequently used to optimize total production processes and increase production flexibility. Let's look at an example of the SMED method from a manufacturing sector: For example, an automotive manufacturing plant has a production line where different car models are produced. Machine set-ups (tool changes) are required for the production of each model, and these set-up times can affect the total production time. The following steps can be followed when implementing SMED. Firstly, identifying internal and external tool change. Internal tool change is to identify operations that can be performed while the machine is running. For example, changing a part on a workstation or making adjustments while the machine is running. External tool change is identifying operations that must be performed when the machine stops. For example, physically changing a mould. Secondly, determine the steps that can carry out internal and external tool change processes in parallel. By prioritizing, identifying critical steps and trying to accelerate these steps with priority. Lastly, standardizing tool change processes. Ensuring that the same steps and processes are applied in each change. Standardization ensures that the process is consistent and fast.

Estimating and planning changeover times can be difficult, especially depending on variable factors. By analysing past changeover time data and relevant factors, machine learning can predict future changeover times. This can be used to plan changeover times more effectively with AI. Furthermore, it can be difficult to understand which factors affect changeover times and where improvements need to be made to increase efficiency. By analysing factors associated with changeover times and productivity indicators, AI algorithms can identify which factors have the most impact. This can be used to identify opportunities for improvement. In addition to this, AI can be used to better understand the factors that influence changeover times and their interactions. Thus, it can create an optimized changeover plan and reduce waste in the process. Lastly, when abnormalities or delays in changeover times are not identified, problems can occur in the process. AI can identify abnormal situations by analysing historical changeover times and transaction data. In this way, early detection and intervention of abnormal situations can be possible.

2.6 *Poka-Yoke*

Poka-Yoke is a Japanese term meaning “error prevention” or “mistake prevention”. It is a technique used in lean production and quality management [32]. Its purpose can be summarized as preventing or correcting errors and inaccuracies before they occur in processes. Poka-Yoke aims to minimize the possibility of operators or workers making mistakes. Poka-Yoke techniques usually involve simple and low-cost solutions. These solutions are designed to prevent errors such as incorrect assembly of a process or a product, being in the wrong position or taking a wrong step. In this way, it is possible to improve quality and keep errors to a minimum [33].

It is possible to understand Poka-Yoke through a few examples. On an assembly line, two identical parts are mixed between different models. A mark or label of a specific colour or shape is placed on each part, allowing workers to select the correct part. Another example, USB devices can often be damaged by trying to plug them in the wrong way. The USB connector can be designed so that it can only be plugged in the correct direction. This way, users cannot make the mistake of plugging in the wrong direction. These examples show how Poka-Yoke techniques are designed to prevent errors. Poka-Yoke can have various applications in production processes and everyday life and is used as a measure to improve quality by preventing errors in advance.

Manual detection of errors in visual inspection processes can be difficult. AI can automatically detect defective products using visual recognition algorithms. This speeds up quality control processes and makes them more precise. Defects can occur when abnormal conditions that occur during the process are not detected. By analysing process data, AI can identify normal operating conditions and detect deviations from these conditions. This can be used to identify abnormal conditions at an

early stage. Lastly, defects in production equipment cannot be predicted in advance. By analysing sensor data of production equipment, machine learning can predict potential fault conditions in advance. This is valuable for planned maintenance and error prevention.

3 Conclusion and Future Works

This study presented the growing popularity of artificial intelligence (AI) in the production sector, driven by technological advancements. AI is crucial for faster and more effective problem-solving during production, enabling both solutions to current issues and proactive prevention of potential problems. Lean manufacturing, pioneered by Japanese companies, focuses on efficiency and waste reduction, employing methods like Kaizen, Heijunka and Kanban. AI enhances lean manufacturing by streamlining processes and improving usability, aiming to demonstrate its effectiveness in production systems and discussing potential contributions. After reaching a certain stage, the use of artificial intelligence techniques will accelerate the processes. With the help of artificial intelligence techniques, continuous improvement studies can be carried out more easily and quickly.

In this study, some widely used lean technique methods (such as Kaizen, Heijunka, Yamazumi, SMED, Kanban and Poka-Yoke) are mentioned. General information about these methods is mentioned. In addition, a general study on how artificial intelligence techniques can be used is presented. In particular, this study will be the first step for many studies that can be done in the future. Especially for large-scale companies, the use of artificial intelligence techniques may be in question. Lastly, application studies will make advantage of artificial intelligence.

References

1. American Society for Quality. ASQ (2024)
2. El-Namrouty KA, Abushaaban MS (2013) Seven wastes elimination targeted by lean manufacturing case study Gaza strip manufacturing firms. *Int J Econ Finance Manage Sci* 1(2):68–80
3. Abdel-Aty TA, McFarlane D, Brooks S, Salter L, Abubakar AS, Evans S, Hawkrigide G, Thomas KP, Negri E, Mukherjee A et al (2024) The role of low-cost digital solutions in supporting industrial sustainability. *Sustainability* 16(3):1301
4. Kang HS, Lee JY, Choi S, Kim H, Park JH, Son JY, Kim BH, Noh SD (2016) Smart manufacturing: past research, present findings, and future directions. *Int J Precis Eng Manuf Green Technol* 3:111–128
5. Zhou J, Li P, Zhou Y, Wang B, Zang J, Meng L (2018) Toward new-generation intelligent manufacturing. *Engineering* 4(1):11–20
6. Buchmeister B, Palcic I, Ojstersek R (2019) Artificial intelligence in manufacturing companies and broader: an overview. In: DAAAM International scientific book, pp 81–98
7. Wuest T, Weimer D, Irgens C, Thoben K-D (2016) Machine learning in manufacturing: advantages, challenges, and applications. *Prod Manuf Res* 4(1):23–45

8. Pham DT, Afify AA (2005) Machine-learning techniques and their applications in manufacturing. *Proc Inst Mech Eng Part B: J Eng Manuf* 219(5):395–412
9. Susto GA, Schirru A, Pampuri S, McLoone S, Beghi A (2014) Machine learning for predictive maintenance: a multiple classifier approach. *IEEE Trans Ind Inf* 11(3):812–820
10. Hansson K, Yella S, Dougherty M, Fleyeh H (2016) Machine learning algorithms in heavy process manufacturing. *Am J Intell Syst* 6(1):1–13
11. Cho S, May G, Tourkogiorgis I, Perez R, Lazaro O, de La Maza B, Kiritsis D (2018) A hybrid machine learning approach for predictive maintenance in smart factories of the future. In: *Advances in production management systems. Smart manufacturing for industry 4.0: IFIP WG 5.7 International conference, APMS 2018, Seoul, Korea, August 26–30, 2018. Proceedings, Part II*. Springer, pp 311–317
12. Azevedo BE, Rocha AMAC, Pereira AI (2024) Hybrid approaches to optimization and machine learning methods: a systematic literature review. *Mach Learn* 1–43
13. Ali Y, Hussain F, Mazharul Haque Md (2024) Advances, challenges, and future research needs in machine learning-based crash prediction models: a systematic review. *Accid Anal Prev* 194:107378
14. Tarkan K, Aksungur S (2016) Comparative analysis of traditional production systems with flexible manufacturing systems. *Int J Appl Math Electron Comput (Special Issue 1)*:374–378
15. Martin M, Anizar A, Ishak A (2024) Implementation of lean manufacturing methodology and its application: a literature review. *Jurnal Sistem Teknik Industri* 26(1):35–46
16. Kumar N, Singh A, Gupta S, Kaswan MS, Singh M (2024) Integration of lean manufacturing and industry 4.0: a bibliometric analysis. *TQM J* 36(1):244–264
17. Vargas GB, de Oliveira Gomes J, Vallejos RV (2024) A framework for the prioritization of industry 4.0 and lean manufacturing technologies based on network theory. *J Manuf Technol Manage* 35(1):95–118
18. Deshmukh M, Gangele A, Gope DK, Dewangan S (2022) Study and implementation of lean manufacturing strategies: a literature review. *Mater Today: Proc* 62:1489–1495
19. Imai M (1986) *Kaizen-the key to Japan's competitive success*. Kaizen Institute, London
20. Davis JW (2011) *Progressive Kaizen: the key to gaining a global competitive advantage*. CRC Press
21. Coimbra EA (2013) *Kaizen in logistics and supply chains*
22. Abdulmouti H (2015) The role of Kaizen (continuous improvement) in improving companies' performance: a case study. In: *2015 International conference on industrial engineering and operations management (IEOM)*. IEEE, pp 1–6
23. Boutbagha M, El Abbadi L (2024) Heijunka-levelling customer orders: a systematic literature review. *Int J Prod Manage Eng* 12(1):31–41
24. Bohnen F, Buhl M, Deuse J (2013) Systematic procedure for leveling of low volume and high mix production. *CIRP J Manuf Sci Technol* 6(1):53–58
25. Szabó P, Janík S, Măkva M, Czifra G (2024) Waste elimination in the assembly process using lean tools. *Acta Polytechnica Hungarica* 21(4)
26. Talapatra S, Sharif-Al-Mahmud ZZZ, Kabir I (2018) Overall efficiency improvement of a production line by using Yamazumi chart: a case study. In: *Proceedings of the International conference on industrial engineering and operations management, vol 1*, p 3166
27. Junior ML, Filho MG (2010) Variations of the Kanban system: literature review and classification. *Int J Prod Econ* 125(1):13–21
28. De La Cruz H, Altamirano E, del Carpio C (2020) Lean model to reduce picking time delays through Heijunka, Kanban, 5s and Jit in the construction sector. In: *Proceedings of the LAC-CEI International multi-conference for engineering, education and technology*. Buenos Aires, Argentina, pp 27–31
29. Sendil Kumar C, Panneerselvam R (2007) Literature review of Jit-Kanban system. *Int J Adv Manuf Technol* 32:393–408
30. Moreira AC, Pais GCS (2011) Single minute exchange of die: a case study implementation. *J Technol Manage Innov* 6(1):129–146

31. Da Silva IB, Filho MG (2019) Single-minute exchange of die (SMED): a state-of-the-art literature review. *Int J Adv Manuf Technol* 102:4289–4307
32. Zhang A (2014) Quality improvement through Poka-Yoke: from engineering design to information system design. *Int J Six Sigma Compet Advantage* 8(2):147–159
33. Saurin TA, Duarte Ribeiro JL, Vidor G (2012) A framework for assessing poka-yoke devices. *J Manuf Syst* 31(3):358–366

Chapter 12

A Multiobjective Metaheuristic Algorithm Perspective to Design Riesz Digital Differentiator Exhibiting Zero-Phase Response



Chandan Nayak

1 Introduction

In recent years, there has been a notable surge in interest in the investigation of employing evolutionary algorithms to realise digital differentiators [1]. All the researchers have considered the design of digital differentiator as a single-objective optimisation problem and aimed to minimise the magnitude response error [1–3]. The eventual aim of a single-objective optimisation is to discover the optimum result that fulfils the problem's requirements, which leads to the minimisation or maximisation of the objective function under the maintained constraints. However, in the digital differentiator-type optimisation problems, there are two distinct objectives: magnitude response error and phase response error, which must be optimised simultaneously. Neglecting to account for the phase response in the cost function of the digital differentiator-type optimisation problem may result in distorted output and dependency on prior knowledge [4]. As more than one objective needs to be optimised in a digital differentiator-type multiobjective optimisation problem, it is necessary to understand the idea of multiobjective optimisation to get near the proper Pareto optimal solution. The single-objective function typically has an extremum point as its optimal result; however, getting such a result in multiobjective issues is problematic [5]. This is a fundamental difference between single-objective and multiobjective optimisation problems. As a result, the multiobjective optimisation problem's solution is not just a single point but rather a collection of points representing the compromises among the problem's many objectives [6]. The aforementioned collection of points is also known as the Pareto set [6]. Based on new preference data, the

C. Nayak (✉)

School of Electronics Engineering, VIT-AP University, Inavolu, Beside AP Secretariat, Amaravati, Andhra Pradesh 522237, India
e-mail: chandanayak234@gmail.com

decision-maker must choose the best option. Without this preference information, all Pareto optimum solutions are equivalent [7].

Generally, a posteriori or a priori preference specification method addresses multi-objective challenges. In the priori preference specification method, the designer transforms the multiobjective optimisation problem into a single-objective optimisation problem by allocating weights to the cost function. The priority level assigned to each objective is based on the designer's previous experience [8]. The priority preference specification approach offers a straightforward solution that requires little memory and is much more computationally efficient [9]. But, it has several drawbacks [10]. The optimisation algorithm should be run multiple times with different weights to acquire the Pareto optimal set and Pareto optimal front because the designer does not always have access to the prior information. Moreover, an evenly disseminated set of weights might not fully cover the search space. Since all of the weights in this method are supposed to be positive, the non-convex regions of the Pareto front are inaccessible.

Additionally, the additive weights used in these methods make it impossible to detect the concave regions of the Pareto optimum front. Finally, before the algorithm's execution, the preference information should be defined in a priori approach. Any changes to the priori information will result in a new algorithm run.

In contrast to a priori techniques, the posteriori approach would offer the designer with a wide variety of Pareto optimal results rather than just one solution in a single algorithm run. Additionally, in the posteriori approach, any Pareto optimum front can be approximated, and the researchers do not need to determine the weight. Also, it preserves the problem's multiobjective character and examines its performance. The coverage (solutions distribution) objective is crucial in the posteriori approach since the designer must choose a solution from the available Pareto optimal solutions. The limitations of a priori methods can be successfully overcome by metaheuristic-based approaches. Population-based multiobjective metaheuristic approaches can perfectly satisfy the convergence and coverage goals since they can simultaneously explore the search area for many objectives and offer various results in a single run.

In the literature on stochastic optimisation, multiobjective optimisation approaches have been added with improved mechanisms to locate a precise approximation of the real Pareto optimal front and uniformly distributed solutions. Evolutionary algorithms form the foundation for some of the most well-known posteriori techniques, such as decomposition-based multiobjective evolutionary algorithm (DBMOEA) [11], multiobjective differential evolution (MODE) [12], non-dominated sorting genetic algorithm (NSGA) [13], NSGA version II (NSGA-II) [14], and multiobjective particle swarm optimisation (MOPSO) [15]. However, the multiobjective salp swarm algorithm (MOSSA) [16] is recently proposed and is a highly efficient posteriori method. It finds application in two-dimensional airfoil design [16], marine propeller design [16], feature selection [17], virtual machine placement [18], energy demand management [19], etc.

The broad applicability of the MOSSA has motivated the authors to employ MOSSA to design a zero-phase response (ZPR) FIR-type digital differentiator. Due to the relationship it has to the Riesz potential, the ZPR differentiator is also known

as the Riesz differentiator. The FIR-type Riesz differentiator is a multimodal, multi-dimensional, nonuniform, nonlinear, and multiobjective design problem addressed in this article.

Even though many studies have reported the digital differentiator design, the phase response is the component of differentiator design that receives the least attention [1, 20]. Numerous applications of a system with ZPR can be found in biomedicine, data transfer, and the processing of video and audio signals [4]. Phase distortion in the digital differentiator of the infinite impulse response (IIR) type cannot be rectified directly in any way. However, the issue can be partially handled with finite impulse response (FIR)-digital differentiator by employing the symmetry criterion. Only a zero-phase Riesz differentiator, which adds no delay to the processed signal, can eliminate phase distortion, which is focussed in this study.

The key points of the current research contribution are outlined below.

As per authors' best knowledge, this is the initial report in which an odd order, specifically the first-order digital Riesz differentiator, has been designed, which claims the novelty of this research work.

- (i) Moreover, this is the first time a metaheuristic multiobjective optimisation algorithm has been used to design the Riesz digital differentiator. Two different properly balanced cost functions have been developed for this multiobjective optimisation problem, which is minimised simultaneously by using some of the benchmark algorithms like NSGA-II, MOPSO, and the proposed MOSSA.
- (ii) Then, using the optimised coefficients, DRFODs are designed with nearly optimal magnitude along with nearly ZPR characteristics.
- (iii) After comparing the statistical results of NSGA-II and MOPSO-based DRFODs, performance superiority has been established for the proposed MOSSA-based DRFOD design approach.
- (iv) Moreover, the practical utility and applicability of the ZPR characteristic of the proposed DRFOD are visually investigated by differentiating several ECG signal excerpts taken from PhysioNet ECG data sets [21].
- (v) Finally, the differentiated output of the proposed MOSSA-based DRFOD with the ZPR is further examined in real time using a widely used hardware accelerator called TMS320C6713 digital signal processor.

The content of the remaining of this article is arranged as follows: Sect. 2 introduces the problem formulation for the proposed DRFOD. Section 3 briefly describes the employed multiobjective evolutionary computation technique, and extensive MATLAB-based simulation results are explained in Sect. 4. Additionally, Sect. 4 explains the hardware implementation of the suggested paradigm for producing feature signals linked to QRS. At the conclusion, the final insights of this article are presented in Sect. 5.

2 Problem Statement and Performance Metrics

The ideal digital differentiator, denoted as $D_i(\omega) = (j\omega)^p$, is defined with respect to the frequency, ω , and the order of the digital differentiator, indicated as p . The typical differentiator $D_i(\omega)$ introduces a phase shift of $((1/2) \times (\pi p))$ in its output, causing phase alteration if precise phase characteristics are not attained for the intended differentiator. This phase alteration is greatly undesirable in various applications such as biomedicine, data transmission, and video and audio signal processing. Consequently, there is strong inclination towards designing a ZPR digital differentiator, which introduces no phase delay to the processed signal. This necessity has stimulated the author to realise zero-phase differentiator.

Equation (1) illustrates the frequency response (FR) of the zero-phase differentiator.

$$G_j(\omega) = |\omega|^p \quad (1)$$

where $\omega \in [0, 1]$ denotes the normalised frequency, and p represents the order of the digital differentiator.

The Riesz kernel is derived by performing the Fourier transform (FT) on Eq. (1), as demonstrated in Eq. (2) [22, 23].

$$K(t) = (1/2\Gamma(-p) \cos(p\pi/2)|t|^{-p-1}) \quad (2)$$

where the gamma function, denoted by $\Gamma(\cdot)$, is a complex-valued function that generalises the factorial function to non-integer values.

For the signal $x(t)$, the p th-order Riesz derivative is derived by convolving $x(t)$ and the Riesz kernel as shown in (3) [22].

$$D^p x(t) = x(t) \times K(t) \quad (3)$$

$$D^p x(t) = \frac{1}{2\Gamma(-p) \cos(p\pi/2)} \int_{-\infty}^{\infty} |t - \tau|^{-p-1} x(\tau) d\tau \quad (4)$$

The FT of (4) is calculated using (5).

$$\text{FT}[D^p x(t)] = X(\omega)|\omega|^p \quad (5)$$

where $X(\omega)$ denotes the FT of $x(t)$.

The created Riesz differentiator $G_{dt}(\omega)$ is a very close approximation of the perfect Riesz differentiator, $G_i(\omega)$, only when the designed differentiator's FR $H_d(\omega)$ substantially resembles that of the ideal differentiator $H_i(\omega)$ FR and is arithmetically linked as per (6) [22–30].

$$G_{dt} = \frac{H_d(\omega) + H_d(-\omega)}{2 \cos(p\pi/2)} \quad (6)$$

The significant restriction of Riesz digital differentiator (DD), $G_{dt}(\omega)$, as defined in (6), is that it cannot be used to design first as well as other odd-order differentiators as $G_{dt}(z) \rightarrow \infty$ for all the odd values of p . However, here the main concern is to design a digital Riesz first-order ($p = 1$) differentiator (DRFOD), $G_{dpropfo}(\omega)$, as shown in (7), which has exactly zero-phase characteristic and near-ideal magnitude response. However, to date, the researchers have neither designed the DRFOD, which has ZPR, nor the phase response has been demonstrated.

$$G_{dpropfo}(\omega) = H_d(\omega) + H_d(-\omega) \quad (7)$$

Notably, a perfect optimal differentiator response that abides by the mathematical relationship in (7) can produce a flawless ZPR [22]. However, due of the size restriction of the impulse response number, it is impossible to have a perfect differentiator response. Thus, as a substitute of the ideal ZPR, a close approximation can only be achieved by approximately formulated multiobjective functions and efficiently chosen multiobjective evolutionary computation techniques with suitable control parameters.

In this article, the constraint mentioned in (7) for odd values of p for the design of the DRFOD is overcome by employing a meticulous multiobjective evolutionary computation technique called MOSSA [16]. A very nearly zero-phase characteristics and extraordinarily accurate magnitude response characteristics of the proposed DRFOD are obtained compared to the ideal zero-phase differentiator counterpart for the full Nyquist frequency band.

An N th-order FIR-type differentiator is defined by (8).

$$H_d(z) = \sum_{n=0}^N \frac{h(n)}{z^n} \quad (8)$$

where the set of coefficients that need to be optimised for $H_d(z)$ is depicted by $h(n)$.

The equivalent FR of $H_d(z)$ is denoted by (9).

$$H_d(\omega) = \sum_{n=0}^N \frac{h(n)}{e^{j\omega n}} \quad (9)$$

The flow chart of the integer order DRFOD design approach using 11th-order $H_d(z)$ is depicted in Fig. 1. The FR of the realised DRFOD, $G_{dpropfo}(\omega)$, comes close to the FR of the ideal ZPR differentiator $G_i(\omega)$ by adopting the NSGA-II, MOPSO, and the MOSSA with the efficiently developed multiobjective function J_{DRFOD} presented in (10).

$$\text{Minimise } J_{\text{DRFOD}} = \{ (J_{\text{MAG_DRFOD}}), (J_{\text{PHG_DRFOD}}) \} \quad (10)$$

where objective1: J_{MAG_DRFOD} represents the maximum of normalised magnitude response error as shown in (11), and objective2: J_{PHG_DRFOD} represents the phase response error as shown in (12).

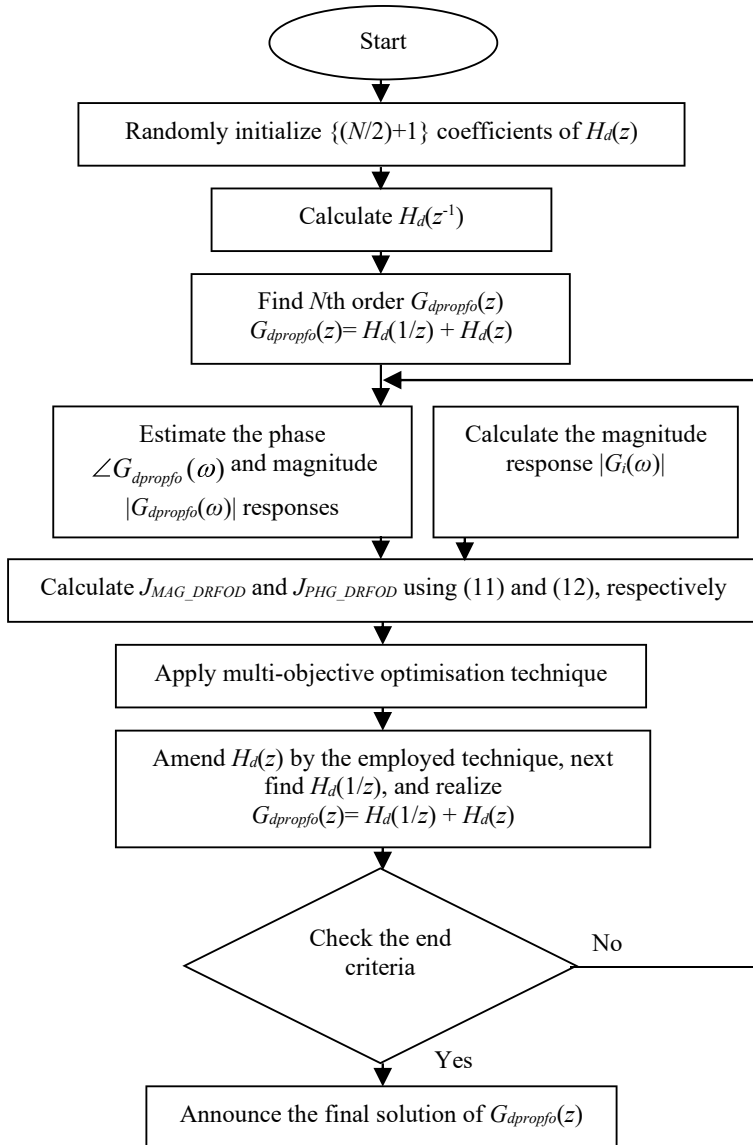


Fig. 1 Flow chart of the multiobjective optimisation technique based DRFOD design methodology

$$J_{MAG_DRFOD} = \max \left[\frac{||G_i(\omega)| - |G_{dpropfo}(\omega)||}{\pi^P} \right] \quad (11)$$

$$J_{PHG_DRFOD} = \sum_{k=1}^{512} |\angle G_{dpropfo}(\omega_k)| \quad (12)$$

The performance of the realised DRFODs is estimated by employing total phase error T_{phase} , normalised maximum absolute magnitude error ($MAME_{norm}$), total normalised magnitude error (TME_{norm}), maximum phase error (MPE), and average group delay (τ_{total}) metrics as defined in (13)–(17), respectively.

$$T_{phase} = \sum_{k=1}^{512} |\angle G_{dpropfo}(\omega_k)| \quad (13)$$

$$MAME_{norm} = \max \left[\frac{||G_i(\omega)| - |G_{dpropfo}(\omega)||}{\pi^P} \right] \quad (14)$$

$$TME_{norm} = \sum \left[\frac{||G_i(\omega)| - |G_{dpropfo}(\omega)||}{\pi^P} \right] \quad (15)$$

$$MPE = [|\angle G_{dpropfo}(\omega)|] \quad (16)$$

$$\tau_{total}(\omega) = \frac{1}{512} \sum_{k=1}^{512} \tau_{dpropfo}(\omega_k) \quad (17)$$

where $\tau_{dpropfo}(\omega) = -\frac{d\theta_{dpropfo}(\omega)}{d\omega}$ and $\theta_{dpropfo}(\omega) = \angle G_{dpropfo}(\omega)$ is the phase response of the $G_{dpropfo}(\omega)$.

2.1 Definitions for Multiobjective Optimisation Problems

Multiobjective optimisation considers multiple objectives. To solve multiobjective problems, each objective should be optimised simultaneously. The proposed multiobjective DRFOD design problem is shown in (10)–(12). The nature of multiobjective optimisation problems does not allow the designer to compare solutions using relational operators because of several objectives.

Using relational operators, researchers safely determine whether one solution is superior to another when there is just one goal, but when there are multiple objectives, the need for another operator called Pareto optimum dominance is essential. It is also the leading operator to compare two solutions when considering numerous objectives [31].

2.1.1 Pareto Domination

Let's assume two vectors: $\vec{z} = (z_1, z_2, z_3, \dots, z_m)$ and $\vec{y} = (y_1, y_2, y_3, \dots, y_m)$. y dominates z ($y \prec z$) if the following criteria is achieved.

$$\begin{aligned} \forall_j \in \{1, 2, 3, \dots, m\}, \quad [f_j(\vec{y}) \leq f_j(\vec{z})] \wedge \\ \exists_j \in \{1, 2, 3, \dots, n\}, \quad [f_j(\vec{y}) < f_j(\vec{z})] \end{aligned} \quad (18)$$

Analysing (18), it is revealed that in a multiobjective optimisation problem, a solution y is only regarded as superior or dominates a solution z if it outperforms the solution z in at least one of the objective values while the other objectives are held equal. In this instance, it is argued that one solution predominates the other. If the above condition is not satisfied, two solutions are denoted to as Pareto optimal or non-dominated solutions. The non-dominated results are essential for a multiobjective optimisation problem. In multiobjective optimisation, the Pareto optimality is also significant and is described as follows [32].

2.1.2 Pareto Optimality

Let us assume $\vec{y} \in Y$. The solution $\vec{y}$ is Pareto optimal if

$$\{ \nexists \vec{z} \in Y \mid \vec{z} \prec \vec{y} \} \quad (19)$$

For each multiobjective problem, a collection of Pareto optimal sets denotes the optimum trade-offs among the various objectives. In multiobjective optimisation, this set of solutions is known as the Pareto optimum set. On the other hand, the Pareto optimum front describes how the Pareto optimal sets are projected into the objective space. The following defines these two sets.

2.1.3 Pareto Optimal Set

The Pareto set comprises Pareto optimal outcomes.

$$P_{set} = \{ \vec{y}, \vec{z} \in Y \mid \nexists \vec{z} \prec \vec{y} \} \quad (20)$$

2.1.4 Pareto Optimal Front

This set contains the Pareto solutions set's objective values.

$$\forall_j \in \{1, 2, 3, \dots, n\}, P_{front} := \{f_j(\vec{y}) \mid \vec{y} \in P_{set}\} \quad (21)$$

A multiobjective evolutionary algorithm results in a set of solutions evenly distributed over the Pareto optimal front. The decision-maker can then choose a solution that best fits the tolerable criteria by comparing solutions using the four definitions above. The proposed DRFOD-type multiobjective and multimodal optimisation problem is solved independently by employing NSGA-II [14], MOPSO [15], and MOSSA [16] algorithms to obtain the near-the-Pareto optimal solution.

3 Multiobjective Evolutionary Computation Techniques

The DRFODs are individually designed by using NSGA-II [14], MOPSO [15], and the recently proposed MOSSA [16]. The widely used NSGA-II and MOPSO, among other multiobjective evolutionary methods are well-known and find applications in several domains [32–35]. Thus, the algorithm steps for the design of the DRFOD using NSGA-II and MOPSO are not discussed in this article. However, a brief description of MOSSA and its step-by-step procedure to realise DRFOD is discussed in Sect. 3.1.

3.1 Multiobjective Salp Swarm Algorithm (MOSSA)

A unique and highly efficient bio-inspired optimiser known as MOSSA [16] has been proposed by Mirjalili et al. in recent multiobjective optimisation literature to address a variety of complex multiobjective optimisation problems with uncertain search areas. The MOSSA mimics the salps' unique swarming behaviour called salp chain while navigating and hunting in the deep water. Salps are thought to create the salp chain to improve hunting strategy and mobility.

In MOSSA, the salp population is divided into leaders and followers. This segregation of salps into two independent groups forms the foundation for the mathematical model of the salp chain. The leader in the salp chain is found first and directs the remainder of the flock (followers). The follower salps build a chain-like structure by following one another, which improves hunting and mobility.

The salps are directed towards the food source using the MOSSA, and the food source is modified throughout epochs. A repository of food sources is equipped with the MOSSA. The top non-dominated results generated during optimisation are kept in this repository. Using Pareto dominance operators, each salp is matched with all the repository members during optimisation. If a salp leads a solution in the repository, then the repository member is replaced by this salp. If a salp completely dominates a group of members in the repository, it is included, and all the previously stored repository members are excluded. A salp should be immediately rejected if

at least one of the repository members dominates it in the new population. A salp must be included in the repository if found non-dominated compared to all repository members. By following these principles, it is possible to ensure that the repository will always contain the non-dominated results that the algorithm has thus far produced. The step-by-step mechanism of MOSSA for the design of the DRFOD is discussed below.

Step 1: The locations of the n_p salps (candidate solutions) are randomly initialised within the upper (ub) and lower (lb) bounds of the search landscape as depicted by (22).

$$x_j^i = lb + (ub - lb) \times \text{rand}, \quad i = 1, 2, \dots, n_p; j = 1, 2, \dots, D \quad (22)$$

where D , x_j^i , and $\text{rand} \in [0, 1]$ are the maximum dimension of the search landscape, j th dimension of the search environment, where the i th salp is located, and the random number, respectively.

Step 2: Calculate J_{DRFOD} for each salp by employing (10). Decide the non-dominated salps. By using the newly acquired non-dominated salps, amend the repository.

Step 3: If the repository is packed, then execute Steps 4–5; else, jump to Step 6.

Step 4: Adopt the repository amendment method to discard one repository member.

Step 5: Include the non-dominated salp in the repository update.

Step 6: From the repository, select a food source F .

Step 7: Utilise (23) to determine c_1 .

$$c_1 = 2e^{-(4l/L)^2} \quad (23)$$

By decreasing the value of c_1 throughout iterations, the steady diversification and intensification phases are maintained in the current research. L and l are the total and active iterations, respectively.

By decreasing the value of c_l during iterations, the consistent exploitation and diversification phases are preserved in the current study.

Step 8: If ($i = 1$), execute Step 9 for each salp x^i , where $i = 1, 2, \dots, n_p$; otherwise, execute from Step 10.

Step 9: Using (24), adjust the leading salp position x_j^1 with respect to F .

$$x_j^1 = \begin{cases} F + (lb + (ub - lb)c_2)c_1 & \text{if } c_3 \geq 0.5 \\ F - (lb + (ub - lb)c_2)c_1 & \text{if } c_3 < 0.5 \end{cases} \quad (24)$$

where the two randomly generated numbers, c_3 and c_2 , fall between 0 and 1.

Step 10: Amend the follower salp position by utilising (25).

$$x_j^i = 0.5(x_j^{i-1} + x_j^i) \quad (25)$$

where x_j^i for $i \geq 2$ represents the position of the j th dimension's i th follower salp.
 Step 11: Restore the salp's location within lb and ub if it leaves the search area.
 Step 12: Continue Steps 2 through 11 until you reach the extreme number of function evaluations (FEs) $f_{\max}(= D \times 10,000)$ [36].
 Step 13: Return the repository once $f_{\max}$ has been attained and reveal the Pareto optimal DRFOD solutions.

4 Results and Discussions

The designed DRFODs' performance using NSGA-II, MOPSO, and the MOSSA techniques are compared through extensive experimental analysis in subsequent subsections. FR, convergence speed, execution time, robustness analysis, and percentage improvement (PI) are rigorously analysed to ensure the efficiency of the proposed filter design problem. All the simulations are done in the MATLAB environment (MATLAB R2022b), powered by a Windows 10 operating system, a core i5 9th gen Intel processor, and 8 GB of memory.

4.1 Selection of Optimal Candidate Solution from the Pareto Optimal Front

All algorithms' standard control variables are kept identical in the design process to compare the evolutionary computation techniques (ECTs) in similar conditions. The standard control variables are $n_p (=100)$, $f_{\max} (=10,000 \times D)$, and minimum ($lb = -2$) and maximum ($ub = 2$) values of the search landscape. The best values of control variables of the NSGA-II, MOPSO, and MOSSA, which are selected by rigorously running the algorithms several times, are presented in Table 1.

The individual Pareto optimal fronts resulted by MOSSA, MOPSO, and NSGA-II for the zero-phase DRFOD design are depicted in Figs. 2, 3, and 4, respectively, to qualify the results. For improved illustration, a combined Pareto optimal fronts resulted by MOSSA, MOPSO, and NSGA-II are shown in Fig. 5. In Figs. 2, 3, 4, and 5, the Pareto optimal fronts are achieved by optimising two objective functions, $J_{\text{MAG_DRFOD}}$ (refer to (11)) and $J_{\text{PHG_DRFOD}}$ (refer to (12)), simultaneously. In Figs. 2, 3, 4, and 5, the x-axis denotes the $\text{MAME}_{\text{norm}}$ metric, and the y-axis denotes the MPE (degree) metric. As shown in Fig. 2, the proposed MOSSA obtained more uniformly distributed approximate Pareto optimal sets in the search arena. However, the Pareto optimal fronts obtained by MOPSO (refer to Fig. 3) and NSGA-II (refer to Fig. 4) are discrete, incomplete, and fail to obtain all Pareto sets. The Pareto fronts obtained by MOPSO (refer to Fig. 3) and NSGA-II (refer to Fig. 4) have some gaps in the middle, which harm the coverage of the obtained Pareto front. However, compared to MOPSO and NSGA-II, the distribution of Pareto fronts obtained by MOSSA is the most uniform and near the true Pareto front. It demonstrates the high

Table 1 Optimal internal control parameters of the NSGA-II, MOPSO, and MOSSA for the realisation of the 22nd-order DRFOD

Control parameter	NSGA-II	MOPSO	MOSSA
Mutants number N_m	40	–	–
Rate of mutation M_r	0.02	–	–
Percentage of crossover P_c	0.7	–	–
Percentage of mutation P_m	0.4	–	–
size of repository N_r	–	100	100
φ_1	–	2.05	–
φ_2	–	2.05	–
φ	–	$\varphi_1 + \varphi_2 = 2.05 + 2.05 = 4.10$	–
w	–	$2/(\varphi - 2 + \sqrt{\varphi^2 - 4\varphi})$	–
Coefficient co_1	–	$\varphi_1 \times w$	–
Coefficient co_2	–	$\varphi_2 \times w$	–
Grids in each dimension N_g	–	10	–
Adaptive coefficient	–	–	$c_1 = 2e^{-(4l/L)^2}$

coverage ability of MOSSA. In summary, all the above results indicate that due to balanced exploration and exploitation capability, MOSSA has obtained improved Pareto optimal and well-distributed solutions in the decision area of the DRFOD optimisation problem compared to MOPSO and NSGA-II.

The optimal coefficients of three instances (refer to candidate solutions A, B, and C in Figs. 2, 3, and 4) resulted from the Pareto optimal front of $MAME_{\text{norm}}$ and MPE (degree) for NSGA-II, MOPSO, and MOSSA along with its corresponding performance metrics are given in Tables 2, 3, and 4, respectively. For the proposed DRFOD design, the candidate solution with the lowest $MAME_{\text{norm}}$ metric and an acceptable MPE metric is selected and considered the optimal solution for the DRFOD design. In Table 2, for the NSGA-II-based design, the candidate solution B with $MAME_{\text{norm}}$ metric of 0.0669 and MPE metric of 9.6703E-014 degree is selected as the optimal solution of the DRFOD design problem. In Table 3, for the MOPSO-based design, the candidate solution B with $MAME_{\text{norm}}$ metric of 0.0652 and MPE metric of 5.8672E-014 degree is selected as the optimal solution of the DRFOD design problem. In Table 4, for the MOSSA-based design, the candidate solution B with $MAME_{\text{norm}}$ metric of 0.0110 and MPE metric of 4.9648E-014 degree is selected as the optimal solution of the DRFOD design problem. These optimal NSGA-II, MOPSO, and MOSSA solutions are used to realise the proposed DRFODs.

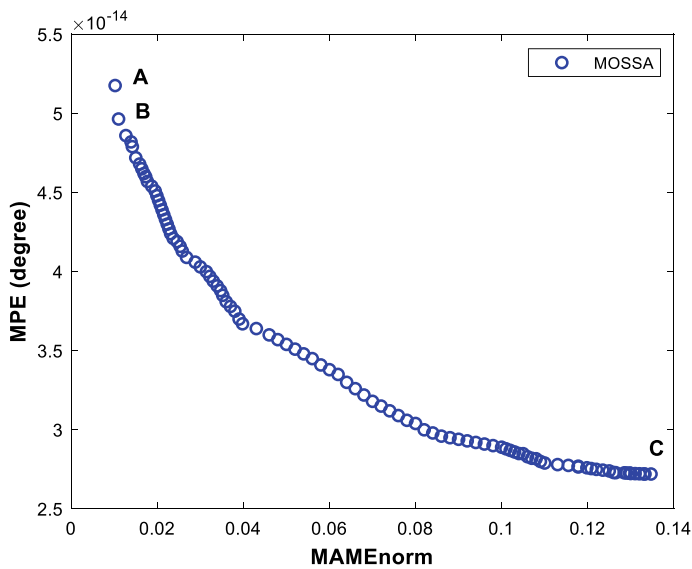


Fig. 2 Pareto optimal front of $MAME_{norm}$ and MPE (degree) for MOSSA-based DRFOD design

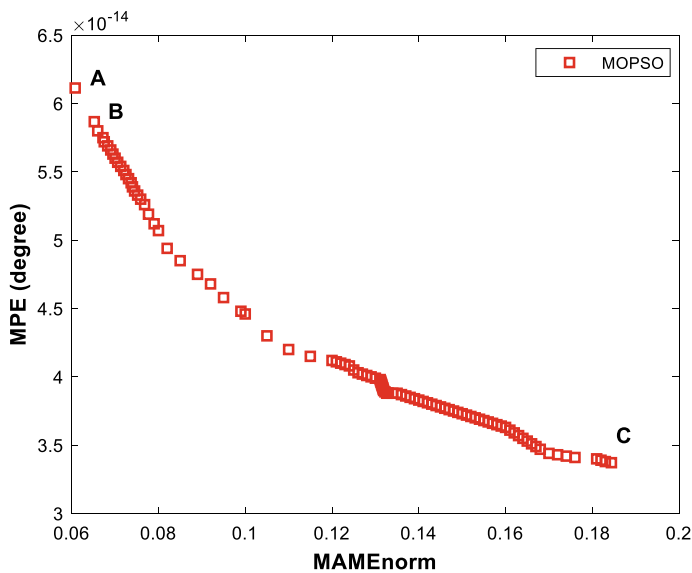


Fig. 3 Pareto optimal front of $MAME_{norm}$ and MPE (degree) for MOPSO-based DRFOD design

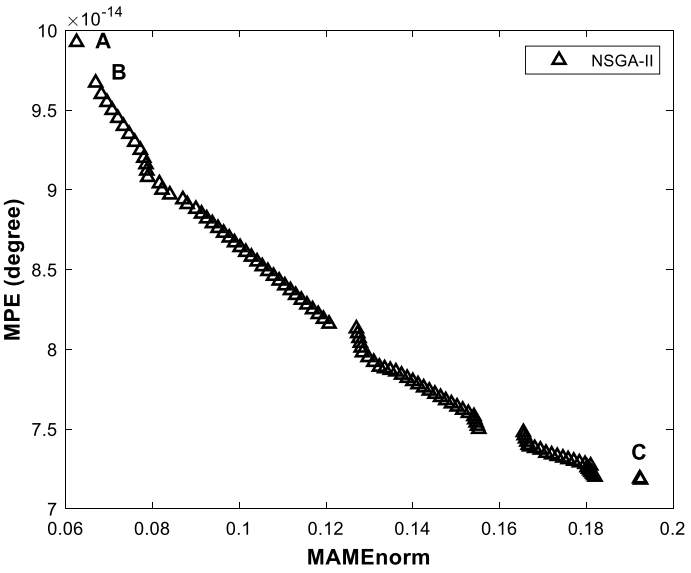


Fig. 4 Pareto optimal front of $MAME_{norm}$ and MPE (degree) for NSGA-II-based DRFOD design

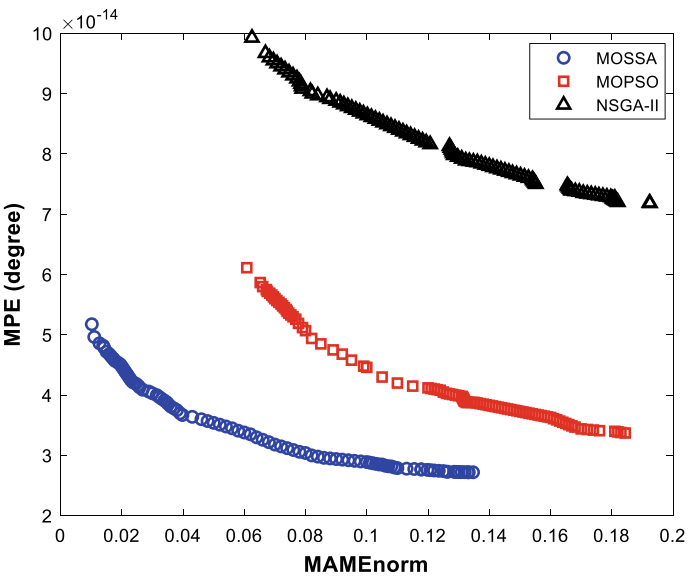


Fig. 5 Combined Pareto optimal front of $MAME_{norm}$ and MPE (degree) for MOSSA, MOPSO, and NSGA-II-based DRFOD design

Table 2 Optimised coefficients and performance metrics of the 22nd-order DRFOD using NSGA-II

$h(n)$	A: NSGA-II	B: NSGA-II	C: NSGA-II
$h(22) = h(0)$	− 0.035725360262	− 0.016078340120	− 0.029354412472458
$h(21) = h(1)$	0.002165129169	0.040124896818	0.081297362249453
$h(20) = h(2)$	0.009382631619	− 0.042221668239	− 0.031215025941272
$h(19) = h(3)$	0.002697981910	− 0.015613313412	− 0.020664884085262
$h(18) = h(4)$	− 0.017075551784	− 0.023212807994	− 0.094687829095061
$h(17) = h(5)$	− 0.023337706522	− 0.009410239443	0.007021384924658
$h(16) = h(6)$	− 0.038128844386	− 0.039396880528	− 0.004592380821448
$h(15) = h(7)$	0.006758936995	0.012806006061	0.047450998692530
$h(14) = h(8)$	− 0.073239927836	− 0.040415568385	0.003801547066346
$h(13) = h(9)$	− 0.000284650348	0.000416222361	0.001989808814252
$h(12) = h(10)$	− 0.663126339106	− 0.635522324696	− 0.536422448603026
$h(11)$	1.552728992526	1.541024964537	1.541233916645824
MPE (degree)	9.9266E-014	9.6703E-014	7.1820E-014
MAME _{norm}	0.0625	0.0669	0.1925

Table 3 Optimised coefficients and performance metrics of the 22nd-order DRFOD using MOPSO

$h(n)$	A: MOPSO	B: MOPSO	C: MOPSO
$h(22) = h(0)$	− 0.010706506257	− 0.001295308686	0.025217089044
$h(21) = h(1)$	0.006939623713	0.014266951744	0.027852764058
$h(20) = h(2)$	0.012593523344	0.006385002537	0.029416723635
$h(19) = h(3)$	0.028055276947	0.018335858662	− 0.020529743988
$h(18) = h(4)$	− 0.012699395994	0.016159118033	0.038336925161
$h(17) = h(5)$	0.003547606453	− 0.006187420867	− 0.008562607779
$h(16) = h(6)$	− 0.029003174554	− 0.013590172169	− 0.026309378847
$h(15) = h(7)$	− 0.032161299189	− 0.011600274401	− 0.007014460428
$h(14) = h(8)$	− 0.066579469287	− 0.057258973881	− 0.015688732381
$h(13) = h(9)$	− 0.019398162952	− 0.018515079934	0.008795002367
$h(12) = h(10)$	− 0.607236897118	− 0.655535575099	− 0.607887628336
$h(11)$	1.562106840901	1.552065819113	1.447381165420
MPE (degree)	6.1135E-014	5.8672E-014	3.3719E-014
MAME _{norm}	0.0608	0.0652	0.1844

Table 4 Optimised coefficients and performance metrics of the 22nd-order DRFOD using MOSSA

$h(n)$	A: MOSSA	B: MOSSA	C: MOSSA
$h(22) = h(0)$	– 0.008639574200	– 0.008664701108	– 0.025692490018
$h(21) = h(1)$	0.001009335950	0.000889522827	– 0.000600753804
$h(20) = h(2)$	– 0.011182123645	– 0.009303284761	– 0.009764001895
$h(19) = h(3)$	0.001515949078	0.000000088944	0.021055730143
$h(18) = h(4)$	– 0.016841401796	– 0.015291361288	0.024212790763
$h(17) = h(5)$	– 0.000497103188	0.000040946633	– 0.009184658331
$h(16) = h(6)$	– 0.025869195151	– 0.026637426615	– 0.031111802546
$h(15) = h(7)$	– 0.002918620720	0.000233774253	– 0.028867627249
$h(14) = h(8)$	– 0.069737081021	– 0.071876732254	– 0.070369611631
$h(13) = h(9)$	0.000750571572	0.000868459021	– 0.010768995301
$h(12) = h(10)$	– 0.637702724615	– 0.637664804058	– 0.538827233188
$h(11)$	1.572420263640	1.569503263635	1.6566495570964
MPE (degree)	5.1763E-014	4.9648E-014	2.7205E-014
MAME _{norm}	0.0102	0.0110	0.1347

4.2 FR Analysis of the Designed DRFODs

The proposed NSGA-II, MOPSO, and MOSSA-based DRFODs' marvellously detailed FR comparison charts are displayed in Figs. 6, 7, and 8. The FR performance of the designed DRFODs is visually investigated in terms of absolute magnitude error (AME), magnitude response (MR), group delay response, phase delay response ($-\theta_{\text{drop}}(\omega)/\omega$), phase response, and absolute phase error (APE). From the phase response plots (refer to Fig. 6a), it can be observed that almost ZPR is obtained for all the cases, and it is only due to the incorporation of fitness function $J_{\text{PHG_DRFOD}}$ (refer to (12)). As a result, a very small APE is observed in Fig. 6b. Also, the designed differentiators' phase and group delay responses are shown in Fig. 7a-b, from which almost zero delay is observed for all the cases. Thus, any signal that passes through will produce the differentiated output with zero-phase distortion.

Figure 8a presents the MR comparison plots of all designed DRFODs and the ideal Riesz differentiator. It can be seen that, in comparison with the NSGA-II and MOPSO-based DRFODs, the MR of the proposed MOSSA-based DRFOD more closely resembles the ideal Riesz differentiator MR over the whole Nyquist frequency band. Figure 8b illustrates the AME comparison plots of the designed DRFODs in the dB scale, and improved performance is observed for the MOSSA-based design approach.

The performance of all designed DRFODs is further investigated numerically using the T_{phase} in degree, τ_{total} in samples, TME_{norm}, and MPE (in degree), MAME_{norm}, and MAME_{norm} (in dB) metrics to study its design efficacy. The results of the performance comparison are depicted in Table 5. Table 5 shows that

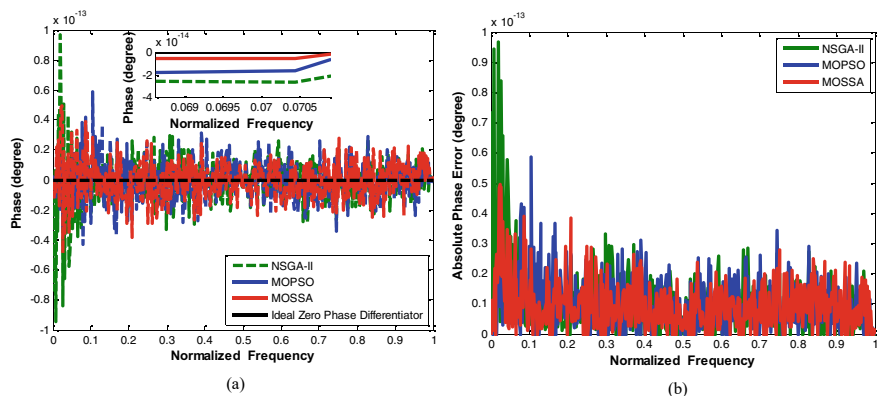


Fig. 6 **a** Phase response and **b** absolute phase error plots of the NSGA-II, MOPSO, and the proposed MOSSA-based DRFODs

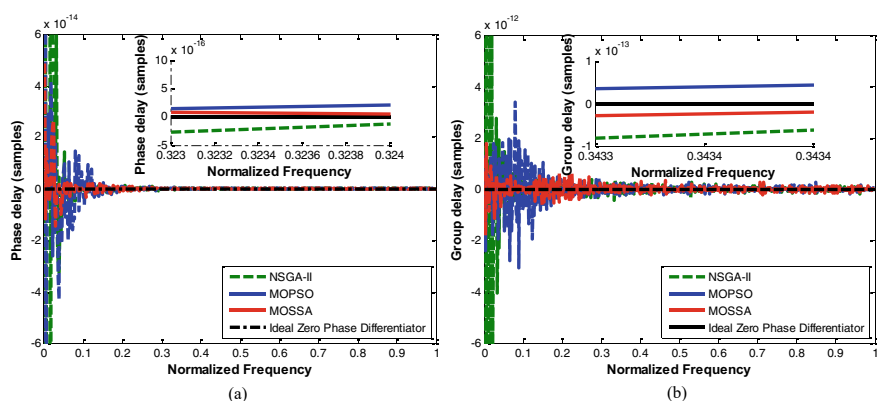


Fig. 7 **a** Phase delay and **b** group delay plots of the NSGA-II, MOPSO, and the proposed MOSSA-based DRFODs

regarding the MPE, T_{phase} , and τ_{total} performance metrics, the proposed MOSSA-based DRFOD results have sensibly superseded the corresponding metrics resulting from the NSGA-II and MOPSO-based designed DRFODs.

It is also noticed from Table 5 that the $\text{MAME}_{\text{norm}}$, $\text{MAME}_{\text{norm}}$ (in dB), and the TME_{norm} metrics obtained by the MOSSA-based DRFOD have outperformed the corresponding metrics resulted by the MOPSO and NSGA-II-based DRFODs. Therefore, from the discussions on the simulation results mentioned above, it is easily understood that with an almost ZPR and near-ideal MR characteristics, the proposed MOSSA-based DRFOD is much superior to NSGA-II and MOPSO-based DRFODs.

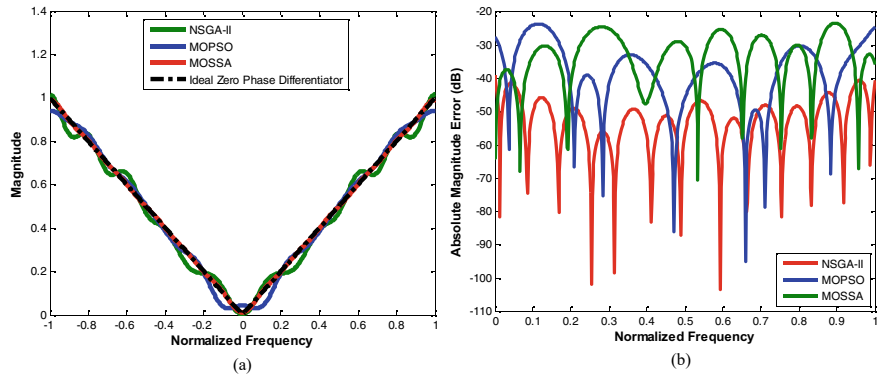


Fig. 8 **a** Magnitude response and **b** absolute magnitude error plots of the NSGA-II, MOPSO, and the proposed MOSSA-based DRFODs

Table 5 Performance comparison of all designed DRFODs

<i>Phase response</i>			
Algorithms	MPE(degree)	T_{phase} (degree)	τ_{total} (samples)
NSGA-II	9.6703E-014	5.3291E-012	7.4917E-014
MOPSO	5.8672E-014	5.0466E-012	6.5171E-014
MOSSA	4.9648E-014	4.8990E-012	5.9812E-014
<i>Magnitude Response</i>			
Algorithms	MAME _{norm}	MAME _{norm} (dB)	Total Magnitude Error
NSGA-II	0.0669	−23.4882	13.8406
MOPSO	0.0652	−23.7098	9.5587
MOSSA	0.0110	−39.1384	1.5076

4.3 Execution Time and Convergence Profile Analysis for Comparing MOSSA with NSGA-II and MOPSO

The rate at which an evolutionary algorithm advances towards the global optimum is known as convergence speed. The efficacy of any evolutionary algorithm can be evaluated from the number of FEs needed to attain the minimum cost value. To estimate the optimised set of DRFOD coefficients, an algorithm convergence profile is also reported for each case of program execution (refer to Fig. 9). The number of FEs and execution times to find the optimal solutions is presented in Table 6. From Fig. 9, it is observed that MOSSA consumes the lowest number of 9100 FEs to reach the near-global optimal solution. The execution time evaluated for finding the optimal solution represents the searching efficacy of the optimisation technique. Table 6 shows that the MOSSA requires the lowest execution time of 58.7245 s to execute 9100 FEs compared to other algorithms.

Fig. 9 Convergence profiles of MOSSA, MOPSO, and NSGA-II-based DRFODs

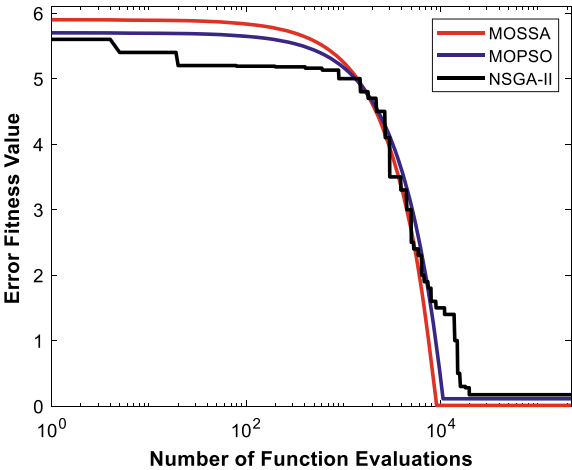


Table 6 Statistical data of DRFODs

Statistical Data	Best	Worst	Average	Median	SD
Algorithm	MAME (dB)				
NSGA-II	− 23.4882	− 8.9713	− 11.2459	− 11.7214	1.9854
MOPSO	− 23.7098	− 10.2216	− 13.6219	− 13.9160	1.2849
MOSSA	− 39.1384	− 32.1068	− 37.5614	− 37.7548	0.5672
Algorithm	Computational time in second				
NSGA-II	76.9725	101.8842	95.7824	96.2243	0.6827
MOPSO	133.7560	264.9842	246.1243	251.1078	1.8741
MOSSA	58.7245	109.7058	87.1192	90.2214	1.2172
Algorithm	Number of function evaluations needed				
NSGA-II	19,750	26,000	24,500	24,800	239
MOPSO	10,600	21,000	19,500	19,900	268
MOSSA	9100	17,000	13,500	14,000	120

4.4 The Reliability and Robustness Analysis

The reliability and robustness of the NSGA-II, MOPSO, and MOSSA in terms of average-case performance are examined with the help of the MAME metric, computational time, and the number of FEs required to execute the algorithm for 50 separate times, to evaluate various statistical parameters, namely standard deviation (SD), median, average, worst, and best values. The experimental results are showcased in Table 6. The robustness of the MOSSA for designing DRFOD is justified by the fact that the average MAME (dB) value is almost closer to the best MAME (dB) value. Moreover, the lowest SD value of 120 in terms of the number of FEs needed and an

acceptable SD value of 1.2172 s in terms of computational time justify the robust behaviour of MOSSA for the DRFOD-type search optimisation problem.

4.5 Hypothesis Testing

Both nonparametric (Wilcoxon rank-sum test) and parametric (two-sample t -test) statistical hypothesis tests have been conducted using pairwise combinations of the proposed MOSSA and the cutting-edge employed NSGA-II and MOPSO algorithms to investigate the consistency in the performance of the proposed evolutionary algorithm for the realisation of digital differentiators.

4.5.1 Two-Sample t -test

The two-sample t -test examines the null hypothesis that two samples with the same mean are drawn from the same population with a normal distribution [37–41]. The null hypothesis is rejected at a specific significance (α) or $(1 - \alpha) \times 100\%$ confidence level (CL), implying that the mean of the two samples of various lengths chosen from populations generated by two separate systems differs significantly. Below is the method for conducting a t -test to assess the effectiveness of the proposed algorithm.

Step 1: As seen below, define the alternative (H_a) and null (H_0) hypotheses.

H_0 : There is no notable distinction in the utilised performance metric between the two data sets acquired for the benchmark and proposed algorithms.

H_a : Between the two data sets, there is a significant disparity.

Step 2: Calculate the standard deviations (S_1, S_2) and the means (μ_1, μ_2) of the data sets with p_1 and p_2 sample lengths, respectively. The benchmarked algorithm is represented by subscript 1, and the proposed method is represented by subscript 2.

Step 3: Use (26) and (27) to calculate the degrees of freedom (d) and t_{cal} , respectively.

$$d = p_1 + p_2 - 2 \quad (26)$$

$$t_{cal} = \frac{\mu_1 - \mu_2}{\sqrt{\left(\frac{S_1^2}{p_1}\right) + \left(\frac{S_2^2}{p_2}\right)}} \quad (27)$$

Step 4: From the t distribution chart, infer and choose the critical value of t ($t_{\alpha, d}$) for $\alpha = 0.01$.

4.5.2 Wilcoxon Rank-Sum Test

Unlike the two-sample t -test, which requires a normal distribution of the data set, the Wilcoxon rank-sum test examines the null hypothesis that two samples are obtained from the same population without a specific kind of distribution [37]. In this study, the Wilcoxon rank-sum test is employed to assess the performance of the proposed algorithm compared to the benchmark algorithm in terms of the $MAME_{\text{norm}}$ metric for digital differentiators, conducted pairwise. The following is the technique for doing this test.

Step 1: Define the alternate hypothesis (H_a) and the null hypothesis (H_0).

H_0 : Regarding $MAME_{\text{norm}}$, no significant difference exists between the two data sets acquired for the benchmark algorithm and the suggested technique.

H_a : Between the two data sets, there is a significant disparity.

Step 2: Assume that n_1 and n_2 are the sample size for the benchmark and the proposed algorithms, respectively. Arrange the total $(n_1 + n_2)$ samples with their corresponding ranks in ascending order. The mean ranks of contradictory observations should be assigned.

Step 3: Let W_1 be the total of the ranks for the benchmark algorithm's data. Then, W_2 is described by

$$W_2 = (0.5) (n_1 + n_2 + 1) (n_1 + n_2) - W_1 \quad (28)$$

Step 4: Determine whether the null hypothesis should be accepted or rejected based on the critical value (U_C) at a particular α from the Wilcoxon rank-sum test table [37]. This study employs two distinct values of α , such as 0.01 and 0.05.

The solutions obtained from the two-sample t -test and Wilcoxon rank-sum test for the pairwise union of NSGA-II, MOPSO, and MOSSA, as depicted in Table 7, are the pieces of evidence that the MOSSA rejects the null hypothesis with a reasonably high value of CL. Hence, compared to NSGA-II and MOPSO-based designs, the MOSSA consistently attains the lowest value of the MAME metric.

4.6 Comparison of Percentage Improvement Among All the Optimally Designed DRFODs

Figures 10 and 11 depict the percentage improvement attained by the proposed MOSSA compared to NSGA-II and MOPSO-based DRFODs in terms of MPE (PI_{MPE}) and $MAME_{\text{norm}}$ ($PI_{\text{MAME}_{\text{norm}}}$) metrics, respectively.

It is ensured from Fig. 10 that in terms of the $PI_{\text{MAME}_{\text{norm}}}$ metric, the proposed MOSSA-based DRFOD achieves 84 and 83% compared to NSGA-II and MOPSO-based design approaches, respectively.

Table 7 Results of Wilcoxon rank-sum test and *t*-test for the proposed DRFODs for 100 independent runs of algorithms

<i>p</i>	Algorithms	Wilcoxon rank-sum test							Two-sample <i>t</i> -test	
		<i>n</i> ₁	<i>n</i> ₂	Critical value		<i>W</i> ₁	<i>W</i> ₂	Accept/ reject the null hypothesis (CL)	<i>t</i> _{cal}	Accept/ reject the null hypothesis (CL)
				0.05	0.01					
1	NSGA-II-MOSSA	7	10	42	37	32	121	Reject (99.00%)	6.7491	Reject (99.99%)
		6	8	29	25	25	80	Reject (99.00%)		
		5	9	22	18	19	86	Reject (95.00%)		
	MOPSO-MOSSA	7	10	42	37	34	119	Reject (99.00%)	5.6415	Reject (99.99%)
		6	8	29	25	25	80	Reject (99.00%)		
		5	9	22	18	19	86	Reject (95.00%)		

Fig. 10 Illustration of the percentage improvement in MAME_{norm} resulting from the MOSSA-based DRFOD compared with the NSGA-II and MOPSO-based DRFODs

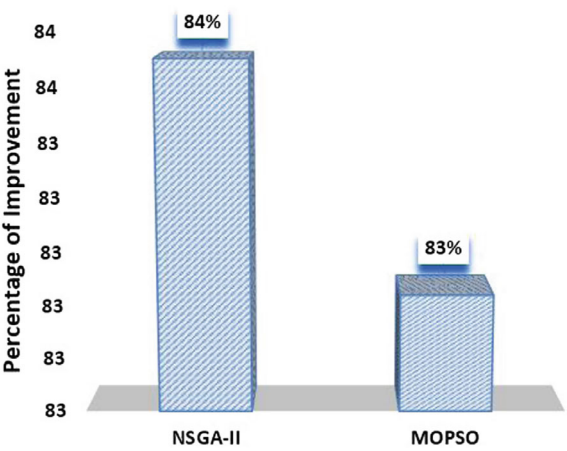
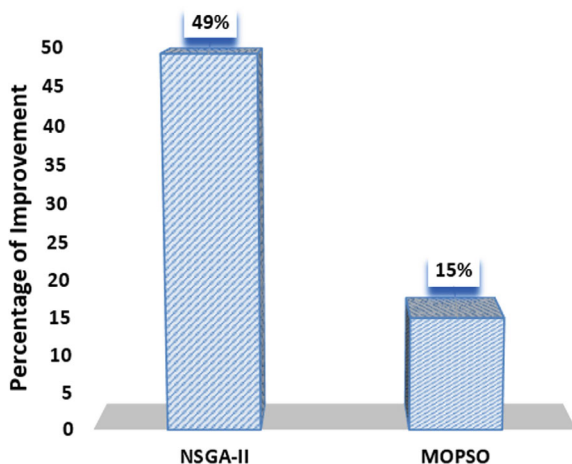


Figure 11 shows that the proposed MOSSA-based zero-phase DRFOD achieves a PI_{MPE} of 49 and 15% compared to NSGA-II and MOPSO-based differentiators, respectively.

Fig. 11 Illustration of the percentage improvement in MPE resulting from the MOSSA-based DRFOD compared with the NSGA-II and MOPSO-based DRFODs



4.7 Comparative Study of the Proposed MOSSA-Based DRFOD with Other Reported Riesz Differentiators

A comprehensive comparative analysis between the proposed DRFOD and digital Riesz DDs documented by other researchers is provided in Table 8, emphasising the importance of this study. In [28], Tseng et al. have employed the classical DD design techniques such as discrete cosine transform (DCT), Tustin's method, and fractional differencing (FD) method to realise the fractional-order digital Riesz differentiators of order $p = 0.3$ and 0.5 . Similarly, Kaur et al. employed the FD approach to design the fractional-order digital Riesz differentiator of order $p = 0.6$ in [22]. However, the design approach presented in [28] and [22] cannot be employed to design first order ($p = 1$) as well as other odd orders digital Riesz differentiators as $G(z) \rightarrow \infty$ for odd values of p (refer to (6)). In this study, the design challenge, as discussed above, has been overcome by employing a meticulous multiobjective ECT called MOSSA.

In [28], a 99th-order FIR- and the 15th-order IIR-type digital Riesz fractional-order differentiators (FODs) are designed. To realise 15th-order IIR-type Riesz FOD, $32 (= 2(15 + 1))$ coefficients are required. Similarly, to realise a 99th-order FIR Riesz FOD, $100 (= 99 + 1)$ coefficients are required. The real-time implementation of these higher-order digital Riesz FODs has a higher computational burden due to an enhanced mathematical complexity. However, for a practical application like online ECG pre-processing, the differentiator should be computationally efficient, highly accurate, and lower order [4]. In this scenario, the proposed 22nd-order FIR-type DRFOD needs only $23 (= 22 + 1)$ coefficients which ensures the computational superiority compared to the higher-order differentiators of [22, 28] for practical applications.

Moreover, the design method outlined in both Sects. 2 and 3 can be applied to implement both lower-order ($N < 22$) and higher-order ($N > 22$) DRFODs. It has been noted that the FR of the differentiator improves with increasing order, however,

Table 8 Comparison of the proposed and the reported digital Riesz differentiators

References	Technique	p	Model Name	Filter Types	N	MPE (degree)	T_{phase} (degree)	τ_{total} (samples)	$MAME_{norm}$	PI_{MAME} (%)	MAME (dB)	TME
Tseng et al [28] (Fig. 1)	FD	0.3	FD0.3	IIR	15	N/R	N/R	N/R	> 0.0600*	82	> -24.4369*	N/R
		0.5	FD0.5	IIR	15	N/R	N/R	N/R	> 0.3500*	97	> -9.1186*	N/R
Tseng et al [28] (Fig. 3)	Tustin's	0.3	Tustin0.3	IIR	15	N/R	N/R	N/R	> 0.6000*	98	> -4.4370*	N/R
		0.5	Tustin0.5	IIR	15	N/R	N/R	N/R	> 1.2000*	99	> 1.5836*	N/R
Tseng et al [28] (Fig. 4)	DCT	0.3	DCT0.3	FIR	99	N/R	N/R	N/R	> 0.2000*	95	> -13.9794*	N/R
		0.5	DCT0.5	FIR	99	N/R	N/R	N/R	> 0.1500*	93	> -16.4782*	N/R
Kaur et al. [22]	FD	0.6	FD0.6	IIR	N/R	N/R	N/R	N/R	Δ	Δ	Δ	N/R
Present work	MOSSA	1	MOSSA1	FIR	22	4.9648E-014	4.8990E-012	5.9812E-014	0.0110	N/A	-39.1384	1.5076

N/R denotes not reported metrics, [] denotes the reference number, () denotes the figure number in the reference number where the Riesz differentiator has been reported, * denotes the approximate calculated values from visual inspection of the magnitude plot, Δ denotes that due to insufficient data, metrics evaluation is not possible, N/A denotes not applicable

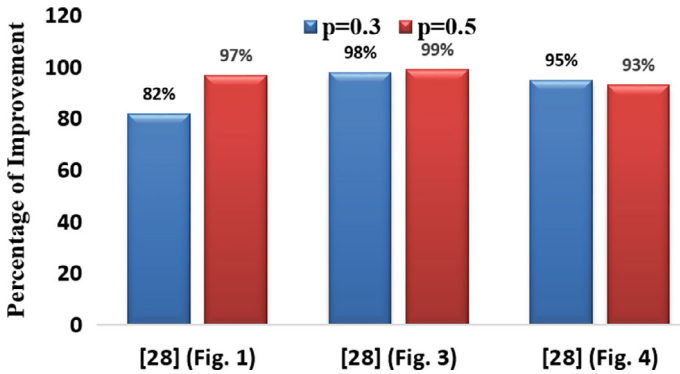


Fig. 12 Illustration of the percentage improvement in MAMEnorm resulting from the MOSSA-based DRFOD compared with the reported DRFODs

with an increase in computational overhead. Hence, it has become a significant challenge for researchers to efficiently construct the differentiator, achieving the desired frequency response with minimal order using optimization techniques.

The 22nd-order DRFOD that is created in this article to produce extremely accurate MR and accurate zero-phase effectively tackles this design trade-off. Moreover, Table 8 shows that compared to the existing digital Riesz FODs [28], the proposed DRFOD achieves sensibly enhanced design performances in terms of $MAME_{norm}$ and $MAME_{norm}$ (dB) metrics besides its computationally efficient design feature. The approximate $MAME_{norm}$ and $MAME_{norm}$ (dB) metrics of the reported digital Riesz FODs [28] are calculated from pictorial examination of the respective magnitude plots. Compared to the digital Riesz FOD of order $p = 0.3$ designed by the FD method (model name: FD0.3), the proposed DRFOD achieves a percentage improvement ($PI_{MAMEnorm}$) of 82% ($= (0.0600 - 0.0110)/0.0600$) in terms of $MAME_{norm}$ metric. Table 8 and Fig. 12 show the PI_{MAME} attained by the proposed DRFOD over the existing digital Riesz FODs in terms of the $MAME_{norm}$ metric.

4.8 Experiment Results and Analysis

The real-time performance of the proposed DRFOD is studied against several real ECG signals under noisy conditions. PhysioNet QTDB [21] ECG signals are employed for this purpose. The digital differentiator has played a pivotal role in the ECG QRS complex detection application, where the digital differentiator is used to generate the feature signal (differentiated output) corresponding to the QRS complex region [3]. The subsequent stages further process the generated feature signal to determine the exact location of the QRS complex [3]. The QRS detection accuracy is directly proportional to the feature signal generation capacity of the employed digital differentiator [3].

As the noise is inherent in the ECG signal, random noise is generated and added to several real ECG signals. The noise-contaminated ECG signals representing several arrhythmia situations are fed to the proposed DRFOD to examine its robust feature signal generation capability, as shown in Figs. 13, 14, 15, and 16. In each figure, the first subplot represents the raw ECG signal representing several arrhythmias (ECG corrupted by high-grade muscle artefacts depicted in Fig. 13(a), irregular heart rate shown in Fig. 14a, high amplitude T-waves presented in Fig. 15a, and complex ECG pathology shown in Fig. 16a); the second subplot represents the random noise; the third subplot represents the noise-contaminated ECG signals, and the fourth subplot represents the differentiated output. Most of the reported QRS detectors fail to identify the exact locations of the QRS complex region for the aforementioned ECG arrhythmias [3]. However, the differentiated outputs shown in each figure are good evidence of the robust feature signal generation capability of the proposed DRFOD for all the aforementioned ECG arrhythmia conditions due to its almost zero-phase and near-ideal magnitude characteristics. Thus, it is noteworthy that the proposed DRFOD can be used for precise ECG QRS complex detection tasks.

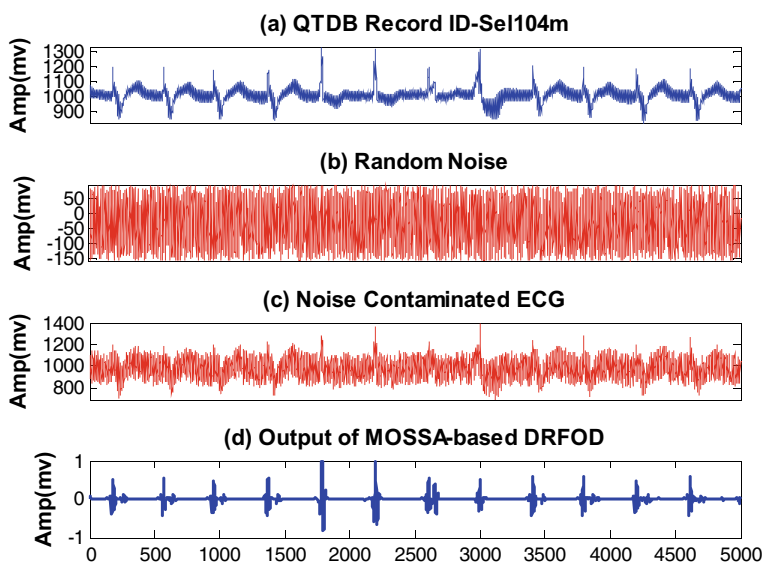


Fig. 13 Application demonstration of the proposed DRFOD in real noise-contaminated ECG signal: **a** Raw ECG of QTDB record ID-Sel104m corrupted by high-grade muscle artefacts, **b** Random noise, **c** Noise-contaminated ECG, **d** Output of the proposed DRFOD

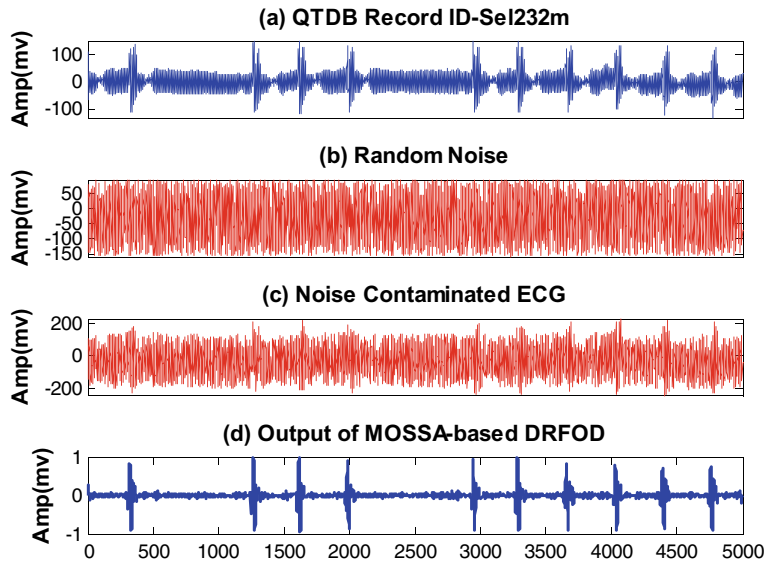


Fig. 14 Application demonstration of the proposed DRFOD in real noise-contaminated ECG signal **a** Raw ECG of QTDB record ID-Sel232m representing irregular heart rate contaminated with muscle artefacts, **b** Random noise, **c** Noise-contaminated ECG, **d** Output of the proposed DRFOD

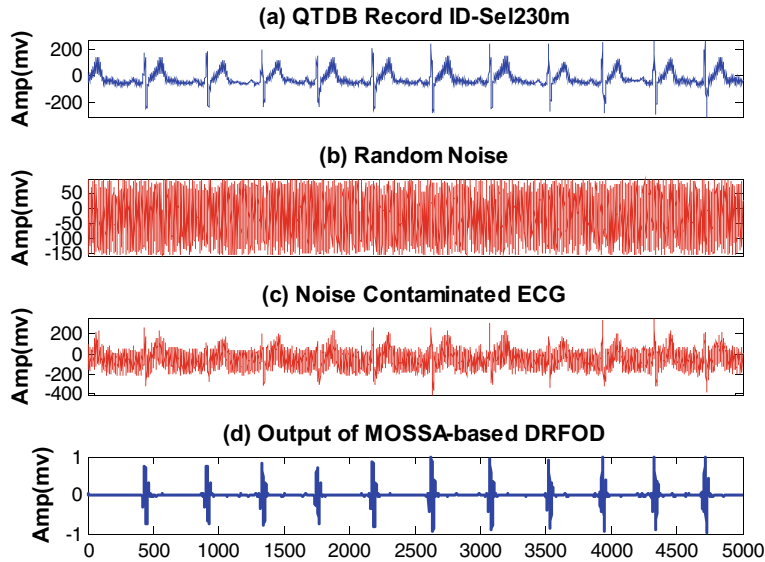


Fig. 15 Application demonstration of the proposed DRFOD in real noise-contaminated ECG signal **a** Raw ECG of QTDB record ID-Sel230m representing high amplitude T-waves, **b** Random noise, **c** Noise-contaminated ECG, **d** Output of the proposed DRFOD

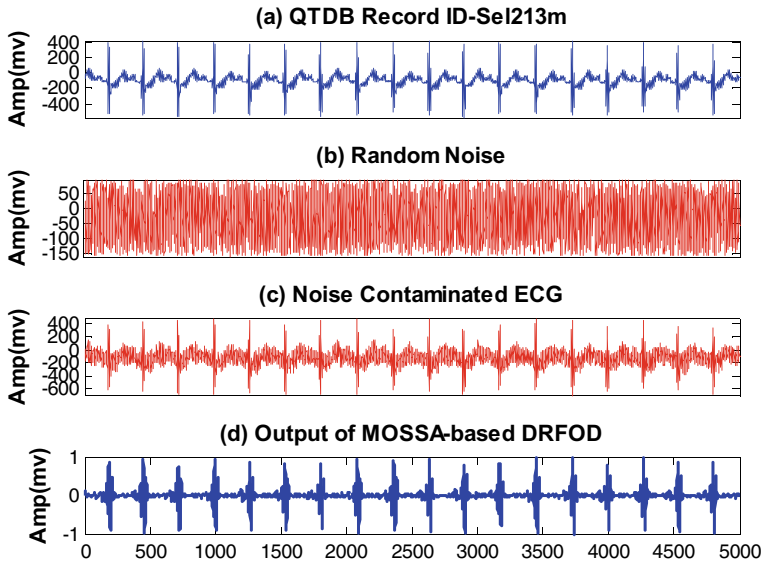
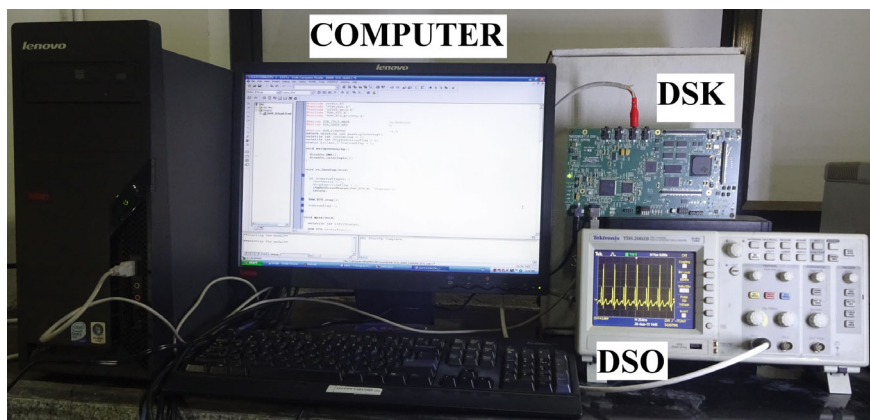


Fig. 16 Application demonstration of the proposed DRFOD in real noise-contaminated ECG signal **a** Raw ECG of QTDB record ID-Sel213m representing complex ECG pathology, **b** Random noise, **c** Noise-contaminated ECG, **d** Output of the proposed DRFOD

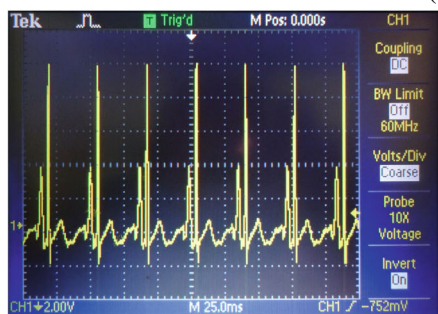
4.9 Hardware Realisation of the Proposed Model for QRS-Related Feature Signal Generation

The differentiated output of the proposed DRFOD with the ZPR is further examined in real time by using a widely used hardware accelerator called TMS320C6713 digital signal processor starter kit (DSK) (flash memory: 512 Kbytes, DRAM: 16 Mbytes, and clock frequency: 225 MHz) of Texas Instrument (TI). The employed DSK has superior computational capability along with the power-optimised feature. Thus, the industry uses the DSK for mobile-based applications to extend the battery life. Figure 17a shows the snapshot of the experimental setup.

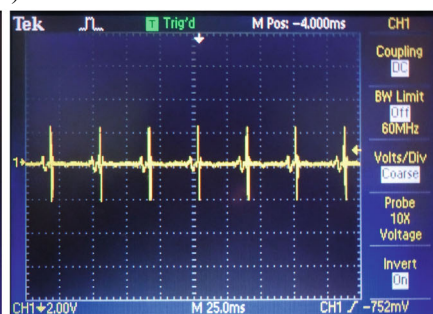
In the experiment, firstly, the optimised coefficients (refer to the third column of Table 4) of the MOSSA-based DRFOD are used to realise the differentiation stage in C programming language using the TI's code composer studio (CCS) v3.1 integrated development environment (IDE). After debugging, the program is built and dumped to the digital signal processor on the DSK by using an embedded JTAG emulator with a USB host interface. Finally, a five-second MITDB ECG excerpt with a sampling frequency of 360 Hz, as shown in Fig. 17b, is applied to the differentiation stage realised on DSK. The corresponding differentiated output (Fig. 17c) is routed to the Tektronix TDS 2002B digital storage oscilloscope (DSO) through the line output jack of the on-board AIC23 codec. In Fig. 17c, it can be seen that the feature signal is generated corresponding to each QRS complex region in the input ECG signal. Thus, it ensures the accurate zero-phase feature of the proposed MOSSA-based



(a)



(b)



(c)

Fig. 17 a Experimental set-up; (b) MITDB raw ECG record ID-100 and (c). Differentiated zero-phase output

DRFOD. To process the ECG signal of 5 s duration, the TMS320C6713 digital signal processor needs only $3.86613\text{E} + 06$ clock cycles. As the TMS320C6713 digital signal processor operates with a clock rate of 225 MHz, the differentiated output is calculated in approximately 17.183 ms ($= (1/225 \text{ MHz}) \times (3.86613\text{E} + 06)$ clock cycles). Moreover, it needs only $9.55 \mu\text{s}$ ($= 17.183 \text{ ms} / (5 \times 360 \text{ samples/s})$) to calculate each sample, which is relatively low compared to the sampling interval of $2777.77 \mu\text{s}$ ($= 1/360 \text{ Hz}$).

5 Conclusion

This article uses the multiobjective salp swarm algorithm (MOSSA) to implement the FIR-type zero-phase digital Riesz first-order differentiator (DRFOD). The MOSSA applies an evenly distributed control mechanism to enable simultaneous local exploitation and global exploration of the search landscape. This innovative control mechanism of MOSSA amplifies the search capability. It circumvents premature convergence by escaping from the local optima to arrive closer to the global best solution in the minimum number of epochs. Due to this, accurately zero-phase and highly accurate magnitude response (MR) of the proposed DRFOD is achieved, and it outperforms the conventional non-dominated sorting genetic algorithm version II (NSGA-II) and multiobjective particle swarm optimisation (MOPSO)-based DRFOD design approaches. The maximum phase error (MPE) and the maximum normalised absolute magnitude error ($MAME_{norm}$) of the proposed MOSSA-based DRFOD are as low as $4.9648E-014$ degrees and 0.0110, respectively. Regarding MPE, the proposed DRFOD achieves a percentage improvement (PI) of 49 and 15% compared to NSGA-II and MOPSO-based DRFODs, respectively.

Similarly, regarding $MAME_{norm}$, the proposed DRFOD design achieves a PI of 84 and 83% compared to NSGA-II and MOPSO-based designs, respectively. Finally, applying the MOSSA-based Riesz differentiator for generating feature signals related to the ECG QRS complex region is thoroughly investigated to justify the accurate zero-phase response of the proposed DRFOD.

Funding The authors declare that there is no institute/agency funding for this work.

Data Availability The data sets analysed during the current study are available in the PhysioNet repository, [<https://archive.physionet.org/cgi-bin/atm/ATM>].

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Krishna BT (2011) Studies on fractional order differentiators and integrators: a survey. *Signal Process* 91(3):386–426
2. Barsainya R, Rawat TK, Kumar M (2017) Design of minimum multiplier fractional order differentiator based on lattice wave digital filter. *ISA Trans* 66:404–413
3. Nayak C, Saha SK, Kar R, Mandal D (2019) An efficient and robust digital fractional order differentiator based pre-processor design for QRS detection. *IEEE Trans Biomed Circuits Syst* 13(4):682–696
4. Ifeakor EC, Jervis BW (2002) *Digital signal processing—a practical approach*, 2nd ed. Pearson Education Limited, Delhi, pp 1–925
5. Cao B, Zhang Y, Zhao J, Liu X, Skonieczny L, Lv Z (2022) Recommendation based on large-scale many-objective optimisation for the intelligent Internet of things system. *IEEE Internet Things J* 9(16):15030–15038

6. Khishe M, Orouji N, Mosavi MR (2023) Multiobjective chimp optimiser: an innovative algorithm for multiobjective problems. *Expert Syst Appl* 211:118734
7. Zhao W, Zhang Z, Mirjalili S, Wang L, Khodadadi N, Mirjalili SM (2022) An effective multi-objective artificial hummingbird algorithm with dynamic elimination-based crowding distance for solving engineering design problems. *Comput Methods Appl Mech Eng* 398:115223
8. Kumar M, Rawat TK (2015) Optimal design of FIR fractional order differentiator using cuckoo search algorithm. *Expert Syst Appl* 42(7):3433–3449
9. Mirjalili S, Jangir P, Saremi S (2017) Multiobjective ant lion optimiser: a multiobjective optimisation algorithm for solving engineering problems. *Appl Intell* 46(1):79–95
10. Deng W, Zhang X, Zhou Y, Liu Y, Zhou X, Chen H, Zhao H (2022) An enhanced fast non-dominated solution sorting genetic algorithm for multiobjective problems. *Inf Sci* 585:441–453
11. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
12. Sarker R, Abbass HA (2004) Differential evolution for solving multiobjective optimisation problems. *Asia-Pacific J Oper Res* 21(2):225–240
13. Srinivas N, Deb K (1994) Multiobjective optimisation using nondominated sorting in genetic algorithms. *Evol Comput* 2(3):221–248
14. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
15. Coello CAC, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimisation. *IEEE Trans Evol Comput* 8(3):256–279
16. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimiser for engineering design problems. *Adv Eng Softw* 114:163–191
17. Aljarah I, Habib M, Faris H, Al-Madi N, Heidari AA, Mafarja M, Elaziz MA, Mirjalili S (2020) A dynamic locality multiobjective Salp Swarm algorithm for feature selection. *Comput Ind Eng* 147:106628
18. Alreshdedi SS, Lu S, Elaziz MA, Ewees AA (2019) Improved multiobjective salp swarm optimisation for virtual machine placement in cloud computing. *Hum Centric Comput Inf Sci* 9(15):1–24
19. Zhao Z, Xia C, Chi L, Chang X, Li W, Yang T, Zomaya AY (2021) An adaptive multiobjective salp swarm algorithm for efficient demand side management. In: IEEE 17th International conference on mobile ad hoc and sensor systems (MASS). <https://doi.org/10.1109/MASS50613.2020.00044>
20. Shyu JJ, Pei SC, Huang YD (2009) Least-squares design of variable maximally linear FIR differentiators. *IEEE Trans Signal Process* 57(11):4568–4573
21. Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE (2000) PhysioBank, PhysioToolkit, and PhysioNet components of a new research resource for complex physiologic signals. *Circulation* 101(23):e215–e220
22. Kaur A, Kumar S, Agarwal A, Agarwal R (2020) An efficient R-peak detection using Riesz fractional-order digital differentiator. *Circ Syst Signal Process* 39:1965–1987
23. Kaur K, Jindal N, Singh K (2021) Fractional Fourier transform-based Riesz fractional derivative approach for edge detection and its application in image enhancement. *Signal Process* 180:107852
24. Kaur K, Jindal N, Singh K (2019) Riesz fractional order derivative in fractional Fourier transform domain: an insight. *Digital Signal Process* 93:58–69
25. Nayak C, Saha SK, Kar R, Mandal D (2020) Efficient design of zero-phase Riesz fractional order digital differentiator using manta-ray foraging optimisation for precise electrocardiogram QRS detection. *IEEE Open J Circ Syst* 1:280–292
26. Tseng CC, Lee SL (2014) Digital image sharpening using Riesz fractional-order derivative and discrete Hartley transform. In: Proceedings of IEEE Asia Pacific conference on circuits and systems, pp 483–486
27. Kumar M, Rawat TK, Anand R, Karwayun R, Jain A (2016) Design of Riesz fractional-order differentiator using discrete sine transform. In: Proceedings of IEEE 3rd International conference on signal processing and integrated networks (SPIN). Noida, India, pp 702–706

28. Tseng CC, Lee SL (2014) Design of digital Riesz fractional-order differentiator. *Signal Process* 102:32–45
29. Nayak C, Saha SK, Kar R, Mandal D (2021) Optimal design of zero-phase digital Riesz FIR fractional order differentiator. *Soft Comput* 25:4261–4282
30. Nongpiur RC, Shpak DJ, Antoniou A (2014) Design of IIR digital differentiators using constrained optimisation. *IEEE Trans Signal Process* 62(7):1729–1739
31. Coello CAC (2009) Evolutionary multiobjective optimisation: some current research trends and topics that remain to be explored. *Front Comput Sci China* 3(1):18–30
32. Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of state of the art. *Swarm Evol Comput* 1(1):32–49
33. Han W, Li H, Gong M (2022) Automatic binary and ternary change detection in SAR images based on evolutionary multiobjective optimisation. *Appl Soft Comput* 125:109200
34. Sundaram A (2022) Multiobjective multi-verse optimisation algorithm to solve dynamic economic emission dispatch problems with transmission loss prediction by an artificial neural network. *Appl Soft Comput* 124:109021
35. Jha K, Saha S (2021) Incorporation of multimodal multiobjective optimisation in designing a filter-based feature selection technique. *Appl Soft Comput* 98:106823
36. Liang JJ, Qu BY, Suganthan PN, Hernandez-Daz AG (2012) Problem definitions and evaluation criteria for the CEC 2013 special session on real parameter optimisation. Technical Report, Zhengzhou University and Nanyang Technological University, Zhengzhou China and Singapore
37. Montgomery DC, Runger GC (2003) *Applied statistics and probability for engineers*, 3rd edn. Wiley, New York, USA
38. Nayak C, Saha SK, Kar R, Mandal D (2019) An optimally designed digital differentiator based pre-processor for R-peak detection in electrocardiogram signal. *Biomed Signal Process Control* 49:440–464
39. Nayak C, Saha SK, Kar R, Mandal D (2019) Optimal SSA-based wideband digital differentiator design for cardiac QRS complex detection application. *Int J Numer Model Electron Netw Devices Fields* 32(2):e2524
40. Nayak C, Saha SK, Kar R, Mandal D (2018) Automated QRS complex detection using MFO-based DFOD. *IET Signal Proc* 12(9):1172–1184
41. Nayak C, Saha SK, Kar R, Mandal D (2018) An efficient QRS complex detection using optimally designed digital differentiator. *Circ Syst Signal Process* 38:716–749

Chapter 13

Utilizing Cobb-Douglas Production Function in the Modeling of Joint Security and Quality of Service (QoS) in 5G Network



Tiew On Ting^{ID}, Su Fong Chien^{ID}, and Arash Bozorgchenani^{ID}

1 Introduction

In this modern era, the advent of 5G networks promises unprecedented connectivity, enabling a wide range of applications from autonomous vehicles to smart cities. However, this increased connectivity also exposes the network infrastructure to new security threats [1–3]. Intrusion detection systems (IDS) play a critical role in safeguarding 5G networks by identifying and mitigating numerous forms of threats. Simultaneously, maintaining high-quality service delivery is essential to meet user expectations, ensuring high throughput of the wireless network.

Traditionally, security and quality of service (QoS) have been treated as separate concerns. Security mechanisms, such as firewalls and encryption, are deployed to protect against attacks, while QoS mechanisms ensure efficient data transmission. However, the interplay between security and QoS is complex. For instance, deploying stringent security measures may impact network performance, leading to latency or packet loss. Conversely, prioritizing QoS over security may leave the network vulnerable to attacks, compromising its integrity.

In this context, our research aims to bridge the gap between security and QoS in 5G networks. By addressing the trade-offs between security and QoS, we pave the way for more resilient and responsive 5G networks. We propose a joint modeling

T. O. Ting (✉)

Department of Digital Media and Technology, Stony Brook Institute at Anhui University (SBIAHU), Anhui University, Hefei, Anhui Province, People's Republic of China
e-mail: toting@ahu.edu.cn

S. F. Chien

Multidisciplinary Research and Innovation Centre, MIMOS Berhad, Kuala Lumpur, Malaysia
e-mail: sf.chien@mimos.my

A. Bozorgchenani

School of Computing, University of Leeds, Leeds, UK
e-mail: a.bozorgchenani@leeds.ac.UK

approach that considers both aspects simultaneously. By adopting the notion of integrating security requirements with QoS constraints, we seek to enhance intrusion detection while minimizing the impact on service quality. Our work contributes to the development of efficient and adaptive intrusion response mechanisms that strike the right balance between security and user experience.

The rest of the book chapter is structured as follows. Section 2 presents the original formulation of joint security and QoS in 5G network. Section 3 reformulates the optimization model, taking the inspiration from the exponential term available in the Cobb-Douglas Production function. Furthermore, Sect. 4 presents the parameter sensitivity analysis for the two parameters of interest, namely α_1 and α_2 . In Sect. 5, we design a small case study and demonstrate feasible solutions via Binary Integer Programming and Quadratic Unconstrained Binary Optimization (QUBO) methodologies. Lastly Sect. 6 concludes the paper with suggestions to future works.

2 Original Formulation

The modeling starts by presenting the excerpts from the original formulation, adopted from previous work [4]. However, the firmware utility attacks were not considered in the work as only the software utility attacks were considered. In this work, both firmware attacks H and software attacks K are considered in the problem formulation. Prior to this, the knowledge of the nature of both attacks is essential to understanding the insights of the problem formulation presented in this work. As such, both threats are discussed in the following paragraphs.

Firmware is the low-level code that runs directly on hardware components, such as baseband processors in 5G devices or network infrastructure components like routers and base stations. Some common attacks involving firmware are:

- i. **Firmware Tampering**—Attackers may modify or replace the firmware on network devices,
- ii. **Firmware Contamination**—Attackers embed malicious code in the firmware during the manufacturing of the device, and
- iii. **Firmware Exploits**—Attackers exploit vulnerabilities in the firmware to gain control or execute arbitrary code.

In the context of 5G networks, the software includes the operating systems, applications, and protocols that manage and facilitate network operations and user interactions. The vast applications of the software also mean that it is more exposed to vulnerabilities, such as malware, ransomware, Denial-of-Service (DoS), Man-in-the-Middle (MitM) attacks.

Hereby, it is worthwhile noting that this work does not aim to present an accurate model representing the actual scenario. Rather, our focus here is to investigate the

effective solution to the optimization model by incorporating the Cobb-Douglas production function and conducting the parameter sensitivity analysis from the proposed model.

2.1 Security Function

A heterogeneous network architecture comprises of many network components such as smartphones, tablets, laptops, IoT devices, base stations, servers. Notably, these devices are susceptible to cyberattacks. To mitigate the exploding network attacks, the Security Agent (SA) is required to monitor, detect, and respond to security threats. These security agents are either software or hardware component deployed within an organization's IT infrastructure to protect systems by providing continuous surveillance and enforcing security policies.

Taking the inspiration from previous works in [4, 5], we consider the following security utility function:

$$\gamma_m^{\text{sec}}(L_m) = \frac{K'_{mn}(L_m)}{K + \bar{K}} + \frac{H'_{mn}(L_m)}{H + \bar{H}} \quad (1)$$

where

- L_m is an integer number representing the security level detected by an m th Security Agent (SA) where $L_m \in \{1, \dots, N\}$,
- $K'_{mn}(L_m)$ is the number of detected software attacks at L_m detection level,
- K is the number of known software attacks,
- $\bar{K}$ is the number of unknown software attacks,
- $H'_{mn}(L_m)$ is the number of detected firmware attacks at L_m detection level,
- H is the number of known firmware attacks, and
- $\bar{H}$ is the number of unknown firmware attacks.

In terms of network security, the aim here is to maximize (1), which entails finding the maximum number of attacks in a network. This ensures the high security of a 5G network against any potential cyberattacks.

2.2 Quality of Service (QoS) Function

QoS is a network measure that determines quality and ranks the priority of data traffic transmission on a network. It ensures various data traffic such as voice, video, and data, receive appropriate service levels. The QoS of a network consists of three main components, given by:

$$\gamma_m^{\text{QoS}}(L_m) = B(L_m) + \eta(L_m) + \Psi(L_m) \quad (2)$$

where

$B(L_m)$ is the bandwidth consumption (in unit [bps]) by a SA for the detection level L_m ,

$\eta(L_m)$ is the consumed computational resources (in unit [s]) by a SA for the detection level L_m , and

$\Psi(L_m)$ is the monetary cost (in unit [\$]) by a SA for the detection level L_m .

To acquire the equally contributing quantitative fraction of each component above, all three components are normalized using the ubiquitous formula:

$$\tilde{X} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (3)$$

Choosing $X_{\min} = 3.33$ and $X_{\max} = 33.33$ will constrain the normalized values within the range $3.33 \leq \tilde{X} \leq 33.33$, and thereby $10 \leq \gamma_m^{\text{QoS}}(L_m) \leq 100$. Hence, performing normalization to the three quantities in (1) results in:

$$\gamma_m^{\text{QoS}}(L_m) = \tilde{B}(L_m) + \tilde{\eta}(L_m) + \tilde{\Psi}(L_m) \quad (4)$$

We aim to minimize the expression (4) above. However, to be consistent with the maximization of the security function (1), the minimization of (4) is converted into a maximization through inversion, as follows:

$$\gamma_m^{\text{QoS}}(L_m) = \frac{1}{\tilde{B}(L_m) + \tilde{\eta}(L_m) + \tilde{\Psi}(L_m)} \quad (5)$$

2.3 Security and QoS Function

With respect to security and QoS, both objective functions are now combined into a single objective function by direct addition, which can be written as follows:

$$\gamma_m(L_m) = \gamma_m^{\text{sec}}(L_m) + \gamma_m^{\text{QoS}}(L_m) \quad (6)$$

Here, by introducing two positive weights, α and β ($\alpha + \beta = 1$), multiplying each weight to each of the terms in (6), we obtain the following function:

$$\gamma_m(L_m) = \alpha \gamma_m^{\text{sec}}(L_m) + \beta \gamma_m^{\text{QoS}}(L_m) \quad (7)$$

Further, the aforementioned optimization problem can be written as:

$$\underset{L_m}{\text{Maximize}} \left\{ \sum_{m=1}^M \gamma_m(L_m) \right\} \quad (8)$$

It is known that $\mathbf{L}_m \in \mathbb{R}^M$ is the decision vector that holds the detection level for the m^{th} SA. In this case, $n \in \{1, \dots, N\}$. The vector $\mathbf{L}_m$ can be further expanded into a two-dimensional matrix, i.e., $\mathbf{L} \in \mathbb{R}^{M \times N}$.

Next, considering the decision matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ where each element $x_{m,n} \in \{0, 1\}$ is a binary number. This means that when $x_{m,n} = 1$, the n^{th} detection level in matrix $\mathbf{L}_{m,n}$ is selected by the m^{th} SA.

Taking into account the binary decision variables, which is the inclusion of matrix $\mathbf{X}$ into (8), resulting in a complete optimization model:

$$\mathbf{P2} : \underset{\mathbf{X}, \mathbf{L}}{\text{Maximize}} \left\{ \sum_{m=1}^M \sum_{n=1}^N [\alpha \gamma_{m,n}^{\text{sec}}(L_{m,n}) + \beta \gamma_{m,n}^{\text{QoS}}(L_{m,n})] x_{m,n} \right\} \quad (9)$$

subject to

$$\mathbf{C1} \quad \alpha + \beta = 1$$

$$\mathbf{C2} \quad \sum_{l=1}^L x_{m,l} \leq 1 \quad \forall m,$$

where

C1 ensures total of the two coefficients to be unity, and

C2 assures that each SA is assigned either none or only one detection level.

The optimization problem above assigns the detection level to the SAs while considering the trade-off between the security detection efficiency and the QoS.

The following section illustrates basic principles of Cobb-Douglas production function and demonstrates how we can utilize this function to convert the optimization model depicted in (9) into a new optimization model. The advantages of this new model will be discussed in the same section.

3 Cobb-Douglas Production Function

The Cobb-Douglas production function is an econometric model that demonstrates how input affect production [6–9]. It is a specific functional form of the production function, widely used to represent the relationship between the input quantities, particularly physical capital and labor, and the resulting outputs. Developed and tested against statistical evidence by Charles Cobb and Paul Douglas between 1927 and 1947, it became known as the Cobb-Douglas production function [6, 7]. In a nutshell, the Cobb-Douglas production function can be written as

$$Q(K, L) = AK^\alpha L^\beta \quad (10)$$

where

- Q is the output produced from inputs K and L ,
- L is the labor investment: the amount of labor expended, which is typically expressed in hours,
- K represents the capital investment: the amount of physical capital input, such as the number of hours for a particular process,
- A denotes the total factor productivity (TFP) that measures the change in output that is independent of the inputs, and
- α, β are constants reflecting the output elasticity of the inputs, with $\alpha + \beta = 1$

From the production function above, it is observed that the Cobb-Douglas function has two distinctive parameters α and β that control the elasticity of two input factors, namely labor L and capital K . This is in-line with the two fundamental functions in the context of a secure 5G network, considering both security S and QoS (denoted as Q) in the problem formulation. The following subsection describe this in detail.

3.1 Novel Formulation of Security and QoS Based on Cobb-Douglas Theory

By adopting a similar strategy as in our previous work [5], we reformulate the optimization model in (9) based on Cobb-Douglas theory. Upon comparison of the Cobb-Douglas function with (9), we highlight some interesting observations below:

- i. There are two main components in both functions (9) and (10): security & QoS and K & L ,
- ii. Both functions contain coefficients α and β , which play similar roles in general, and
- iii. In both functions, $\alpha + \beta = 1$.

We start by applying the natural log to (10),

$$\begin{aligned}
 \ln Q(K, L) &= \ln (AK^\alpha L^\beta) \\
 &= \ln A + \ln K^\alpha + \ln L^\beta \\
 &= \ln A + \alpha \ln K + \beta \ln L
 \end{aligned} \tag{11}$$

By comparing (11) to (7),

$$\begin{aligned}
 \ln A &= 0 \\
 A &= e^0 \\
 A &= 1
 \end{aligned}$$

$$\begin{aligned}
\alpha \ln K &= \alpha \gamma_m^{\text{sec}}(L_m) \\
\ln K &= \gamma_m^{\text{sec}}(L_m) \\
K &= \exp(\gamma_m^{\text{sec}}(L_m))
\end{aligned}$$

$$\begin{aligned}
\beta \ln L &= \beta \gamma_m^{\text{QoS}}(L_m) \\
\ln L &= \gamma_m^{\text{QoS}}(L_m) \\
L &= \exp(\gamma_m^{\text{QoS}}(L_m))
\end{aligned}$$

As such, substituting the values above into (10) gives the following:

$$Q(L_m) = [\exp(\gamma_m^{\text{sec}}(L_m))]^\alpha [\exp(\gamma_m^{\text{QoS}}(L_m))]^\beta \quad (12)$$

3.2 New Optimization Model Based on Cobb-Douglas Function

Referring to original optimization model in (9), the maximization model can thus be revised as:

$$\mathbf{P2} : \text{Max}_X \left\{ \sum_{m=1}^M \sum_{n=1}^N \left[\left(\frac{K'_{mn}}{K + \bar{K}} + \frac{H'_{mn}}{H + \bar{H}} \right)^{\alpha'_1} \times \left(\frac{1}{\tilde{B} \times \tilde{\eta}_{mn} \times \tilde{\Psi}_{mn}} \right)^{\alpha'_2} \right] x_{mn} \right\} \quad (13)$$

$$\mathbf{C1.1} : \sum_{n=1}^N x_{mn} \leq 1, \quad \forall m$$

$$\mathbf{C1.2} : \sum_{n=1}^N \left(\frac{H'_{mn}}{H + \tilde{H}} \right) x_{mn} \geq \delta_s, \quad \forall m$$

$$\mathbf{C1.3} : \sum_{n=1}^N B_{mn} x_{mn} \leq \delta_b, \quad \forall m$$

$$\mathbf{C1.4} : \sum_{n=1}^N \eta_{mn} x_{mn} \leq \delta_\eta, \quad \forall m$$

$$\mathbf{C1.5} : \sum_{n=1}^N \Psi_{mn} x_{mn} \leq \delta_\Psi, \quad \forall m$$

$$\mathbf{C1.6} : x_{mn} \in \{0, 1\}, \quad \forall m \text{ and } n$$

where $\mathbf{X} \in \mathbb{R}^{M \times N}$ is the decision matrix where each element is binary (i.e., $x_{mn} \in \{0, 1\}$) representing the n th detection level for the m th SA. The decision vector $\mathbf{L} \in \mathbb{R}^M$ represents the detection level for the SAs, and coefficients α_1 and α_2 are specified in general form,

$$\alpha'_i = \frac{\alpha_i}{1 - \alpha_i} \quad (14)$$

Here, the summation $\alpha_1 + \alpha_2 = 1$. From the optimization model (13), we observe the following:

- i. The software and firmware securities are both a decimal value well below 1.
- ii. The second term is a value within one as all the QoS factors ($\tilde{B}_{mn}$, $\tilde{\eta}_{mn}$, and $\tilde{\Psi}_{mn}$) are values larger than one. We can keep the initial normalization in [3.3–33.3] for each component, thereby ensuring the second term to be well below zero.

We would like to note that the mathematical model presented here is slightly different from model in our previous work, given by equation (14) in [5]. Note that the preliminary terms in (13) are added instead of multiplied as in [5]. Based on our analysis, this minor difference does not affect the final solution obtained.

4 Graphical Parameter Analysis

This section investigates the sensitivity of relevant parameters to better understand their effect in the new optimization model.

4.1 Analysis on α parameter

Here, a graphical analysis of the impact of α is conducted to demonstrate its effect on the optimization problem given in (9), considering both α'_1 and α'_2 .

$$\alpha'_i = \frac{\alpha_i}{1 - \alpha_i} \quad (15)$$

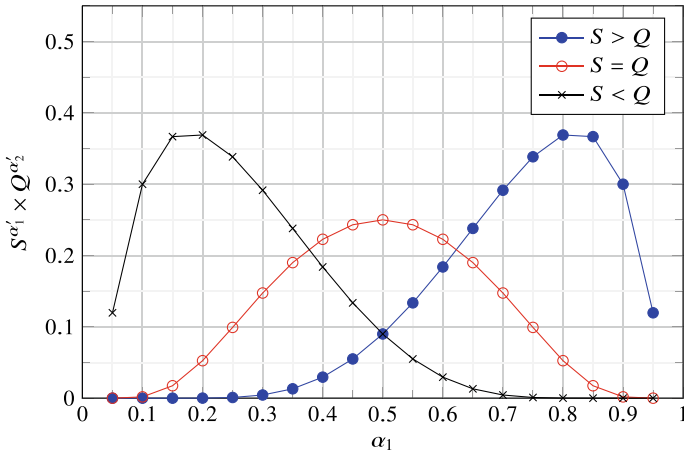
The values of α'_i are calculated based on the formula above for the respective α_i values, as depicted in Table 1. For simplicity, we assume the following simplified notations:

$$S = \frac{K'_{mn}}{K + \bar{K}} + \frac{H'_{mn}}{H + \bar{H}}$$

$$Q = \frac{1}{\tilde{B} \times \tilde{\eta}_{mn} \times \tilde{\Psi}_{mn}}$$

Table 1 Respective α'_1 and α'_2 values calculated from (15)

α_1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
α'_1	1.00	0.11	0.25	0.43	0.67	1.00	1.50	2.30	4.00	9.00	∞
α_2	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0
α'_2	∞	9.00	4.00	2.30	1.50	1.00	0.67	0.43	0.25	0.11	1.00

**Fig. 1** Observations for α_1 versus $S^{\alpha'_1} \times Q^{\alpha'_2}$ for three conditions of S and Q pairs

Hence, the core part of (9) is then

$$S^{\alpha'_1} \times Q^{\alpha'_2} \quad (16)$$

Furthermore, we are interested in observing the effect of α_1 versus $S^{\alpha'_1} \times Q^{\alpha'_2}$ under three different conditions:

- i. $S > Q$: Let $S = 0.9$ and $Q = 0.1$
- ii. $S = Q$: Let $S = 0.5$ and $Q = 0.5$
- iii. $S < Q$: Let $S = 0.1$ and $Q = 0.9$

One important observation from the three curves depicted in Fig. 1 is that all three plots are concave, each having a maximum point. This leads to the insight that we can locate a maximum point at the turning point where

$$\frac{d(S^{\alpha'_1} \times Q^{\alpha'_2})}{d\alpha_1} = 0$$

With this strategy, we obtain the optimal value of α_1 as provided in (27). The proof from the derivation process is presented in Sect. 4.3.

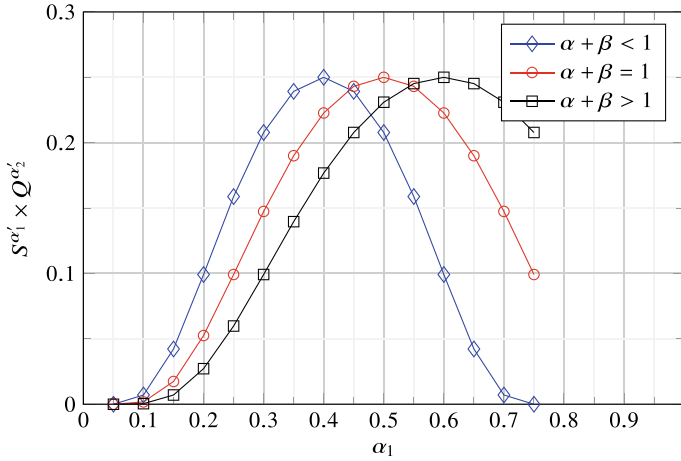


Fig. 2 Three elasticity scenarios in Cobb-Douglas function

4.2 Output Elasticity Based on Cobb-Douglas Function

Output elasticity in the context of the Cobb-Douglas production function and quality of service (QoS) in network systems is highly relevant because it helps understand how changes in inputs affect outputs and, consequently, the quality of the service provided. In the Cobb-Douglas production function, elasticity measures how responsive an output is to changes in inputs, specifically:

- i. **Output Elasticity of Capital (α)**—This measures the change in output resulting from the change in the capital input.
- ii. **Output Elasticity of Labor (β)**—This measures the change in output resulting from the change in the labor input.

Further, we investigate the elasticity of the output based on α and β . Output elasticity measures the responsiveness of output to a change of (10) by either capital K or labor L in production, as interpreted below:

- i. Elasticity 1: $\alpha + \beta = 1$, scaling of inputs K and L will also scale the output. This means doubling usage of capital K and labor L function display constant returns to scale, meaning that doubling the usage of capital K and labor L will also double its output.
- ii. Elasticity 2: $\alpha + \beta < 0$, returns to scale are decreasing. It means that the increase in capital K and labor L will produce slightly less output.
- iii. Elasticity 3: $\alpha + \beta > 0$, returns to scale are increasing. It means that the increase in capital K and labor L will produce slightly higher output.

The graphs in Fig. 2 are drawn based on different values of $\alpha + \beta$: $\alpha + \beta = 1$, $\alpha + \beta = 0.8$, and $\alpha + \beta = 1.2$. This analysis adopts relevant values of $S = 0.5$ and $Q = 0.5$. The interpretation from the observations of the three graphs is that increasing $\alpha + \beta$ shifts the maximum point to the right side.

4.3 The Optimal α_i Parameter Value

By equating both α'_1 and α'_2 in terms of α_1 , from (15), we know that

$$\begin{aligned}\alpha'_1 &= \frac{\alpha_1}{1 - \alpha_1} \\ \alpha'_2 &= \frac{\alpha_2}{1 - \alpha_2}\end{aligned}\tag{17}$$

Substituting $\alpha_2 = 1 - \alpha_1$, we obtain

$$\begin{aligned}\alpha'_2 &= \frac{1 - \alpha_1}{1 - (1 - \alpha_1)} \\ &= \frac{1 - \alpha_1}{\alpha_1}\end{aligned}\tag{18}$$

Now, assuming

$$\begin{aligned}S &= \frac{K'_{mn}}{K + \bar{K}} + \frac{H'_{mn}}{H + \bar{H}} \\ Q &= \frac{1}{\tilde{B} \times \tilde{\eta}_{mn} \times \tilde{\Psi}_{mn}}\end{aligned}$$

Thus we have the following functions

$$\begin{aligned}s(\alpha_1) &= S^{\frac{\alpha_1}{1-\alpha_1}} \\ q(\alpha_1) &= Q^{\frac{1-\alpha_1}{\alpha_1}}\end{aligned}$$

The derivation for the above functions are

$$\begin{aligned}s'(\alpha_1) &= S^{\frac{\alpha_1}{1-\alpha_1}} \cdot \ln S \cdot \frac{1}{(1 - \alpha_1)^2} \\ &= \frac{S^{\frac{\alpha_1}{1-\alpha_1}}}{(1 - \alpha_1)^2} \cdot \ln S \\ q'(\alpha_1) &= Q^{\frac{1-\alpha_1}{\alpha_1}} \cdot \ln Q \cdot \left(-\frac{1}{\alpha_1^2}\right) \\ &= -\frac{Q^{\frac{1-\alpha_1}{\alpha_1}}}{\alpha_1^2} \cdot \ln Q\end{aligned}$$

Next, we substitute $s(\alpha_1)$, $Q(\alpha_2)$, (17), and (18) into (13), resulting in

$$f(\alpha_1) = S^{\frac{\alpha_1}{1-\alpha_1}} \times Q^{\frac{1-\alpha_1}{\alpha_1}}$$

To find the maximum value of $f(\alpha_1)$, we find its derivative, which is equivalent to zero. Applying the chain rule for the derivation of (19)

$$\begin{aligned} \frac{df(\alpha_1)}{d\alpha_1} &= S^{\frac{\alpha_1}{1-\alpha_1}} (q') + Q^{\frac{1-\alpha_1}{\alpha_1}} (s') \\ &= S^{\frac{\alpha_1}{1-\alpha_1}} \left(-\frac{Q^{\frac{1-\alpha_1}{\alpha_1}}}{\alpha_1^2} \cdot \ln Q \right) + Q^{\frac{1-\alpha_1}{\alpha_1}} \left(\frac{S^{\frac{\alpha_1}{1-\alpha_1}}}{(1-\alpha_1)^2} \cdot \ln S \right) \\ &= \left(-\frac{S^{\frac{\alpha_1}{1-\alpha_1}} Q^{\frac{1-\alpha_1}{\alpha_1}}}{\alpha_1^2} \cdot \ln Q \right) + \left(\frac{S^{\frac{\alpha_1}{1-\alpha_1}} Q^{\frac{1-\alpha_1}{\alpha_1}}}{(1-\alpha_1)^2} \cdot \ln S \right) \end{aligned} \quad (19)$$

To find the optimal α_1 , the (19) above is equivalent to zero, results in

$$\frac{S^{\frac{\alpha_1}{1-\alpha_1}} Q^{\frac{1-\alpha_1}{\alpha_1}}}{(1-\alpha_1)^2} \cdot \ln S = \frac{S^{\frac{\alpha_1}{1-\alpha_1}} Q^{\frac{1-\alpha_1}{\alpha_1}}}{\alpha_1^2} \cdot \ln Q \quad (20)$$

Simplifying the equation, noting that the terms with S and Q can be canceled out on both sides, the following equation is obtained:

$$\frac{\ln S}{(1-\alpha_1)^2} = \frac{\ln Q}{\alpha_1^2} \quad (21)$$

Next, moving all squared terms to the left side results in

$$\left(\frac{\alpha_1}{1-\alpha_1} \right)^2 = \frac{\ln Q}{\ln S} \quad (22)$$

Perform square root to both sides,

$$\frac{\alpha_1}{1-\alpha_1} = \sqrt{\frac{\ln Q}{\ln S}} \quad (23)$$

Multiply both sides by $(1-\alpha_1)$ gives

$$\alpha_1 = (1-\alpha_1) \sqrt{\frac{\ln Q}{\ln S}} \quad (24)$$

Expanding the term on the right side,

$$\alpha_1 = \sqrt{\frac{\ln Q}{\ln S}} - \alpha_1 \sqrt{\frac{\ln Q}{\ln S}} \quad (25)$$

Then collecting the similar factor α_1 on the left side results in

$$\alpha_1 \left(1 + \sqrt{\frac{\ln Q}{\ln S}} \right) = \sqrt{\frac{\ln Q}{\ln S}} \quad (26)$$

Finally, we obtain the factor α_1 ,

$$\alpha_1 = \frac{\sqrt{\frac{\ln Q}{\ln S}}}{\left(1 + \sqrt{\frac{\ln Q}{\ln S}} \right)} \quad (27)$$

In order to prove the maximum point, we substitute $S = Q$ and obtain $\alpha_1 = 0.5$. From this outcome, it is evident from Fig. 1 that this is indeed the maximum point.

5 Results and Discussions

We adopt Quadratic Unconstrained Binary Optimization (QUBO) approach in finding a feasible solution to the formulated problem. Prior to the solution, a small case study representing the real world scenario is designed and presented in this section.

5.1 Quadratic Unconstrained Binary Optimization (QUBO)

QUBO is a mathematical formulation used in the field of quantum computing and optimization problems [10–15]. It represents optimization problems in a specific format suited for quantum annealing devices, such as D-Wave's quantum annealers. In a QUBO formulation, the goal is to minimize a quadratic objective function subject to binary variables. These binary variables can only take on two values, typically denoted as 0 and 1. The general form of a QUBO problem is as follows:

$$\min_x \left(\sum_i^n Q_i x_i + \sum_i^n Q_{ij} x_i x_j \right) \quad (28)$$

where

- x_i are the binary variables,
- Q_i are the linear coefficients,
- Q_{ij} are the quadratic coefficients.

The objective in this optimization model is to find the assignment of binary values x_i that minimizes the overall value of the quadratic expression.

QUBO problems arise in various fields, including combinatorial optimization, machine learning, and logistics, among others. Quantum annealers, such as those provided by D-Wave, are specialized devices designed to solve QUBO problems efficiently by exploiting quantum effects like quantum tunneling and entanglement. Researchers and practitioners often transform optimization problems from their original form into QUBO form to leverage quantum annealing capabilities or other optimization techniques suited for QUBO problems.

To solve the function in the previous subsection $f(x) = 4.00x_1 + 3.63x_2 + 3.08x_3 + 2.86x_4 + 2.67x_5$ using QUBO, we need to express it in quadratic form, then construct the corresponding QUBO matrix.

First, we need to identify the quadratic terms in the function. Since there are no quadratic terms (no x_i^2 terms), we only have linear terms. To represent linear terms in a QUBO matrix, we need to square them. This allows us to create a quadratic function in terms of binary variables. We introduce penalty terms to ensure that each variable is either 0 or 1. Here's how it is accomplished:

- i. Square each linear term:

$$\begin{aligned}(4.00x_1)^2 &= 16.00x_1^2, \\ (3.63x_2)^2 &= 13.1769x_2^2, \\ (3.08x_3)^2 &= 9.4864x_3^2, \\ (2.86x_4)^2 &= 8.1796x_4^2, \\ (2.67x_5)^2 &= 7.1289x_5^2\end{aligned}$$

- ii. Introduce penalty terms for each variable to ensure they are binary (0 or 1):
 $-2x_1 + 2x_2^2 - 2x_3^2 + 2x_4^2 - 2x_5^2$

Now, we combine all terms to form the QUBO function:

$$f(x) = 16x_1^2 + 13.1769x_2^2 + 9.4864x_3^2 + 8.1796x_4^2 + 7.1289x_5^2 - 2x_1 + 2x_2 - 2x_3 + 2x_4 - 2x_5$$

To represent this in a QUBO matrix form, we set the coefficients of the quadratic terms along the diagonal of the matrix, and the linear coefficients as the off-diagonal elements. Since there are no cross-product terms, the off-diagonal elements will be zero.

$$Q = \begin{bmatrix} 16 & 0 & 0 & 0 & 0 \\ 0 & 13.1769 & 0 & 0 & 0 \\ 0 & 0 & 9.4864 & 0 & 0 \\ 0 & 0 & 0 & 8.1796 & 0 \\ 0 & 0 & 0 & 0 & 7.1289 \end{bmatrix}$$

The linear terms can be stored in a separate vector, h :

$$h = [-2, 2, -2, 2, -2]$$

With this QUBO matrix Q and linear vector h , the QUBO solver is utilized to find the optimal solution $\mathbf{X} = [0, 0, 0, 0, 1]$ and $f(\mathbf{X}) = 2.67$, which is similar to the solution by Binary Integer Programming.

6 Conclusions

In this work, we present the formulation of the optimization problem, considering both joint security and QoS utility functions based on Cobb-Douglas production function theory. Subsequently, we conduct a series of parameter analyses to investigate the effect on the cost function with variations in relevant parameter values (α_1 and α_2). Additionally, we provide a mathematical proof for the optimality of α_1 . Before concluding, we present a small case study demonstrating a feasible solution using the Binary Integer Programming method and the prominent Quadratic Unconstrained Binary Optimization (QUBO) strategy.

References

1. Liu Y, Wang Y, Li K, Chen W, Xia F (2020) A survey of intrusion detection systems in 5G networks: challenges and solutions. *IEEE Access* 8:139687–139700
2. Alabbas H, Al-Khatib WG, Liew CS, Albeshri A (2020) A survey of intrusion detection systems in 5G: solutions, challenges, and opportunities. *IEEE Access* 8:35151–35171
3. Baydoun M, Awad M (2020) A survey on security and privacy in 5g networks: potential solutions, recent advances and future directions. *IEEE Access* 8:153863–153879
4. Bozorgchenani A, Zarakovitis C, Chien SF, Lim HS, Ni Q, Gouglidis A, Mallouli W (2022) Joint security-vs-qos framework: optimizing the selection of intrusion detection mechanisms in 5G networks. In: *ARES 2022: The 17th International conference on availability, reliability and security*, pp 1–6
5. Bozorgchenani A, Zarakovitis CC, Chien SF, Ting TO, Ni Q, Mallouli W (2023) Novel modeling and optimization for joint cybersecurity-vs-qos intrusion detection mechanisms in 5G networks. *Comput Netw* 237:110051
6. Cobb CW, Douglas PH (1928) A theory of production. *Am Econ Rev* 18(1):139–165
7. Douglas PH (1948) Cobb-Douglas production functions. *Rev Econ Stat* 30(4):312–320
8. Solow RM (1957) Technical change and the aggregate production function. *Rev Econ Stat* 39(3):312–320
9. Mankiw NG, Romer D, Weil DN (1992) A contribution to the empirics of economic growth. *Q J Econ* 107(2):407–437
10. Smith JE, Stecklein GA, Amin MH (2018) Quantum annealing and computation: a brief documentary note. *arXiv preprint* [arXiv:1806.01554](https://arxiv.org/abs/1806.01554)
11. Lucas A (2014) Quadratic unconstrained binary optimization (qubo) formulation. *arXiv preprint* [arXiv:1406.0774](https://arxiv.org/abs/1406.0774)
12. Michalewicz Z, Fogel DB (2004) Quadratic unconstrained binary optimization. *Institut fur Angewandte Mathematik, Universitat Heidelberg*, pp 635–640
13. King AD, Raymond C, Tang E, Egger DJ (2018) Quantum optimization using Grover's algorithm on near-term devices. *arXiv preprint*: [arXiv:1805.05224](https://arxiv.org/abs/1805.05224)

14. Bunyk P, Hoskinson EM, Johnson MW, Tolkacheva E, Altomare F, Mastrangelo C, Neufeld R, Rose G (2014) Solving quadratic unconstrained binary optimization (qubo) problems with quantum annealers. arXiv preprint [arXiv:1401.2910](https://arxiv.org/abs/1401.2910)
15. Denchev VS, Ding N, Isakov SV, Lanting T, Martin RO, Amin M, Smelyanskiy M, Williams C (2016) A review of quantum annealing. *Quantum Sci Technol* 1(1):01LT01

Chapter 14

Enhancing Low-Dose SPECT 2D Reconstruction from One-Dimensional Projections By Using Conditional Generative Adversarial Networks for Improved Image Quality



Adis Alihodzic 

1 Introduction

SPECT is a medical procedure that utilizes single-photon emission computed tomography [6, 19, 46]. This imaging technique was established on gamma-ray projection and is across the board in clinical practice and research. In general, it is well-known that gamma rays from SPECT devices induce a confident capacity of danger to human health and seriously increase the chance of cancer in people [6]. Consequently, in clinical practice, it is strongly advised that the radiation dose be reduced to the lowest possible level [5]. Deflating radioactive radiation indicates generating poorer projection data that include a smaller number of photons, which ultimately results in the reconstructed images being affected by high Poisson noise, extreme antiques, and small resolution, directly affecting the diagnostic accuracy [49]. Hence, it is a very challenging problem in SPECT to achieve a high-quality reconstructed image while keeping the radiation dose low. Algorithms used for 2D image reconstruction from 1D projections have been divided into two groups. The first group consists of algorithms based on mathematical analytic techniques, such as Filtered back projection [3] and Fourier's slice theorem [14], while the second group of algorithms refers to algebraic iterative reconstruction techniques [7, 22, 38, 42]. Until recently, filtered back projection was the prevailing method for reconstructions in X-ray CT and most SPECT and PET reconstructions [13, 25, 50]. The disadvantages of filtered back projection algorithms are that they require much scanning and filtering, i.e., it is hard to establish a compromise between blur and noise [21]. Alternatively, algebraic reconstruction algorithms based on iterations are more adaptable but less efficient than filtered-back projection algorithms. Compared to analytical techniques for image reconstruction (such as filtered back projection), the algebraic iterative reconstruction techniques perform reconstruction with a lower dose of radiation, so

A. Alihodzic (✉)

Department of Mathematical and Computer Sciences, University of Sarajevo,
Zmaja od Bosne, 33-35 Sarajevo, Bosnia and Herzegovina
e-mail: adis.alihodzic@pmf.unsa.ba

the quality of the reconstructed image is kept. However, some studies show that these techniques still generate images in which the loss of low-contrast abnormalities was observed, as well as changes in the appearance, streak artifacts, and structure of the image itself [29]. In addition to the mentioned deterministic algorithms for tomography, metaheuristics are also used in the literature, such as genetic algorithm [45] and evolutionary algorithm [4]. In the last few years, in addition to the mentioned approaches, several other modifications have been proposed to obtain better reconstructions at low doses of radiation, that is, for a small number of projections [11, 15, 27, 44]. Guided by the ALARA (as low as reasonably achievable) paradigm, artificial intelligence algorithms have been exploited in medical imaging in the last few years to achieve the best possible quality of the reconstructed image at low radiation doses. These algorithms promise to preserve the quality of the reconstructed image for very low doses of radiation. These techniques can be classified into three classes [40, 41]. The first class contains algorithms combining traditional iterative reconstruction algorithms with deep neural networks to create new architectures of deep neural networks [52]. The second class of algorithms is related to the non-filtering or pre-processing of projection data (sinograms) utilizing deep neural networks to decrease noise on the original sinogram and accomplish the reconstruction of the enhanced (filtered) sinogram by operating classic approaches of image tomography [12, 30]. The third class uses reconstructed images obtained at low radiation doses as input to a convolution network to generate high-resolution images. This class is characterized by the exploitation algorithms, such as deep convolutional neural networks (DCNN) [10, 26], DenseNet and deconvolution [53], a deep encoder-decoder (EA) network [9, 23], and generative adversarial networks (GAN) [17, 51] and variants [34]. The ensuing sections in this paper are positioned as obeys. In Sect. 2, the techniques for image reconstruction are presented in detail. Section 3 illustrates the problem of SPECT reconstruction and how generative adversarial networks (GANs) and conditional generative adversarial networks (CGANs) are involved in SPECT reconstruction at low radiation doses. In Sect. 4, an experimental analysis related to data preparation, evaluation of the obtained results, and their comparison with conventional algorithms for low-dose image reconstruction was performed. This section has also been devoted to discussing the outcomes achieved. The last section is booked for a summary of the chapter.

2 Conventional Techniques for Image Reconstruction

2.1 Filtered Back Projection

Traditional techniques for 2D image reconstruction from 1D projections (sinogram) initially employed back projection $B(x, y)$ in medical image analysis (CT, SPECT, et al.) [3, 20, 37]. Back projection is defined related the sinogram, i.e., the Radon transformation $\hat{f}(\rho, \theta)$ as follows

$$B(x, y) = \int_0^\pi \tilde{f}(x \cos \theta + y \sin \theta) d\theta, \quad (1)$$

where the parameters ρ and θ are utilized to determine the location of the line l pierced by the parallel lines (rays) perpendicular to it. Since back projection generates visible image blur, filtered back projection (FBP) is operated instead based on high-pass filters, such as Ram-Lak, Butterworth, Hann, Shepp-Logan, and Cosine [3]. Although iterative methods for image reconstruction are preferred for better reconstruction quality and noise reduction, filtered back projection is still used because of its speed, simple implementation, shallow use of computer resources, and more straightforward mathematical interpretation compared to iterative methods. The FBP is defined by

$$f(x, y) = \int_0^\pi \bar{\tilde{f}}(x \cos \theta + y \sin \theta, \theta) d\theta, \quad (2)$$

where operator $\bar{\tilde{f}}$ is given by

$$\bar{\tilde{f}}(\rho, \theta) = \frac{1}{\sqrt{2\pi}} \mathcal{F}_{s \rightarrow \rho}^{-1} \left[|s| \mathcal{F}_{\rho \rightarrow s}(\tilde{f}(\rho, \theta)) \right], \quad (3)$$

while $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the 1D and the inverse 1D Fourier transforms respectively, and $|s|$ is high-frequency filter.

2.2 Iterative Reconstruction Techniques

Unlike filtered back projection (FBP), iterative reconstructive techniques consume incomparably more resources and are usually evaluated on graphics processing units. However, in most cases, these techniques successfully solve the problems caused by FBP techniques. One frequently used iterative technique for image reconstruction is the SIRT (Simultaneous Iterative Reconstruction Technique) [8, 18]. This technique iteratively updates the image so the reconstruction rate is gradually sweetened from iteration to iteration. The description of the steps of this technique is given in Algorithm 1.

Algorithm 1: Pseudocode of SIRT technique

Input: Projection data, i.e., the sinogram $\mathbf{p}$ of the image being reconstructed.

Step 1: Initialization of the reconstruction image $\mathbf{x}^0$ to arbitrary gray level values, e.g., all pixels of the image $\mathbf{x}^0$ can have an intensity equal to zero. The number of iterations n and the convergence parameter α are set to predefined values.

Step 2: for $i = 0, 1, 2, \dots, n$ **do**

- a) Projection calculation $\mathbf{p}^{(i)} = \mathbf{W}\mathbf{x}^{(i)}$, where the matrix $\mathbf{W}$ is the image transformation in the projection space.
- b) The determination of the error $\mathbf{e}^i = \mathbf{p} - \mathbf{p}^i$ as the difference between the original $\mathbf{p}$ and the estimated sinogram $\mathbf{p}^i$.
- c) Iterative update of the reconstruction image $\mathbf{x}^i$ by using the formula

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha \mathbf{W}^T \mathbf{e}^i.$$

Output: The reconstructed image $\mathbf{x}^{n+1}$ due to the projection data $\mathbf{p}$.

In the pseudo-code of Algorithm 1, the application of the matrix $\mathbf{W}$ to the image $\mathbf{x}^{(i)}$ represents a forward projection, while the multiplication of the matrix $\mathbf{W}^T$ with the error $\mathbf{e}^{(i)}$ refers to the inverse operation, i.e., to the weighted back projection.

2.3 Iterative Statistical Reconstruction Techniques

In this section, statistical iterative reconstruction techniques will be described. With these techniques, projection data are not assumed to be modeled by line integrals, as with analytical methods (filtered back projection). During reconstruction, these methods consider factors such as the SPECT or PET device's attenuation, scattering, and detector resolution. It is known that detectors produce fogging or noise during operation, which is usually modeled using some distribution. For example, the Poisson distribution is used in the OSEM (Ordered Subset Expectation Maximization) method [54]. Since the OSEM algorithm is a modification of the MLEM (Maximum Likelihood Expectation Maximization) algorithm, the MLEM algorithm will be first described below [43]. Let it be necessary to find the reconstruction image $\mathbf{x}$ if the weight matrix $\mathbf{A}$ and the measured data (projections) $\mathbf{p}$ are known. The matrix $\mathbf{A}$ tells about the probability of detecting an event through a particular projection. More specifically, it contains information about how each pixel in the image contributes to the measurement at each detector for each projection angle. Namely, for each pair of detectors and pixels in the image, the value in the matrix tells how much the pixel contributes to the measurement at that detector. It can be determined by factors such as the distance between the pixel and the detector, the absorption of photons as they pass through the tissue, and other characteristics of the SPECT scanning system. A Poisson distribution with a mean value of $\mathbf{A}\mathbf{x}$ is always associated with measurements or projections $\mathbf{p}$. Approximate determination of the image $\mathbf{x}$ is reduced to solving the following optimization problem using maximum likelihood estimation

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{4}$$

where f is the negative log-likelihood function defined by

$$f(\mathbf{x}) = \mathbf{1}^T \mathbf{A}\mathbf{x} - \mathbf{p}^T \ln \mathbf{A}\mathbf{x}. \quad (5)$$

Since the function $f(\mathbf{x})$ is convex, each of its local minimums is also global on the domain where it is defined, so searching for the reconstruction image $\mathbf{x}$ is reduced to the use of the projected subgradient method

$$\mathbf{x}^{k+1} = [\mathbf{x}^k - \alpha^k \nabla f(\mathbf{x}^k)]_+ \quad (6)$$

where the parameter α indicates the speed of convergence, i.e., the step size of the algorithm, while the operator $[\cdot]_+$ is used to set the negative elements of the vector to zero. The gradient $\nabla f(\mathbf{x}^k)$ of the function f at the point $\mathbf{x}^k$ is given by

$$\nabla f(\mathbf{x}^k) = A^T (\mathbf{1} - \mathbf{p}/\mathbf{A}\mathbf{x}^k). \quad (7)$$

The equation that describes the classic MLEM algorithm is obtained when the gradient $\nabla f(\mathbf{x}^k)$ is included from the last equation in Eq. (6), and for the step size α^k in k iteration takes this estimate $\alpha^k = \mathbf{x}^k / \mathbf{A}^T \mathbf{1}$. Thus, the reconstruction of image $\mathbf{x}$ is iteratively found using the following MLEM algorithm

$$\mathbf{x}^{k+1} = c \frac{\mathbf{x}^k}{\mathbf{A}^T \mathbf{1}} \mathbf{A}^T \frac{\mathbf{p}}{\mathbf{A}\mathbf{x}^k}, \quad (8)$$

where c represents the scaling constant, while the expressions both $\mathbf{p}/\mathbf{A}\mathbf{x}^k$ and $\mathbf{x}^k \mathbf{A}^T \mathbf{p}/\mathbf{A}\mathbf{x}^k$ represent forward projection and back projection, respectively.

The OSEM algorithm improves the original MLEM algorithm, which relies on enhancing the speed and efficiency of iterative image reconstruction. The OSEM algorithm, unlike the MLEM algorithm, instead of utilizing all projections at once, splits the data into smaller subsets, which are usually chosen and established on some order, for example, even and odd projections. OSEM algorithm, exploiting smaller subsets, encourages quicker convergence of the algorithm. In addition, the division of projections into subsets supports their independent processing, so their processing can be parallelized and decrease the reconstruction time, ultimately affecting the overall time needed to acquire the final image. Also, the OSEM algorithm can possess additional regularization techniques to improve the stability and quality of image reconstruction. It is usually more flexible than the classic MLEM algorithm and can be adapted for different detector geometries or different characteristics of the measured data (projections). In this paper, for comparative analysis, several types of statistical iterative techniques will be exploited, such as OSEM (Ordered Subset Expectation Maximization) [54], BSREM (Block Sequential Regularized Expectation Maximization) [31, 47] and KEM (Kernel expectation maximum) [16, 48].

3 Application of Conditional Generative Adversarial Networks (CGANs) for SPECT Reconstruction

3.1 Generative Adversarial Networks (GANs)

Generative adversarial networks (GANs) are unsupervised training models in which the generator G (neural network) tries to learn patterns in data to produce new (unseen) data. These networks consist of two parts (generator and discriminator) that fight against each other. The generator $G(\cdot)$ at the input receives a vector of random values $\mathbf{z} \sim p_{\mathbf{z}}$ and transforms it into a fake (generated) data $G(\mathbf{z}) \sim p_g$ which tries to approximate the accurate data $\mathbf{x} \sim p_{\text{data}}$. The discriminator $D(\cdot, \cdot)$ acquires accurate data $\mathbf{x} \sim p_{\text{data}}$ and generated data $G(\mathbf{z}) \sim p_g$, and it tries to determine which of them are generated (false) or real. It indicates that if the generator $G(\mathbf{z})$ deceives the discriminator $D(\mathbf{x}, G(\mathbf{z}))$, the discriminator (binary classifier) will not classify the generated data as false, so it must be further improved. On the other hand, whenever the discriminator succeeds in distinguishing the generated from real data, the generated data must be updated (fixed) because it is not good enough. Since the discriminator is a binary classifier, it can be found by maximizing the loss function (binary cross-entropy) as follows

$$\max_D \left[\log D(\mathbf{x}) + \log(1 - D(G(\mathbf{z}))) \right]. \quad (9)$$

Knowing that the generator $G(\cdot)$ wants to trick the discriminator $D(\cdot, \cdot)$, i.e., to make him believe that the data generated by $G(\cdot)$ is authentic, it implies that $G(\cdot)$ can be got in this way

$$\min_G \left[\log(1 - D(G(\mathbf{z}))) \right]. \quad (10)$$

Based on the last two relations, the generator and discriminator can be found as follows

$$\min_G \max_D \left[\log(D(\mathbf{x})) + \log(1 - D(G(\mathbf{z}))) \right]. \quad (11)$$

According to the last relation, the goal of GANs is to find $G(\cdot)$ and $D(\cdot, \cdot)$ such that the generated data $G(\mathbf{z})$ is as similar as possible to the actual data $\mathbf{x}$. In other words, training the network aims to achieve the global optimum, i.e., $p_g = p_{\text{data}}$. Alternating GAN training involves training the discriminator first, followed by the generator, and repeating this process. During the discriminator's training, the generator's parameters are fixed, and the discriminator learns to determine the weaknesses of the present generator in order to successfully distinguish the true data from the generated ones. Analogously, the discriminator's parameters were fixed while the generator was training. Due to the concurrent search for gradient descent of the generator and discriminator loss functions, the disadvantages of generative adversarial networks

can be organized into two groups. The first one refers to the vanishing gradient problem that occurs when the discriminator begins to distinguish real examples from generated ones very successfully. Then, due to the discriminator's good performance, the error function's gradient tends to zero, so propagating the gradient through the model does not achieve the desired effect; that is, the parameters of the model change very little or even remain unchanged. The second category refers to the discriminator getting stuck in one of the local minima, which means the generator finds one or more examples that the discriminator accepts as accurate, so it learns to map each input vector, up to slight variations, to those examples.

Algorithm 2: Pseudocode of training GANs by stochastic gradient descent

Input: Initialization of the number of steps to m , the total number of iterations to n , and weights and parameters of generator (G) and discriminator (D) randomly.

for $i = 1, 2, \dots, n$ **do**

for $j = 1, 2, \dots, m$ **do**

Sample a set of k examples $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$ from the distribution p_z .

Sample a set of k examples $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$ from the distribution p_{data} .

Compute the discriminator loss and update its weights by stochastic gradient ascent, where the gradient of the empirical error is calculated as

$$\nabla_{\theta_D} \frac{1}{k} \sum_{i=1}^k \left[\log D(\mathbf{x}_i) + \log(1 - D(G(\mathbf{z}_i))) \right].$$

end for

Sample a set of k examples $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$ from the distribution p_z .

Compute the generator loss and update its weights by stochastic gradient descent, where the gradient of the empirical error is calculated as

$$\nabla_{\theta_G} \frac{1}{k} \sum_{i=1}^k \log(1 - D(G(\mathbf{z}_i))).$$

end for

Output: Set of learned weights for generator (G) and discriminator (D).

3.2 Conditional Generative Adversarial Networks (CGANs)

Conditional generative adversarial networks (CGANs) were introduced by researchers Mehdi Mirza and Simon Osindero [34]. These networks are a variation of generative adversarial networks (GANs) that incorporate conditional information into the generation process. These networks enable the controlled generation of images by adding additional input data that determine the desired characteristics of

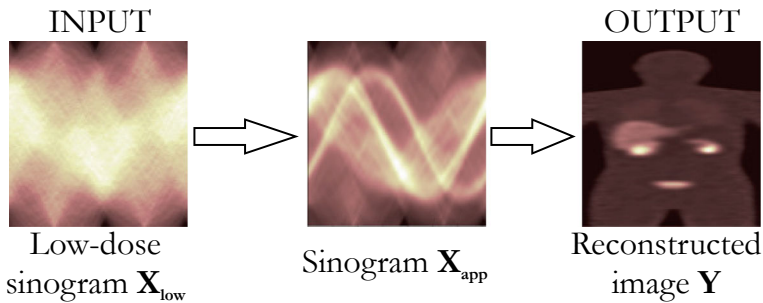


Fig. 1 Configuration of the proposed approach for low-dose reconstruction in two stages

the generated images. This technique allows for more precise control over the image generation process and enables the generation of images with specific properties or styles. For example, the CGANs can be used to generate images of people. In that case, additional information is provided, such as age, gender, hair color, or facial expression, as a condition for generating the image. The network will use this conditional information and the latent noise vector to generate an image that satisfies the given conditions. This type of network also fits well in SPECT applications, where, based on conditional information (sinogram for a regular dose of radiation), a sinogram of better quality is generated by a sinogram created from a low dose of radiation. More about these networks can be found in the paper [2].

3.3 SPECT Reconstruction

This section describes a hybrid approach based on conditional generative adversarial networks (CGANs) [33] and well-known traditional analytic and algebraic iterative methods to produce a high-quality reconstruction of images in SPECT imaging if low radiation doses are available. In other words, modern approaches such as GANs and conventional reconstruction methods will produce high-quality SPECT-reconstructed images based on low-radiation sinograms.

3.3.1 Mathematical Formulation of the Problem

Give a sinogram $\mathbf{X}_{\text{low}}$ generated at low radiation doses by a SPECT device (e.g., 60 views). In addition to the sinogram $\mathbf{X}_{\text{low}}$, let there also be a sinogram $\mathbf{X}_{\text{high}}$ produced at normal radiation doses (120 views) so that the sinogram $\mathbf{X}_{\text{high}}$ will represent a label in the dataset. Let f represent the mapping that the sinogram $\mathbf{X}_{\text{low}}$ maps into the sinogram $\mathbf{X}_{\text{app}}$ so that the sinogram $\mathbf{X}_{\text{app}}$ very well approximates the high-quality sinogram $\mathbf{X}_{\text{high}}$, i.e., holds $\|\mathbf{X}_{\text{app}} - \mathbf{X}_{\text{high}}\| \rightarrow 0$. The mapping of f in this paper, for

which $\mathbf{X}_{\text{app}} = f(\mathbf{X}_{\text{low}})$ is realized through CGAN, which generates a high-quality sinogram $\mathbf{X}_{\text{app}}$, which also represents the first stage of our approach. In the second phase of our approach, the reconstruction of the original image is conducted based on the sinogram $\mathbf{X}_{\text{app}}$ by exploiting traditional methods, such as Filtered Back Projection and others [1, 20, 37]. In other words, traditional methods refer to discovering the function g to hold $Y = g(\mathbf{X}_{\text{app}})$, where $\mathbf{Y}$ is the reconstruction of the original image, as shown in Fig. 1. The two-phase process for SPECT image reconstruction is depicted in Fig. 1. In the remainder of the section, the first phase of our approach based on conditional generative adversarial networks (CGANs) will be described in detail since the traditional methods utilized in the second phase of our approach were explained earlier.

3.3.2 The CGAN Model for Sinogram Enhancement

The first phase of our two-phase approach uses the CGAN architecture, which consists of two convolutional neural networks shown in Fig. 2. The first neural network is the generator's network $G(\cdot)$, and it induces a sinogram of enhanced quality $G(\mathbf{X})$ based on previously constructed low-quality sinograms $\mathbf{X}$ (low radiation dose) utilizing the SIMIND simulator [35]. Thus, the generator network $G(\cdot)$ creates false high-quality sinograms $G(\mathbf{X})$ compared to real high-quality sinograms $\mathbf{Y}$ generated by the SIMIND simulator at regular radiation doses. The second convolutional network is the discriminator network $D(\cdot, \cdot)$. This network evaluates the false sinograms $G(\mathbf{X})$ by comparing them with the real SPECT sinograms, i.e., with the labels $\mathbf{Y}$. The main task of the discriminator network is to try to learn to classify the fake high-quality sinograms $G(\mathbf{X})$ from the accurate labels $\mathbf{Y}$ as well as possible. For comparison requirements, the l_1 metric is often the best choice, and it needs to be as small as possible, i.e., $\|G(\mathbf{X}) - \mathbf{Y}\|_1 \rightarrow 0$.

3.3.3 Architecture of the Generator Network

It is known for SPECT sinograms that the edges are not clearly defined, and that is why the gradients are very uniform, which indicates the presence of projected low-frequency components. It most often happens due to various factors, such as absorption, scattering, and detector limitations. As a result, the edges of sinograms between different human body organs cannot be located. In addition, to reconstruct high-frequency components in SPECT, such as contrast and sinusoidal shapes, it is necessary to do much preprocessing with image processing techniques in the frequency domain concerning the Fourier transform. High-frequency components indicate faster changes in intensity, e.g., impulsive transitions between organs or tissues and changes expressed in the concentration of radioactive substances in small areas. On the other hand, low-frequency components in SPECT refer to soft edges between tissues or organs and the presence of radioactive substances. Unlike the high-frequency components, these components indicate slower and more gradual changes.

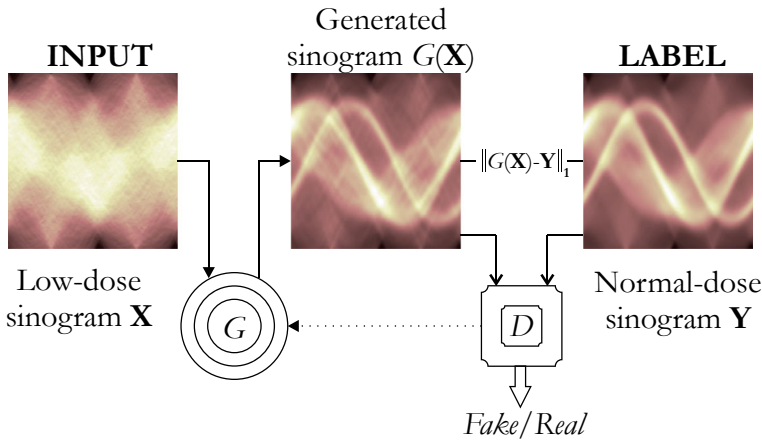


Fig. 2 CGAN framework based on generator (G) and discriminator (D) was developed to enhance the quality of sinograms derived from low-dose imaging

In order to create a better-quality SPECT sinogram through the generator network, it is necessary to reserve a certain number of layers for both high-frequency and low-frequency components. In other words, it needs to represent the generator network as an encoder-decoder architecture, where the encoder extracts low frequencies, and the decoder serves to restore high-frequency components. This kind of architecture is precisely realized through the U-NET network [39]. So, with this network, the encoder performs a DownSampling operation in which the spatial resolution of the sinogram is reduced using convolution, thus losing details on the SPECT sinogram but preserving information about low frequencies. On the other hand, the decoder in the U-NET network performs the UpSampling operation through deconvolution, which increases the spatial resolution, thus restoring the details or high-frequency components lost through the encoder. Thus, the combination of encoder and decoder allows the U-NET network to efficiently learn the representation of the input sinogram at low radiation dose to generate a new sinogram that reflects the features of the labeled sinogram, even when the input sinogram is drastically poorer in detail than the labeled sinogram. In this paper, the encoder and decoder are realized over eight blocks, so the first four blocks are exploited for the DownSampling operation and the remaining four for UpSampling. Therefore, the input-depleted SPECT sinogram is passed through the encoder to extract its features through four DownSampling blocks. Each of the four blocks in the encoder comprises two connected layers followed by convolution operations, Batch normalization, and LeakyReLU activation, as shown in Fig. 3. In order to achieve a better extraction of features from the SPECT sinogram for each block, the size of the convolution filters is set to three, while the stride is equal to one. In this way, the size of the sinogram through the blocks is reduced by factor two so that the encoder sinogram's fourth block has a dimension of 16×16 . Along with reducing the sinogram, the number of filters increases so that in the

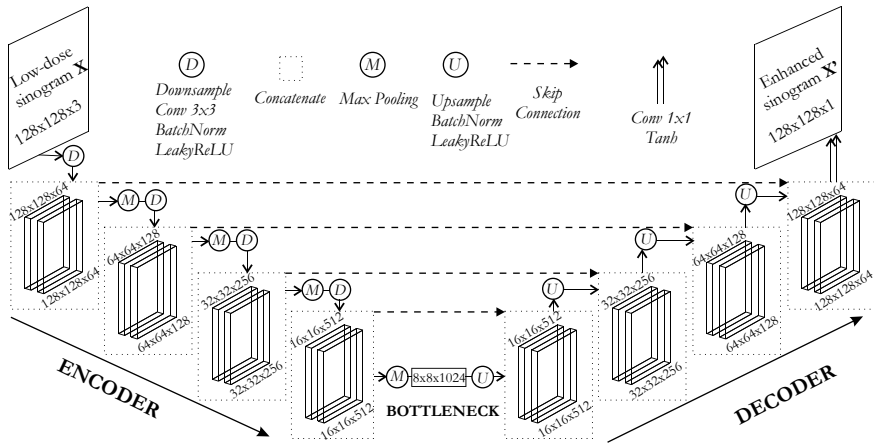


Fig. 3 Design of the generator network (G)

fourth block of the encoder, it is equal to the number 512. After the encoder, there is a bottleneck layer at which the sinogram reduces its dimension to 8×8 while the number of filters increases to 1024. Behind the bottleneck layer, the training process moves to the decoding stage, where through four blocks of the decoder, the sinogram is enlarged from 8×8 to 128×128 using deconvolution. In this case, the number of filters is reduced from 1024 to 64. Bicubic interpolation was used to rescale the decoder's sinogram to a larger dimension to create the desired high-quality sinogram. Batch normalization and LeakyReLU activation are utilized for each decoder block to achieve stability and speed during training. Apart from the mentioned configuration, it is necessary to establish a symmetrical skip connection between pairs of DownSampling and UpSampling blocks in order to obtain finer details (edges) on the generated sinogram. The skip connection also ensures that high-resolution information is exchanged directly between the encoder and the decoder. Another convolutional layer is usually added after the decoder to generate a sinogram of the initial dimensions.

3.3.4 Architecture of the Discriminator Network

As stated so far, the goal of the discriminator network is to assemble a clear distinction between the labeled sinogram (created at moderate radiation dose) and the generated sinogram (created from the low radiation sinogram and the labeled sinogram). In this paper, the Patch2GAN discriminator model is employed to realize the discriminator since it is more accurate, efficient, and better distinguishes the generated sinograms of labeled ones, which is crucial in medical applications [24]. Namely, Patch2GAN operates with local blocks of the image and not with the entire image, which GAN does, so it can better concentrate on the details of the image, such as

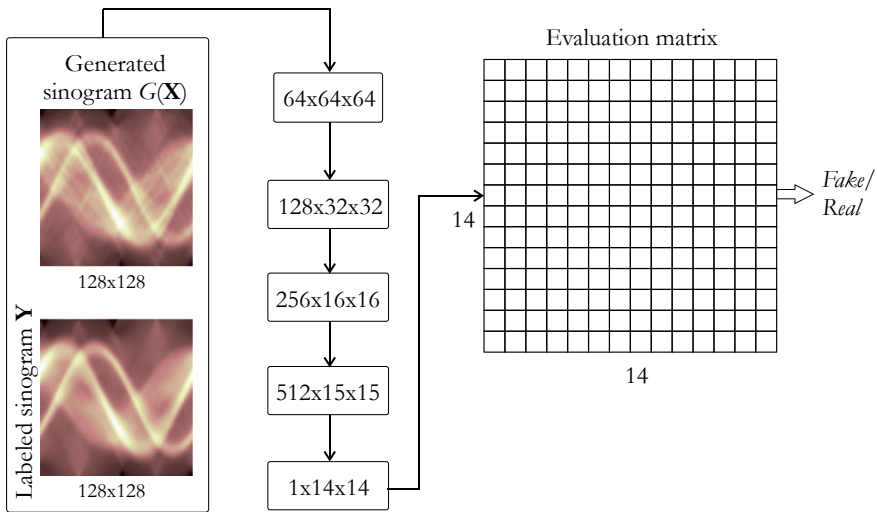


Fig. 4 Layout of the discriminator (D) with specifications of convolutional blocks. The input into the discriminator is the matrix $\mathbf{Y}$, which represents the labeled sinogram obtained from normal-dose radiation and the sinogram $G(\mathbf{X})$ generated by the generator (G)

sinusoidal features of the sinogram expressed through low and high spatial frequencies, edges, contrasts. In addition, Patch2GAN better catches artifacts, which usually arise in generated sinograms constructed from low doses of radiation. Unlike the classic discriminator network, Patch2GAN simulates a significantly smaller number of parameters. At the input, the Patch2GAN discriminator model accepts two images: a generated one using a low-quality sinogram and a labeled image (sinogram) built at normal radiation doses. The output from Patch2GAN is a probability matrix of usually square format in which each element indicates a predictive value relative to a particular “patch” of the generated sinogram. The outcome from the PatchGAN network is obtained by averaging the elements of the evaluation matrix.

Based on simulation tests, it has shown that the best outcomes are reached when the evaluation matrix is in 14×14 format because then the resolution and contrast of the sinogram are drastically improved, i.e., the structure of the generated sinogram is fully preserved compared to the labeled sinogram. In the experimental analysis, the discriminator network established on the Patch2GAN model consists of five parts, depicted in Fig. 4. Since this discriminator network is based on the “Downsampling” operation, each part reduces the dimensionality of the passed inputs. Thus, the first part of the network is realized via a 2D convolutional layer with 64 filters of size 5, using stride 2 to reduce the feature map. The second and third parts of the network are adjusted analogously, with 128 and 256 filters allocated for the second and third parts, respectively. The second and third parts of the network are equipped with two more layers each for Batch normalization and LeakyReLU operation. The fourth part of the network uses 512 filters of size 3, and for map reduction, it uses stride 1.

This part also includes the LeakyReLU layer. The fifth part is realized through the convolutional layer and has only one filter of size 3, while the stride is set to the value 3. Behind this layer is the evaluation matrix of the format 14×14 .

3.3.5 Loss Function

In this section, the loss function $L(\cdot, \cdot, \cdot)$ for the CGAN model will be presented as a sum of two terms, where the first one is the adversarial loss denoted as $l_A(\cdot, \cdot)$ and the second one is L_1 loss marked as $l_S(\cdot)$, which constrains the generator network to penalize the diversity between the generated $G(\mathbf{X})$ and the labeled sinogram $\mathbf{Y}$. It is formally defined as follows

$$L(G, D, \lambda) = l_A(G, D) + \lambda l_S(G), \quad (12)$$

where λ is the hyperparameter used to establish the balance between the above losses. The loss $l_A(\cdot, \cdot)$ enables the generator $G(\cdot)$ to yield convincing sinograms, while the loss l_S directs to generating sinograms that will be very similar to the labeled sinograms. Therefore, digging for the generator $G(\cdot)$ and the discriminator $D(\cdot, \cdot)$ reduces to solving of the next problem

$$\min_G \max_D L(G, D, \lambda). \quad (13)$$

Formulation of loss $l_A(\cdot, \cdot)$

In the CGAN model, the assignment of the loss l_A is to conduct the generator network $G(\cdot)$ to create a proper sinogram $G(\mathbf{X})$ established on the current sinogram $\mathbf{X}$ induced at a low radiation dose. In doing so, the generator network $G(\cdot)$ should confuse the discriminator $D(\cdot, \cdot)$ so that the discriminator can hardly distinguish whether it is a generated or marked sinogram. In the fight against fraud by the generator, the generated sinogram $G(\mathbf{X})$ reveals the quality characteristics of the labeled sinogram by preserving sinusoidal details and contrast. Formally, the loss l_A is defined as follows

$$l_A(G, D) = E_{\mathbf{X}, \mathbf{Y}} \left[\log D(\mathbf{X}, \mathbf{Y}) \right] + E_{\mathbf{X}} \left[\log(1 - D(\mathbf{X}, G(\mathbf{X}))) \right]. \quad (14)$$

Consequently, the loss of $l_A(G, D)$ points that the generator $G(\cdot)$ sweetens the quality of the generated sinogram by minimizing the classification accuracy of the discriminator $D(\cdot, \cdot)$, while the discriminator $D(\cdot, \cdot)$ maximizes the classification accuracy in terms of increasing the distinction between the generated $G(\mathbf{X})$ and the labeled sinogram $\mathbf{Y}$. The competition between the generator and the discriminator lasts until a calm state or equilibrium is reached.

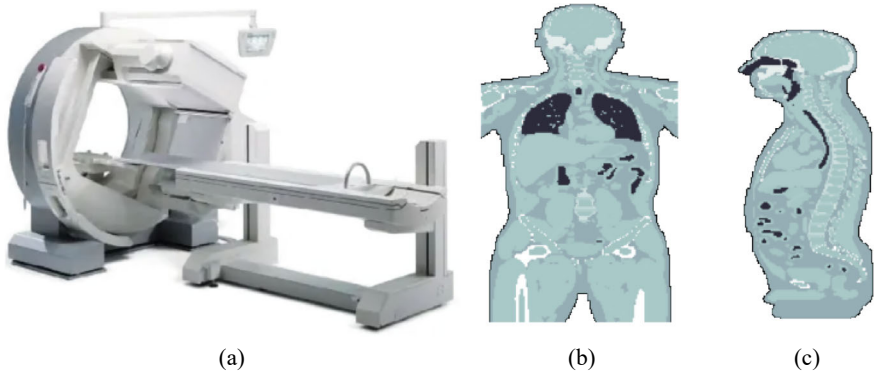


Fig. 5 **a** SIEMENS E. CAM device. **b** WB phantom. **c** HT phantom

Formulation of loss $l_S(\cdot)$

In the CGAN model, the purpose of the l_S loss is to enhance the consistency between images $G(\mathbf{X})$ and $\mathbf{Y}$. During the building of the sinogram $G(\mathbf{X})$, it is vital to note that due to the appearance of sums in the sinogram $\mathbf{X}$, the quality of the restored sinogram $G(\mathbf{X})$ may decrease, particularly in comparison with the sinogram $\mathbf{Y}$ (generated at standard radiation dose). In order to discourage this issue, the loss l_S must be realized as a metric based on the pixel difference between the labeled sinogram $\mathbf{Y}$ and the generated sinogram $G(\mathbf{X})$ to remove the high sum of the sinogram $\mathbf{X}$, and sweetened low contrast, since the loss of l_A preserves details and high contrast. Therefore, the loss l_S is formally expressed as follows.

$$l_S(G) = E_{\mathbf{X}, \mathbf{Y}} \left[\|\mathbf{Y} - G(\mathbf{X})\|_1 \right]. \quad (15)$$

Consequently, the loss of $l_S(G)$ in the sinogram domain penalizes the difference at the pixel level. It, together with the loss of l_A , ensures acceptable performance in detail and contrast restoration and overall sum reduction.

4 Performance Evaluation and Results

4.1 Creating Datasets

In this section, the SIMIND Monte Carlo simulator was employed to construct the dataset in order to simulate the process of the γ camera of the original SIEMENS E. CAM device¹ (see Fig. 5a.). To assemble the dataset for our two-phase approach, two

¹ <https://ttgimagingolutions.com/product/siemens-e-cam/>.

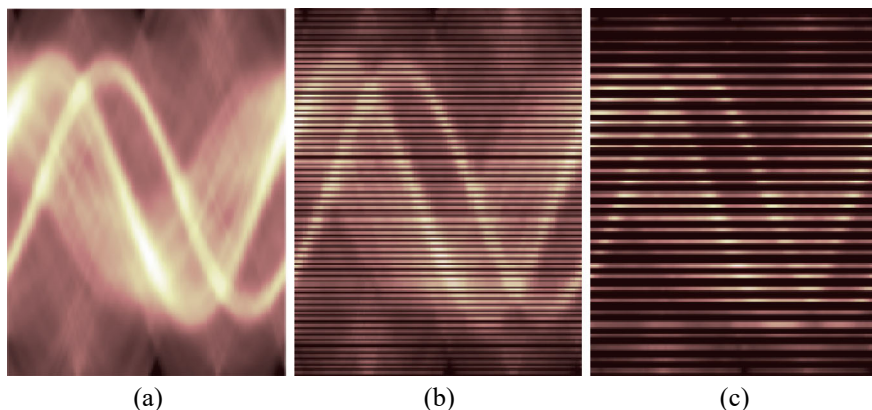


Fig. 6 **a** 120-views sinogram. **b** 60-views sinogram. **c** 30-views sinogram

digital phantoms are simulated using the SIMIND simulator [32]. The first phantom is the Whole Body (WB), and the second is the Human Trunk (HT). The exposition of these models is given in Fig. 5b, c. Through the SIMIND simulation, it is assumed that the orbit of the detector is circular and covers 360° , and the radius of rotation is set to 15 cm. In addition, the number of detectors is equal to 128, each 2.2mm wide. In the simulation, the energy window that determines the range of gamma-ray energies the detectors can detect is 18% around the gamma-photon center energy, which is set at 141 keV. With this configuration, the phantoms are radiated from 120 angles/views, so the number of photons for the WB phantom is 2.81×10^8 , and for the HT, 1.16×10^8 . Slices are taken from the simulated phantoms where the projected data quality is acceptable, resulting in 250 2D sinograms for the WB phantom and 300 2D sinograms for the HT phantom. So, after this, the dataset has a total of 550 2D sinograms, with 80% of them used for training and the remaining 20% for testing. All extracted sinograms are 120×128 in size. In Fig. 6, there are three 2D sinograms. The first one from the left to right is the reference sinogram, and the remaining two are used in the first phase of training the CGAN model to generate better quality sinograms that will serve as input to the second phase of our model, on which the analytic filtered back projection will be applied. Since all sinograms are matrix-represented, the sinogram presented in Fig. 6b is created by sampling a reference sinogram to take the odd rows of its matrix, and the elements of the missing even rows in the resulting sinogram are set to zero. Therefore, this sinogram has sampled by 60 views regarding the reference sinogram. The last sinogram shown in Fig. 6c was created analogously, and it has been sampled from 30 views due to the reference sinogram shown in Fig. 6a.

The sinogram from any sparse-view level in the test set is fed into the generator (U-NET) network. The network's output is then presented as the synthesized sinogram. In Fig. 7b, the synthesized sinogram of the chest slice of the WB phantom was obtained by operating the CGAN, which as input uses the low-dose sinogram shown in Fig. 7a. According to the content depicted in Fig. 7b, the CGAN delivers

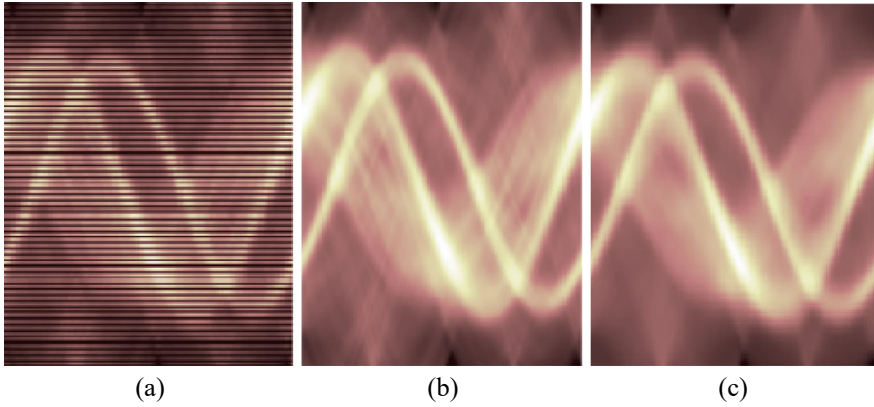


Fig. 7 **a** Low-dose sinogram. **b** Sinogram produced by CGAN. **c** Normal-dose sinogram

an adequate rate compared to the regular-dose sinogram shown in Fig. 7c. Based on this, the CGAN reasonably restores the normal-dose sinogram when a low radiation dose is fed to its input. The restored sinogram also keeps precise original sinusoidal waves and contrasts. Consequently, it affects the accuracy of global reconstruction and structure restoration.

4.2 Metrics for Qualitative Evaluation of Reconstructed Images

To evaluate the experimental results of the two-phase algorithm based on the CGAN model and traditional FBP, metrics such as NMSE (Normalized Mean Square Error), PSNR (Peak Signal-to-Noise Ratio), SSIM (structural similarity). NMSE metric is calculated by averaging the squared difference of the intensity pixels between the reconstructed image R_I and ground truth G_T as follows

$$NMSE = \frac{\sum_{i=1}^M \sum_{j=1}^N (R_I(i, j) - G_T(i, j))^2}{\sum_{i=1}^M \sum_{j=1}^N G_T(i, j)^2}, \quad (16)$$

where MN is the size of the reconstructed image.

PSNR serves as an accurate measure for estimating image distortion or the existence of noise. It can be expressed as

$$PSNR = 10 \times \log \frac{MAX_I^2}{MSE} (dB), \quad (17)$$

where MAX_I is the maximum pixel intensity of an image, I is the ground truth, and MSE is defined as follows

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (R_I(i, j) - G_T(i, j))^2. \quad (18)$$

SSIM is one of the most widely exploited metrics for measuring the similarity of two images in medical imaging. It is defined as

$$SSIM(I, J) = \frac{(2\mu_I\mu_J + C_1)(2\sigma_{IJ} + C_2)}{(\mu_I^2 + \mu_J^2 + C_1)(\sigma_I^2 + \sigma_J^2 + C_2)}, \quad (19)$$

where I and J are reconstructed image and ground-truth, respectively, μ_I is the average of I , μ_J is the average of J , σ_I^2 and σ_J^2 represent the variance of I and J , respectively, and σ_{IJ} is the covariance of I and J . At the same time, C_1 and C_2 are stabilization constants preventing division by zero in case the denominators are close to zero.

4.3 The Discussion of Experimental Results

In this section, the experimental outcomes fetched utilizing the two-phase CGAN-FBR approach and the other four conventional algorithms, where FBP is an analytic method. At the same time, OSEM, BSREM, and KEM are iterative statistical methods. The mentioned algorithms compete in image reconstruction based on sinograms obtained through 30-view and 60-view sampling of the reference sinogram generated from 120 views/angles. The following global metrics measure the best-achieved results: PSNR, SSIM, and NMSE. In implementing the CGAN-FBR approach, the weight parameters are optimized using the ADAM method [28], where the momentum Adam parameters β_1 and β_2 are set to values of 0.45 and 0.95, respectively. In addition, the learning rates for the generator and discriminator networks are set to the numbers 0.00001 and 0.001, respectively. The CGAN-FBR model is trained for 2000 epochs, while the mini-batch size is 32. This model is implemented in Python with the help of the PyTorch library. Other algorithms competing with the CGAN-FBR model are also implemented in Python using the PyTomography library [36]. All algorithms are executed on machine AMD Ryzen 5 5600 X Core AM4 4.6GHz processor with 64GB DDR5 4800MHz RAM, MSI GeForce RTX 4070 Ti Gaming X Slim 12G graphics card, and Windows 11 $\times$ 64 operating system. Training the CGAN-FBR model on this machine required about 10h. The results of the comparative analysis between the mentioned algorithms are shown in Tables 1 and 2. For all algorithms, PSNR, SSIM, and NMSE metrics were evaluated on the test data set, so only average results are shown in Tables 1 and 2. In these tables, the global metrics, such as PSNR and SSIM, validate the reconstructed images' quality and quantitatively compare their accuracy.

Table 1 Quality metrics were evaluated for reconstructing images using five competing methods from projection data consisting of 30 views

Algorithms	PSNR	SSIM	NMSE
<i>CGAN-FBP</i>	40.51	0.93	0.0009
<i>OSEM</i>	36.51	0.54	0.0167
<i>BSREM</i>	36.01	0.45	0.0769
<i>KEM</i>	35.89	0.39	0.0594
<i>FBP</i>	33.76	0.37	0.0698

The best-achieved results are marked in bold letters

Table 2 Quality metrics were evaluated for reconstructing images using five competing methods from projection data consisting of 60 views

Algorithms	PSNR	SSIM	NMSE
<i>CGAN-FBP</i>	46.09	0.97	0.0001
<i>OSEM</i>	42.07	0.83	0.0030
<i>BSREM</i>	41.91	0.79	0.0109
<i>KEM</i>	41.50	0.75	0.0110
<i>FBP</i>	39.79	0.71	0.0308

The best-achieved results are marked in bold letters

According to the outcomes of the simulation presented in Table 1, it is observed, based on global metrics, such as PSNR, SSIM, and NMSE, that the proposed CGAN-FBP approach, even at low radiation doses (30 angles/views), gives realistic reconstructions. In addition, it produces far better results than other methods (OSEM, BSREM, KEM, FBP) when reconstructions are made over sinograms created at low

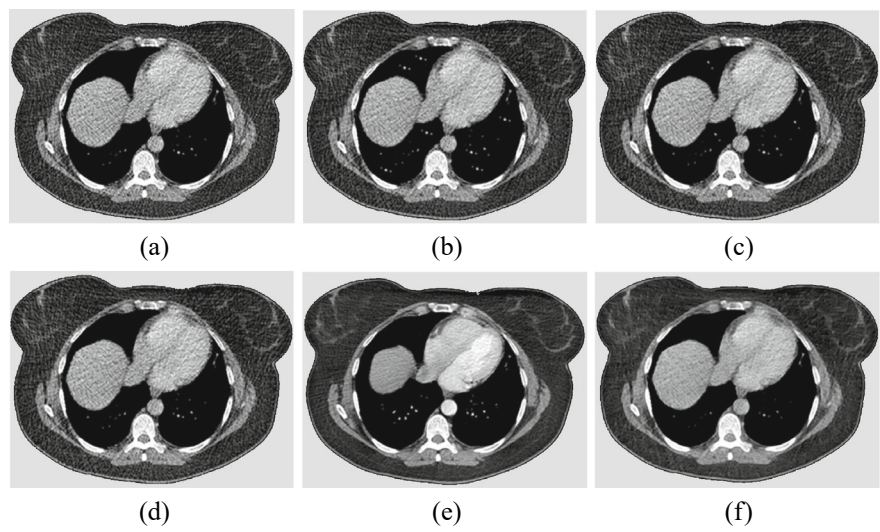


Fig. 8 Transaxial cross-sections through chest parts were generated employing **a** low-dose FBP and **b** Low-dose OSEM. **c** Low-dose BSREM. **d** Low-dose KEM. **e** Low-dose CGAN-FBP. **f** Normal-dose FBP

radiation doses. Thus, in SPECT imaging with very low doses, traditional methods have a much lower reconstruction performance than the proposed CGAN-FBP method, which suggests that it is a more robust method and, therefore, can better maintain high-quality image reconstruction in SPECT imaging at very low doses of radiation. The increase in the radiation dose from 30 angles/views to 60 angles/views directly affects the quality of the reconstruction itself, which is recorded tabularly through Table 1, as well as visually depicted in Fig. 8 where the need for visual comparison uses a transaxial section of the chest region. It is not difficult to see that the qualitative results of the reconstruction accuracy shown in Table 2 coincide with the visual contents shown in Fig. 8 from (a) to (e). Images from (a) to (e) were taken at a low radiation dose (60 angles/views), while the image under (f) was taken with FBP versus a normal sinogram (120 angles/views). In this case, the proposed CGAN-FBP model is also superior to other techniques regarding visual appearance and quantitative analysis of the data presented in Table 2. It is essential to say that in this experimental analysis, Poisson noise was equally represented in both training and testing data, which is usually not the practice case. Therefore, the developed method must be robust to noise, making it more robust on unseen data. Also, reducing the radiation dose in SPECT imaging implies fewer photons, further affecting the increase in Poisson noise. The increase in the Poisson noise directly reduces the contrast in the image, which is directly reflected in the quality of the reconstructed image. Although the proposed CGAN-FBP method gives satisfactory results in terms of the grade of the reconstructed image, it is still very demanding in terms of time and hardware, so in further research, emphasis should be placed on discovering a more optimal framework, which will undoubtedly represent a challenge in further research in this area.

5 Conclusion

This chapter introduces a combined CGAN-FBP model based on hybridizing conditional generative adversarial networks (CGANs) and analytic techniques for image reconstruction from projections such as Filtered Back Projection (FBP). The primary intent was to exploit CGAN to generate better-quality sinograms based on low-dose sinograms to get enhanced reconstructed images in SPECT imaging utilizing conventional techniques. It is revealed that the CGAN-FBP model based on the generator and discriminator network can habilitate the complicated sinusoidal elements and almost wholly retrieve the standard sinogram's authentic distinction, decreasing the noise and patting the contrast in the reconstructed image. Additionally, it was found that, compared to iterative methods for image reconstruction, the proposed hybridized model achieves superior qualitative and quantitative results. Consequently, the CGAN-FBP model can reduce the dose of radioactive tracer required for SPECT imaging while maintaining the quality of the reconstructed images.

References

1. Avenido HGD, Crisostomo RV (2022) Image reconstruction from a large number of projections in proton and ^{12}C ions computed tomography using sequential and parallel art algorithms. *Proc Comput Sci* 197:126–134. <https://doi.org/10.1016/j.procs.2021.12.126>.
2. Babu CNG, Dutt AGG, Pushpa SK, Manjunath TN (2023) Conditional generative adversarial networks for image transformation. In: *Advances in Data-driven Computing and Intelligent Systems*, pp. 547–560. Springer Nature Singapore
3. Basu S, Bresler Y (2000) $O(n/\sup 2/\log/\sub 2/n)$ filtered backprojection reconstruction algorithm for tomography. *IEEE Trans Image Process* 9(10):1760–1773. <https://doi.org/10.1109/83.869187>
4. Batenburg K (2005) An evolutionary algorithm for discrete tomography. *Discret Appl Math* 151(1):36–54. <https://doi.org/10.1016/j.dam.2005.02.021>.
5. Bevelacqua JJ (2010) Practical and effective alara. *Health Phys* 98(2):S39–S47. <https://doi.org/10.1097/HP.0b013e3181d18d63>
6. Brenner DJ, Sachs RK (2006) Estimating radiation-induced cancer risks at very low doses: rationale for using a linear no-threshold approach. *Radiat Environ Biophys* 44(4):253–256. <https://doi.org/10.1007/s00411-006-0029-4>
7. Bruyant PP (2002) Analytic and iterative reconstruction algorithms in spect. *J Nucl Med* 43(10):1343–1358
8. Chen D, Goris B, Bleichrodt F, Mezerji HH, Bals S, Batenburg KJ, de With G, Friedrich H (2014) The properties of sirt, tvmm, and dart for 3d imaging of tubular domains in nanocomposite thin-films and sections. *Ultramicroscopy* 147:137–148. <https://doi.org/10.1016/j.ultramic.2014.08.005>
9. Chen H, Zhang Y, Kalra MK, Lin F, Chen Y, Liao P, Zhou J, Wang G (2017) Low-dose ct with a residual encoder-decoder convolutional neural network. *IEEE Trans Med Imaging* 36(12):2524–2535. <https://doi.org/10.1109/TMI.2017.2715284>
10. Chen H, Zhang Y, Zhang W, Liao P, Li K, Zhou J, Wang G (2017) Low-dose CT via convolutional neural network. *Biomed Opt Express* 8(2):679–694. <https://doi.org/10.1364/BOE.8.000679>
11. Chen Y, Huang J, Li S, Lu Y, Xu Y (2020) A content-adaptive unstructured grid based integral equation method with the tv regularization for spect reconstruction. *Inverse Prob Imag* 14(1):27–52. <https://doi.org/10.3934/ipi.2019062>
12. Chrysostomou C, Koutsantonis L, Lemesios C, Papanicolas CN (2020) Spect angle interpolation based on deep learning methodologies. In: *2020 IEEE nuclear science symposium and medical imaging conference (NSS/MIC)*, pp. 1–4. <https://doi.org/10.1109/NSS/MIC42677.2020.9507966>
13. Fleischmann D, Boas FE (2011) Computed tomography-old ideas and new technology. *Eur Radiol* 21(3):510–517. <https://doi.org/10.1007/s00330-011-2056-z>
14. Fuh CS, Lin SS (1998) The fourier slice theorem for range data reconstruction. *Image and Vision Computing* 16(9):689–701
15. Geyer LL, Schoepf UJ, Meinel FG, Nance JW, Bastarrika JG, Leipsic JA, Paul NS, Rengo M, Laghi A, Cecco CND (2015) State of the art: iterative ct reconstruction techniques. *Radiology* 276(2):339–357. <https://doi.org/10.1148/radiol.2015132766>
16. Gong K, Catana C, Qi J, Li Q (2019) Pet image reconstruction using deep image prior. *IEEE Trans Med Imaging* 38(7):1655–1665. <https://doi.org/10.1109/TMI.2018.2888491>
17. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. *Commun ACM* 63(11):139–144. <https://doi.org/10.1145/3422622>.
18. Gregor J, Benson T (2008) Computational analysis and improvement of sirt. *IEEE Trans Med Imaging* 27(7):918–924. <https://doi.org/10.1109/TMI.2008.923696>
19. Groch MW, Erwin WD (2001) Single-photon emission computed tomography in the year 2001: instrumentation and quality control. *J Nucl Med Technol* 29(1):12–18

20. Guo J, Qi H, Xu Y, Chen Z, Li S, Zhou L (2016) Iterative image reconstruction for limited-angle ct using optimized initial image. *Computational and Mathematical Methods in Medicine* **vol. 2016**(Article ID 5836410), 1–9. <https://doi.org/10.1155/2016/5836410>
21. Hansen PC, Jørgensen JS, Lionheart WRB (2021) Computed Tomography: Algorithms, Insight, and Just Enough Theory. SIAM. <https://doi.org/10.1137/1.9781611976670.ch1>.
22. Hutchins G, Rogers W, Chiao P, Raylman R, Murphy B (1990) Constrained least squares filtering in high resolution pet and spect imaging. *IEEE Trans Nucl Sci* 37(2):647–651. <https://doi.org/10.1109/23.106692>
23. Häggström I, Schmidtlein CR, Campanella G, Fuchs TJ (2019) DeepPET: a deep encoder-decoder network for directly solving the pet image reconstruction inverse problem. *Med Image Anal* 54:253–262
24. Isola P, Zhu JY, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5967–5976. <https://doi.org/10.1109/CVPR.2017.632>
25. Jha AK, Purandare NC, Shah S, Agrawal AD, Rangarajan V (2014) Pet reconstruction artifact can be minimized by using sinogram correction and filtered back-projection technique. *Indian J radiol Imag* 24(2):103–106. <https://doi.org/10.4103/0971-3026.134379>
26. Jiang G, Wei J, Xu Y, He Z, Zeng H, Wu J, Qin G, Chen W, Lu Y (2021) Synthesis of mammogram from digital breast tomosynthesis using deep convolutional neural network with gradient guided cgans. *IEEE Trans Med Imaging* 40(8):2080–2091. <https://doi.org/10.1109/TMI.2021.3071544>
27. Jiang Y, Li S, Xu Y (2019) A higher-order polynomial method for spect reconstruction. *IEEE Trans Med Imaging* 38(5):1271–1283. <https://doi.org/10.1109/TMI.2018.2881919>
28. Kingma D, Ba J (2015) Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR). San Diego, CA, USA
29. Li H, Wan Z (2021) A modified algebraic reconstruction algorithm for sparse projection. *Annals of Translational Medicine* 9(18), 1–7. <https://doi.org/10.21037/atm-21-3529>
30. Li S, Ye W, Li F (2022) Lu-net: combining lstm and u-net for sinogram synthesis in sparse-view spect reconstruction. *Math Biosci Eng* 19(4):4320–4340. <https://doi.org/10.3934/mbe.2022200>
31. Lindström E, Lindsjö L, Sundin A, Sörensen J, Lubberink M (2020) Evaluation of block-sequential regularized expectation maximization reconstruction of 68ga-dotatoc, 18f-fluoride, and 11c-acetate whole-body examinations acquired on a digital time-of-flight pet/ct scanner. *EJNMMI Physics* 7(1):1–14
32. Ljungberg M, Strand SE (1989) A monte carlo program for the simulation of scintillation camera characteristics. *Comput Methods Programs Biomed* 29(4):257–272. [https://doi.org/10.1016/0169-2607\(89\)90111-9](https://doi.org/10.1016/0169-2607(89)90111-9)
33. Ma L, Qu S (2023) Application of conditional generative adversarial network to multi-step car-following modeling. *Frontiers in Neurorobotics*. <https://doi.org/10.3389/fnbot.2023.1148892>
34. Mirza M, Osindero S (2014) Conditional generative adversarial nets. <https://doi.org/10.48550/arXiv.1411.1784>
35. Morphis M, van Staden JA, du Raan H, Ljungberg M (2021) Modelling of energy-dependent spectral resolution for spect monte carlo simulations using simind. *Heliyon* 7(2):e06097
36. Polson L, Fedrigo R, Li C, Sabouri M, Dzikunu O, Ahamed S, Rahmim A, Uribe C (2023) Pytomography: A python library for quantitative medical image reconstruction. <https://doi.org/10.48550/arXiv.2309.01977>
37. Qi H, Chen Z, Zhou L (2014) Ct image reconstruction from sparse projections using adaptive t_{pv} regularization. *Mathematical Methods and Applications in Medical Imaging* 2015:1–8. <https://doi.org/10.1155/2015/354869>
38. Riddell C, Buvat I, Savi A, Gilardi MC, Fazio F (2002) Iterative reconstruction of spect data with adaptive regularization. *IEEE Trans Nucl Sci* 49(5):2350–2354. <https://doi.org/10.1109/TNS.2002.803677>

39. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MIC-CAI 2015, pp. 234–241. Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28
40. Shao W, Pomper MG, Du Y (2021) A learned reconstruction network for spect imaging. *IEEE Transactions on Radiation and Plasma Medical Sciences* 5(1):26–34. <https://doi.org/10.1109/TRPMS.2020.2994041>
41. Shao W, Rowe SP, Du Y (2020) Spectnet: a deep learning neural network for spect image reconstruction. *Annals of Translational Medicine* 9(9). <https://atm.amegroups.org/article/view/50692>
42. Shcherbinin S, Celler A, Trummer M, Humphries T (2009) An apd-based iterative reconstruction method for simultaneous technetium-99m/iodine-123 spect imaging. *Physica Med* 25(4):192–200. <https://doi.org/10.1016/j.ejmp.2009.01.003>
43. Shepp LA, Vardi Y (1982) Maximum likelihood reconstruction for emission tomography. *IEEE Trans Med Imaging* 1(2):113–122. <https://doi.org/10.1109/TMI.1982.4307558>
44. Singh S, Kalra MK, Gilman MD, Hsieh J, Pien HH, Digumarthy SR, Shepard JAO (2011) Adaptive statistical iterative reconstruction technique for radiation dose reduction in chest ct: a pilot study. *Radiology* 259(2):565–573. <https://doi.org/10.1148/radiol.11101450>
45. Valenti C (2008) A genetic algorithm for discrete tomography reconstruction. *Genet Program Evolvable Mach* 9(1):85–96. <https://doi.org/10.1007/s10710-007-9051-9>
46. Vangestel C, de Wiele CV, Mees G, Mertens K, Staelens S, Reutelingsperger C, Pauwels P, Damme NV, Peeters M (2012) Single-photon emission computed tomographic imaging of the early time course of therapy-induced cell death using technetium 99m tricarbonyl his-annexin a5 in a colorectal cancer xenograft model. *Mol Imaging* 11(2):135–147
47. Wang G, Qi J (2015) Pet image reconstruction using kernel method. *IEEE Trans Med Imaging* 34(1):61–71. <https://doi.org/10.1109/TMI.2014.2343916>
48. Wang G, Qi J (2015) Pet image reconstruction using kernel method. *IEEE Trans Med Imaging* 34(1):61–71. <https://doi.org/10.1109/TMI.2014.2343916>
49. Wells RG (2020) Dose reduction is good but it is image quality that matters. *J Nucl Cardiol* 27(1):238–240
50. Willemink MJ, Noël PB (2019) The evolution of image reconstruction for ct-from filtered back projection to artificial intelligence. *Eur Radiol* 29(5):2185–2195. <https://doi.org/10.1007/s00330-018-5810-7>
51. Yang Q, Yan P, Zhang Y, Yu H, Shi Y, Mou X, Kalra MK, Zhang Y, Sun L, Wang G (2018) Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss. *IEEE Trans Med Imaging* 37(6):1348–1357. <https://doi.org/10.1109/TMI.2018.2827462>
52. Zhang H, Dong B, Liu B (2019) Jsr-net: A deep network for joint spatial-radon domain ct reconstruction from incomplete data. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3657–3661. <https://doi.org/10.1109/ICASSP.2019.8682178>
53. Zhang Z, Liang X, Dong X, Xie Y, Cao G (2018) A sparse-view ct reconstruction method based on combination of densenet and deconvolution. *IEEE Trans Med Imaging* 37(6):1407–1417. <https://doi.org/10.1109/TMI.2018.2823338>
54. Zhu YM (2018) Ordered subset expectation maximization algorithm for positron emission tomographic image reconstruction using belief kernels. *J Med Imag* 5(4):1–8. <https://doi.org/10.1117/1.JMI.5.4.044005>

Chapter 15

Performance of Weightless Swarm Algorithm on Numerical Benchmark Functions



Yu Dou and Tiew On Ting

1 Introduction

Particle Swarm Optimization (PSO) is an evolutionary computation proposed by Eberhart and Kennedy, which was inspired by the social and cooperative behavior of bird flocks [16]. The key feature of PSO is finding optimal solutions through collaborative information sharing among its population. PSO simulates birds by designing particles with only two properties: speed and position. Each particle separately searches for the locally optimal solution in the search space, which is recorded as the current local best position. This information is then shared with other particles, aiding in identifying the current global best position. At each iteration, all the particles learn from the local and global best positions to adjust their movement trajectories and speeds in the next iteration. As the iterations proceed, the population gradually converges towards the global optimum. PSO has the advantages of fast convergence speed, relatively stable performance, and simple operation. It is now widely used in engineering [22, 36], physics [31, 47], chemistry [8, 27], medicine [1, 3], and computer science [26, 41].

Since its establishment, PSO has seen extensive development. It is essential to highlight that PSO is currently advancing along four main trends as follows [5]:

- (1) Adaptive parameters. In canonical PSO, three adjustable parameters exist: w , c_1 , and c_2 . The concept of inertia weight w was first proposed by Shi et al. [39] to balance the global search and local exploitation. Additionally, Shi et al., provided guidelines for selecting the inertia weight. They suggest a linear decrement of w

Y. Dou (✉)

School of Engineering, University of Leicester, Leicester, UK
e-mail: yd116@leicester.ac.uk

T. O. Ting

Stony Brook Institute at Anhui University (SBIAHU),
Hefei, Anhui Province, People's Republic of China
e-mail: toting@ahu.edu.cn

from 0.9 to 0.4 during the optimization [38]. PSO with adaptive inertia weight factor (PSOAIWF), proposed by Liu et al., adjusts the inertia weight based on a particle's fitness compared to the average fitness. This algorithm assigns lower inertia values to particles with higher fitness and higher inertia weights to those with lower fitness [21]. A similar modification of inertia weight can also be found in literature [7, 29, 54]. The essence of these algorithms lies in using adaptive inertia weight based on time, fitness, or success rate to enhance PSO performance. Besides, the adjustment of acceleration coefficients c_1 and c_2 is also common. In literature [33], Ratnaweera et al. first proposed the concept of time-varying acceleration coefficients. PSO with individual coefficients adjustment (PSO-SAIC), proposed by Wu *et al.*, adjusts each particle's inertia weight and social acceleration coefficient based on the relationship between its fitness and global optimal fitness. This strategy facilitates diversity injection, mainly when particles are close to the global best position [48]. In addition, some papers utilize adaptive particle's velocity to improve the performance of PSO [50, 51].

- (2) Designing varied topological structures and altering particles' learning modes are vital to preserving population diversity in PSO. PSO's topology, which describes a particle's neighborhood relations and interactions, influences information transmission and directly impacts the optimization and convergence abilities of the particles. Generally, there are two particle swarm topologies: static and dynamic. In static topology, the neighborhood of each particle remains unchanged during the optimization process. Examples include global topology, local topology, Von Neumann topology [13], wheel topology [9], and four clusters topology. In dynamic topologies, the neighborhood of some particles changes during iterations, examples of which include Clan topology [10], multi-ring topology [4], and random population topology [28]. According to Elliackin et al. [11], in PSO, particles converge faster using global topology compared to other topologies.
- (3) Combine PSO with other optimization algorithms or strategies to form a hybrid algorithm. This strategy amplifies the strengths and mitigates the weaknesses of both PSO and the combined algorithms. A common approach involves integrating PSO with algorithms like Genetic Algorithm (GA) [2], Differential Evolution (DE) [23], Artificial Bee Colony algorithm (ABC) [18], Grey Wolf Optimizer (GWO) [40], and Gravitational Search Algorithm (GSA) [24]. Another approach involves incorporating specific strategies into PSO. Sun et al. [42] developed a quantum behavior PSO (QPSO) by combining quantum theory and PSO. This algorithm ensures the discovery of the global optimum within the search space. In addition, the niching technique utilized in PSO can effectively solve multi-modal optimization problems [19, 20]. The primary objective of this technique is to maintain a diverse population. While the niching technique aids in finding more accurate solutions, it typically increases computational costs due to slower convergence.
- (4) Multi-swarm optimization is a variant of PSO that employs multiple sub-swarms rather than a single standard swarm. The core principle of multi-swarm optimization involves each sub-swarm concentrating on a specific region, with a unique

diversification method dictating sub-swarm initiation. This framework is adept at optimizing multi-modal problems characterized by numerous local minima. In recent research, Xia et al. introduced MSPSO, a model based on a multiple-swarm framework, which integrates three strategies (DNS, SRS, PDS) to balance early and late exploitation abilities [49]. Kumar proposed the multi-stage, multi-swarm PSO (MS2PSO), targeting common PSO issues like premature convergence, the disproportionate influence of the global best, and oscillations [17]. Additional applications and explorations of multi-swarm optimization are detailed in studies [6, 35, 53].

Most improvements to PSO involve adding new strategies or enhancing existing ones in the canonical framework. While these variants enhance PSO's performance, they simultaneously increase the algorithm's complexity and computational demands, which may overshadow PSO's inherent advantages of rapid convergence and straightforward implementation. Is there an algorithm that maintains accuracy and speed of convergence while reducing the complexity of PSO? Ting et al. proposed a more straightforward and faster algorithm based on PSO named Weightless Swarm Algorithm (WSA) [44, 46]. In contrast to canonical PSO, WSA eliminates the inertia weight and individual acceleration coefficients, accelerating convergence and reducing computational costs. Simultaneously, WSA retains the characteristics of robust stability and high accuracy. Therefore, in most circumstances, WSA is an efficient substitution for PSO.

In recent years, the effectiveness of WSA has been demonstrated across various applications. Ting et al. utilized WSA to implement Maximum Power Point Tracking (MPPT) in photovoltaic modules, effectively adapting to variations in environmental temperature and irradiance [45]. The results demonstrated superior accuracy compared to conventional methods such as P&O and IncCond, particularly under rapidly changing conditions. Ting applied WSA to optimize the Material Reduction Rate (MRR) during drilling processes, highlighting its effectiveness in identifying feasible solutions without violating constraints while emphasizing its stability when compared to PSO [43]. Jayaprakasam et al. [15] employed WSA to minimize Peak Sidelobe Level (PSL) in open-loop Collaborative Beamforming (CBF), surpassing GA and PSO in terms of effectiveness. Reddy et al. [34] investigated the application of WSA as a transformation technique to enhance the performance of machine learning classifiers for SONAR data classification. The study showcased that WSA significantly improved the accuracy of classifiers, including Random Forest, Stochastic Gradient Descent, and Decision Tree, with the Random Forest classifier's Mathews Correlation Coefficient increasing from 75.27% to 81.81%, outperforming traditional dimensionality reduction techniques like PCA and LDA.

The chapter aims to compare the performances of WSA and four alternative algorithms while also investigating the sensitivity of the social acceleration coefficient in WSA. The subsequent sections are organized as follows: Sect. 2 outlines the specific steps and strategies WSA employs. Section 3 provides comprehensive details regarding the simulations conducted. Section 4 presents a comparative analysis of five algorithms and an examination of WSA's parameter sensitivity. Finally, Sect. 5 offers a concise conclusion.

2 Algorithms

This section explains how PSO and WSA work in detail and offers a comparison. It also includes a case study to illustrate the differences better. Specifically, in what situation can WSA be more efficient than PSO?

2.1 Particle Swarm Optimization (PSO)

As introduced in Sect. 1, PSO involves the navigation of particles representing potential solutions within a search space, guided by personal and global best positions. Through iterative convergence driven by individual and collective experiences, these particles progressively approach an optimal solution. The following equations show the iteration process of the canonical PSO [37]:

$$V_i^{t+1} = w V_i^t + c_1 r_1 (Pbest_i^t - X_i^t) + c_2 r_2 (Gbest^t - X_i^t), \quad (1)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2)$$

where subscript i denotes the i th particle, superscript t denotes the iteration time, w is the inertia weight, V is the velocity, X is the individual position, $Pbest$ is the local best position, $Gbest$ is the global best position, c_1 is the cognitive acceleration coefficient, c_2 is the social acceleration coefficient, and r_1, r_2 are the random numbers within $[0, 1]$. As shown below, Fig. 1 illustrates the PSO process in a flowchart, while Algorithm 1 provides the corresponding pseudo-code for this process.

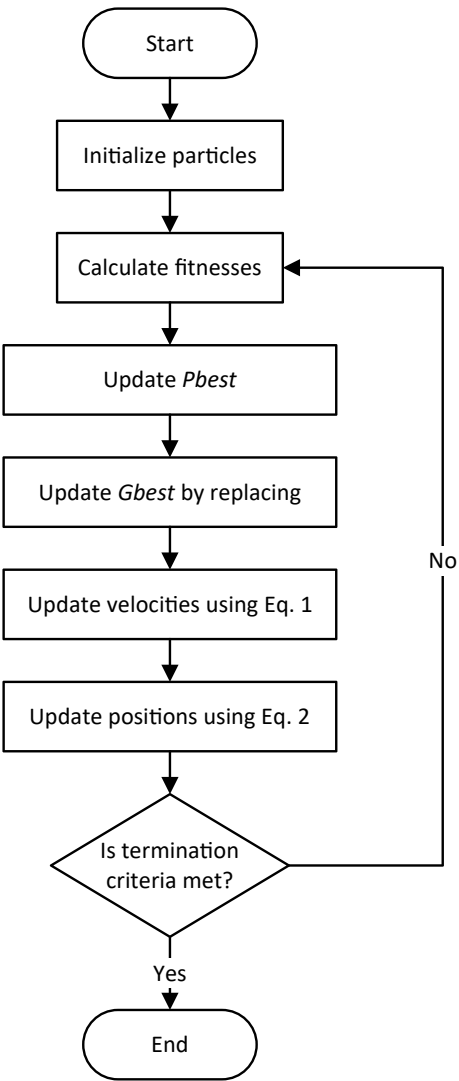
Algorithm 1 Particle Swarm Optimization (PSO)

```

1: Initialize swarm with population size  $N$ , individual positions  $X_i$ , and velocities  $V_i$  for each particle  $i$ 
2: Initialize  $Pbest_i$  and  $Gbest$  with corresponding initial values
3: while termination condition not met do
4:   for each particle  $i$  do
5:     Calculate the fitness  $F_i$  of particle  $i$ 
6:     if  $F_i$  is better than the fitness of  $Pbest_i$  then
7:       Update  $Pbest_i$  to  $X_i$ 
8:     end if
9:     if  $F_i$  is better than the fitness of  $Gbest$  then
10:      Update  $Gbest$  to  $X_i$ 
11:    end if
12:    Update the velocity  $V_i$  using Eq. 1
13:    Update the position  $X_i$  using Eq. 2
14:   end for
15: end while

```

Fig. 1 Flowchart for particle swarm optimization (PSO)



2.2 Weightless Swarm Algorithm (WSA)

The term “Weightless Swarm Algorithm” is used because the algorithm omits the inertia weight term w in PSO, rendering it weightless. In WSA, a unique approach is employed to update the global best position Gbest, which differs from the PSO method. In PSO, Gbest is updated by directly replacing it with a better position upon discovery. However, WSA adopts a swapping strategy between Gbest and the individual position X . This swapping involves alternating between the global best

and individual particle positions for updating Gbest, fostering a more diverse search pattern and minimizing the chances of invalid updates. This methodological variation influences the convergence speed.

Furthermore, WSA deviates from PSO by excluding inertia weight w and cognitive acceleration coefficient c_1 . The omission of these parameters simplifies the operational framework of WSA, impacting the balance between exploration and exploitation. It potentially leads to faster convergence due to focused reliance on social learning and increases the risk of entrapment in local optima. The governing equations of WSA are as follows:

$$V_i^{t+1} = c_2 r_2 (Gbest_i^t - X_i^t), \quad (3)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (4)$$

where c_2 corresponds to the social acceleration coefficient in PSO, emphasizing the global search ability of the swarm. As shown below, Fig. 2 illustrates the WSA process in a flowchart, while Algorithm 2 provides the corresponding pseudo-code for this process.

Algorithm 2 Weightless Swarm Algorithm (WSA)

```

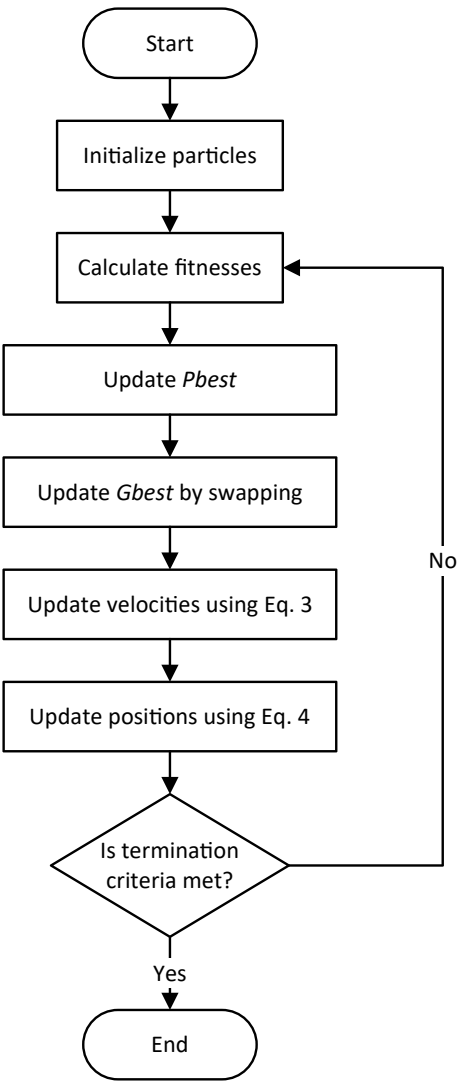
1: Initialize swarm with population size  $N$ , individual positions  $X_i$ , and velocities  $V_i$  for each
   particle  $i$ 
2: Initialize  $Pbest_i$  and  $Gbest$  with corresponding initial values
3: while termination condition not met do
4:   for each particle  $i$  do
5:     Calculate the fitness  $F_i$  of particle  $i$ 
6:     if  $F_i$  is better than the fitness of  $Pbest_i$  then
7:       Update  $Pbest_i$  to  $X_i$ 
8:     end if
9:     if  $F_i$  is better than the fitness of  $Gbest$  then
10:      Swap  $F_i$  and  $Gbest$ 
11:    end if
12:    Update the velocity  $V_i$  using Eq. 3
13:    Update the position  $X_i$  using Eq. 4
14:   end for
15: end while

```

2.3 Comparison Between PSO and WSA: A Case Study

A simple case study illustrates a 1-D optimization function defined as $y = x^2$. The study involves two swarms, denoted as S_1 and S_2 , with positions X_1 and X_2 respectively. Replacement and swapping strategies are employed to assess the algorithm's

Fig. 2 Flowchart for weightless swarm algorithm (WSA)



effectiveness. The objective is to determine the number of iterations required for algorithm convergence. In the replacement scenario described in Table 1, it is observed that during iteration 2 and iteration 4, while updating the global best position X_2 , three updates become invalid due to $c_2r_2(Gbest - X_2)$ equating to zero. These ineffective updates commonly encountered in PSO are effectively circumvented by WSA, highlighting its capability to eliminate such inefficiencies.

Table 1 An illustrative example demonstrating the benefits of the swapping strategy

		Replacing			Swapping		
		X_1	X_2	Gbest	X_1	X_2	Gbest
Initialization		3	5	3	3	5	3
Iteration 1	$c_2r_2 = 2.0$ for X_1	3	1	3	3	1	3
	$c_2r_2 = 2.0$ for X_2	3	1	1	3	3	1
Iteration 2	$c_2r_2 = 0.5$ for X_1	2	1	1	2	0	1
	$c_2r_2 = 1.5$ for X_2	2	1	1	2	1	0
Iteration 3	$c_2r_2 = 1.5$ for X_1	1.5	1	1			
	$c_2r_2 = 1.5$ for X_2	1.5	1	1			
Iteration 4	$c_2r_2 = 0.5$ for X_1	0.5	1	1			
	$c_2r_2 = 0.5$ for X_2	0.5	1	1			
Iteration 5	$c_2r_2 = 1.0$ for X_1	0.5	0	0.5			
	$c_2r_2 = 0.5$ for X_2	0.5	0	0			

3 Simulation

This study aims to achieve two primary objectives. Firstly, we strive to compare the performance of various optimization algorithms on different standard test functions, including WSA, PSO, APSO [52], QPSO [42], and PSOGSA [24]. Secondly, we conduct Parameter Sensitivity Analysis (PSA) to determine the optimal cognitive acceleration coefficient c_2 by evaluating its performance across all benchmark functions. This test is crucial for ensuring that WSA operates at its maximum potential.

3.1 Environment

For all the simulations mentioned in this study, we used MATLAB R2022a. The tests were run on a computer with Windows 11 Home. The computer is powered by an AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10GHz and 16.0 GB RAM.

Table 2 Benchmark functions for testing optimization algorithms

Function	Mathematical expression	Input domain	$\mathbf{x}^*$	$f(\mathbf{x}^*)$
Rosenbrock	$f_1(\mathbf{x}) = \sum_{i=1}^{d-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	$[-30, 30]$	$(1)^d$	0
Sphere	$f_2(\mathbf{x}) = \sum_{i=1}^d x_i^2$	$[-5.12, 5.12]$	$(0)^d$	0
Ackley	$f_3(\mathbf{x}) = -20e^{-0.02\sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}} - e^{d-1} \sum_{i=1}^d \cos(2\pi x_i) + 20 + e$	$[-35, 35]$	$(0)^d$	0
Michalewicz	$f_4(\mathbf{x}) = -\sum_{i=1}^d \sin(x_i) \sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$	$[0, \pi]$	$(2.20, 1.57)$	-1.8013
Qing	$f_5(\mathbf{x}) = \sum_{i=1}^d (x_i^2 - i)^2$	$[-500, 500]$	$(\pm 1, \pm 1)$	0
Rastrigin	$f_6(\mathbf{x}) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$	$[-5.12, 5.12]$	$(0)^d$	0
Schwefel	$f_7(\mathbf{x}) = 418.9829d - \sum_{i=1}^d x_i \sin\left(\sqrt{ x_i }\right)$	$[-500, 500]$	$(420.9687)^d$	0
Styblinski-Tang	$f_8(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^d (x_i^4 - 16x_i^2 + 5x_i)$	$[-5, 5]$	$(-2.903534)^d$	-39.16599d

3.2 Benchmark Test Functions

In this section, we present a selection of eight benchmark test functions, carefully chosen from existing literature [12, 14, 25, 32]. Detailed information about these functions, including their mathematical expressions, the input domain, and the global minimums, are systematically outlined in Table 2. Note that functions f_1 and f_2 are unimodal, functions f_3, f_4, f_5, f_6, f_7 , and f_8 are multi-modal. These functions are an important tool for assessing and comparing the performance of the algorithms discussed in this study.

The plots in Fig. 3 depict the 3-D representation of all benchmark test functions. Figure 3(a) depicts the Rosenbrock function's global minimum within a narrow, parabolic-shaped valley. While this valley can be relatively easily identified, achieving convergence at the minimum point poses a significant challenge for many gradient-based optimization algorithms, as referenced in [30]. Moreover, the Michalewicz and Schwefel functions exhibit a higher level of complexity due to their numerous local minima, presenting additional challenges for optimization.

3.3 Parameter Settings

The computed precision ε is set as 0.001. Maximum iteration times T_{max} is 1000. All algorithms use the same population size N . The specific population can be seen in Table 3. Besides, the social acceleration coefficient c_2 of WSA is adaptive to different test functions and dimensions. The other four algorithms use default parameters. Note that all tests in this study were replicated 100 times to minimize the influence of randomness.

3.4 Performance Indices

In this chapter, the indexes referred to as “performance indices” are utilized to evaluate the effectiveness of algorithms.

1. Success Rate (SR): This is the ratio of the number of successes to the total number of trials, reflecting the convergence ability. The expression of success rate is:

$$SR = \frac{m}{n} \quad (5)$$

where m is the number of successes and n is the total number of trials.

2. Average Iteration Running Time (AIRT): This is the average CPU time elapsed when converges successfully, reflecting the convergence speed. The expression of average iteration running time is:

$$AIRT = \frac{t}{ns} \quad (6)$$

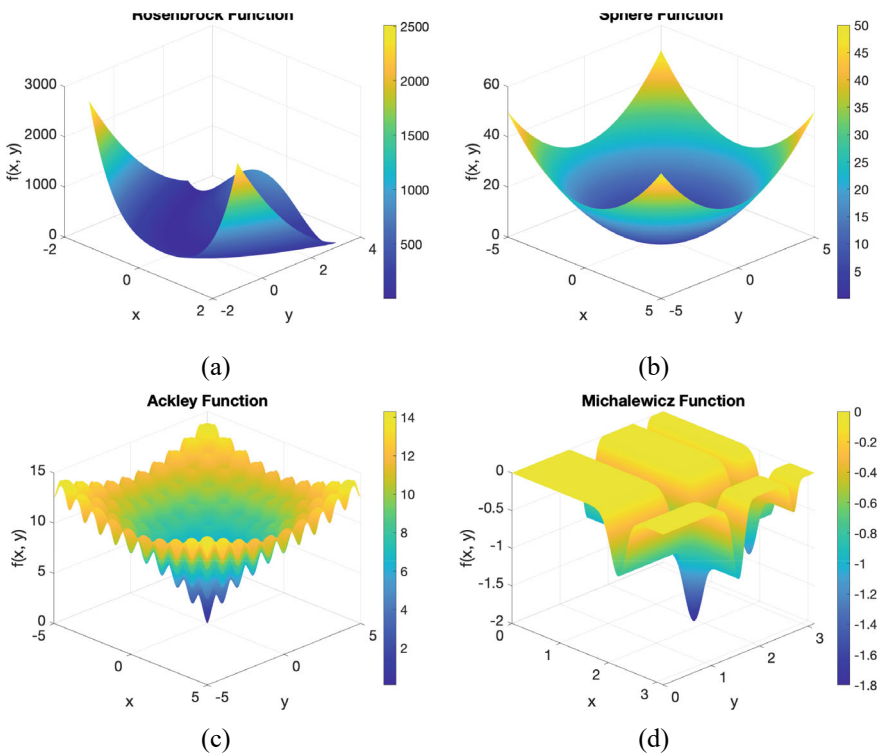


Fig. 3 3-D plots of test functions **a** f_1 —Rosenbrock. **b** f_2 —Sphere. **c** f_3 —Ackley. **d** f_4 —Michalewicz. **e** f_5 —Qing. **f** f_6 —Rastrigin. **g** f_7 —Schwefel. **h** f_8 —Styblinski-Tang

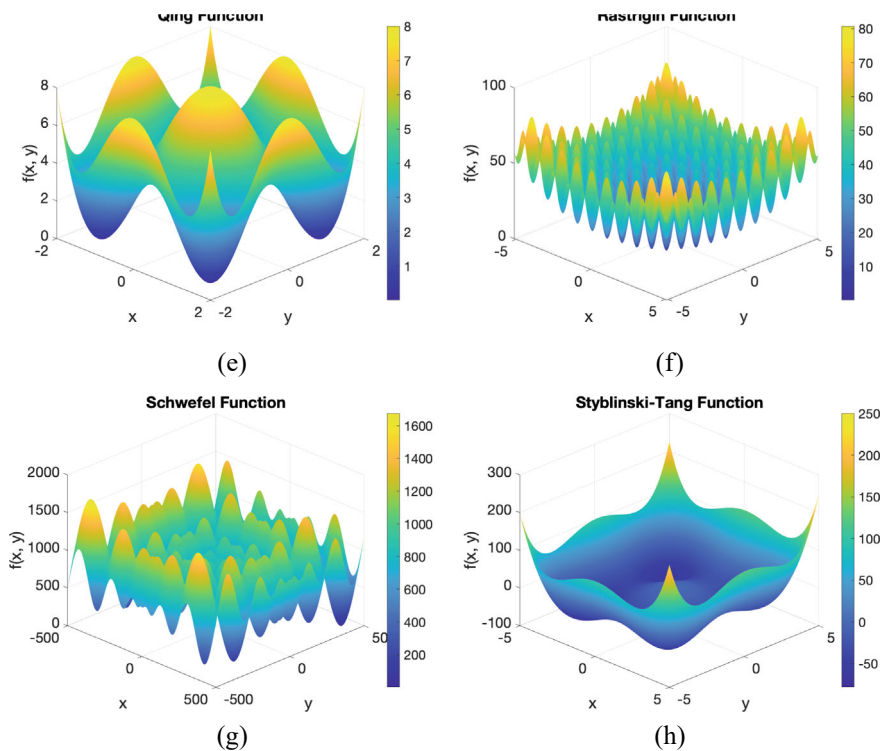


Fig. 3 (continued)

where t is the total CPU time elapsed and ns is the total number of trials.

3. **Average Fitness Evaluation Times (AFET)**: This is the average fitness evaluation times required if it converges successfully, reflecting the convergence speed. The expression of success rate is:

$$AFET = p \frac{i}{ns} \quad (7)$$

where p is the population, i is the iteration times required, and ns is the total number of trials.

4 Results

This section compares the performances of five algorithms on all benchmark test functions and investigates the sensitivity of the social acceleration coefficient c_2 in WSA.

Table 3 Numerical results of optimization on test functions

Function	Dim	Pop	WSA			PSO		
			SR (%)	AFET	AIRT (s)	SR (%)	AFET	AIRT (s)
f_1	2	250	98	1.30E+04	5.70E-04	100	2.57E+03	3.18E-04
	5	500	1	2.76E+05	1.64E-02	2	1.13E+04	1.36E-03
	10	1000	–	–	–	–	–	–
f_2	2	3	97	2.43E+01	8.51E-05	62	7.60E+01	3.30E-04
	5	5	99	8.18E+01	1.15E-04	23	7.68E+02	1.21E-03
	10	10	100	5.41E+02	3.63E-04	28	2.62E+03	2.56E-03
f_3	2	5	100	1.87E+01	6.66E-05	85	1.81E+02	4.10E-04
	5	7	100	2.64E+02	3.32E-04	16	1.10E+03	1.54E-03
	10	10	99	2.33E+03	2.11E-03	2	3.50E+03	3.63E-03
f_4	2	150	97	8.95E+02	6.86E-04	99	1.47E+03	1.10E-03
	5	750	90	6.02E+04	3.47E-02	67	5.25E+04	3.06E-02
	10	2500	92	3.93E+05	3.62E-01	3	3.45E+05	3.04E-01
f_5	2	15	100	4.03E+02	2.35E-04	98	4.61E+02	4.67E-04
	5	50	100	3.70E+03	1.01E-03	93	2.77E+03	9.94E-04
	10	150	100	1.72E+04	2.91E-03	41	2.48E+04	5.23E-03
f_6	2	3	95	4.88E+01	1.09E-04	19	6.60E+01	2.60E-04
	5	5	100	1.34E+02	1.96E-04	–	–	–
	10	10	100	1.09E+03	1.88E-03	–	–	–
f_7	2	500	89	1.04E+05	6.25E-03	97	7.50E+03	6.99E-04
	5	10000	83	2.43E+06	1.41E-01	29	3.38E+05	2.55E-02
	10	50000	86	2.25E+07	4.28E+00	–	–	–
f_8	2	100	99	5.83E+03	2.45E-03	100	1.02E+03	6.90E-04
	5	500	96	1.82E+04	6.17E-03	88	1.15E+04	4.12E-03
	10	1500	99	5.78E+04	3.11E-02	15	1.55E+05	6.93E-02

4.1 Performance Test

The performance of five algorithms, including WSA, PSO, APSO, QPSO, and PSO-GSA, is systematically quantified in Tables 3 and 4. The assessment focuses on benchmark test functions across 2, 5, and 10 dimensions. The results demonstrate the clear superiority of WSA over its competitors in terms of SR, AIRT, and AFET on test functions f_1 , f_2 , f_3 , f_4 , f_5 , and f_8 . WSA achieves an impressive average SR exceeding 95% while demonstrating significantly fewer AIRT and AFET metrics than other algorithms. For test function f_6 , WSA exhibits superior performance in 2D optimization, particularly in convergence speed. For test function f_7 , despite the SR being below 90%, WSA outperforms the other algorithms.

Table 4 Numerical results of optimization on test functions

Function	APSO			QPSO			PSOGSA		
	SR (%)	AFET	AIRT (s)	SR (%)	AFET	AIRT (s)	SR (%)	AFET	AIRT (s)
f_1	100	5.74E+03	1.77E-02	100	5.66E+04	1.27E-01	100	6.55E+03	1.08E-01
	1	9.55E+04	2.82E-01	–	–	–	97	1.53E+05	1.12E+01
	–	–	–	–	–	–	100	4.49E+05	1.28E+02
f_2	100	1.55E+02	1.39E-03	100	3.85E+01	6.10E-04	100	1.08E+02	1.45E-03
	100	4.17E+02	2.55E-03	100	1.45E+02	1.19E-03	100	5.34E+02	4.23E-03
	100	9.82E+02	5.00E-03	100	4.84E+02	3.02E-03	100	1.48E+03	1.11E-02
f_3	85	6.75E+02	3.25E-03	100	2.19E+02	1.90E-03	92	1.04E+03	4.55E-03
	38	1.18E+03	5.39E-03	100	5.46E+02	3.26E-03	56	1.96E+03	7.79E-03
	4	1.77E+03	6.93E-03	100	1.21E+03	5.68E-03	20	3.26E+03	1.72E-02
f_4	100	5.32E+03	3.27E-02	59	5.94E+04	1.47E-01	100	5.63E+03	7.10E-01
	16	7.80E+04	5.14E-01	–	–	–	21	1.41E+05	1.74E+01
	–	–	–	–	–	–	–	–	–
f_5	95	6.93E+02	2.92E-03	49	7.92E+03	1.96E-02	100	7.05E+02	4.16E-03
	64	4.66E+03	1.44E-02	–	–	–	100	7.50E+03	6.64E-02
	4	1.80E+04	4.98E-02	–	–	–	100	3.32E+04	1.44E+00
f_6	10	3.44E+02	1.68E-03	100	1.50E+02	1.30E-03	11	4.41E+02	5.53E-03
	–	–	–	100	5.25E+02	2.81E-03	–	–	–
	–	–	–	100	1.69E+03	7.33E-03	–	–	–
f_7	90	5.01E+03	1.77E-02	1	1.13E+05	3.24E-01	93	6.09E+03	1.80E-01
	7	2.73E+05	1.16E+00	–	–	–	–	–	–
	–	–	–	–	–	–	–	–	–
f_8	96	3.30E+03	1.24E-02	34	4.98E+04	1.40E-01	99	3.17E+03	2.83E-02
	40	4.05E+04	1.53E-01	–	–	–	65	7.17E+04	5.25E+00
	1	1.56E+05	7.10E-01	–	–	–	–	–	–

4.2 Parameter Sensitivity Analysis

In canonical PSO, achieving better performance relies on carefully balancing the acceleration coefficients c_1 , c_2 , and the inertia weight w . However, in WSA, which eliminates w and c_1 , emphasis is placed solely on tuning the parameter c_2 . This exclusive focus enables a comprehensive sensitivity analysis of the social acceleration coefficient c_2 to evaluate its impact on WSA’s performance.

The results presented in Fig. 4 illustrate how variations in c_2 influence WSA’s success rate with 2, 5, and 10 dimensions. Notably, functions f_2 , f_3 , and f_6 exhibit an increasing trend in success rate as c_2 increases, reaching nearly 100% when c_2 exceeds 4.5. Conversely, other functions demonstrate Omega-shaped curves in their performance metrics. An important observation is that the appropriate range of c_2 varies among different dimensions of the same function. The range tends to be narrow as dimensionality increases for functions f_3 , f_4 , f_5 , f_7 , and f_8 . These findings emphasize the significance of the social acceleration coefficient c_2 on the success rate.

4.3 Application of WSA on Robotic Arm: A Case Study

This case study aims to use Iterative Learning Control (ILC) to update the control inputs of a 7-degree-of-freedom WAM Arm continuously. This robotic arm is subjected to repetitive external disturbances and needs to accurately track a circular trajectory with a radius of 0.2 m.

ILC involves several parameters, such as learning gain and feedback gain, that significantly impact the robotic arm's tracking performance and convergence speed. To speed up the convergence of tracking errors, we used the WSA technique to search for optimal gain coefficients in ILC. This method enhances the system's convergence speed and tracking accuracy while also effectively mitigating external disturbances and optimizing the control strategy during iterations.

We compared three methods: ILC, PSO-optimized ILC, and WSA-optimized ILC. To evaluate their performance, we quantified tracking errors using Root Mean Square Error (RMSE). Figure 5 shows the simulation model of the WAM Arm and the desired

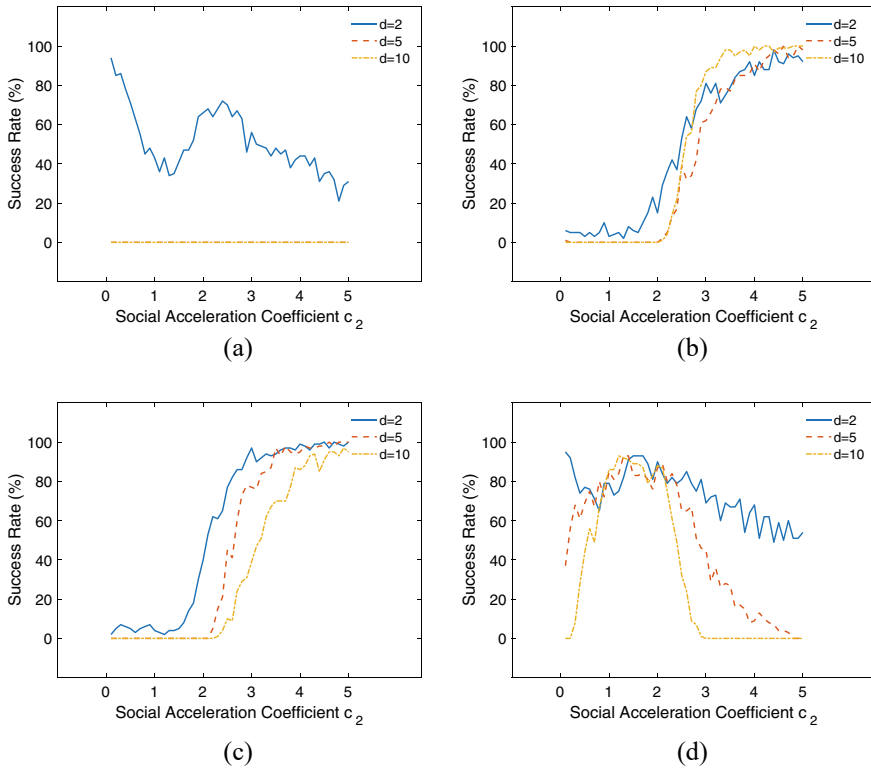


Fig. 4 Success rate of test functions with different social acceleration coefficients **a** f_1 —Rosenbrock. **b** f_2 —Sphere. **c** f_3 —Ackley. **d** f_4 —Michalewicz. **e** f_5 —Qing. **f** f_6 —Rastrigin. **g** f_7 —Schwefel. **h** f_8 —Styblinski-Tang

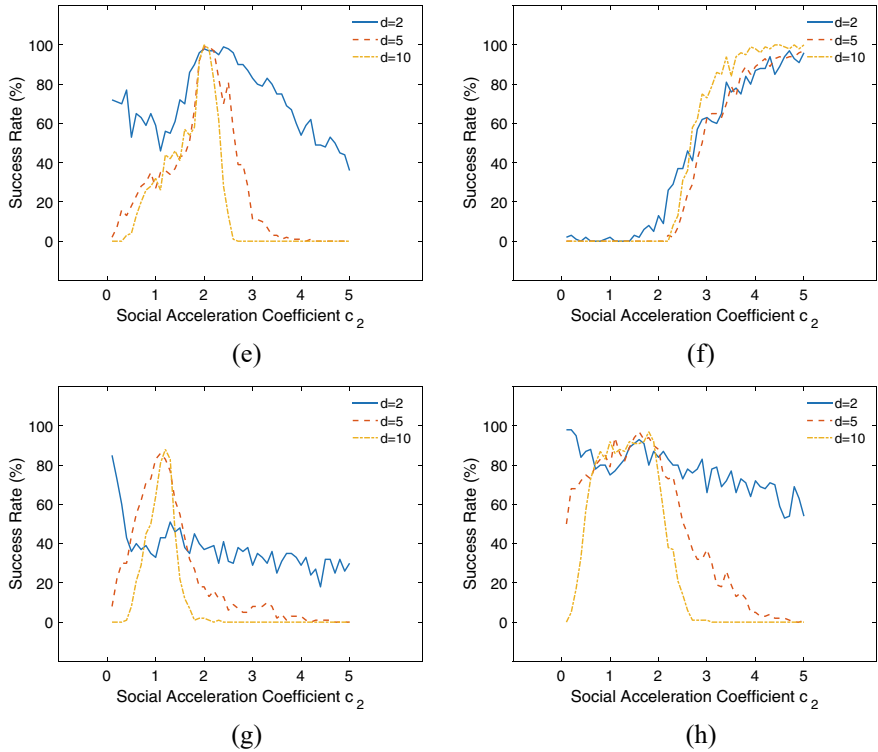


Fig. 4 (continued)

trajectory. Figure 6 shows that WSA-optimized ILC outperforms the other methods in both convergence speed and tracking accuracy, demonstrating its effectiveness and superiority.

5 Conclusion

According to the results, WSA performs better than traditional methods by eliminating the cognitive acceleration coefficient and inertia weight, accelerating convergence, and significantly reducing computational time. This accomplishment can be attributed to its unique design, which removes unnecessary parameters and incorporates a swapping mechanism. Through eight benchmark tests, WSA achieves the highest success rate in seven, highlighting its robustness in solving high-dimensional numerical optimization problems.

Besides, the social acceleration coefficient significantly influences WSA's performance, impacting both the speed and accuracy of convergence. While a clear

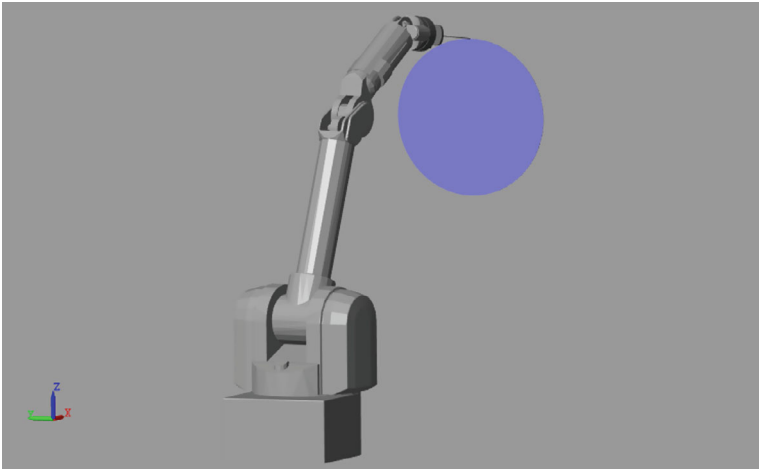


Fig. 5 Simulation interface of WAM arm

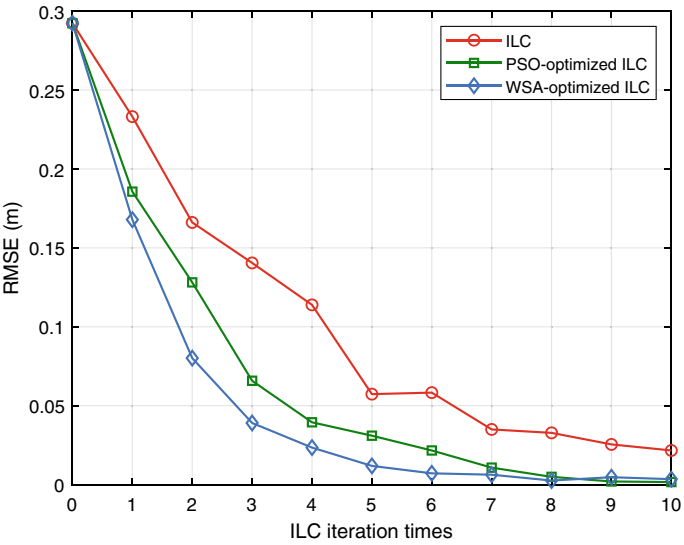


Fig. 6 Comparison of ILC, PSO-optimized ILC, and WSA-optimized ILC

correlation exists between performance and the coefficient value, its impact is not entirely positive or negative, indicating that further empirical research is needed. Optimizing this coefficient is crucial for fully harnessing WSA's potential.

Compared to algorithms like PSO, which necessitate extensive tuning of three different parameters, WSA's single-parameter model simplifies the tuning process and makes it particularly suitable for optimization challenges characterized by low dimensions and smaller population sizes. Thus, it is user-friendly and highly effective for practical optimization tasks.

References

1. Ait-Aoudia S, Guerrout EH, Mahiou R (2014) Medical image segmentation using particle swarm optimization. In: 2014 18th International conference on information visualisation, pp 287–291. <https://doi.org/10.1109/IV.2014.68>
2. Ali AF, Tawhid MA (2017) A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems. *Ain Shams Eng J* 8(2):191–206. <https://doi.org/10.1016/j.asej.2016.07.008>
3. Azadifar S, Ahmadi A (2021) A graph-based gene selection method for medical diagnosis problems using a many-objective PSO algorithm. *BMC Med Inf Decis Mak* 21. <https://doi.org/10.1186/s12911-021-01696-3>
4. Bastos-Filho CJA, Carvalho DF, Caraciolo MP, Miranda PBC, Figueiredo EMN (2009) Multi-ring particle swarm optimization. In: dos Santos WP (ed) *Evolutionary computation*, chap. 28. IntechOpen, Rijeka. <https://doi.org/10.5772/9597>
5. Borowska B (2022) Learning competitive swarm optimization. *Entropy* 24(2). <https://doi.org/10.3390/e24020283>
6. Chakraborty R, Sushil R, Garg ML (2020) Mutual-inclusive learning-based multi-swarm PSO algorithm for image segmentation using an innovative objective function. *Int J Comput Sci Eng* 21(4):483–494. <https://doi.org/10.1504/IJCSE.2020.106864>
7. Chen HH, Li GQ, Liao HI (2009) A self-adaptive improved particle swarm optimization algorithm and its application in available transfer capability calculation. In: 2009 Fifth International conference on natural computation, vol 3, pp 200–205. <https://doi.org/10.1109/ICNC.2009.214>
8. Di Vito A, Pecchia A, Auf der Maur M, Campanari V, Martelli F, Di Carlo A (2021) Role of phase nanosegregation in the photoluminescence spectra of halide perovskites. *J Phys Chem Lett* 12(48):11659–11665. <https://doi.org/10.1021/acs.jpclett.1c03378>
9. Essiet IO, Sun Y, Wang Z (2019) Improved genetic algorithm based on particle swarm optimization-inspired reference point placement. *Eng Optim* 51(7):1097–1114. <https://doi.org/10.1080/0305215X.2018.1509961>
10. Ferreira de Carvalho D, José Albanez Bastos-Filho C (2009) Clan particle swarm optimization. *Int J Intell Comput Cybern* 2(2):197–227. <https://doi.org/10.1108/17563780910959875>
11. Figueiredo EM, Ludermitr TB (2014) Investigating the use of alternative topologies on performance of the pso-elm. *Neurocomputing* 127:4–12. *Advances in Intelligent Systems*. <https://doi.org/10.1016/j.neucom.2013.05.047>
12. Hoffmeister F, Bäck T (1991) Genetic algorithms and evolution strategies: similarities and differences. In: Schwefel HP, Männer R (eds) *Parallel problem solving from nature*. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 455–469
13. Holden N, Freitas A (2008) A hybrid PSO/ACO algorithm for discovering classification rules in data mining. *J Artif Evol App* 2(1–2):11. <https://doi.org/10.1155/2008/316145>
14. Jamil M, Yang XS (2013) A literature survey of benchmark functions for global optimisation problems. *Int J Math Model Numer Optim* 4(2):150. <https://doi.org/10.1504/ijmmno.2013.055204>

15. Jayaprakasam S, Abdul Rahim SK, Leow CY, Ting TO (2017) Sidelobe reduction and capacity improvement of open-loop collaborative beamforming in wireless sensor networks. *PLoS ONE* 12(5):e0175510
16. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of ICNN'95 - International conference on neural networks*, vol 4, pp 1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>
17. Kumar A (2021) Multi-stage, multi-swarm PSO for joint optimization of well placement and control. <https://doi.org/10.48550/ARXIV.2106.01146>
18. Kiran MS, Gündüz M (2013) A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems. *Appl Soft Comput* 13(4):2188–2203. <https://doi.org/10.1016/j.asoc.2012.12.007>
19. Li M, Lin D, Kou J (2012) A hybrid niching PSO enhanced with recombination-replacement crowding strategy for multimodal function optimization. *Appl Soft Comput* 12(3):975–987. <https://doi.org/10.1016/j.asoc.2011.11.032>
20. Li Y, Chen Y, Zhong J, Huang Z (2019) Niching particle swarm optimization with equilibrium factor for multi-modal optimization. *Inf Sci* 494:233–246. <https://doi.org/10.1016/j.ins.2019.01.084>
21. Liu B, Wang L, Jin YH, Tang F, Huang DX (2005) Improved particle swarm optimization combined with chaos. *Chaos, Solitons & Fractals* 25(5):1261–1271. <https://doi.org/10.1016/j.chaos.2004.11.095>
22. Majdi A, Beiki M (2019) Applying evolutionary optimization algorithms for improving fuzzy c-mean clustering performance to predict the deformation modulus of rock mass. *Int J Rock Mech Min Sci* 113:172–182. <https://doi.org/10.1016/j.ijrmms.2018.10.030>
23. Mallick RK, Nahak N (2016) Hybrid differential evolution particle swarm optimization (de-psy) algorithm for optimization of unified power flow controller parameters. In: *2016 IEEE Uttar Pradesh Section International conference on electrical, computer and electronics engineering (UPCON)*, pp 635–640. <https://doi.org/10.1109/UPCON.2016.7894729>
24. Mirjalili S, Hashim SZM (2010) A new hybrid psgsa algorithm for function optimization. In: *2010 International conference on computer and information application*, pp 374–377. <https://doi.org/10.1109/ICCIA.2010.6141614>
25. Mühlenbein H, Schomisch M, Born J (1991) The parallel genetic algorithm as function optimizer. *Parallel Comput* 17(6):619–632. [https://doi.org/10.1016/S0167-8191\(05\)80052-3](https://doi.org/10.1016/S0167-8191(05)80052-3)
26. Nagano K, Collins T, Chen CA, Nakano A (2015) GPU-based inverse rendering with multi-objective particle swarm optimization. In: *SIGGRAPH Asia 2015 visualization in high performance computing*, SA '15. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/2818517.2818523>
27. Namasivayam V, Bajorath J (2012) Searching for coordinated activity cliffs using particle swarm optimization. *J Chem Inf Model* 52(4):927–934. <https://doi.org/10.1021/ci3000503>
28. Ni Q, Yin X, Tian K, Zhai Y (2017) Particle swarm optimization with dynamic random population topology strategies for a generalized portfolio selection problem. *Nat Comput: Int J* 16(1):31–44. <https://doi.org/10.1007/s11047-016-9541-x>
29. Nickabadi A, Ebadzadeh MM, Safabakhsh R (2011) A novel particle swarm optimization algorithm with adaptive inertia weight. *Appl Soft Comput* 11(4):3658–3670. <https://doi.org/10.1016/j.asoc.2011.01.037>
30. Picheny V, Wagner T, Ginsbourger D (2013) A benchmark of kriging-based infill criteria for noisy optimization. *Struct Multidiscipl Optim* 48. <https://doi.org/10.1007/s00158-013-0919-4>
31. Prasad J, Souradeep T (2012) Cosmological parameter estimation using particle swarm optimization. *Phys Rev D* 85:123008. <https://doi.org/10.1103/PhysRevD.85.123008>
32. Rahnamayan S, Tizhoosh HR, Salama MM (2007) A novel population initialization method for accelerating evolutionary algorithms. *Comput Math Appl* 53(10):1605–1614. <https://doi.org/10.1016/j.camwa.2006.07.013>
33. Ratnaweera A, Halgamuge S, Watson H (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans Evol Comput* 8(3):240–255. <https://doi.org/10.1109/TEVC.2004.826071>

34. Reddy JS et al (2021) Weightless swarm algorithm as transformation technique for improving the performance of machine learning algorithms. In: 2021 5th International conference on trends in electronics and informatics (ICOEI). IEEE, pp 1491–1494
35. Rosso MM, Cucuzza R, Aloisio A, Marano GC (2022) Enhanced multi-strategy particle swarm optimization for constrained problems with an evolutionary-strategies-based unfeasible local search operator. *Appl Sci* 12(5). <https://doi.org/10.3390/app12052285>
36. Sadoughi M, Pourdadaashnia A, Farhadi-Kangarlu M, Galvani S (2022) Pso-optimized she-pwm technique in a cascaded h-bridge multilevel inverter for variable output voltage applications. *IEEE Trans Power Electron* 37(7):8065–8075. <https://doi.org/10.1109/TPEL.2022.3146825>
37. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: 1998 IEEE International conference on evolutionary computation proceedings. In: IEEE World congress on computational intelligence (Cat. No.98TH8360), pp 69–73. <https://doi.org/10.1109/ICEC.1998.699146>
38. Shi Y, Eberhart R (1999) Empirical study of particle swarm optimization. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406), vol 3, pp 1945–1950. <https://doi.org/10.1109/CEC.1999.785511>
39. Shi Y, Eberhart RC (1998) Parameter selection in particle swarm optimization. In: Evolutionary programming VII. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 591–600
40. Singh DN, Singh S (2017) Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *J Appl Math* 2017:15. <https://doi.org/10.1155/2017/2030489>
41. Sukpancharoen S, Srinophakun TR, Hirunlabh J (2018) The application of a mixed coding approach to address mixed integer linear and non-linear programming problems using particle swarm optimization (PSO) with an artificial bee colony (ABC) algorithm. In: Proceedings of the 2nd International conference on intelligent systems, metaheuristics & swarm intelligence, ISMSI '18. Association for Computing Machinery, New York, NY, USA, pp 78–83. <https://doi.org/10.1145/3206185.3206201>
42. Sun J, Lai CH, Xu W, Ding Y, Chai Z (2007) A modified quantum-behaved particle swarm optimization. In: Computational science—ICCS 2007. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 294–301
43. Ting T (2015) Optimization of drilling process via weightless swarm algorithm. In: Emerging research on swarm intelligence and algorithm optimization. IGI Global, pp 181–190
44. Ting T, Man K, Guan SU, Seon J, Jeong T, Wong PW (2012) Maximum power point tracking (mppt) via weightless swarm algorithm (WSA) on cloudy days. In: 2012 IEEE Asia Pacific conference on circuits and systems, pp 336–339. <https://doi.org/10.1109/APCCAS.2012.6419040>
45. Ting T, Man KL, Guan SU, Seon J, Jeong TT, Wong PW (2012) Maximum power point tracking (MPPT) via weightless swarm algorithm (WSA) on cloudy days. In: 2012 IEEE Asia Pacific conference on circuits and systems. IEEE, pp 336–339
46. Ting TO, Man KL, Guan SU, Nayel M, Wan K (2012) Weightless swarm algorithm (WSA) for dynamic optimization problems. In: Network and parallel computing. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 508–515
47. Weerathunga TS, Mohanty SD (2017) Performance of particle swarm optimization on the fully-coherent all-sky search for gravitational waves from compact binary coalescences. *Phys Rev D* 95:124030. <https://doi.org/10.1103/PhysRevD.95.124030>
48. Wu Z, Zhou J (2007) A self-adaptive particle swarm optimization algorithm with individual coefficients adjustment. In: 2007 International conference on computational intelligence and security (CIS 2007), pp 133–136. <https://doi.org/10.1109/CIS.2007.95>
49. Xia X, Gui L, Zhan ZH (2018) A multi-swarm particle swarm optimization algorithm based on dynamical topology and purposeful detecting. *Appl Soft Comput* 67:126–140. <https://doi.org/10.1016/j.asoc.2018.02.042>
50. Xu G (2013) An adaptive parameter tuning of particle swarm optimization algorithm. *Appl Math Comput* 219(9):4560–4569. <https://doi.org/10.1016/j.amc.2012.10.067>
51. Xu L, Song B, Cao M (2021) An improved particle swarm optimization algorithm with adaptive weighted delay velocity. *Syst Sci Control Eng* 9(1):188–197. <https://doi.org/10.1080/21642583.2021.1891153>

52. Yang XS, Deb S, Fong S (2011) Accelerated particle swarm optimization and support vector machine for business optimization and applications. In: Networked digital technologies. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 53–66
53. Yu X, Zhang X (2017) Multiswarm comprehensive learning particle swarm optimization for solving multiobjective optimization problems. PLoS ONE 12(2):1–21. <https://doi.org/10.1371/journal.pone.0172033>
54. Zhu H, Hu Y, Zhu W (2019) A dynamic adaptive particle swarm optimization and genetic algorithm for different constrained engineering design optimization problems. Adv Mech Eng 11(3):1687814018824930. <https://doi.org/10.1177/1687814018824930>

Chapter 16

On Complex Neural Networks



Momin Jamil Sandhu[✉] and Xin-She Yang[✉]

1 Introduction

Complex numbers are not an abstract mathematical concept. They occur ubiquitously, either naturally or by design, in a variety of signal and image processing applications. The data generated in these applications is inherently complex. Complex numbers are a convenient way of representing signals in many applications, including but not limited to communication, radar, and magnetic resonance imaging (MRI). Further, simpler and more effective data processing is possible in the complex domain $\mathbb{C}$, even for real-valued data applications. For instance, speech processing applications often transform real-valued data into complex spectrograms to visualize frequencies and harmonics. Similarly, computer vision and image processing applications convert the data into a frequency domain for interpretation and efficient processing. Real numbers, among others, are a subset of complex numbers. Almost every application prefers to implement real numbers due to their low computational complexity and implementation effort. As a result, they have become a formidable choice of implementation in machine learning (ML) models based on neural networks (NN).

NN based on real values are commonly known as real-valued neural networks (RvNN's). They are most widely used in almost every application compared to their complex-valued counterpart. In RvNN's, the parameters $\mathbf{x}$, $\mathbf{y}$, $\mathbf{b}$, $\mathbf{w} \in \mathbb{R}$ represent input, output, bias, and weights, respectively. In contrast to RvNN's, the parameters of complex-valued neural networks (CvNN's) $\mathbf{x}$, $\mathbf{y}$, $\mathbf{b}$, $\mathbf{w} \in \mathbb{C}$. In other words, CvNN's can process complex-valued parameters. The output of CvNN's is either real or complex, depending on the application at hand [1].

M. J. Sandhu (✉)

Harman Becker Automotive Systems GmbH, Becker Göring Str. 16, Karlsbad, Germany
e-mail: momin.jamil@harman.com

X.-S. Yang

Middlesex University, The Burroughs, Hendon, London NW4 4BT, UK
e-mail: x.yang@mdx.ac.uk

A nonzero complex number can be defined by its magnitude and phase. The majority of signal processing algorithms rely on absolute values, complex conjugates, and phase rotations. It is, therefore, preferable to use CvNN's as opposed to RvNN's for signal and image processing applications. In these applications, the significance of phase cannot be ignored. In [2], it was illustrated that the phase contains information related to shapes and edges, while the contrast information and contribution of certain frequencies are contained in the magnitude of the signal. This can be illustrated with the help of Fig. 1b and c, respectively. These figures represent the reconstruction of an original image in Fig. 1a using the inverse Fourier transform. Figure 1b and c shows the reconstructed magnitude-only and phase-only images, respectively. Figure 1b illustrates that the magnitude-only image is a faded image with only a little indication of the image details and features. On the contrary, Fig. 1c demonstrates that the phase-only reconstructed image preserves the image outlines and contains far more signal-dependent information. One way to deal with complex input data is to separate it into real/imaginary, or magnitude/phase parts. They can then be used as an input to RvNN's, thus providing a way to deal with problems with complex characteristics [3, 4]. However, such approaches are not practical as they neglect the statistical coupling or coherence between the real and imaginary components [3, 6–8].

Despite frequent complex signal representation in many practical signal and image processing applications, the use of CvNN's is marginalized. Their marginalization can be attributed to factors like (i) convoluted design architecture, (ii) differentiability of complex activation functions, (iii) complex network parameters, and (iv) increased computational complexity [3, 4].

The first formal description of CvNN was introduced by Clarke in [9]. A CvNN with complex-valued $\mathbf{x}$, $\mathbf{y}$, $\mathbf{b}$, $\mathbf{w}$ was proposed in [10]. A commonly used back-propagation algorithm was extended to complex domain by [11–14]. After their conception in the early 1990s, basic ideas, fundamentals, and applications of CvNN's gained significant attention, see, e.g., [3, 4, 6–8, 15, 16]. CvNN's are recognized for their ability to facilitate more efficient optimization [17], improved generalization properties [18], accelerated learning [19, 21], resilient memory mechanisms

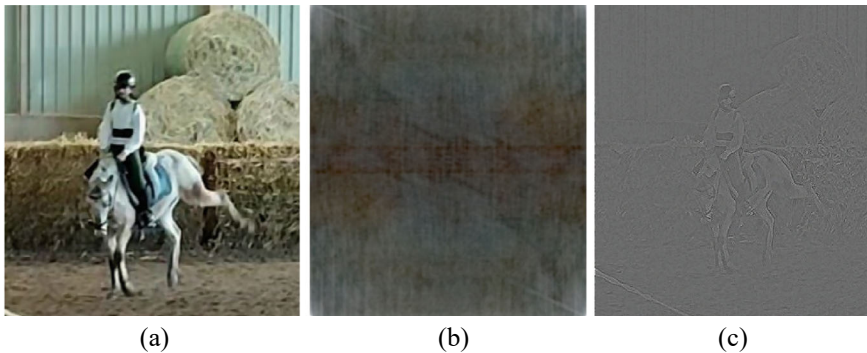


Fig. 1 Plots of the **a** original image **b** the magnitude-only image, and **c** the phase-only image

against noise [19], and enhanced representational capacity in recurrent neural networks (RNN's) [19, 20]. The complex-valued long short-term memory (LSTM) network and its advantages over real-valued LSTM were introduced and discussed in [21].

CvNN's can deal with both amplitude and phase information in many practical applications. The extraction of both amplitude and phase information is possible, when training is performed using CvNN's as opposed to RvNN's. Therefore, CvNN's are the natural and most compatible choice for signal and image processing applications, among others. Soon after the conception of CvNN's, there was a realization that the complex domain extension of NN was not straightforward due to a lack of complex activation functions.

The subsequent sections of the chapter are organized in the following manner. Section 2 explores the rationale behind CvNN's, various categories of CvNN's, and its fundamental components. In Sect. 3, a review of the existing state-of-the-art development tools for CvNN's is provided. The dataset issues for signal processing applications, both in general and specifically for radio applications, are discussed in Sect. 4. The viability of using CvNN's for real-world applications is discussed in Sect. 5, taking into account the current research findings. Lastly, Sect. 6 concludes the chapter.

2 Complex-Valued Neural Networks

2.1 Notations

In this chapter, we use italic boldface upper- and lower-case letters to denote matrices, and column vectors, respectively. The boldface upper- and lower-case letters denoted stacked matrices and vectors, respectively. The subscript n denotes the n -th element of a vector. The convolutional operator, partial derivative, and expectation operator are denoted by $*$, ∂ , and $\mathbb{E}[\cdot]$, respectively. The sets of real and complex domains are denoted by the symbols $\mathbb{R}$ and $\mathbb{C}$, respectively. We use the symbols z , r , ϕ , and j to denote complex numbers, magnitude, phase, and imaginary part of complex number.

2.2 Why Complex-Valued Neural Networks?

The conceptual difference between RvNN's and CvNN's is depicted in Fig. 2a–b, respectively. CvNN's possess complex-valued input and work directly with complex values without the need to pre-process them by casting them to real numbers. Each layer of the CvNN's operates analogously to a real-valued layer. The difference is that the operations (addition, multiplication, convolution, etc.) and trainable parameters (weights, bias, etc.) are all in the complex domain. The activation functions are also defined in the complex domain, see Fig. 4 and Sect. 2.4.3 for an overview and discussion on activation functions. The questions arise: (i) Why are CvNN's needed

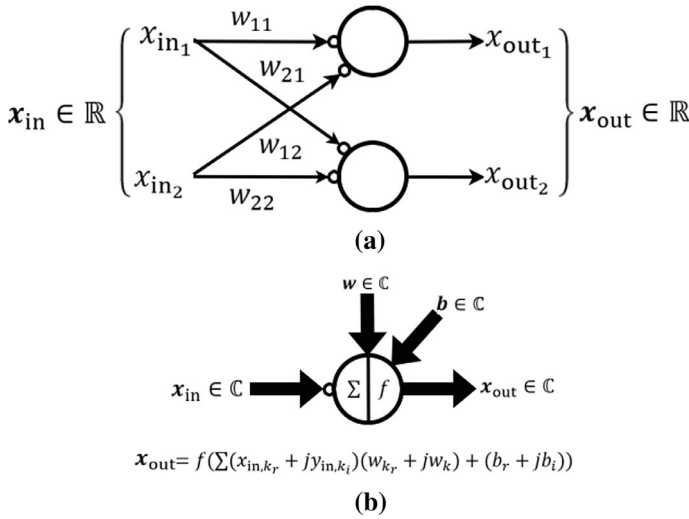


Fig. 2 Conceptual difference between RvNN and CvNN [5]. The final output of the CvNN is a weighted sum of the complex inputs followed by applying the activation function f

when a complex number $z = a + jb$ can be expressed by an ordered pair of real-valued numbers (a, b) ? (ii) Why not use a RvNN's with two channels (as shown in Fig. 2a)? Two main reasons for their advocacy are:

- In image and signal processing, the real and imaginary components of a signal exhibit statistical coherence. In the time domain, multiplying a signal by a scalar quantity is equivalent to multiplying the signal's magnitude in the frequency domain. A linear phase shift in the frequency domain, on the other hand, results from a circular rotation of a signal in the time domain. This indicates a statistical correlation between the real and imaginary components of the signal during a phase transition or alteration.
- If the information regarding the relationship between magnitude and phase and its significance to the learning objective is known in advance, it is more appropriate to deploy a CvNN.

Complex numbers have two degrees of freedom (DoF), usually expressed as the real and imaginary components, or by the magnitude and phase, as given by (1)

$$z = a + jb = |z|e^{j\angle z} \quad (1)$$

where $j = \sqrt{-1}$, $\mathbf{a} = (a_1, a_2, \dots, a_n) \in \mathbb{R}^n$ and $\mathbf{b} = (b_1, b_2, \dots, b_n) \in \mathbb{R}^n$ are the real and imaginary parts of z , $|z|$ and $\angle z$ represent magnitude and phase of z , respectively. The $\mathbf{a}$ & $\mathbf{b}$ denote the set of all ordered n -tuples of distinct real-valued entities. It means that in computer simulations, the complex arithmetic is simulated using real-valued arithmetic.

The complex number can also be represented as an ordered pair of real numbers, so z can be represented as

$$z \equiv (a, b). \quad (2)$$

The input value can be split into either rectangular (real and imaginary) or polar coordinates (phase and magnitude), as shown in (3) and (4), respectively. This is an effective way to represent CvNN's to a double-dimension RvNN's.

$$\{z_1, z_2\} = \{(a_1, b_1), (a_2, b_2)\} \quad (3)$$

$$\{z_1, z_2\} = \{(r_1, \phi_1), (r_2, \phi_2)\}, \quad (4)$$

where $z_{1,2} \in \mathbb{C}$, while r and ϕ represent the magnitude and phase of z .

CvNN's deal with complex numbers as opposed to RvNN's. The input data mapping of RvNN's and CvNN's can be represented as $f(z) : \mathbb{R}^{2N} \rightarrow \mathbb{R}$ and $f(z) : \mathbb{C} \rightarrow \mathbb{C}$, respectively [4, 26]. The former acts as a multiplexer, splitting the real and imaginary components of complex-valued data into two independent streams. This equates to doubling the number of learnable parameters compared to the latter. A comparison of learnable parameters in the complex-valued function approximation was studied in [27]. The study demonstrated that CvNN's outperform RvNN's in terms of simplicity and generalization ability.

2.3 Types of CvNN's

Figure 3 illustrates many forms of CvNN's and their variants [4] and references therein. Fully CvNN's with complex-valued weights and activation functions were proposed in [5, 8]. The rationale behind fully utilizing CvNN's is their capacity to retain the information conveyed by the phase component. Some of the applications deploying full CvNN's include estimation of direction of arrival and beam-forming [28, 29], magnetic resonance imaging (MRI) image reconstruction [30], synthetic aperture radar (SAR) image interpretation [31], channel equalization [32–34]. The use of CvNN's also presents several challenges. These include, but are not limited to, increased computational complexity, the difficulty of obtaining complex-differentiable (or holomorphic) activation functions [4, 13], susceptibility to noise [1], a lack of appropriate training datasets [35], and limited availability of evaluation metrics for comparative studies [36].

The definition in (2) forms the basis of split-CvNN's, where a and b serve as inputs into the RvNN's. Split-CvNN's are further divided into two categories, as shown in Fig. 3. The two categories are (i) those with real-valued weights and activation functions, and (ii) those with complex-valued weights and real-valued activation functions. The main drawback of the first configuration is that the input data dimen-

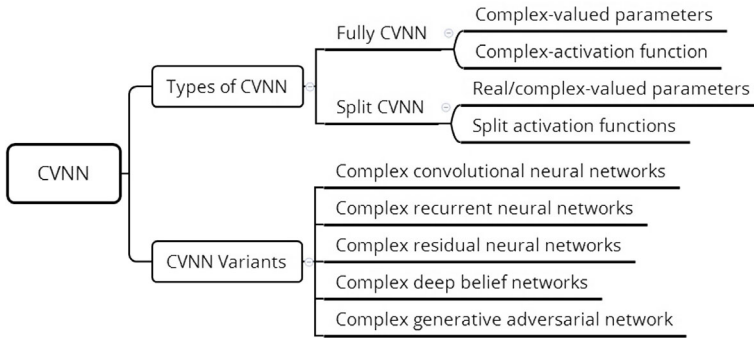


Fig. 3 Types of CvNN's and its variants

sions and learnable parameters increase the network size. The lack of coherence between the real and imaginary parts introduces a phase distortion [6, 22]. Further, such configuration approximates the output by using real-valued gradients instead of true complex gradients [24, 33]. Furthermore, inconsistencies in the approximation of the input were also reported in [37]. The sensitivity analysis and convergence of such a configuration strongly depend on proper initialization and the choice of learning rate [33]. On the other hand, deploying the second configuration can address the drawbacks of the first configuration, such as phase distortion.

2.4 Building Blocks of CvNN's Architecture

In this section, we briefly explore and present an introduction of basic layers that serve as the main building blocks of CvNN's.

2.4.1 Complex Convolution Layer

Complex numbers can be represented as ordered pairs of real numbers according to (2). Consequently, the addition and multiplication of z_1 and z_2 can be mathematically expressed by (5) and (6) in a complex domain, as [7]

$$(a_1, b_1) + (a_2, b_2) = (a_1 + a_2, b_1 + b_2) \quad (5)$$

$$(a_1, b_1) \cdot (a_2, b_2) = (a_1 a_2 - b_1 b_2, a_1 b_2 + a_2 b_1). \quad (6)$$

For two-dimensional real values, the above equations, addition and multiplication, can be expressed as (7) and (8)

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2) \quad (7)$$

$$(x_1, y_1) \cdot (x_2, y_2) = (x_1 x_2, y_1 y_2). \quad (8)$$

By comparing, it is clear that from (5) to (8), the addition process in both the $\mathbb{R}$ and $\mathbb{C}$ domains is identical; however, the multiplication of two complex numbers is unique. The cross-terms in (6) represent both phase rotation and amplitude amplification or attenuation [7], which makes the design of CvNN's a challenging task. It further indicates that complex parameters would increase the CvNN's computational complexity compared to the RvNN's. CvNN's would require two additions and four real multiplications, as opposed to only a single multiplication for RvNN's. It is also an important result, which indicates that RvNN's cannot be interpreted as just a double dimensional extension of CvNN's. In other words, simply increasing the number of real-valued parameters by two will not produce the same outcome as in CvNN's.

A distinguishing feature of CvNN's compared to RvNN's is the presence of a complex-valued convolutional layer. Each convolutional block in the layer consists of a complex kernel or weight matrix represented by $\mathbf{W} = \mathbf{M} + j\mathbf{N}$ and a complex input vector represented by $\mathbf{z} = \mathbf{a} + j\mathbf{b}$. The complex convolution, as dictated by the distributive property, can be represented by equation (9)

$$\mathbf{W} * \mathbf{h} = (\mathbf{M} * \mathbf{a} - \mathbf{N} * \mathbf{b}) + j(\mathbf{N} * \mathbf{a} + \mathbf{M} * \mathbf{b}). \quad (9)$$

The above equation can also be represented in vector form as:

$$\begin{bmatrix} \text{Re}(\mathbf{W} * \mathbf{z}) \\ \text{Im}(\mathbf{W} * \mathbf{z}) \end{bmatrix} = \begin{bmatrix} \mathbf{M} & -\mathbf{N} \\ \mathbf{N} & \mathbf{M} \end{bmatrix} \times \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \quad (10)$$

The complex-valued convolutional layer needs two real-valued convolutional weight matrices, $\mathbf{M}$ and $\mathbf{N}$, stacked together in a three-dimensional array, as shown in equation (9). The output of the complex-valued convolution operator, applied to the complex-valued input vector $\mathbf{z}$, consists of four components: $(\mathbf{M}, \mathbf{N}, \mathbf{a}, \mathbf{b})$. The difference between them is that the $\mathbf{O}_r$ is obtained by subtracting $\mathbf{N} * \mathbf{b}$ from $\mathbf{M} * \mathbf{a}$, while the $\mathbf{O}_i$ is obtained by adding the cross-terms $\mathbf{N} * \mathbf{a}$ and $\mathbf{M} * \mathbf{b}$. Here, $\mathbf{O}_{r,i}$ represent the real and imaginary parts of the output, respectively. In other words, unlike basic linear transformations in RvNN's, the complex-valued full connection carries out the convolution of the real and imaginary components of neurons. The primary distinction between CvNN's and RvNN's is the interleaving of real and imaginary components of CvNN.

2.4.2 Complex Batch Normalization

The standard batch normalization is applicable to real numbers. Its role is to accelerate learning and optimize models in some cases [38]. Due to the statistical interdependence between the real and imaginary components of a complex number, the real

and imaginary parts cannot be independently normalized [18]. A batch normalization formulation for complex values was proposed by [40]. In order to effectively manage the normalization process in the complex domain, it was suggested to consider complex numbers as two-dimensional vectors [40]. Then, re-scale the 2D vectors by the square root of their variance along each of the two principal components. Lastly, multiplying the $\mathbf{0}$ -centered data $\mathbf{x} - \mathbf{E}[\mathbf{x}]$ by the inverse square root of the 2×2 covariance matrix $\mathbf{K}$ [40]:

$$\tilde{\mathbf{x}} = (\mathbf{K})^{-\frac{1}{2}} (\mathbf{x} - \mathbf{E}[\mathbf{x}]), \quad (11)$$

where the covariance matrix $\mathbf{K}$ is given as

$$\begin{bmatrix} \mathbf{K}_{r,r} & \mathbf{K}_{r,i} \\ \mathbf{K}_{i,r} & \mathbf{K}_{i,i} \end{bmatrix} = \begin{bmatrix} \text{cov}(\mathbf{x}_r, \mathbf{x}_r) & \text{cov}(\mathbf{x}_r, \mathbf{x}_i) \\ \text{cov}(\mathbf{x}_i, \mathbf{x}_r) & \text{cov}(\mathbf{x}_i, \mathbf{x}_i) \end{bmatrix}. \quad (12)$$

The normalization procedure allows to de-correlate of the imaginary and real parts, which avoids co-adaptation between the two components and reduces the risk of over-fitting [39]. Finally, complex batch normalization is defined as [40]:

$$BN(\mathbf{x}) = \gamma \cdot \mathbf{x} + \beta, \quad (13)$$

where γ and β are scaling and shift parameters, respectively, see [40] and references within for further discussion of this topic. When batch normalization is not feasible, appropriate complex weight initialization becomes crucial to mitigate the potential vanishing or exploding gradients [42].

2.4.3 Types of Activation Functions

An artificial neuron receives input signals from other neurons. It then assigns a specific weight to each input signal and calculates the sum of the weighted input signals. Finally, using an activation function, the neuron produces an output signal. Activation functions could possess linear or nonlinear characteristics. Nonlinear activation functions are desirable over linear activation functions. The latter possesses limited processing capability compared to the former. NN achieves its versatile processing capabilities by applying nonlinear activation functions.

Choosing a suitable activation function is the most challenging aspect of the overall CvNN design. Liouville's theorem of complex analysis dictates limitations on the selection of activation functions. According to this theorem, a complex function is constant if it is bounded and analytical (holomorphic) everywhere; i.e., a function is differentiable at every point in a complex plane. CvNN's integrate non-holomorphic features through two methods: (i) loss function minimization over complex parameters and (ii) the commonly used activation function. Non-holomorphic functions do not possess complex derivatives. This restricts the activation function's selection or choice. Non-holomorphism is unavoidable if: (i) the loss function is real-valued; (ii)

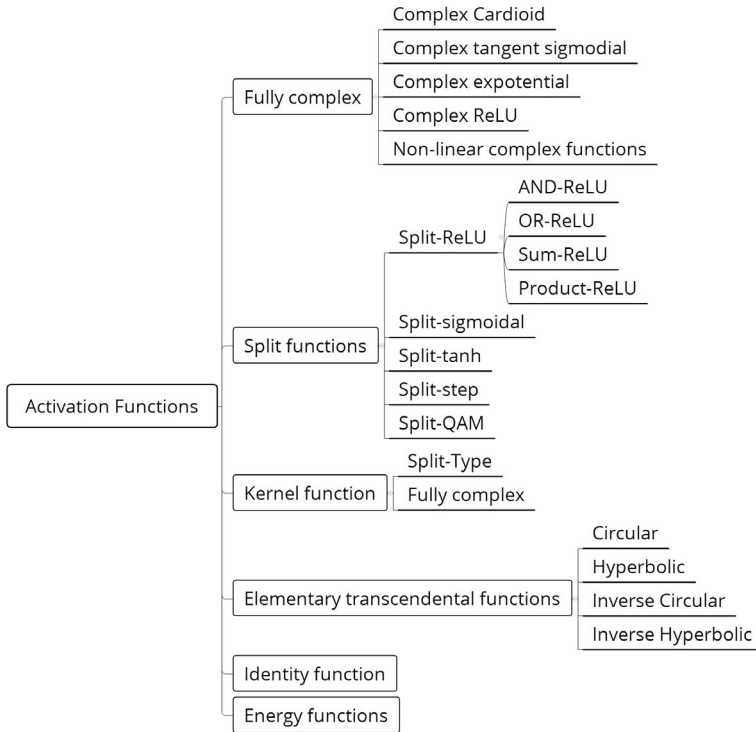


Fig. 4 Types of activation functions

boundedness and analyticity of the loss function at the same time. In the complex domain, boundedness and analyticity cannot be simultaneously achieved. According to [23], boundedness is often preferred over analyticity for activation functions in complex domain. A simple, systematic, and elegant solution to deal with non-holomorphic complex-valued functions is *Wirtinger calculus* [24].

Figure 4 depicts the classification of different activation functions, see [25] and references therein. These activation functions are divided into two main categories: split and fully complex. In the former, real and imaginary parts are treated independently, as given by (14), while in the latter, real and imaginary parts are treated as a single entity. Depending on the nature of the final output, the split activation function can be further divided either into a complex domain or converted into a real domain from a complex domain. The complex domain conversion is performed by computing the absolute value of the real and imaginary parts using (15). Figures 5, 6, and 7 depict exemplary fully complex, split-sigmoidal, and amplitude-phase-type activation functions.

$$f(z) = f_r(\mathbf{x}) + j f_r(\mathbf{y}) \quad (14)$$

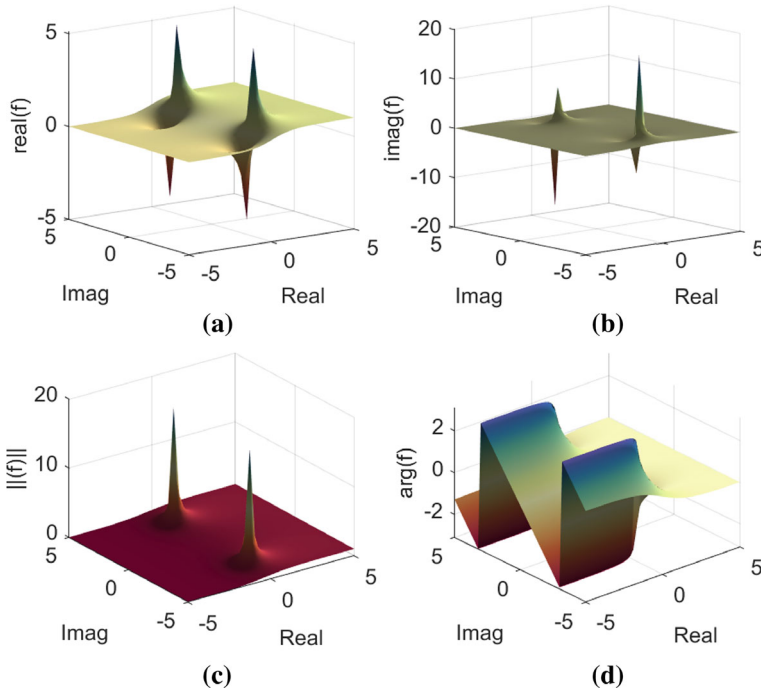


Fig. 5 Complex-valued function $f(z) = 1/(1 + e^{-z})$, where $z = x + jy$. **a** real part, **b** imaginary part **c** amplitude, and **d** phase of the function

$$f(z) = \sqrt{(f_r(x))^2 + j(f_r(y))^2}. \quad (15)$$

2.4.4 Complex Pooling

The pooling layer is crucial for compressing feature information by removing redundancy and enhancing the representation power of a neural network through non-linear mapping. Complex maximum pooling and complex average pooling have been reported in [41]. The real-valued maximum pooling selects the highest value among the features in the neighborhood. However, complex-valued maximum pooling (CMP) takes into account the direct correspondence between the magnitude and phase of complex values [41].

Real-valued average pooling averages the elements in a feature map through sliding windows. Complex average pooling (CAP), on the other hand, separately averages the real and imaginary parts of complex values [40, 41] and references within for more information.

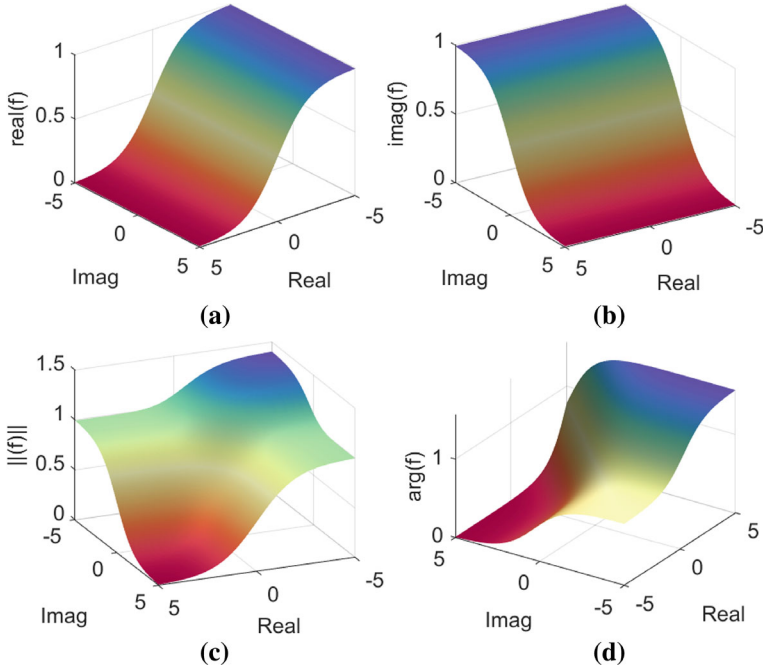


Fig. 6 Split complex activation function $f(z) = \log(1/(1 + e^{-z}))$: **a** real part **b** imaginary part **c** amplitude, and **d** phase of the function

2.4.5 Complex Differentiability

A function $f(z)$ for any complex-valued variable $z = a + jb$ can be written as

$$f(z) = f(z, z^*) = \Re(f) + \Im(f) = p(a, b) + j \cdot q(a, b) \quad \text{where } j = \sqrt{-1}. \quad (16)$$

If the partial derivatives pa , pb , qa , and qb exist and satisfy the Cauchy-Riemann equations, i.e., $pa = qb$ and $pb = -qa$, then $f(z)$ is called a holomorphic function and its derivative with respect to z exists. On the contrary, it is called a non-holomorphic function and its derivative w.r.t. the complex-valued variable z does not exist, i.e., $q(a, b) = 0$

A simple, systematic, and elegant solution to deal with non-holomorphic complex-valued functions is *Wirtinger calculus* [24]. The Wirtinger calculus is an extension of the concept of complex derivative that incorporates the use of conjugate coordinates. Wirtinger introduced conjugate coordinates to derive the derivative of a non-holomorphic function $f(z)$ with respect to z in the complex domain, as follows:

$$\frac{\partial f}{\partial z} = \frac{1}{2} \left(\frac{\partial f}{\partial a} - j \frac{\partial f}{\partial b} \right) \quad (17)$$

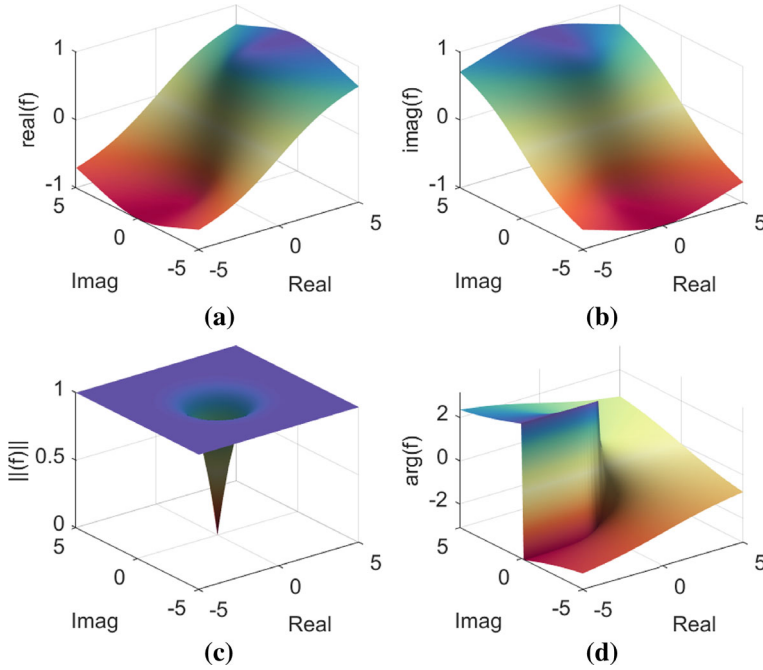


Fig. 7 Split complex activation function $f(z) = \tanh(|z|)e^{j\angle z}$: **a** real part **b** imaginary part **c** amplitude, and **d** phase of the function

$$\frac{\partial f}{\partial z^*} = \frac{1}{2} \left(\frac{\partial f}{\partial a} + j \frac{\partial f}{\partial b} \right), \quad (18)$$

where $\left(\frac{\partial f}{\partial z}\right)$ and $\left(\frac{\partial f}{\partial z^*}\right)$ are known as $\mathbb{R}$ and conjugate $\mathbb{R}$ derivative, respectively. In addition, the variables z and z^* are independent of each other. This implies that the derivatives of the function $f(z)$ w.r.t z are not necessarily equal to the derivative of $f(z)$ w.r.t z^* . This property allows us to solve the derivative of a non-holomorphic function with respect to the complex-valued variable.

2.4.6 Complex-Valued Back Propagation

One of the most fundamental building blocks in NN is the back-propagation algorithm. It aims to minimize the cost function by adjusting NN weights and biases. The gradients of the cost function w.r.t those parameters determine the required level of adjustment. The gradient of a function determines how much the parameter needs to change (in a positive or negative direction) to minimize the cost function at hand.

The back-propagation algorithm uses the chain rule to calculate the gradients of the objective function w.r.t. all adjustable parameters. The gradient optimization algorithm, such as gradient descent, updates the parameters. Open-source software frameworks such as TensorFlow and PyTorch support complex-valued back-propagation algorithm to calculate complex gradients [51, 52]. Back-propagation has been extensively studied, and well documented and has been extended to CvNN's such as complex-valued feedforward neural networks (CvFNNs) [5, 11–13]. For the sake of brevity, detailed discussion and derivation are omitted here.

3 Complex-Valued Neural Networks Libraries Current State of Development

Much of the work in signal processing often relies on RvNN's and renounces potential performance benefits that can possibly be achieved with CvNN's. Despite complex signal representation in practical signal processing applications, the use of CvNN's seems to be marginalized. Their marginalization can be attributed to: (i) convoluted design architecture, (ii) differentiability of complex activation functions, (iii) complex network parameters, (iv) increased computational complexity, (v) availability of appropriate datasets, and (vi) most importantly, a lack of off-the-self CvNN's models compared to their real-valued counter-parts [3, 4].

Ever since the introduction of complex-valued deep NN in 2018 [40], there has been a growing interest in the adoption of CvNN's for several applications, see, e.g., [4]. The introduction of key concepts such as complex convolutions, complex batch normalization, complex weight initialization strategies, and complex differentiation in [40] and their application to convolutional feedforward have sparked an increasing interest in CvNN's. The work in [40] showed that CvNN's are better than RvNN's in a number of different types of experiments, including computer vision, music transcription, and speech spectrum prediction.

Recent years have seen the rapid popularization and development of numerous ML and deep learning (DL) frameworks and algorithms. These frameworks provide interfaces, libraries, tools, and pre-built optimized functions. They are instrumental in deploying training routines from high-level languages to massively concurrent graphical processing unit (GPU) architectures. This allows effective and efficient ways to develop ML and DL models by speeding up the development process by eliminating the need to build the models from scratch in a plethora of applications.

Frameworks such as Chainer [43], Caffe [44], DeepLearning4j [45], Keras [46], Theano [47], TensorFlow [48], PyTorch [49], scikit-learn [50] are most commonly used in the creation of deep and machine learning models. Theano [47] and TensorFlow [48] allow high level algorithm definition in the Python programming language. Both allow automatic differentiation of training loss functions through very large networks. Compilation arising from the network's forward and backward passes into hardware-optimized concurrent dense matrix algebra kernels can be handled by them.

Although CvNN's have shown superior modeling performance for many tasks across the signal and image processing domain, however, the ML frameworks described above do not adequately support CvNN's. These frameworks only handle real-valued numbers, restricting their use to real-domain digital signal processing (DSP). In recent years, there have been some efforts with a strong focus on the implementation of modules to support CvNN's in Python; however, the progress remains slow.

A Python tool to deal with the implementation of CvNN's models using TensorFlow at the back-end was reported in [51]. A PyTorch wrapper for modeling complex-valued deep learning in PyTorch was reported in [52]. A new Python-based open-source framework known as RosenPy has been introduced in [53]. This library incorporates different CvNN's, such as CV-FFNN (complex-valued feedforward neural network) [34], SC-FFNN (split-complex feedforward neural network) [34], CV-RBFNN (complex-valued radial basis function neural network) [54], FC-RBFNN (fully complex radial basis function neural network) [55], and Deep PT-RBFNN (deep phase-transmittance radial basis function neural network). The package has been designed to reduce the level of difficulty in implementing CvNN's, enabling users to model and train these networks by simply setting of their hyperparameters [53].

4 Dataset Challenges

Cutting-edge research in finance, natural language processing, computer vision, voice recognition, and medical imaging is propelling ML to the forefront of innovation. Regrettably, recent research has mostly overlooked radio signal processing. However, there is significant potential for applying ML to the radio domain. Successful implementation of ML-based systems requires overcoming many challenges. The first and most formidable challenge is data preparation or collection. For many applications, the datasets are readily available, and for others, they do not. Radio communication is one of those domains that not only differentiates from others but also presents several challenges compared to the other domains. Some of the effects that are universally present in all over-the-air systems include the following [62]:

- Unknown propagation delays, combined with asynchronous traffic and protocol schedules, result in a random signal arrival time.
- Error in random symbol rate caused by free running sample clocks in a detached radio system.
- Propagation multipath interference, reflectors, and other effects introduce non-impulsive delay spread.
- The movement of emitters, reflectors, and/or receivers introduces Doppler offsets.
- Gaussian thermal and impulsive noise, co-channel, and adjacent channel interference.

Several applications in the current machine learning literature are not the same as radio domain data due to the challenges it poses. Radio communication signals are artificially produced by a transmitter. During the transmission, they go through a number of harsh channel effects. This results in a diversity of anomalies that are present in the received copies of the radio communications signals, which poses some unique challenges [62]. Existing datasets, such as RadioML[62], are used to train deep learning models for narrow-band radio frequency (RF) signals. RadioML datasets lack enough diversity in signal types and channel impairments to sufficiently assess the performance of ML models in a real-world scenario [63].

Datasets play an integral part in ML applications. Radio signal processing is an algorithmically matured application, but the non-availability of high-quality training datasets has slowed its advance in the ML domain. Although there are various datasets for radio signal recognition, such as RadioML 2016.10A, RadioML 2016.04C, and RadioML 2018.10A [64], these data are outdated, not being updated and maintained. Further, RadioML 2016.10A and RadioML 2016.04C are simulated datasets and don't capture real-world scenarios. These existing datasets have been extensively used in research; however, they may not be suitable for many practical applications. Most categories of these datasets are generated by simulation, which could be very different from those captured from real applications. There are different approaches to generate datasets depending on the application, such as live signal capture and a combination of hardware and software. The limitations of RadioML 2016.10A, RadioML 2016.04C, and RadioML 2018.10A [64] were addressed in TorchSig datasets [65]. TorchSig is a general-purpose, complex-valued signal processing machine learning toolkit [65]. TorchSig dataset represents 53 digital modulations useful for signal classification research. It can also generate and interface with the widebandSig53 dataset, which is a wideband extension to Sig53. Regardless of the generation method, any new dataset must fulfill the following conditions:

- It should be collected in a real-world scenario with a large number of categories.
- It should be easy to acquire and at a low expense.
- Finally, it should be easy to update in the future. Therefore, the source of the radio signal is very important for the dataset.

Producing or generating a dataset for the application at hand could be time-consuming, expensive, and require human resources as well. Given the unique characteristics of signals, e.g., in radio signal processing, labeling a large-scale and high-quality dataset could be quite cumbersome. Therefore, an automatic collection and labeling system would be desirable to capture live signals without minimal or no human intervention. Further, post-processing would be required to ensure the feature richness and balance of the collected data.

5 Are Complex-Valued Neural Network Better Than Real-Valued Neural Networks?

CvNN's are found to be superior in applications where data is complex-valued [8]. If, however, input data to CvNN's is real-valued, CvNN's did not show any performance gains over RvNN's [8, 56]. In fact, RvNN's were shown to outperform CvNN's on similar tasks [56]. When RvNN's and CvNN's of similar size were compared, the performance of CvNN's was found to be at par or slightly worse, but not better. The work reported in [57] demonstrates that CvNN's are not effective in accurately determining the horizontal lateral positions of musical audio items within binaural audio. In another example, CvNN's did not outperform the proposed auto-encoder architecture in a speech enhancement application [57]. The complex models did not outperform comparable real models, nor did they outperform a state-of-the-art baseline model in the experimental work in [57]. Methods for building and optimizing complex networks for audio applications were presented in [57]. The authors have reported that the naive implementation of Adam optimization is not suitable to handle complex random variables. Further, it was shown that the selection of input and output representations has a significant impact on the performance of a CvNN's.

The work in [51] introduced a comprehensive tool that enables the implementation of CvNN's to effectively utilize the complex-valued properties of the data. The authors have presented a straightforward comparison with real equivalent networks. A series of results have shown that CvNN's are an attractive choice over conventional RvNN's on complex-valued datasets. They have also shown that CvNN's were able to obtain better performance on non-circular complex-valued data commonly encountered in signal processing applications. It was shown that CvNN's statistically outperformed RvNN's with higher accuracy, smaller variance, and less over-fitting, regardless of the model architecture and hyper-parameters [51]. They also reported a few exceptions or outlier cases where RvNN's were found to be competitive against CvNN's. These exceptions were observed when the size of the dataset was relatively small or when the statistical coherence between the real and imaginary parts, i.e., the correlation coefficient $|\rho|$, was close to zero. In such cases, it was almost impossible to distinguish between feature vectors. For these exceptions, CvNN's and RvNN's were found not to be useful [51]. The findings in [51] do not provide sufficient evidence to support the claim that CvNN's outperform RvNN's. Additional research is needed to investigate the effects of learning rates, feature vector size, and multi-class datasets on the performance of these models.

The work in [60] presents a comparison between RvNN's and an "equivalent" CvNN's for wireless data classification. In order to guarantee a fair comparison, the number of parameters for both types of NN was equal. In their study, they also explored the effect of changing the set of inputs to the NN by using multiple configurations such as in-phase and quadrature (I & Q) components and polar representation as data inputs to NN in their study. The investigation focuses on three configurations for the I and Q samples: (i) using only the input I (ii) using only the input Q, and (iii) using both I and Q (referred to as IQ) as the input. For the polar representation of IQ

samples (i) use only the input magnitude, (2) use only the phase as an input, and (iii) both magnitude & phase were used as input. For configurations that only required a single input, the other input was set to zero [60].

In general, CvNN's are found to be more accurate for wireless device classification than RvNN's [60]. The CvNN's consistently outperformed the RvNN's while having a same number of parameters. The primary advantage of CvNN's compared to RvNN's is their ability to utilize the combined information from both the I & Q components of the signals due to the cross-terms generated by the complex number multiplication and convolution operations, according to (9).

A performance comparison of RvNN's and CvNN's for digital self-interference cancelation was reported in [61]. They concluded that CvNN's with certain non-holomorphic activation functions performed significantly better than those with holomorphic activation functions. Further, their proposed real-input CvNN (RIn-CvNN) was found to perform equally well as CvNN's with non-holomorphic activation functions. Furthermore, it was concluded that RvNN's required more neurons than RIn-CvNN and CvNN to achieve the same performance.

Contrary to some of the sampled work above, the question of the appropriateness of CvNN's for speech enhancement was addressed in [59]. From the presented experiments, it was concluded that CvNN's do not perform dramatically better than RvNN's on intrinsically complex tasks. The number of multiplication operations was higher than RvNN's. The authors concluded that split-RvNN's using the real and imaginary parts of features as inputs were preferred for complex regression problems.

A comparative study of the CvNN's and RvNN's trained with gradient descent and random search algorithms was the subject of investigation in [58]. According to their findings, there was no difference in the quality of results between CvNN's and RvNN's for the application in their work.

6 Conclusions

This work presents a brief overview of CvNN's. Compared to RvNN's, research on CvNN's has progressed at a slower pace, and only a few applications have utilized them. The argument for using CvNN's in applications where data is complex by design or occurs naturally has merit. Some applications, such as radio signal processing and wireless communications, are not able to leverage the benefits of existing open-source NN libraries in their existing development frameworks. This can be attributed to a lack of factors such as the availability of appropriate datasets, appropriate complex-valued activation functions, differentiability of complex-valued functions, well-defined loss functions, optimization algorithms, weight initialization strategies, and numerical issues during the training process. Furthermore, most of the publicly available software development frameworks are unable to directly integrate complex-valued data into applications or libraries developed for real-valued net-

works. In the absence of complex-valued deep learning software implementations, the appropriateness of deploying real-valued solutions for handling complex-valued data remains an open research problem.

The benefits of CvNN's in signal, image processing, and wireless communications cannot be ignored in light of research activities for the 6G wireless communication standard. Several industry players, such as Rohde & Schwarz and NVIDIA are partnering and transitioning from simulations to implementing and testing AI models for 6G applications.¹ This may accelerate the adoption of CvNN's for signal processing applications for 6G systems. It can be foreseen that a neural receiver may replace signal processing blocks in the physical layer of a wireless communications system with trained machine learning models. These efforts will also expedite the generation of datasets, which are an absolute prerequisite for training NN models.

Acknowledgements We thank Dr. Rolf Esslinger for useful discussion on the topic and for reviewing the manuscript.

References

1. Zhang H, Gu M, Jiang XD et al. (2021) An optical neural chip for implementing complex-valued neural networks. *Nature Commun* 12(1):1–11. <https://doi.org/10.1038/s41467-020-20719-7>
2. Oppenheim AV, Lim JS (1981) The importance of phase in signals. In: *Proceedings of the IEEE* 69(5):529–541
3. Hirose A, Yoshida S (2011) Comparison of complex-and real-valued feedforward neural networks in their generalization ability. In: Lu BL, Zhang L, Kwok J (eds) *Neural information processing*, pp 526–531
4. Lee C, Hasegawa H, Gao S (2022) Complex-valued neural networks: a comprehensive survey. *IEEE/CAA J Automatica Sinica* 9(8):1406–1426. <https://doi.org/10.1109/JAS.2022.105743>
5. Hirose A (1992) Proposal of fully complex-valued neural networks. In: *International joint conference on neural networks*. Baltimore, USA, pp 152–157. <https://doi.org/10.1109/IJCNN.1992.227274>
6. Hirose A (1994) Applications of complex-valued neural networks to coherent optical computing using phase-sensitive detection scheme. *Inform Sci Appl* 2(2):103–117
7. Hirose A (2006) In: *Complex-valued neural networks*. Springer
8. Hirose A (2009) Complex-valued neural networks: the merits and their origins. In: *International joint conference on neural networks*, Atlanta, USA, pp 1237–1244. <https://doi.org/10.1109/IJCNN.2009.5178754>
9. Clarke TL (1990) Generalization of neural networks to the complex plane. In: *International joint conference on neural networks*. San Diego, USA, pp 435–440. <https://doi.org/10.1109/IJCNN.1990.137751>
10. Hirose A (1992) Dynamics of fully complex-valued neural networks. *Electron Lett* 28(16):1492–1494
11. Leung H, Haykin S (1991) The complex backpropagation algorithm. *IEEE Trans Signal Process* 39(9):2101–2104

¹ https://www.rohde-schwarz.com/us/about/news-press/all-news/towards-6g-rohde-schwarz-showcases-ai-ml-based-neural-receiver-with-nvidia-at-mwc-barcelona-press-release-detailpage_229356-1330882.html.

12. Benvenuto N, Piazza F (1992) On the complex backpropagation algorithm. *IEEE Trans Signal Process* 40(4):967–969
13. Georgiou G, Koutsougeras C (1992) Complex domain backpropagation. *IEEE Trans Circuits and Syst II: Analog and Digital Signal Process* 39(5):330–334
14. Hirose A (1992) Continuous complex-valued back-propagation learning. *Electron Lett* 28(20):1854–1855
15. Mandic DP, Goh VSL (2009) Complex valued nonlinear adaptive filters—noncircularity, widely linear and neural models. Wiley
16. Aizenberg I (2011) In: Complex-valued neural networks with multi-valued neurons. Springer
17. Nitta T (1992) On the critical points of the complex-valued neural network. In: International conference on neural information processing, Singapore, pp 1099–1103. <https://doi.org/10.1109/ICONIP.2002.1202792>
18. Hirose A, Yoshida S (2012) Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. *IEEE Trans Neural Netw Learn Syst* 23(4):541–551
19. Arjovsky M, Shah A, Bengio Y (2016) Unitary evolution recurrent neural networks. In: International conference on machine learning. New York, USA, pp 1120–1128
20. Wisdom S, Powers T, Hershey J, Le Roux J, Atlas L (2016) Full-capacity unitary recurrent neural networks. In: International conference on neural information processing systems. Barcelona Spain, pp 4887–4895
21. Danihelka I, Wayne G, Uria B, Kalchbrenner N, Graves A (2016) Associative long short-term memory. In: International conference on machine learning. New York, USA, pp 1986–1994
22. Hirose A (2013) Complex-valued neural networks—advances and applications. In: Hirose A (ed) Wiley & Sons
23. Nitta T (1997) An extension of the back-propagation algorithm to complex numbers. *Neural Networks* 10(8):1391–1415
24. Amin FM (2012) Complex-valued neural networks: learning algorithms and applications. In: Doctoral Thesis, University of Fukui, Japan
25. Bassey J, Li X, Qian L (2021) A survey of complex-valued neural networks. In: arXiv preprint (2021). [arXiv:2101.12249v1](https://arxiv.org/abs/2101.12249v1)
26. Arena P, Fortuna L, Re R, Xibilia MG (1995) Multilayer perceptrons to approximate complex valued functions. *Int J Neural Syst* 6(4):435–446
27. Amin MF, Savitha R, Amin MI, Murase K (2011) Complex-valued functional link network design by orthogonal least squares method for function approximation problems. In: International joint conference on neural networks. San Jose, USA, pp 1489–1496. <https://doi.org/10.1109/IJCNN.2011.6033400>
28. Yang WH, Chan KK, Chang PR (1994) Complex-valued neural-network for direction of-arrival estimation. *Electron Lett* 30(7):574–575
29. Chang A-C, Jen C-W, Su I-J (2007) Robust adaptive array beamforming based on independent component analysis with regularized constraints. *IEICE Trans Commun* E90(7):1791–1800. <https://doi.org/10.1049/el:19940400>
30. Dedmari MA, Conjeti S, Estrada S, Ehses P, Stöcker T, Reuter M (2018) Complex fully convolutional neural networks for MR image reconstruction. In: Knoll F, Maier A, Rueckert D (eds) Machine learning for medical image reconstruction, Lecture notes in computer science, vol 11074. Springer, pp 30–38
31. Zhang Z, Wang H, Xu F, Jin Y-Q (2017) Complex-valued convolutional neural network and its application in polarimetric SAR image classification. *IEEE Trans Geosci Remote Sens* 55(12):7177–7188
32. Li M-B, Huang G-B, Saratchandran P, Sundararajan N (2005) Fully complex extreme learning machine. *Neurocomputing* 68:306–314. <https://doi.org/10.1016/j.neucom.2005.03.002>
33. Savitha R, Suresh S, Sundararajan N, Saratchandran P (2009) A new learning algorithm with logarithmic performance index for complex-valued neural networks. *Neurocomputing* 72(16–18):3771–3781

34. Scardapane S, Van Vaerenbergh S, Hussain A, Uncini A (2020) Complex-valued neural networks with nonparametric activation functions. *IEEE Trans Emerg Topics in Comput Intell* 4(2):140–150
35. Ceylan M, Yaşar H (2016) A novel approach for automatic blood vessel extraction in retinal images: complex ripplelet-I transform and complex valued artificial neural network. *Turkish J Electri Eng Comput Sci* 24(4):3212–3227
36. Pu Y-F, Xie X, Cao J, Chen H, Zhang K, Wang J (2022) An input weights dependent complex-valued learning algorithm based on wirtinger calculus. *IEEE Trans Syst Man and Cybernet: Syst* 52(5):2920–2932
37. Yang S-S, Ho C-L, Siu S (2007) Sensitivity analysis of the split-complex valued multilayer perceptron due to the errors of the i.i.d. inputs and weights. *IEEE Trans Neural Netw* 18(5):1280–1293
38. Sergey I, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the international conference on machine learning*. Lille, France, pp 448–456
39. Cogswell M, Ahmed F, Girshick R, Zitnick L, Batra D (2015) Reducing overfitting in deep networks by decorrelating representations. In: *International conference on learning representations*. San Diego, USA, pp 1–12
40. Trabelsi C, Bilaniuk O, Zhang Y, Serdyuk D, Subramanian S, Santos JF, Mehri S, Rostamzadeh N, Bengio Y, Pal CJ (2018) Deep complex networks. In: *International conference on learning representations*. Vancouver, Canada, pp 1–19
41. Xiao C, Yang S, Feng Z (2023) Complex-valued depthwise separable convolutional neural network for automatic modulation classification. *IEEE Trans Instrumentatio and Measurment* 72:1–10. <https://doi.org/10.1109/TIM.2023.3298657>
42. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the international conference on artificial intelligence and statistics*. vol 9. Sardinia, Italy, pp 249–256
43. <https://chainer.org/>
44. <https://caffe.berkeleyvision.org/>
45. <https://deeplearning4j.konduit.ai/>
46. <https://keras.io>
47. <https://pypi.org/project/Theano/>
48. <https://www.tensorflow.org>
49. <https://pytorch.org>
50. <https://scikit-learn.org>
51. Barrachina JA, Ren C, Morisseau C, Vieillard G, Ovarlez J-P (2021) Complex-valued versus real-valued neural networks for classification perspectives: an example on non-circular data. In: *IEEE International conference on acoustics, speech and signal processing*. Toronto, Canada, pp 2990–2994. <https://doi.org/10.1109/ICASSP39728.2021.9413814>
52. Smith JW (2023) Complex-valued neural networks for data-driven signal processing and signal understanding. In: *arXiv preprint*. [arXiv:2309.07948](https://arxiv.org/abs/2309.07948)
53. Cruz AA, Mayer KS, Arantes DS (2022) RosenPy: an open source Python framework for complex-valued neural networks. Available at SSRN: <https://ssrn.com/abstract=4252610>
54. Enriconi MP, De Castro FFC, Müller, De Casrto MCF (2020) Phase transmittance RBF neural network beamforming for static and dynamic channels. *IEEE Antennas and Wireless Propagation Lett* 19(2):243–247
55. Savitha R, Suresh S, Sundararajan N (2012) Metacognitive learning in a fully complex-valued radial basis function neural network. *Neural Comput* 24(5):1297–1328
56. Mönning N, Manandhar S (2018) Evaluation of complex-valued neural networks on real-valued classification tasks. In: *arXiv preprint* (2018). [arXiv: 1811.12351](https://arxiv.org/abs/1811.12351)
57. Sarroff AM (2018) Complex neural networks for audio. In: *Doctoral Thesis*, Dartmouth College, USA

58. Zimmermann HG, Minin A, Kusherbaeva V (2011) Comparison of the complex valued and real valued neural networks trained with gradient descent and random search algorithms. In: European symposium on artificial neural networks. Computational Intelligence and Machine Learning. Bruges, Belgium, pp 213–218
59. Drude L, Raj B, Haeb-Umbach R (2016) On the appropriateness of complex-valued neural networks for speech enhancement. In: INTERSPEECH, San Francisco, USA
60. Chen J, Wong W-K, Hamdaoui B, Elmaghub A, Sivanesan K, Dorrance R, Yang LL (2022) An analysis of complex-valued CNNs for RF data-driven wireless device classification. In: IEEE International conference on communications. Seoul, Republic of Korea, pp 4318–4323
61. Wang Q, He F, Meng J (2019) Performance comparison of real and complex valued neural networks for digital self-interference cancellation. In: International conference on communication technology. Xi'an, China, pp 1193–1199
62. O'Shea TJ, West N (2016) Radio machine learning dataset with GNU radio. In: Proceedings of the GNU radio conference, Boulder, USA
63. Boegner L, Gulati M, Vanhoy G, Vallance P, Comar B, Kokalj-Filipovic S, Lennon C, Miller RD (2018) Large scale radio frequency signal classification. In: arXiv preprint (2022). [arXiv:2207.09918](https://arxiv.org/abs/2207.09918)
64. <https://www.deepsig.ai/datasets/>
65. <https://torchsig.com/>

Chapter 17

Self-adaptive Big Bang-Big Crunch Algorithm for Engineering Designs



Jéssica Cristiane Andrade, Claudemir Mota da Cruz,
Fran Sérgio Lobato, Gustavo Barbosa Libotte,
and Gustavo Mendes Platt

1 Introduction

In the last decades, population-based algorithms have been considered important tools in engineering systems design. During the evolutionary process, the input parameters were usually kept constant. This characteristic simplifies its computational implementation, but does not consider (for example) that, in these algorithms, the population size should vary over time. Thus, it might be interesting to increase this parameter (to a maximum value) during the first generations in order to allow the individuals to explore the design space (increasing exploration skills). On the other hand, at the end of the generations, the population can become homogeneous. In this case, unnecessary evaluations of the objective function are performed, since the individuals with similar characteristics are evaluated [1]. Intuitively, if the population size can be reduced throughout the evolutionary process without losing the capacity to explore the design space, the computational cost can be reduced when compared to the algorithm with a constant population.

J. C. Andrade (✉) · F. S. Lobato
Federal University of Uberlândia, Uberlândia, Brazil
e-mail: jessica.cristiane@ufu.br

F. S. Lobato
e-mail: fslobato@ufu.br

C. M. da Cruz
State University of Santa Cruz, Ilhéus, Brazil
e-mail: cmcruz@uesc.br

G. B. Libotte
Rio de Janeiro State University, Nova Friburgo, Brazil
e-mail: gustavolibotte@iprj.uerj.br

G. M. Platt
Federal University of Rio Grande, Santo Antônio da Patrulha, Brazil
e-mail: gmplatt@furg.br

The use of parameter control was not a common subject of study at the beginning of evolutionary computing, but over the years and with the increasing complexity of problems, its use became necessary to ensure the efficiency and robustness of the algorithms. In addition, the choice of parameter values in any population-based algorithm can significantly influence its performance and has been the subject of several studies [2–4]. This choice of values can be done manually, through previous experience acquired by the researcher or simply by intuition (parameter tuning), or even automatically, through parameter control methods.

The optimal value of the control parameters can change throughout the algorithm execution, i.e., at the beginning of the search process, the optimal values of the control parameters can be different from the optimal values at the end of the execution. The main advantages of this strategy are presented as follows [2]:

- Allows the use of appropriate parameters at different stages of the search process, i.e., increasing exploration at the beginning, and refining the solution at the end;
- Allows the adjustment of parameters when the fitness scenario changes in dynamic problems;
- Allows the algorithm to collect information about the fitness scenario and use this information to adjust parameters at later stages;
- Avoids manual setting of parameters by the user.

As mentioned earlier, population-based algorithms can be divided into two categories with respect to their control parameters [3]: constant and non-constant (variable). In the first category, the control parameters are fixed along the optimization process and are defined after some preliminary runs or, as pointed previously, by the experience of the user. On the other hand, in the second set of algorithms, the control parameters are updated during the execution of the code. Usually, algorithms with variable control parameters are more efficient than those with constant parameters, as they are able to adapt to the characteristics of the problem and the optimization environment [3].

With regard to the algorithms with variable control parameters, these can be classified into three types [3]: (i) deterministic, (ii) adaptive, and (iii) self-adaptive. A deterministic population-based algorithm follows some pre-established rules to alter the control parameters. An adaptive population-based algorithm improves its performance during execution, automatically adjusting its parameters independently of the user intervention. For this, it uses information obtained during the process to adapt the parameters. Finally, a self-adaptive population-based algorithm is a variation of the adaptive strategy, but, in this case, the parameters themselves are encoded. In summary, we can say that the main difference between these three types of evolutionary algorithms lies in the way they deal with randomness and parameter tuning [5].

Parameter control strategies in population-based algorithms can be grouped mainly as control methods for [2]: population, selection, variation (mutation and crossover), and fitness function.

In this context, Lobato and co-workers [1] proposed a self-adaptive Differential Evolution (DE) strategy based on the concept of population diversity. A variant of

the DE algorithm that uses the Particle Swarm Optimization mutation strategy at some stages of evolution to improve performance and avoid premature convergence was proposed by Dixit and co-workers [6]. Zhou and co-workers [7] proposed a self-adaptive DE algorithm using a self-adaptation strategy for both the mutation operator and the crossover and amplification factor, always based on their historical performance over generations. Liu and Lampinen [8] proposed the use of fuzzy logic controllers to adapt the parameters of the mutation operator and the crossover operator in the DE algorithm. Chen and co-workers [9] developed a self-adaptive genetic algorithm that uses reinforcement learning to dynamically adjust mutation and crossover rates.

Another factor that strongly influences the performance of an evolutionary algorithm is the population size, as a small population can lead to premature convergence, while a very large population can reduce the efficiency of the algorithm [2, 4, 10, 11]. With respect to the adjustment of the population size, the works can be classified into 4 groups [2].

- Theoretical studies on population size and the advantages of varying the population size;
- Approaches that aim to drop the population size parameter by introducing a new operator or concept;
- Strategies to determine an optimal population size during its run;
- Operators to control the population size in real time (on-the-fly).

Zhang [10] proposes an adaptive guided DE algorithm considering a new mutation scheme and a nonlinear population size reduction strategy based on hyperbolic tangent function. At the beginning, convergence is slow, and the population size is fixed at the maximum value; in the middle of the process, the number of individuals decreases linearly; finally, at the end of the algorithmic procedure, the curve induces the convergence of the population size to the established minimum. This scheme allows the exploration of new regions of the search space at the beginning, avoiding premature convergence, and fewer evaluations at the end, when the algorithm is already possibly converging. In the work carried out by Lacerda and co-workers [11], a parameterless controller was proposed to adjust the population size in evolutionary and swarm-based algorithms. The controller increases or decreases the speed of population change as fitness improves and adds or removes individuals using a selection mechanism. An adaptive multi-population strategy considering the Non-dominated Sorting Genetic Algorithm (NSGA-II) is proposed by Zhao and co-workers [12]. In this algorithm, the population is divided into sub-populations and the size of each sub-population is controlled according to the quality of individuals, in addition to applying different crossover operators in each sub-population.

This work presents the Big Bang-Big Crunch (BB-BC) algorithm with variable control parameters to solve engineering system design. Unlike other population-based algorithms that use mutation, crossover, selection, or the amplification factor as variable control parameters, BB-BC uses population size as a variable control parameter, thus fitting into the category of self-adaptive algorithms.

In order to reduce the number of objective function evaluations during an evolutionary process, the Self-Adaptive Big Bang-Big Crunch (SA-BB-BC) algorithm is proposed. The proposed methodology consists of two strategies to update the population size and a parameter for controlling the influence of particle positions (center of mass) and the best particle. The quality of obtained results is evaluated considering classical engineering systems design. This work is organized as follows. Section 2 presents a review dedicated to the Big Bang-Big Crunch (BB-BC) algorithm. Section 3 contains the description of the proposed methodology. Section 4 presents the results obtained and discusses them. The conclusions are presented in Sect. 5.

2 Big Bang-Big Crunch Algorithm

The Big Bang-Big Crunch (BB-BC) algorithm, proposed by Erol and Eksin [13], is based on the Big Bang-Big Crunch Theory for the creation of the Universe. In the Big Bang phase, particles are randomly scattered, decreasing their energy content. On the other hand, in the Big Crunch phase, the particles collapse as a consequence of the gravitational action.

From an algorithmic point of view, the candidates in the BB-BC strategy are randomly generated considering the limits defined for the vector of design variables (this step represents the initial Big Bang). As mentioned by Kaveh and Bakhshpoori [14], the Big Bang and the Big Crunch phases are applied sequentially, except for the first population of individuals (which are generated randomly). In the Big Crunch phase, a contraction operator is determined, as a center of mass of the particles in the system (for instance). Then, in a new Big Bang phase, the particles are now dispersed from this new center of mass.

The flowchart of BB-BC is presented in Fig. 1 and the main steps are described as follows [13, 15]:

- Step 1—Define the BB-BC parameters: NP (population size), max_{gen} (maximum number of generations) and the parameter β . Inform the characteristics of optimization problem (objective function, number of design variables, lower (x^l) and upper (x^u) bound for each design variable, constraints, parameters).
- Step 2—Generate (randomly) an initial population, considering the population size and the limits for each design variable.
- Step 3—Evaluate each potential candidate in terms of the objective function and determine the best particle (P_{best}).
- Step 4—Determine the center of mass (x^c) according to Eq. (1):

$$x^c = \frac{\sum_{i=1}^{NP} \frac{1}{f_i} x^i}{\sum_{j=1}^{NP} \frac{1}{f_j}} \quad (1)$$

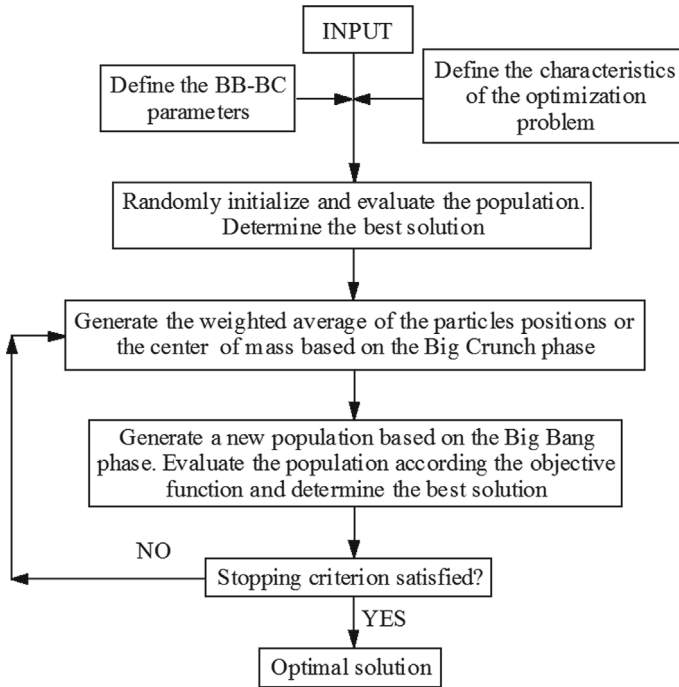


Fig. 1 Flowchart of the BB-BC algorithm

where x^i is a potential candidate and f^i is an objective function evaluated in this point.

- Step 5—Generate new candidates around the center of mass. For this purpose, the following relation is considered:

$$x^{new} = \beta x^c + (1 - \beta) P_{best} + \frac{r(x^u - x^l)}{k} \quad (2)$$

where r is a random number with standard normal distribution, k is the current iteration, and β is a parameter for controlling the influence of the weighted average of the particles positions (or the center of mass (x^c)) and the best particle (P_{best}). Evaluate each generated candidate and determine the best particle (P_{best}).

- Step 6—If the stopping criterion was not satisfied, return to Step 4.

3 Methodology

As previously discussed, in the BB-BC algorithm it is necessary to define NP (population size), max_{gen} (maximum number of generations) and the control parameter β . In practice, these values are defined according to each case study. In addition, all

parameters are constant during the evolutionary process in the canonical form of the algorithm. As an alternative, in this section, two strategies to update the population size and the parameter β are presented. It is important to mention that the max_{gen} is kept fixed along the search process (stopping criterion).

3.1 Population Size Update

Traditionally, the population size is kept fixed throughout generations in the population-based optimization algorithms. This approach simplifies these algorithms, but it represents an assumption that does not follow the biological evolution, i.e., the number of individuals change due to birth or death of individuals. As mentioned by Vellev [16] and Cavalini Jr and co-workers [17], may be interesting to consider a population with as many individuals as possible to ensure phenotype diversity. However, along the evolutionary process, the population can be contracted when the individuals present similar characteristics, i.e., close objective function values.

Based on this principle, in this work, the population size in the BB-BC algorithm is updated according to the convergence rate concept (λ). This is defined as [17].

$$\lambda = \frac{f_{average}}{f_{worst}} \quad (3)$$

where $f_{average}$ and f_{worst} are the average and worst values of the objective function, respectively.

In practice, this concept represents a measure of population homogeneity during the search process. Thus, if λ is low, the objective function value of the worst individual of the population differs significantly from the average value (the BB-BC algorithm has not yet converged). In contrast, if λ is close to one, the population is homogeneous (the BB-BC algorithm converged). As a consequence, this parameter can be used to updating the population size as follows:

$$NP = \text{round}(NP_{min} + (1 - \lambda)(NP_{max} - NP_{min})) \quad (4)$$

where NP_{min} and NP_{max} are the minimum and maximum number of individuals in the population, respectively, and the operator $\text{round}()$ indicates the rounding to the nearest integer. In this case, if λ (see Eq. (4)) is equal to zero, the population size is maximum. Conversely, if λ is equal to one, the population size is minimum. On the other hand, any value between 0 and 1 implies that the population size will be a value between NP_{min} and NP_{max} .

It is important to mention that if the evolutionary process does not converge, the population does not become homogeneous, and, thus, the parameter NP assumes a value close to maximum. In this case, the proposed approach behaves like traditional population-based optimization methods, where the population size is constant. In

addition, it is possible that during the evolutionary process the value of NP increases or decreases. This will depend on the value of the computed convergence rate.

In general, during the updating the population size, three possibilities should be considered:

- If the population size does not change, no action is realized;
- If the population size in the next generation decreases in comparison to the current generation, the individuals that will be part of the new population (with reduced size) are selected from their ordering in relation to value of the objective function, thus forming a new population;
- If the population size in the next generation increases with respect to the current generation, these new individuals are generated randomly, considering the bounds for each design variable. These new individuals are added to the current population.

3.2 Updating of the β Parameter

In order to update the β parameter, a chaotic search pattern based on a one-dimensional map is considered. The interest in use of chaotic search in optimization context is due to the role of randomness introduced by these models [18]. As a consequence, the inserted randomness by these models can help the algorithm to avoid local optima [19].

In the literature, various chaotic search patterns can be found [18]. In this contribution, the traditional Logistic map is considered. This polynomial map is a simple nonlinear dynamical equation defined as [20]:

$$y_{n+1} = \theta y_n(1 - y_n), \quad 0 < \theta \leq 4 \quad (5)$$

where y_{n+1} is the dependent variable, θ is the bifurcation parameter, and n is a counter.

To update the β parameter, the following relation based on Eq. (5) is considered:

$$\beta_{n+1} = \beta_{min} + (\beta_{max} - \beta_{min})\alpha_n \quad (6)$$

$$\alpha_n = \theta\beta_n(1 - \beta_n) \quad (7)$$

where β_{min} and β_{max} represent lower and upper bounds for the parameter β , and α is an auxiliary parameter responsible for introducing chaotic dynamics during the process. The first value for β_n can be chosen from the value suggested by the literature.

4 Applications

In order to evaluate the proposed methodology, some practical points regarding the application of the procedure should be emphasized:

- In order to compare the obtained results by using the proposed methodology, the following optimization strategies are considered: Constraint Correction Method (CCM) [21]; Multiplier Method based on Fletcher's algorithm (MMF) [22]; Genetic Algorithms (GA¹) [23] ($NP=60$ and $max_{gen}=25$ (first sub-population) and $NP=30$ and $max_{gen}=20$ (second sub-population)); Genetic Algorithms (GA²) [24] ($NP=200$ and $max_{gen}=400$); Genetic Algorithms (GA³) [25] (NP and max_{gen} not reported. Stopping criteria: number of function evaluations defined by user); Genetic Algorithms (GA⁴) [26] ($NP=100$ and $max_{gen}=45$); Particle Swarm Optimization (PSO¹) [27] ($NP=20$ and $max_{gen}=55$); Particle Swarm Optimization (PSO²) [28] ($NP=70$ and $max_{gen}=20$); Evolutionary Process Strategy (EPS¹) [29] ($NP_1=80$ and $max_{gen1}=1000$); Evolutionary Process Strategy (EPS²) [30] ($NP_1=100$ and $max_{gen1}=420$); Simulation of Social Behavior (SSB) of species [31] ($NP_1=40$ and $max_{gen1}=1000$); Fractional Nonlinear Programming (FNP) [32]; Differential Evolution (DE) [33] ($NP_1=50$ and $max_{gen1}=480$); Evolutionary Strategies (ES) [34] ($NP_1=50$ and $max_{gen1}=500$); Hybrid Evolutionary Algorithm (HEA) [35] ($NP_1=50$ and $max_{gen1}=480$); Ant Colony Algorithm (ACA) [36] (Not reported); Water Cycle Algorithm (WCA) [37] ($NP=50$ and $max_{gen}=235$ (first case) and $NP=50$ and $max_{gen}=550$ (second case)); Branch-and-Bound Method (BBM) [38]; Lagrange Multiplier Method (LMM) [39]; Algorithm based on the Socio-Behavioral Concept of Society and Civilization (SC) [40] ($NP=100$ and $max_{gen}=125$ (second case) and $NP=100$ and $max_{gen}=190$ (third case)); Cuckoo Search (CS) [41] ($NP=25$ and $max_{gen}=600$ (second case) and $NP=25$ and $max_{gen}=800$ (third case)); Geometric Programming (GP) [42]; Vibrating Particles System (VPS) algorithm [43] ($NP=20$ and $max_{gen}=200$). For comparison purposes, best, average, worst solutions and standard deviation values, inequality constraints evaluated at optimal solution and number of average objective function evaluations are considered.
- The parameters used by BB-BC algorithm are the following [13]: $NP=20$, $\beta=0.1$ and $max_{gen}=200$;
- In order to evaluate the influence of the parameter β , the BB-BC- r algorithm is considered. In this version of the classical BB-BC algorithm, the parameter β is updated considering the following expression:

$$\beta = \beta_{min} + (\beta_{max} - \beta_{min})\text{rand} \quad (8)$$

where β_{min} (0.01) and β_{max} (0.1) represent lower and upper bound for this parameter, and rand is a random number with uniform distribution between (0,1). In addition, NP is equal to 20 and max_{gen} is equal to 200;

- The proposed methodology is called Self-Adaptive Big Bang-Big Crunch (SA-BB-BC) algorithm. The parameters used by SA-BB-BC are the following: NP

defined by Eq. (4) (minimal and maximum numbers of individuals in the population equals to 5 and 20, respectively), β is updated considering the Eq. (6) ($\theta=3.57$, $\beta_{min}=0.01$ and $\beta_{max}=0.1$), and $max_{gen}=200$;

- The maximum number of generations was used as a stopping criterion;
- The Static Penalty Method [44] was employed to deal with the constraints, with a penalty parameter equal to 10^8 in all applications;
- All case studies were run 20 times to obtain the best, average, worst and standard deviation values.
- To evaluate the performance of the proposed methodology, the (decrease or increase) rate (Γ) (%) is defined as:

$$\Gamma = \begin{cases} +100\left(1 - \frac{n_{eval,i}}{n_{eval}}\right), & \text{increase;} \\ -100\left(1 - \frac{n_{eval}}{n_{eval,i}}\right), & \text{decrease} \end{cases} \quad (9)$$

where $n_{eval,i}$ and n_{eval} are the number of objective function evaluations for a i -th strategy (except SA-BB-BC) and SA-BB-BC algorithm, respectively. Signs + and - represent an increase and decrease in the number of objective function evaluations in relation to SA-BB-BC algorithm.

4.1 Tension/Compression Spring Design Problem

The first application considers the minimization of the weight of a tension/compression spring, as presented in Fig. 2 [21]. For this purpose, the design variables are the wire diameter (x_1), the mean coil diameter (x_2), and the number of active coils (x_3).

The objective function (volume) is given by:

$$\min f(x) = (x_3 + 2) x_2 x_1^2 \quad (10)$$

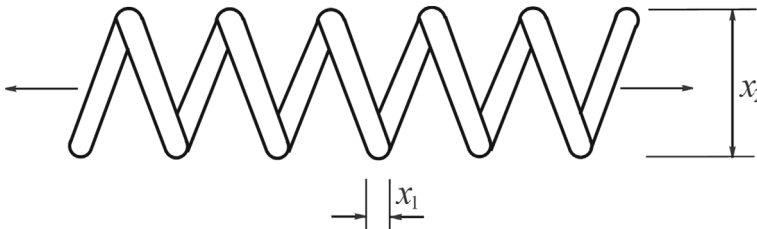


Fig. 2 Tension/compression spring design problem

This engineering design is subjected to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter, and on design variables. These constraints are given by:

$$g_1(x) \equiv 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0 \quad (11a)$$

$$g_2(x) \equiv \frac{4x_2^2 - x_1 x_2}{12566 (x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0 \quad (11b)$$

$$g_3(x) \equiv 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0 \quad (11c)$$

$$g_4(x) \equiv \frac{(x_1 + x_2)}{1.5} - 1 \leq 0 \quad (11d)$$

$$0.05 \leq x_1 \leq 2.00 \quad (11e)$$

$$0.25 \leq x_2 \leq 1.30 \quad (11f)$$

$$2 \leq x_3 \leq 15 \quad (11g)$$

Table 1 presents the obtained results for the tension/compression spring design problem considering the SA-BB-BC algorithm, as well as other optimization strategies. It is possible to observe that the BB-BC, BB-BC-*r* and SA-BB-BC algorithms were able to find a good estimate for the best solution reported in the literature (0.0127 in³).

Table 2 contains the values of the inequality constraints evaluated at optimal solutions, considering the different strategies for the tension/compression spring design problem. It can be observed that both BB-BC, BB-BC-*r*, and SA-BB-BC algorithms were able to satisfy all constraints.

Table 3 presents best, average, and worst solutions, standard deviations (in terms of the objective function), number of average objective function evaluations and percentage decrease/increase rates Γ for the different optimization strategies under consideration. We can observe that both BB-BC, BB-BC-*r* and SA-BB-BC algorithms always converged (as well as the other reported optimization strategies) to a good estimate of the optimal solution, given the order of magnitude of the standard deviation and the worst and average values. In relation to the number of objective function evaluations (n_{eval}) required by SA-BB-BC algorithm, an increase of, approximately, 22.51% and 11.17% was observed considering the PSO¹ and PSO² strategies, respectively. On the other hand, a reduction in the range of 23.93% to 97.93% is found, when considering the comparison with other algorithms. This means that the proposed methodology was able to obtain a good approximation to the best solution reported by literature, however, with a lower number of evaluations in relation to the majority of the optimization strategies (GA¹, GA², SSB, ES, DE, EPS¹, EPS², WCA, VPS, BB-BC, BB-BC-*r*) considered.

Figure 3 presents the evolution of the parameters λ , β and NP as functions of the number of generations (in typical runs of the algorithm). In Fig. 3a (BB-BC-*r*) and b (SA-BB-BC) it is clear as the parameter β varies between the limits defined. In the

Table 1 Optimal solution (f^{opt}) considering different optimization strategies for the tension/compression string design problem

Algorithm	x_1 (in)	x_2 (in)	x_3 (—)	f^{opt} (in ³)
CCM	0.0534	0.3992	9.1854	0.0127
MMF	0.0500	0.3159	14.2500	0.0128
GA ¹	0.0515	0.3517	11.6322	0.0127
PSO ¹	0.0504	0.3215	13.9799	0.0131
GA ²	0.0520	0.3640	10.8905	0.0127
SSB	0.0522	0.3682	10.6484	0.0127
FNP	0.0517	0.3567	11.2890	0.0127
PSO ²	0.0517	0.3576	11.2445	0.0127
ES	0.0516	0.3554	11.3979	0.0127
DE	0.0517	0.3567	11.2890	0.0127
HEA	0.0517	0.3567	11.2883	0.0127
EPS ¹	0.0516	0.3555	11.3332	0.0126
ACA	0.0519	0.3615	11.0000	0.0126
EPS ²	0.0517	0.3567	11.2893	0.0127
WCA	0.0517	0.3565	11.3004	0.0127
VPS	0.0517	0.3577	11.2310	0.0127
BB-BC	0.0500	0.3173	14.0363	0.0127
BB-BC- r	0.0507	0.3355	12.7058	0.0127
SA-BB-BC	0.0506	0.3306	13.1687	0.0127

Table 2 Inequality constraints evaluated at optimal solution considering different optimization strategies for the tension/compression string design problem

Algorithm	$g_1(x)$	$g_2(x)$	$g_3(x)$	$g_4(x)$
CCM	−0.0012	0.0000	−4.1238	−0.6983
MMF	−0.0013	−0.0038	−3.9383	−0.7561
GA ¹	−0.0033	−0.0001	−4.0263	−0.7312
GA ²	−0.0013	0.0000	−4.0613	−0.7227
PSO ²	−0.0008	0.0000	−4.0513	−0.7271
DE	0.0000	0.0000	−4.0538	−0.7277
HEA	0.0000	0.0000	−4.0538	−0.7277
EPS ¹	0.0010	0.0010	−4.0619	−0.7286
ACA	−0.0004	0.0012	−4.0674	−0.7244
WCA	−0.0001	0.0001	−4.0534	−0.7279
VPS	0.0000	0.0000	−4.0555	−0.7270
BB-BC	−0.0003	−0.0000	−3.9663	−0.7550
BB-BC- r	−0.0042	−0.0001	−3.9875	−0.7424
SA-BB-BC	−0.0116	−0.0011	−3.9355	−0.7458

Table 3 Best (f^{opt}), average (f_m^{opt}) and worst (f^w) results, standard deviation (f^{std}), number of average objective function evaluations (n_{eval}) and rate Γ considering different optimization strategies for the tension/compression string design problem

Algorithm	f^{opt} (in ³)	f^w (in ³)	f_m^{opt} (in ³)	f^{std} (in ³)	n_{eval}	Γ (%)
GA ¹	0.0127	0.0128	0.0128	3.9×10^{-5}	2190	−23.93
PSO ¹	0.0131	0.0190	0.0155	–	1291	+22.51
GA ²	0.0127	0.0130	0.0127	5.9×10^{-5}	80200	−97.93
SSB	0.0127	0.0167	0.0129	5.9×10^{-4}	25167	−93.39
PSO ²	0.0127	0.0129	0.0127	5.2×10^{-5}	1480	+11.17
ES	0.0127	0.0165	0.0135	9.7×10^{-4}	25000	−93.34
DE	0.0127	0.0127	0.0127	1.3×10^{-5}	24000	−93.06
HEA	0.0127	0.0127	0.0127	1.4×10^{-9}	24000	−93.06
EPS ¹	0.0126	0.0126	0.0126	8.5×10^{-7}	80000	−97.92
ACA	0.0126	0.0129	0.0127	3.5×10^{-5}	–	–
EPS ²	0.0127	0.0127	0.0127	1.2×10^{-8}	24950	−93.33
WCA	0.0127	0.0130	0.0127	8.1×10^{-5}	11750	−85.83
VPS	0.0127	0.0130	0.0127	7.3×10^{-5}	4020	−58.56
BB-BC	0.0127	0.0128	0.0127	1.3×10^{-5}	4020	− 58.56
BB-BC- <i>r</i>	0.0127	0.0129	0.0127	5.6×10^{-6}	4020	−58.56
SA-BB-BC	0.0127	0.0127	0.0127	3.4×10^{-5}	1666	–

first, purely random values define this parameter, while the second one employs the Logistic map. In Fig. 3c it is possible to observe a typical evolution of the convergence rate. Initially, its value is close to zero due to diversity of population. During the generations, the population tends to become homogeneous and the value of the parameter λ tends to unity. As a consequence, the value of the population size goes from the maximum to the minimum value, as presented in Fig. 3d. This reduction along the evolutionary process implies in a smaller value for the n_{eval} in relation to canonical algorithm.

4.2 Design of a Pressure Vessel

The second application considers a cylindrical vessel capped at both ends by hemispherical heads, as presented in Fig. 4 [39].

In this case, the total cost (material cost, forming, and welding) should be minimized. For this purpose, the following design variables are considered: thickness of the shell (x_1), thickness of the head (x_2), inner radius (x_3) and length of the cylindrical section of the vessel (x_4) (not including the head). In this test case, x_1 and x_2 are integer multiples of 0.0625 in, which are the available thicknesses of rolled steel plates, and x_3 and x_4 are continuous [39]. Mathematically, this problem is formulated

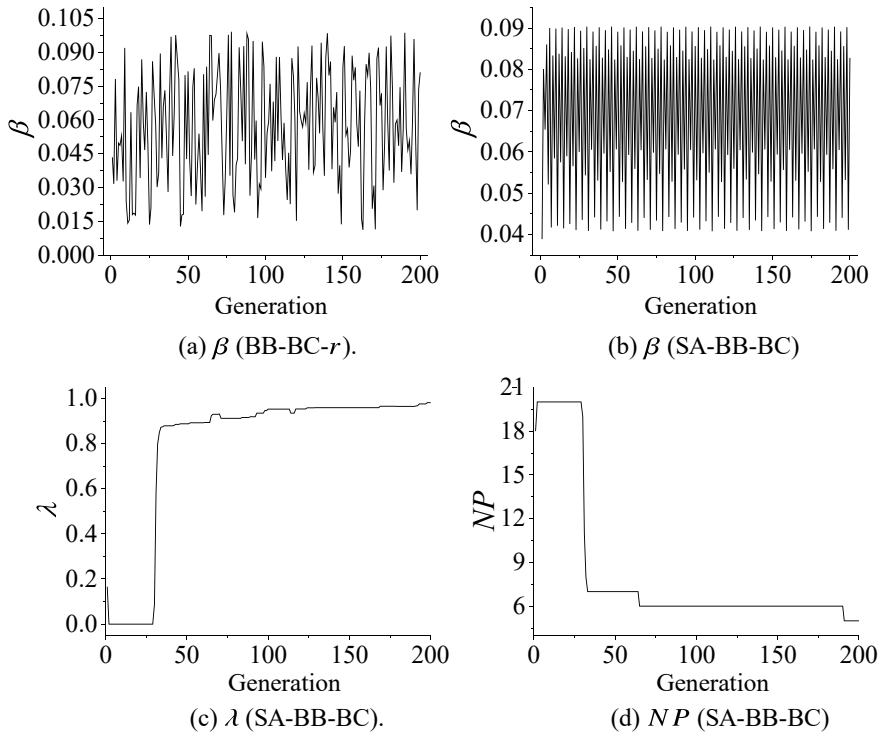


Fig. 3 Parameters λ , β and NP along the evolutionary process for the tension/compression string design problem

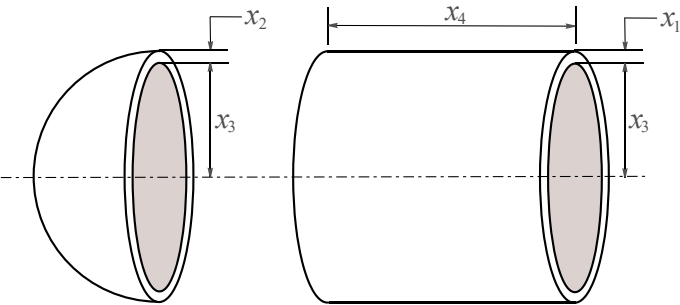


Fig. 4 Design of a pressure vessel problem

as:

$$\min f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (12a)$$

$$g_1(x) \equiv -x_1 + 0.0193x_3 \leq 0 \quad (12b)$$

$$g_2(x) \equiv -x_2 + 0.00954x_3 \leq 0 \quad (12c)$$

$$g_3(x) \equiv -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \quad (12d)$$

$$g_4(x) \equiv x_4 - 240 \leq 0 \quad (12e)$$

$$x_1 = \{0.0625t \mid t \in \mathbb{Z}, 1 \leq t \leq 21\} \quad (12f)$$

$$x_2 = \{0.0625t \mid t \in \mathbb{Z}, 1 \leq t \leq 21\} \quad (12g)$$

$$10 \leq x_3 \leq 200 \quad (12h)$$

$$10 \leq x_4 \leq 200 \quad (12i)$$

where the first (Eq. (12b)) and second (Eq. (12c)) inequality constraints represent the thicknesses of the side and top with the internal radius, respectively. The third (Eq. (12d)) guarantees that the internal volume is greater than a fixed value, and the fourth (Eq. (12e)) defines the maximum length. Equations (12f–12g) represent the discretized domain for x_1 and x_2 . Finally, inequalities indicated by the Equations (12h–12i) stand for the domain for the continuous variables (x_3 and x_4).

Like other population-based optimization strategies, both BB-BC, BB-BC- r and SA-BB-BC are not capable of dealing, directly, with discretized variables. In this case, for the treatment of this type of variable, the methodology proposed by Lobato [45] is considered. Essentially, this consists of transforming a discretized variable into a continuous variable, enabling the direct application of the BB-BC, BB-BC- r and SA-BB-BC algorithms. More details about this strategy can be found in reference [45].

Table 4 presents the obtained results for the pressure vessel problem, considering different optimization strategies. It is possible to observe that the BB-BC, BB-BC- r and SA-BB-BC algorithms were able to find a good approximation in relation to best solution reported by literature (\$ 5885.3327).

Table 5 presents the inequality constraints evaluated at optimal solutions for the different optimization strategies. As was observed for the first application, both BB-BC, BB-BC- r and SA-BB-BC algorithms were able to satisfy all constraints, as well as the other optimization strategies used.

Table 6 presents best, average, worst and standard deviation in terms of the objective function, the number of average objective function evaluations and decrease/increase rates Γ for the pressure vessel problem. One can note that both BB-BC, BB-BC- r and SA-BB-BC algorithms converged to a good estimate of the optimal solution, as found for the standard deviation. With respect to n_{eval} required by SA-BB-BC algorithm, it was observed an increase of 5.56% considering the PSO² strategy. Conversely, in comparison to other strategies, a reduction on the range 28.45% to 98.05% was observed. Thus, the proposed methodology was able to obtain a good approximation of the optimal solution with a lower number of evaluations of the

Table 4 Optimal solution (f^{opt}) considering different optimization strategies for the pressure vessel problem

Algorithm	x_1 (in)	x_2 (in)	x_3 (in)	x_4 (in)	f^{opt} (\$)
BBM	1.1250	0.6250	47.7000	117.7010	8129.1036
LMM	1.1250	0.6250	58.2910	43.6900	7198.0428
GA ³	0.9375	0.5000	48.3290	112.6790	6410.3811
GA ¹	0.8125	0.4375	40.3239	200.0000	6288.7445
SC	0.8125	0.4375	41.9768	182.2845	6171.0000
GA ²	0.8125	0.4375	42.0973	176.6540	6059.9463
PSO ²	0.8125	0.4375	42.0912	176.7465	6061.0777
ES	0.8125	0.4375	42.0980	176.6405	6059.7456
EPS ¹	0.8036	0.3972	41.6392	182.4120	5930.3137
ACA	0.8125	0.4375	42.0983	176.6377	6059.7258
EPS ²	0.8125	0.4375	42.0984	176.6365	6059.7143
WCA	0.7781	0.3846	40.3196	200.0000	5885.3327
CS	0.8125	0.4375	42.0984	176.6365	6059.7143
VPS	0.8125	0.4375	42.0984	176.6366	6059.7146
BB-BC	0.8125	0.4375	42.1006	176.6138	6059.7273
BB-BC- r	0.8125	0.4375	42.0976	176.6323	6059.7145
SA-BB-BC	0.8125	0.4375	42.0984	176.6388	6059.7184

Table 5 Inequality constraints evaluated at optimal solution, considering different optimization strategies for the pressure vessel problem

Algorithm	$g_1(x)$	$g_2(x)$	$g_3(x)$	$g_4(x)$
BBM	-0.2044	-0.1699	54.2260	-122.2990
LMM	0.0000	-0.0689	-21.2201	-196.3100
GA ³	-0.0048	-0.0389	-3652.87	-127.3210
GA ¹	-0.0343	-0.0528	-27.1058	-0.0000
SC	-0.0023	-0.0370	-23420.5	-57.7155
GA ²	0.0000	-0.0359	-27.8861	-63.3460
PSO ²	-0.0001	-0.0359	-116.382	-63.2535
EPS ¹	0.0000	0.0000	-1.5914	-57.5879
WCA	0.0000	0.0000	0.0000	-40.0000
VPS	0.0000	-0.0359	-0.0084	-63.3634
BB-BC	0.0000	-0.0360	-0.0086	-63.3655
BB-BC- r	0.0000	-0.0361	-0.0085	-63.3654
SA-BB-BC	0.0000	-0.0358	-0.0082	-63.3658

Table 6 Best (f^{opt}), average (f_m^{opt}) and worst (f^w) results, standard deviation (f^{std})

Algorithm	f^{opt} (\$)	f^w (\$)	f_m^{opt} (\$)	f^{std} (\$)	n_{eval}	Γ (%)
GA ¹	6288.7445	6308.1497	6293.8432	7.4133	2190	−28.45
SC	6171.0000	6453.6500	6335.0500	–	12630	−87.60
GA ²	6059.9463	6469.3220	6177.2533	130.9297	80200	−98.05
PSO ²	6061.0777	6363.8041	6147.1332	86.4545	1480	+5.56
ES	6059.7456	7332.8799	6850.0049	426.0000	25000	−93.74
EPS ¹	5930.3137	5960.0557	5946.7901	9.1614	80000	−98.05
ACA	6059.7258	6150.1289	6081.7812	67.2418	–	–
EPS ²	6059.7143	6059.7143	6059.7143	0.0000	42100	−96.28
WCA	5885.3327	6590.2129	6198.6172	213.0490	27500	−94.31
CS	6059.7143	6495.3470	6447.7360	502.6930	15000	−89.56
VPS	6059.7146	6141.3115	6080.9478	21.8291	4020	−61.02
BB-BC	6059.7273	6098.9787	6082.3456	5.678	4020	−61.02
BB-BC- <i>r</i>	6059.7145	6099.9897	6082.4367	9.1238	4020	−61.02
SA-BB-BC	6059.7184	6187.5556	6099.7568	12.9451	1567	–

Number of average objective function evaluations (n_{eval}) and rate Γ considering different optimization strategies for the pressure vessel problem

objective function compared to most of the other methods studied (GA¹, SC, GA², ES, EPS¹, EPS², WCA, CS, VPS, BB-BC and BB-BC-*r*).

Figure 5 illustrates the progression of the parameters λ , β and NP along the generations for the pressure vessel problem. Figure 5a and b demonstrate the variation of the parameter β considering a random strategy and the Logistic map, respectively. A typical variation of the convergence rate can be observed in Fig. 5c: initially, the value of this parameter is close to zero due to diversity of population. When the population starts to become homogeneous, the value of this parameter approaches unity. Thus, the parameter NP goes from the maximum to the minimum value (see Fig. 5d). As mentioned earlier, this reduction allows a smaller value for the n_{eval} in relation to canonical algorithm.

Finally, one can conclude that the strategy considered for the treatment of discretized variables and associated with both BB-BC, BB-BC-*r*, and SA-BB-BC algorithms was able to obtain good estimates for the best reported optimal solution.

4.3 Design of a Welded Beam

The last application considers a welded beam design [46]. The aim is to minimize the minimum cost subject to constraints on shear stress (τ), bending stress in the beam (σ), buckling load on the bar (P_c), end deflection of the beam (δ), and side constraints. For this purpose, the following design variables are considered: x_1 , x_2 , x_3 and x_4 , as observed in Fig. 6.

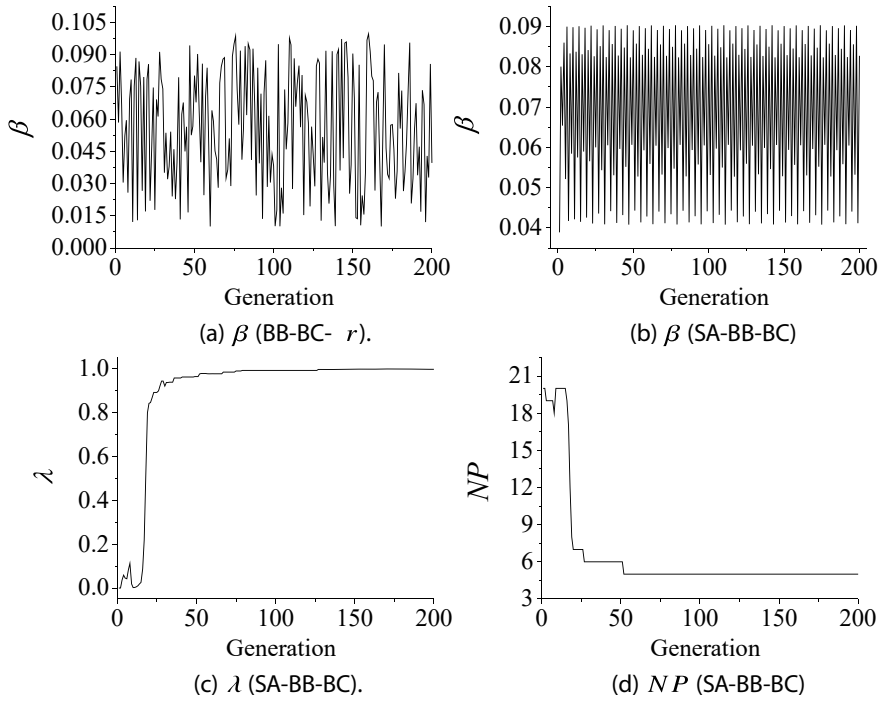
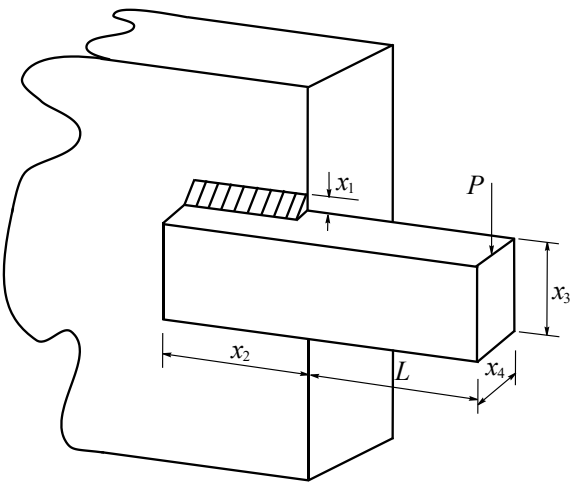


Fig. 5 Parameters λ , β and NP along the evolutionary process for the pressure vessel problem

Fig. 6 Welded beam design problem



From the mathematical point of view, this problem is formulated as:

$$\min f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \quad (13a)$$

$$g_1(x) \equiv \tau(x) - \tau_{max} \leq 0 \quad (13b)$$

$$g_2(x) \equiv \sigma(x) - \sigma_{max} \leq 0 \quad (13c)$$

$$g_3(x) \equiv x_1 - x_4 \leq 0 \quad (13d)$$

$$g_4(x) \equiv 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0 \quad (13e)$$

$$g_5(x) \equiv 0.125 - x_1 \leq 0 \quad (13f)$$

$$g_6(x) \equiv \delta(x) - \delta_{max} \leq 0 \quad (13g)$$

$$g_7(x) \equiv P - P_c(x) \leq 0 \quad (13h)$$

$$0.1 \leq x_1 \leq 2.0 \quad (13i)$$

$$0.1 \leq x_2 \leq 10.0 \quad (13j)$$

$$0.1 \leq x_3 \leq 10.0 \quad (13k)$$

$$0.1 \leq x_4 \leq 2.0 \quad (13l)$$

where:

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R(x)} + (\tau'')^2} \quad (14a)$$

$$\tau'(x) = \frac{P}{\sqrt{2}x_1x_2} \quad (14b)$$

$$\tau''(x) = \frac{M(x)R(x)}{J(x)} \quad (14c)$$

$$M(x) = P \left(L + \frac{x_2}{2} \right) \quad (14d)$$

$$R(x) = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2} \right)^2} \quad (14e)$$

$$J(x) = 2 \left[\frac{x_1x_2}{\sqrt{2}} \left(\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2} \right)^2 \right) \right] \quad (14f)$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2} \quad (14g)$$

$$\delta(x) = \frac{4PL^3}{Ex_4x_3^3} \quad (14h)$$

$$P_c(x) = \frac{4.013\sqrt{\frac{EGx_3^2x_4^6}{36}}}{L^2} \left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}} \right) \quad (14i)$$

Table 7 Optimal solution (f^{opt}) considering different optimization strategies for the welded beam design problem

Algorithm	x_1 (in)	x_2 (in)	x_3 (in)	x_4 (in)	f^{opt} (\$)
GP	0.2455	6.1960	8.2730	0.2455	2.3859
GA ⁴	0.2489	6.1730	8.1789	0.2533	2.4331
SC	0.2407	6.4851	8.2399	0.2497	2.4426
SSB	0.2444	6.2380	8.2886	0.2446	2.3854
DE	0.2444	6.2175	8.2914	0.2444	2.3810
HEA	0.2444	6.2175	8.2915	0.2444	2.3810
VPS	0.2444	6.2175	8.2914	0.2444	2.3810
BB-BC	0.2203	6.9290	8.9527	0.2228	2.3810
BB-BC- r	0.2202	6.9287	8.9525	0.2227	2.3810
SA-BB-BC	0.2202	6.9293	8.9533	0.2227	2.3810

Table 8 Inequality constraints evaluated at optimal solution considering different optimization strategies for the welded beam design problem

Algorithm	$g_1(x)$	$g_2(x)$	$g_3(x)$	$g_4(x)$	$g_5(x)$	$g_6(x)$	$g_7(x)$
GP	-1.4467	-4.7151	0.0000	-3.0203	-0.1205	-0.2342	-74.276
GA ⁴	-11.2070	-255.57	-0.0044	-2.9829	-0.1239	-0.2342	-618.81
SC	-129.8545	-270.40	-0.0090	-2.9663	-0.1157	-0.2343	-372.49
SSB	-32.4102	-3.2454	-0.0001	-3.0201	-0.1194	-0.2342	-13.079
DE	0.0000	0.0000	0.0000	-3.0230	-0.1194	-0.2342	0.0000
HEA	-0.0016	-0.0397	0.0000	-3.0230	-0.1194	-0.2342	-0.0011
VPS	-0.0001	-0.0018	0.0000	-3.0230	-0.1194	-0.2342	-0.0003
BB-BC	-0.0001	-0.0017	0.0000	-3.0203	-0.1172	-0.2342	-0.0003
BB-BC- r	-0.0001	-0.0019	0.0000	-3.0212	-0.1166	-0.2343	-0.0004
SA-BB-BC	-0.0001	-0.0018	0.0000	-3.0211	-0.1174	-0.2344	-0.0002

where τ' and τ'' are two components of the shear stress of the weld, M is the moment of the force P in relation to center of gravity, J is the polar moment of inertia and R is the maximum distance from any point on one of the welds to the center of gravity.

Table 7 contains the obtained results for the welded beam design problem considering the proposed algorithm and different optimization strategies. Once again, both the BB-BC, BB-BC- r and SA-BB-BC algorithms were able to find a good estimate for the optimal solution.

Table 8 shows the values for the inequality constraints evaluated at optimal solutions for the welded beam design problem. As observed for other algorithms, the results obtained by BB-BC, BB-BC- r and SA-BB-BC algorithms satisfy all constraints.

Table 9 presents best, average, worst and standard deviation in terms of the objective function, the number of average objective function evaluations and decrease/

Table 9 Best (f^{opt}), average (f_m^{opt}) and worst (f^w) results, standard deviation (f^{std}), number of average objective function evaluations (n_{eval}) and rate Γ considering different optimization strategies for the welded beam design problem

Algorithm	f^{opt} (\$)	f^w (\$)	f_m^{opt} (\$)	f^{std} (\$)	n_{eval}	Γ (%)
GA ⁴	2.4331	–	–	–	4600	–70.79
SC	2.4426	2.6315	2.5215	–	19259	–93.03
SSB	2.3854	6.3997	3.2551	9.6×10^{-1}	33095	–95.94
DE	2.3810	2.3810	2.3810	3.2×10^{-10}	24000	–94.40
HEA	2.3810	2.3810	2.3810	1.3×10^{-5}	30000	–95.52
VPS	2.3810	2.3810	2.3810	1.0×10^{-5}	10020	–86.59
BB-BC	2.3810	2.3811	2.3810	2.3×10^{-5}	4020	– 66.57
BB-BC- <i>r</i>	2.3810	2.3813	2.3811	4.5×10^{-5}	4020	–66.57
SA-BB-BC	2.3810	2.3813	2.3811	3.2×10^{-5}	1344	–

increase rates for the welded beam design problem. Again, both the BB-BC, BB-BC-*r*, and SA-BB-BC algorithms always converged to a good estimate of the optimal solution. Regarding the value of the n_{eval} required by SA-BB-BC algorithm, a decrease in the range from 66.57% to 95.94% with respect to other optimization strategies (GA⁴, SC, SSB, DE, HEA, VPS, BB-BC, and BB-BC-*r*) was observed.

Figure 7 illustrates the evolution of the parameters λ , β and NP through the generations, for the welded beam design problem. The random and chaotic behavior of the parameter β are observed in Fig. 7a and b, respectively. In Fig. 7c, a typical evolution of the convergence rate is presented. As mentioned earlier, in the first generations, the value of this rate is close to zero and along the evolutionary process the population tends to become homogeneous, and consequently, the value of λ tends to unity. Thus, the value of the population size goes from the maximum to the minimum value, as presented in Fig. 7d. As a consequence, the n_{eval} is reduced in relation to canonical algorithm.

5 Conclusions

In this work, a new optimization strategy based on BB-BC algorithm was proposed, called SA-BB-BC. The SA-BB-BC algorithm is based on the concept of convergence rate associated with a chaotic search pattern, both considered updating dynamically the control parameter β and the population size NP .

The proposed methodology was employed in engineering systems design. In general, for each test case, the SA-BB-BC algorithm was able to find results close to the best solution reported by literature. However, the number of objective function evaluations, for the vast majority of reported results, was much smaller. In addition, the SA-BB-BC strategy proves to be superior when compared with the canonical

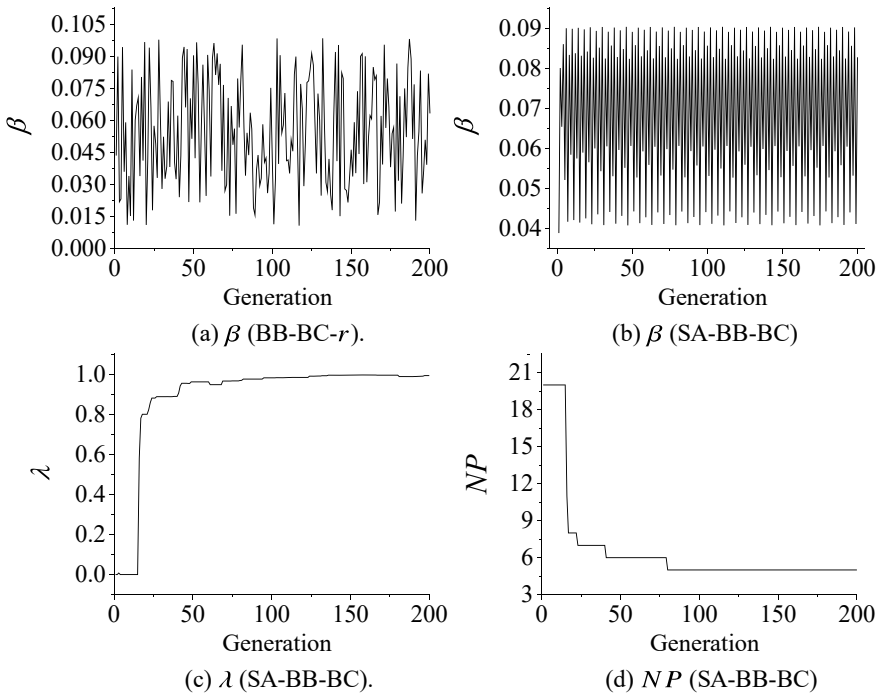


Fig. 7 Parameters λ , β and NP along the evolutionary process for the welded beam design problem

BB-BC algorithm (with constant parameters) and BB-BC- r algorithm (where the parameter β is randomly updated).

Finally, it is important to mention that the SA-BB-BC algorithm can avoid premature convergence, reducing the number of objective function evaluations, but without losing the quality of the solution. Additionally, the incorporation of this strategy into other evolutionary strategies is straightforward.

In future work, we intend to analyze the influence of the parameter values of SA-BB-BC in other engineering design problems.

Acknowledgements The authors are thankful for the financial support provided to the present research by FAPEMIG and CNPq agencies.

References

1. Lobato FS, Steffen V Jr, Silva Neto AJ (2010) Self-adaptive differential evolution based on the concept of population diversity applied to simultaneous estimation of anisotropic scattering phase function, albedo and optical thickness. *Comput Model Eng Sci* 69(1):1–17
2. Karafotias G, Hoogendoorn M, Eiben AE (2015) Parameter control in evolutionary algorithms: trends and challenges. *IEEE Trans Evolut Comput* 19(2):167–187
3. Eiben AE, Smith JE (2015) Introduction to evolutionary computing, natural computing series, 2nd edn. Springer, Berlin, Heidelberg
4. Guan Y, Yang L, Sheng W (2017) Population control in evolutionary algorithms: review and comparison. In: International conference on bio-inspired computing: theories and applications. Springer, pp 143–154
5. Aleti A, Moser I (2016) A systematic literature review of adaptive parameter control methods for evolutionary algorithms. *ACM Comput Surveys* 49(3):56–78
6. Dixit A, Mani A, Bansal R (2021) An adaptive mutation strategy for differential evolution algorithm based on particle swarm optimization. *Evolut Intell* 15:1571–1585
7. Zhou S, Xing L, Zheng X, Du N, Wang L, Zhang Q (2021) A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times. *IEEE Trans Cybernet* 51(3):1430–1442
8. Liu J, Lampinen J (2005) A fuzzy adaptive differential evolution algorithm. *Soft Comput* 9(6):448–462
9. Chen R, Yang B, Li S, Wang S (2020) A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. *Comput Indus Eng* 149:106778–106798
10. Zhang X, Liu Q, Qu Y (2023) An adaptive differential evolution algorithm with population size reduction strategy for unconstrained optimization problem. *Appl Soft Comput* 138:110209–110223
11. Lacerda MGP, Amorim Neto HA, Ludermitr TB, Kuchen H, Lima Neto FB (2022) Towards a parameterless out-of-the-box population size control for evolutionary and swarm-based algorithms for single objective bound constrained real-parameter numerical optimization. *Appl Soft Comput* 123:108920–108932
12. Zhao Z, Liu B, Zhang C, Liu H (2019) An improved adaptive NSGA-II with multi-population algorithm. *Appl Intell* 49(2):569–580
13. Erol OK, Eksin I (2006) A new optimization method: Big Bang-Big Crunch. *Adv Eng Software* 37(2):106–111
14. Kaveh A, Bakhshpoori T (2019) Metaheuristics: outlines, MATLAB codes and examples. Springer, pp 190
15. Camp CV (2007) Design of space trusses using Big Bang-Big Crunch optimization. *J Struct Eng* 133(7):999–1008
16. Vellev S (2008) An adaptive genetic algorithm with dynamic population size for optimizing join queries. In: International conference: intelligent information and engineering systems, INFOS 2008. Varna, Bulgaria, 2008 June–July, pp 1–10
17. Cavalini AA, Lobato FS, Koroishi EH, Steffen V (2015) Model updating of a rotating machine using the self-adaptive differential evolution algorithm. *Inverse Problems in Sci Eng* 1:504–523
18. Tavazoei MS, Haeri M (2007) Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. *Appl Mathemat Comput* 187(2):1076–1085
19. Li B, Jiang W (1998) Optimizing complex functions by Chaos search. *Cybernet Syst* 29(4):409–419
20. Ott E (2002) Chaos in dynamical systems. UK, Cambridge University Press, Cambridge
21. Arora JS (1989) Introduction to optimum design. McGraw-Hill, New York
22. Belegundu AD (1992) A study of mathematical programming methods for structural optimization. University of Iowa
23. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. *Comput Indus* 41(2):113–127

24. Coello CAC, Montes EM (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Adv Eng Inform* 16(3):193–203
25. Deb K (1997) GeneAS: a robust optimal design technique for mechanical component design. In: *Evolutionary algorithms in engineering applications*, Berlin, Springer, pp 497–514
26. Deb K (1991) Optimal design of a welded beam via genetic algorithms. *AIAA J* 29(11):2013–2015
27. Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng Optimizat* 33(6):735–748
28. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng Appl Artif Intell* 20(1):89–99
29. Zahara E, Kao Y-T (2009) Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems. *Expert Syst Appl* 36(2):3880–3886
30. Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Appl Soft Comput* 10(2):629–640
31. Ray T, Liew K-M (2003) Society and civilization: an optimization algorithm based on the simulation of social behavior. *IEEE Trans Evolut Comput* 7(4):386–396
32. Tsai J-F (2005) Global optimization of nonlinear fractional programming problems in engineering design. *Eng Optimizat* 37(4):399–409
33. Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. *Inform Sci* 178(15):3043–3074
34. Montes EM, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *Int J General Syst* 37(5):443–473
35. Wang Y, Cai Z, Zhou Y, Fan Z (2009) Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Struct Multidisciplinary Optimizat* 37(3):395–413
36. Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. *Eng Comput* 27(1):155–182
37. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput Struct* 110:151–166
38. Sandgren E (1998) Nonlinear integer and discrete programming in mechanical design. In: *International design engineering technical conferences and computers and information in engineering conference*. pp 95–105
39. Kannan BK, Kramer SN (1994) An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *J Mech Design* 116(2):405–411
40. Akhtar S, Tai K, Ray T (2002) A socio-behavioural simulation model for engineering design optimization. *Eng Optimizat* 34(4):341–354
41. Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng Comput* 29(1):17–35
42. Ragsdell KM, Phillips DT (1976) Optimal design of a class of welded structures using geometric programming. *ASME J Eng Indus* 98(3):1021–1025
43. Andrade JC (2023) Vibrating particle system algorithm to engineering systems design. Federal University of Uberlândia, Brazil
44. Vanderplaats GN (1984) *Numerical optimization techniques for engineering design with applications*, 1st edn. McGraw-Hill, New York
45. Lobato FS (2008) *Multi-objective optimization to engineering systems design*. Federal University of Uberlândia
46. Rao SS (2009) In: *Engineering optimization: theory and practice*. 4th edn. Hoboken, New Jersey, Wiley & Sons

Chapter 18

Optimizing Camera Placement for Maximum Coverage of Simple Polygons with Holes: Deterministic Approaches and Swarm Intelligence Algorithms



Adis Alihodzic¹, Eva Tuba², and Milan Tuba³

1 Introduction

The Internet of Things (IoT) is revolutionizing the modern world by expanding Internet technologies. It connects objects and facilitates information sharing through recognition, perception, and communication. The widespread adoption of IoT has significantly improved people's daily lives. Many IoT technologies rely on sensor and camera networks as the fundamental hardware for collecting, processing, and transmitting data. These networks find applications in various fields, such as structural health monitoring [17], extensive multimedia data communication [36], target localization and tracking [59], robot motion planning [28], and more. Moreover, the strategic placement of cameras plays a crucial role in IoT and computer vision tasks, enhancing tasks like recognition and tracking [37]. Cameras are also integral to robotics, providing vital information for navigation. An intriguing application involves optimizing camera placement in real-world environments, like campus layouts represented as orthogonal polygons [6]. Additionally, optimizing camera deployment can aid traffic jam monitoring by surveilling specific orthogonal areas to monitor road events effectively.

The problem of optimal camera placement has been analyzed for decades, with roots tracing back to the Art Gallery Problem (AGP) in computational geometry [23]. Stemming in the 1970s, the AGP posed one of the field's earliest and most

A. Alihodzic (✉)

Department of Mathematical and Computer Sciences, University of Sarajevo,
Zmaj od Bosne, 33-35, Sarajevo, Bosnia and Herzegovina
e-mail: adis.alihodzic@pmf.unsa.ba

E. Tuba

Trinity University, San Antonio, TX, USA
e-mail: etuba@ieee.org

M. Tuba

Singidunum University, Belgrade, Serbia
e-mail: tuba@ieee.org

significant challenges. In computational geometry, it handles the visibility dilemma of positioning security cameras to oversee every area of a museum or gallery [16]. AGP concentrates on efficiently covering the interior of a polygonal environment with minimal agents such as cameras, guards, or sensors, often designated as vectors. Initially classified as an intractable NP-hard problem, the AGP aims to determine the minimum number of cameras required to cover the interior of a polygon, whether it has holes or not [38, 40, 46, 52]. Despite significant contributions from researchers such as O'Rourke, Supowit, Lee, Lin, Katz, Rpoisman, Schuchardt, and Hecker, locating the fewest agents capable of surveilling any polygon remains NP-hard. In 1975, Chvátal demonstrated that for simple polygons with n vertices, at most $\lfloor \frac{n}{3} \rfloor$, cameras are always sufficient and sometimes necessary for complete coverage. Using the coloring technique employed by Fisk [31] to prove Chvátal's result, Avis and Toussaint developed an $O(n \log n)$ time complexity algorithm for camera placement in a simple polygon without holes, although it does not guarantee the minimal number of cameras. Additionally, Bjorling-Sachs and Souvaine [19] proposed an $O(n^2)$ time algorithm for placing non-optimal guards in polygons with h holes or obstacles. O'Rourke has shown that for an n -sided polygon with h holes, at most $\lfloor \frac{n+2h}{3} \rfloor$ agents are required. Contrariwise, Bjorling-Sachs, Souvaine, Hoffmann, and others showed that $\lfloor \frac{n+h}{3} \rfloor$ cameras are typically sufficient to cover polygons with n vertices on the outer boundary and h holes inside [19, 35].

With the general deployment of camera networks, the emphasis on optimal camera placement has shifted from theoretical analysis to practical application. Previous research primarily focused on formulating the problem, as various papers showed [12, 18, 20, 34, 65]. However, recent investigations in automated camera placement have centered on two key objectives: accommodating specific user requirements and devising efficient optimization methods. This study specifically addresses the development of optimization algorithms to maximize coverage within a given simple polygon with holes by using a fixed number of cameras as well as different types of cameras. To efficiently address the large search space of moderately sized camera placement problems, swarm intelligence techniques are employed to produce suboptimal camera positions within a practical time frame [22, 29, 42, 64].

This study introduces two algorithmic approaches, suboptimal deterministic and swarm intelligence, to address the camera placement problem. In reference [60], Wang and her colleagues achieved an approximation of the AGP on a campus that was represented as a simple polygon with holes. They employed particle swarm optimization (PSO) on the pixels to locate the approximate solution. In our approach, we use both deterministic and swarm intelligence techniques to seek a suboptimal solution. All proposed techniques are based on a geometric approach that does not depend on pixels except for calculating coverage since, in the literature, there is no polynomial algorithm for determining the surface area of the union of n non-convex polygons. We exploit swarm intelligence algorithms to tackle the NP-hard problem of optimal camera placement due to their global optimization capabilities, which facilitate rapid convergence and reduce CPU processing time. Before applying these algorithms to address the optimal camera placement problem, preprocessing techniques are employed to decompose the polygons into convex components, min-

imizing the number of required cameras. For decomposing a polygon with holes, we use constrained Delaunay triangulation [24] since an unconstrained one is used for polygons without holes. The constraints in constrained Delaunay triangulation will encompass all edges that cut through the holes. In order to evaluate the effectiveness of the proposed techniques, all proposed methods will be implemented and tested on two generated simple polygons as in paper [60]. Experimental findings demonstrate that stochastic approaches outperform the deterministic ones, providing suboptimal solutions within a reasonable timeframe.

The ensuing sections in this paper are positioned as obeys. Section 2 describes the problem formulation for the optimal camera placement of a simple polygon with holes. Section 3 proposes several suboptimal deterministic visibility algorithms. Section 4, a novel contribution to the field, presents four swarm intelligence techniques for solving optimal camera placement problems. Section 5 presents experimental and comparative results of applying our algorithms. Finally, the last section of the paper, Sect. 6, discusses conclusions and suggestions for future work.

2 Formulation of a Mathematical Problem

In this section, a mathematical model will be described for the optimal placement of cameras so that the interior of a simple polygon with holes is covered as much as possible. Optimal camera placement problems are often mathematically formulated as coverage, optimization, or combinatorial geometry problems. Without loss of generality, let the space to be covered or monitored is a plane or some simple polygon P composed of n vertices that contain m disjoint obstacles $O_j (j = \overline{1, m})$ within it. Let p cameras $\{K_i\}_{i=1}^p$ be given so that each of them is defined by parameters such as center (x_i, y_i) , radius r_i , angle orientation θ_i , and type t_i . The problem of optimal placement of cameras is reduced to finding the appropriate values of the mentioned parameters except for the parameter r_i (this parameter is fixed for the camera type and refers to the depth of its visibility) to maximize the cost function $f(\cup_{i=1}^p K_i(x_i, y_i, \theta_i, t_i), P)$, where the function $f(\cdot, \cdot)$ represents the surface area of the union of p regions covered by p cameras $\{K_i\}_{i=1}^p$. In other words, the mathematical model for the optimal camera coverage of the interior of the simple polygon P can be defined as follows:

$$\begin{aligned}
 & \underset{x_i, y_i, \theta_i, t_i}{\text{maximize}} && f\left(\bigcup_{i=1}^p K_i(x_i, y_i, \theta_i, t_i), P\right) \\
 & \text{subject to} && \\
 & && K_i \cap P \subset P \ (\forall i), \\
 & && O_j \subset P \ (\forall j), \\
 & && K_i \cap O_j \neq K_i \ (\forall i)(\forall j), \\
 & && O_j \cap O_k = \emptyset \ (\forall j)(\forall k).
 \end{aligned} \tag{1}$$

It is necessary to consider two scenarios before locating the optimal parameters for cameras $\{K_i\}_{i=1}^p$. The first is the camera model, and the second is the environment model. In this section, we first introduce the camera model used in this paper. We consider three types of cameras and focus on two-dimensional planar areas, where a circle section represents the field of view of a camera. Next, the environment is discretized into square grids, depicted as pixels in the image. Based on this, we propose several versions of deterministic visibility algorithms and one stochastic algorithm to determine whether the camera covers a grid of pixels and calculate the overall coverage.

2.1 The Camera Framework

The camera coverage model involves three primary parameters: Field of View (FoV), Depth of Field (DoF), and Pixel Resolution (PR). The FoV creates a geometric shape resembling a rectangular pyramid, projecting objects within the image plane from the camera's optical center. This chapter simplifies the model by focusing on planar regions and the horizontal FoV angle. The DoF determines the range of distances for sharp imaging, affecting the visibility of objects. The PR is the ratio of pixels to object size, crucial for tasks like face recognition with minimum resolution requirements. Factors like perspective distortion, occlusion, and camera pose also impact the model. Surveillance systems typically use Static Perspective (SP), Pan-Tilt-Zoom (PTZ), and Omnidirectional (OD) cameras, each with specific characteristics. The SP cameras have fixed positions, the PTZ cameras allow orientation adjustments, and the OD cameras offer a 360-degree horizontal FoV angle. Different camera fields can be represented as circles, triangles, sectors, or trapezoids. This study employs sectors for SP and PTZ cameras' coverage and circles for OD cameras, illustrated in Fig. 1. The parameters of the mentioned cameras are separated into geometric parameters and intrinsic parameters. The geometric parameters are the position of camera expressed by coordinates (x, y) in the plane and its orientation expressed by angle θ , where θ is fixed at zero at the OD cameras.

The intrinsic parameters are related to the horizontal field of view expressed by angle θ and maximum viewing distance expressed by distance r . Intrinsic parameters θ and r are determined by the camera type c_i . Thus, the camera coverage model can be represented by the vector $d = [x \ y \ \theta \ c_i]$, where x , y , and θ are continuous variables, while c_i is a discrete variable. In Fig. 1, the maximum viewing distance (DoF) is expressed by r_1 for PTZ camera and by r_2 for OD camera. Also, the field of view (FoV) for PTZ camera expressed by α is set to $\frac{\pi}{3}$, while for OD $\alpha = 2\pi$.

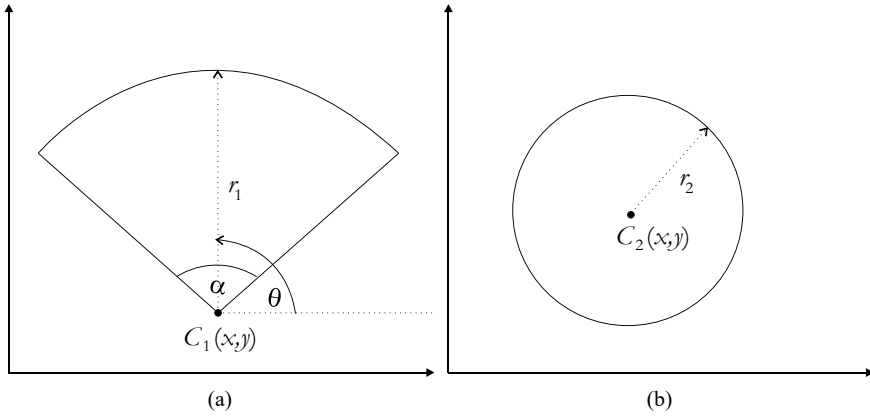


Fig. 1 **a** The illustration of the SP and PTZ cameras. **b** The example of the OD camera

2.2 Environment Model

An environmental model is crucial for the camera framework as it is the basis for further analysis. A complete model has details such as the region's shape, the position and shape of obstacles, parts of observed objects, the importance of different subregions, and communication interference. However, incorporating all these elements makes the model complex and computationally intensive. To simplify researchers often simplify the model. This study uses a two-dimensional instead of an entire 3D space and discretizes the target area into a uniform square grid for easier visibility and coverage calculations. We account for fixed obstacles but not moving ones or mutual occlusion of monitored objects, and we assume uniform importance across all positions.

2.3 An Example of Coverage Calculation for a Simple Polygon with Holes

This section explains the coverage of a simple polygon composed of 12 vertices and five holes (obstacles) inside it, as shown in Fig. 2. Figure 2 also represented two types of cameras, which will be used for a polygon's interior coverage. In the experimental analysis, the PTZ camera is selected so that its radius r_1 is equal to the number 160, and $\alpha = \pi/3$, which means that it covers 4331 pixels. Similarly, for the OD camera, the radius r_2 is taken to be 64, so it is obtained that it covers 4205 pixels. The task of optimal placement of cameras within a simple polygon with holes is to determine the optimal locations and the orientations of the cameras to place them in order to maximize the total coverage. In this chapter, the determination of

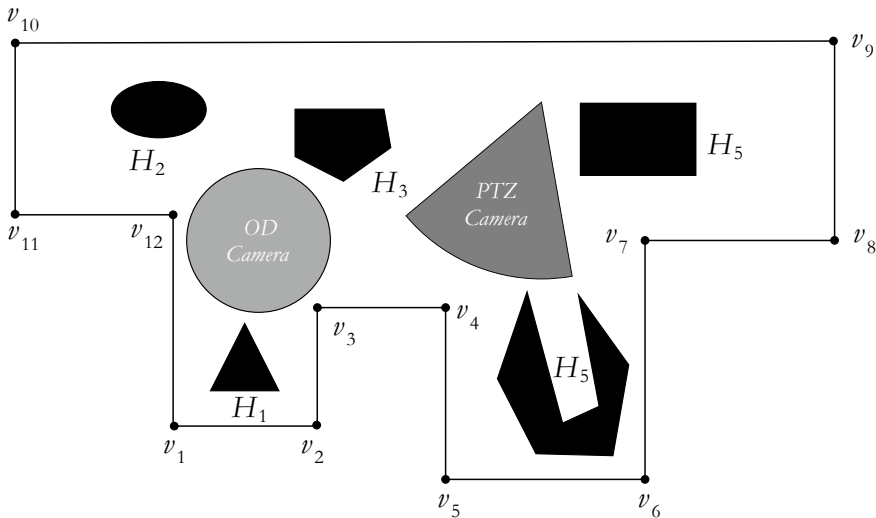


Fig. 2 A representation of a concave polygon composed of twelve vertices with five obstacles inside it

positions and orientations will be carried out through deterministic and stochastic algorithms, emphasizing the application of known metaheuristics, i.e., swarm intelligence techniques. There are two approaches to calculating spatial coverage using cameras: a geometric approach and an approach based on looking at pixel coverage. The geometric approach mainly consists of two steps. The first step is to do a camera union, which implies looking for unions of simple polygons, which ultimately generate several polygons with and without holes. It is essential to point out that joining polygons can generate polygons with one or more holes. After that, the areas of the resulting polygons are calculated and stored. Since some of the resulting polygons cross the boundary of the central polygon, which includes obstacles, and some of them also intersect obstacles, it is necessary to perform polygon clipping between the combined polygons and the boundary line of the main polygon as well as with the obstacles, in the second step. As a result, clipped polygons are obtained, for which surfaces should also be calculated. In the end, the differences between the surfaces of the combined and clipped polygons are summed, which gives information about the coverage of the central polygon. Due to the geometric approach's complexity and the paper's authors' inability to find an efficient algorithm for the union of concave polygons, the second approach will be used for optimal coverage of the polygon's interior.

The second approach for estimating the coverage of polygons with cameras is not directly geometric. It boils down to calculating how many pixels from the polygon's interior are covered by camera pixels, excluding those pixels belonging to obstacles. This approach is based on a matrix design, i.e., associated with adequate creation and

retrieval of matrix blocks. Namely, it can be efficiently solved by initially creating a rectangle R of minimum area that contains the central simple polygon P . After that, the visibility matrix M_v should be created, which contains the intensities of the pixels of the rectangle R , so its elements will be the numbers 0, 255, 100, and 150. Hence, the matrix M_v contains

- the pixels of the polygon's interior,
- the pixels of obstacles inside the polygon, and
- the pixels outside the polygon lie inside the rectangle R .

Consequently, pixels inside the polygon outside the obstacles will be assigned the number 255, and those pixels inside obstacles and outside the polygon P will be assigned the number 0. In contrast, the pixels of the PTZ and OD cameras will be assigned the numbers 100 and 150, respectively. Before drawing the pixels of cameras on the canvas, i.e., inside the polygon P and outside the obstacles, the number of elements of matrix M_v whose value is equal to 255 is calculated. This number will later be used to determine the percentage coverage of the polygon with cameras. It is necessary to know the center $C_K(x_k, y_k)$ of the camera K , radius r_k , and angle θ_k to draw the pixels of a particular camera K . Drawing the camera pixels on the screen is reduced to updating the content of a particular block B_K of the matrix M_v , where the block B_K is a submatrix of the initial matrix M_v composed of its elements. Based on an arbitrary camera pixel planned to be drawn, the appropriate position of the element in the block B_K is determined in constant time. Block B_K is updated using the Ray-Casting algorithm [60] to locate and draw only the visible pixels of the camera, i.e., pixels inside polygon P that are not obscured by obstacle pixels. Accordingly, the elements of the block B_K are passed through, and their content is updated if the pixels at the positions of those elements lie inside the cameras and are not obscured by the pixels of obstacles. This fact implies that the elements of the block B_K must have a value of 255 before updating. In other words, to update the contents of the block B_K , it is not necessary to physically draw the pixels of the cameras on the canvas, but it is enough to check whether they are visible operating the Ray-Casting algorithm. If they are, then depending on whether the PTZ or OD camera is used, the content of the block B_K is updated to the value 100 or 150. Also, if the pixels of the new camera are drawn over the pixels of existing camera, then it is enough to check in the matrix M_v whether there are values different from 0 and 255 at the positions where new pixels are drawn. If there are, then this is an indicator that these pixels should not be drawn, so the content of the corresponding block B_K should not be updated. If it is established that the pixels are visible and were not previously drawn, then the number of changes occurring in the matrix M_v is recorded. The pseudocode of Algorithm 1 represents the above description for drawing visible camera pixels and will be used in deterministic algorithms and metaheuristics.

Algorithm 1: Drawing camera pixels $K(C_K(x, y), \alpha, \theta, r_1, r_2)$

Input: Camera center $C_K(x, y)$, orientation angle θ , radius r_1 of a PTZ camera, radius r_2 of an OD camera, angle α which determines the camera type.

Step 1: Initialize the counter t_K of camera K to zero, i.e. $t_K \leftarrow 0$, and set the angle α to $\pi/4$ or 2π if the camera K is a PTZ or OD camera.

Step 2: A) For each visible pixel $p(x, y)$ of the camera K calculated by the Ray-Casting algorithm ($p(x, y)$ lies on a segment s of the length r_1 (r_2), so the start point of the s is the $C_K(x, y)$), determine the corresponding element (i, j) of the matrix M_v .

B) If $M_v(i, j) = 255$, increase the counter t_K by one ($t_K \leftarrow t_K + 1$), and update the element $M_v(i, j)$ to the number 100 or 150, if the camera K is a PTZ or an OD camera.

Output: Return the counter of changes t_K of camera K made in visibility matrix M_v .

Based on what has been said so far, it is not difficult to see that the described approach for correct placement or drawing of cameras on the canvas can also be parallelized, i.e., at the same time, it is possible to draw the pixels of several cameras on the canvas, that is, to update the content of the matrix M_v by blocks. After updating the content of the blocks of the matrix M_v that are attached to the cameras, it will be numerically recorded how many changes have been made per block, so that number will be used to divide by the total number of pixels inside the polygon beyond the obstacle pixels. Therefore, it is only necessary to count the number of camera pixels within the central polygon P that are not obstacle pixels and divide their number by the number of white pixels within the polygon P . In this way, the percentage coverage of the polygon is obtained. By counting changes within the blocks of the visibility matrix M_v , the total coverage of polygon P will be immediately gained.

3 Deterministic Approaches for Optimal Camera Placement

This part will detail three suboptimal deterministic methods for covering the interior area with holes. First, we examine the greedy algorithm for coloring triangle vertices of Delaunay triangles in three colors. After that, one color is selected to represent camera positions in a triangulated polygon. Algorithm 2 summarizes the steps of the deterministic algorithm based on the coloring of the Delaunay triangulation vertices (DACDT) to maximize the coverage of a simple polygon with obstacles.

Algorithm 2: Total polygon coverage via Delaunay triangles vertex coloring

Input: The central simple polygon P with holes (obstacles), which are represented in the form of simple polygons.

Step 1: Determine the total number of pixels t_P of the simple polygon P excluding the pixels of obstacles. Next, using constrained Delaunay triangulation, triangulate the polygon P into Delaunay triangles.

Step 2: Label all Delaunay triangles constructed in the previous step with three colors utilizing the 3-coloring algorithm.

Step 3: Mark those vertices of Delaunay triangles that are colored with the color with the lowest number of occurrences compared to the remaining two colors.

Step 4: In the marked vertices of the Delaunay triangles from the previous step (let that number be m), place the centers of the cameras K_i and adjust their orientations θ_i so that the directions of the cameras K_i ($i = \overline{1, m}$) coincide with the bisectors of the angles whose legs are coincident with the centers of the cameras.

Step 5: For each placed camera K_i ($i = \overline{1, m}$) in the previous step, using Algorithm 1, calculate pixel coverage, i.e., the number of changes t_{K_i} ($i = 1, \dots, m$).

Step 6: Calculate the percentage coverage of the polygon P as $cov(P) = \frac{1}{t_P} \sum_{i=1}^m t_{K_i}$.

Output: The percentage coverage $cov(P)$ of the central polygon P with holes.

The disadvantage of using the 3-coloring algorithm in the DACDT is that it exploits many cameras, especially if there are many Delaunay triangles with small surface areas. Since this approach requires many cameras, the proposed algorithm pre-determines the number of cameras that can mark a simple polygon P . So, instead of the usage of Delaunay triangles marked with the 3-coloring algorithm as done in the DACDT, we propose a new deterministic approach based on sorting surface areas of Delaunay triangles (DASADT) for covering a simple polygon P with obstacles. The main steps of the DASADT are presented by Algorithm 3. With this DASADT approach, after producing the Delaunay triangulation of the simple polygon P , these triangles are sorted in descending order due to their surface areas. After that, the obtained triangles are used to place the cameras inside them using Algorithm 1. The cameras are placed in triangles with the largest surface areas. It moves toward those with the smallest surface areas by placing PTZ or OD cameras according to the criteria specified in Algorithm 3. Without loss of generality, in the DASADT, the surface area A_{PTZ} of the PTZ camera is assumed to be larger than the surface area A_{OD} of the OD camera.

Algorithm 3: Overall polygon coverage using Delaunay triangle surface areas

Input: The main simple polygon P with holes (obstacles), which are represented in the form of simple polygons. A certain number of the PTZ and OD cameras.

Step 1: Determine the total number of pixels t_P of the simple polygon P excluding the pixels of obstacles and surface areas A_{OD} and A_{PTZ} of the OD and PTZ cameras, respectively.

Step 2: Using constrained Delaunay triangulation, triangulate the polygon P into Delaunay triangles and sort them in descending order, referring to their surface areas.

Step 3: For each Delaunay's triangle D_i and its surface area A_{D_i} **do**

a) **if** cameras number has reached **then break**.

b) **if** $A_{OD} < A_{D_i} < A_{PTZ}$ **then** place the camera OD in the center of the D_i .

c) **else if** $A_{PTZ} \leq A_{D_i} < 1.5A_{PTZ}$ **then** place the camera PTZ on a vertex of D_i .

d) **else if** $1.5A_{PTZ} \leq A_{D_i} < 2.5A_{PTZ}$ **then** place two PTZ cameras on the vertices of D_i .

e) **else if** $A_{D_i} \geq 2.5A_{PTZ}$ **then** place three PTZ cameras on the vertices of D_i .

Step 4: If not all cameras have been consumed and all Delaunay triangles have been processed, place the remaining cameras in those not previously covered triangles.

Output: The percentage coverage $cov(P)$ of the central polygon P with holes.

The DASADT approach can be further improved by taking in Step 3c) in Algorithm 3 the vertex of the triangle D_i , which gives the best coverage until then. A similar conclusion applies when placing two cameras, as done in Step 3d). The coverage can be further increased if, in Step 4) the cameras are carefully placed inside the uncovered triangles at those vertices and centers of the Delaunay triangles so that the overall coverage is the highest possible. This modification of the DASADT approach based on checking the positions of the vertices and centers of Delaunay triangles results in the new EASADT approach. The steps of the EASADT method are summarized in Algorithm 4.

Algorithm 4: Polygon coverage based on check positions of Delaunay triangles

Input: The central concave polygon P with holes, which are represented in the form of simple polygons. A certain number of the PTZ and OD cameras.

Step 1: Determine the total number of pixels t_P of the simple polygon P excluding the pixels of obstacles and surface areas A_{OD} and A_{PTZ} of the OD and PTZ cameras, respectively.

Step 2: Using constrained Delaunay triangulation, triangulate the polygon P into Delaunay triangles and sort them in descending order, referring to their surface areas.

Step 3: For each Delaunay's triangle D_i and its surface area A_{D_i} **do**

a) **if** cameras number has reached **then break**.

b) **if** $A_{OD} < A_{D_i} < A_{PTZ}$ **then** place the camera OD in the center of the D_i .

c) **else if** $A_{PTZ} \leq A_{D_i} < 1.5A_{PTZ}$ **then** place the camera PTZ in that vertex of the triangle D_i so that the total coverage is the largest.

d) **else if** $1.5A_{PTZ} \leq A_{D_i} < 2.5A_{PTZ}$ **then** place two PTZ cameras in those vertices of the triangle D_i which deliver the best coverage.

e) **else if** $A_{D_i} \geq 2.5A_{PTZ}$ **then** place three PTZ cameras on the vertices of D_i .

Step 4: If not all cameras have been consumed and non-processed Delaunay triangles exist, select those triangles with the most significant surface area and place the OD and PTZ cameras in them, choosing the positions (centers and vertices of the triangles) so that the total coverage is maximized.

Output: The percentage coverage $cov(P)$ of the central polygon P with holes.

In the next section, swarm intelligence algorithms will be proposed to solve the camera placement problem. This chapter introduces these algorithms inspired by searching for optimal camera positions in the enhanced version (EASADT) of the DASADT method, as described in steps of Algorithm 4.

4 Swarm Intelligence Algorithms for Optimal Camera Placement

This section will involve the Bat algorithm [62] as the most popular swarm intelligence technique to solve the Camera Placement Problem. Among various approaches, the Bat Algorithm has been particularly noted for its successful application to various problems. Developed by Xin-She Yang in 2010, it is inspired by bats' echolocation behavior. By searching databases in the period from 2011 to 2022, the authors of this paper found that the Bat algorithm has found application in more than 400

papers published by eminent publishers (journals and conferences), such as Elsevier, Springer, and IEEE, and journals and conferences that are searched through the Google Scholar database. Due to the massive volume of works, several areas in which the Bat algorithm has been applied will be briefly listed in this section, and a complete overview of the Bat algorithm with its variants, modifications, and hybridizations can be found in the article [54]. Several of the practical areas where the Bat algorithm has found valuable applications are Engineering Optimization [3, 5, 11, 32, 33], Image Processing [10, 48, 51, 58], Machine Learning and Artificial Intelligence [2, 7–9, 30, 56, 57], Bioinformatics [15, 21], Combinatorial Optimization [4, 25, 26, 47, 50], Robotics and Autonomous Systems [41, 61, 66], Telecommunications [27, 44, 53, 55], Financial Optimization [1, 14, 45], Health and Medicine [13, 39, 43, 49]. Because of its proven performance in solving various optimization problems, we have used the bat algorithm to address our camera placement challenge. We will not describe the basic structure of the Bat algorithm here due to the need for more space for writing. More details related to the Bat algorithm and other swarm intelligence algorithms can be found in the book [63].

4.1 *An Adjusted Discrete Bat Algorithm for Maximum Coverage*

This part will detail an adjusted version of the discrete bat algorithm (ADBA) for maximum coverage (MC) of a simple polygon with holes. Before delving into the detailed description of our proposed method for solving the camera placement problem, it is essential to note that the original Bat Algorithm (BA) was initially developed to solve continuous optimization problems. Consequently, the classic version of this metaheuristic cannot be directly applied to discrete problems such as the camera placement problem, which is the focus of our study. Therefore, certain modifications to the basic bat algorithm are necessary to address the problem. The first modification refers to the discrete representation of the velocity $\mathbf{v}_i$ of the i th agent (bat) $\mathbf{x}_i$, as a random number drawn from a uniform distribution of the closed interval $[0, d_H]$, where d_H is a Hamming distance between the i th agent $\mathbf{x}_i$ and the best agent $\mathbf{x}^*$. Since the discrete version for representing the velocity $\mathbf{v}_i$ of the i th agent (bat) $\mathbf{x}_i$ does not depend on the frequency f_i of the agent $\mathbf{x}_i$, which is used in the continuous version of the original Bat algorithm, it will not be used in this chapter. The second modification refers to moving the i th agent $\mathbf{x}_i^t$ from the current iteration t to the next iteration $t + 1$, i.e., how to determine the position of the i th agent $\mathbf{x}_i^t$ in the next iteration $t + 1$, i.e., find $\mathbf{x}_i^{t+1}$. For these purposes, similarly to the paper [47], the 2-opt operator is used, i.e. holds $\mathbf{x}_i^{t+1} \leftarrow 2\text{-opt}(\mathbf{x}_i^t, \mathbf{v}_i^t)$. In addition, the local solution is generated using 2-opt and 3-opt operators applied to the best bat found so far (in our simulations, one agent (bat) among the 20 best ones). The formulas for updating signal strength A_i and signal pulse r_i for the i th agent $\mathbf{x}_i$ remain the same as in the original Bat algorithm. The central problem for all swarm intelligence methods is designing an

agent for a particular problem. Before designing an arbitrary agent (bat), defining the search space for the discrete version of the bat algorithm is necessary. The search space in this chapter is represented by a vector $\mathbf{S}$ composed of numbers from 1 to m , where m is the total number of vertices of Delaunay triangles and their centers. So, $\mathbf{S} = [1 \ 2 \ \dots \ m]$. The latter implies that a constrained Delaunay triangulation is performed over the initial simple polygon P with obstacles. Then, the Delaunay triangles' vertices and corresponding centers are labeled with numbers, thus forming the search space $\mathbf{S}$. Since the number of cameras n_C will be passed as input to the bat algorithm, as well as some deterministic algorithms, it follows that the creation of its initial population is reduced to the design of an arbitrary agent in the form of a vector $\mathbf{x}$ composed of n_C components. Namely, the vector $\mathbf{x}$ is randomly created by selecting n_C numbers from the search space $\mathbf{S}$. As the numbers indicate the vertices or centers of Delaunay triangles, the placement of PTZ or OD cameras follows immediately based on their content. The placement of PTZ cameras is done analogously, as expressed in Step 4 of Algorithm 2. In contrast, OD cameras are placed in the centers of the Delaunay triangles if they are selected in the search space.

5 Analysis of Experimental Results

In this simulation experiment, we prioritized maximizing the coverage inside a polygonal area. We attempted to do this with as few cameras as possible. To illustrate these results, we have used two types of concave polygons with holes depicted in Fig. 3. Both polygons are placed inside the rectangle R with dimensions $680\text{px} \times 594\text{px}$. The polygon P has 54 vertices and 11 holes, while the polygon Q has 12 vertices and two holes. In Figure 3, the black area within polygons P and Q represents obstacles such as buildings; conversely, the white area represents the region that needs to be monitored. Figure 3 also contains constrained delaunay triangulations of these polygons (see Fig. 3b, d). As we can see in Fig. 3, the polygon P has a very complex geometric structure with 11 obstacles inside it, and because of that, it is very suitable for testing the algorithms we developed. In comparative analysis, all algorithms except ADBA are deterministic, while ADBA is a metaheuristic algorithm. Since the ADBA approach is a swarm intelligence technique, this paper assigns 50 agents (bats) and 500 iterations to implement it, totaling 25,000 functional executions. Also, for the proposed ADBA, the initial values pulse rates r_i^0 and loudnesses A_i^0 for each agent i were respectively set to 0.5 and 0.99, while the constants γ and α were set to 0.5 and 0.7, respectively. Since the ADBA is a metaheuristic algorithm, and the other competing algorithms are deterministic, it will be run in parallel in 10 runs utilizing multi-threads in the simulation, and its average results will be compared to the results of the other three algorithms. All mentioned algorithms have been implemented in the C# programming language in Visual Studio 2022. All algorithms are executed on machine AMD Ryzen 5 5600 X Core AM4 4.6GHz processor with 64GB DDR5 4800MHz RAM, MSI GeForce RTX 4070 Ti Gaming X Slim 12G graphics card, and Windows 11 $\times$ 64 operating system. The simulation outcomes of the compara-

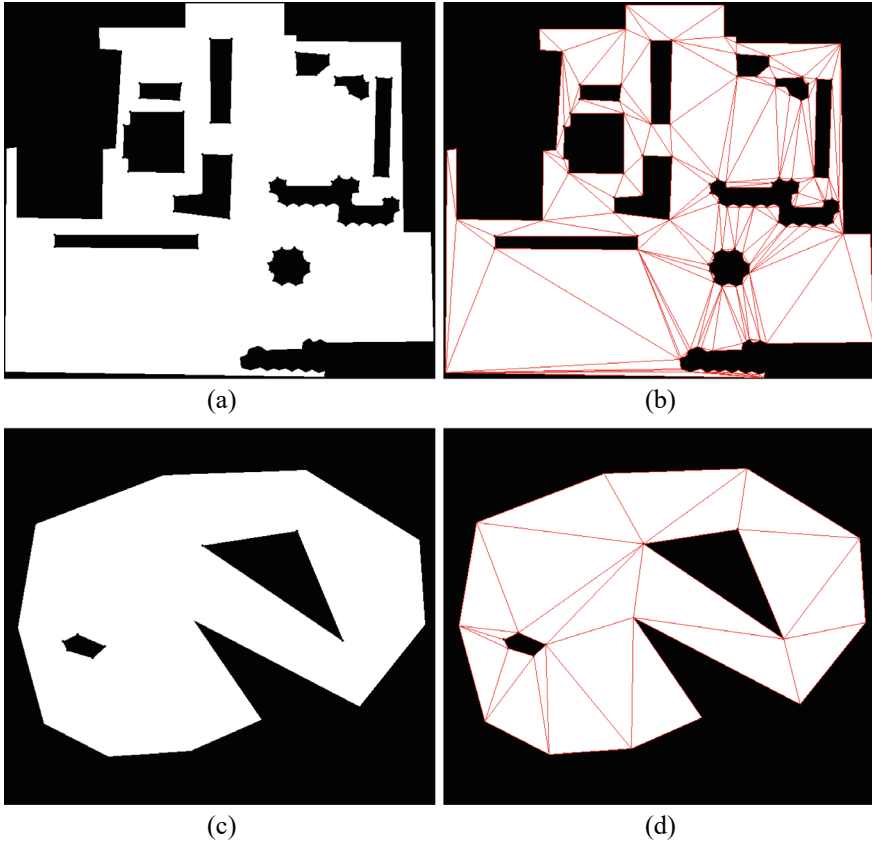


Fig. 3 **a** The polygon P is composed of 54 vertices and 11 holes. **b** Constrained Delaunay triangulation of the polygon P includes 177 triangles. **c** The polygon Q has 12 vertices and two obstacles. **d** Constrained Delaunay triangulation of the polygon Q counts 22 triangles

tive analysis between the mentioned algorithms are shown in Tables 1 and 2. For all algorithms, *Area Coverage*, *No. of cameras*, and *Time of execution* were evaluated, and the produced results are kept in Tables 1 and 2. Also, Figs. 4 and 5 show the visual coverage of the area inside the polygons P and Q achieved by the four proposed algorithms, respectively. In the implementation, the angle α is set to $\frac{\pi}{3}$ for a PTZ camera, and its radius r_1 is set to the value 160, while a radius r_2 of an OD camera is set to number 64. According to the outcomes of the simulation presented in Table 1, related to the seeking for the smallest number of cameras to cover the interior of the polygon P , it can be seen that the DACDT approach, whose operation is based on the placement of cameras in the colored vertices of Delaunay triangulation triangles, achieves coverage of 63.15% in a short time using 50 PTZ cameras, which is graphically shown in Fig. 4a.

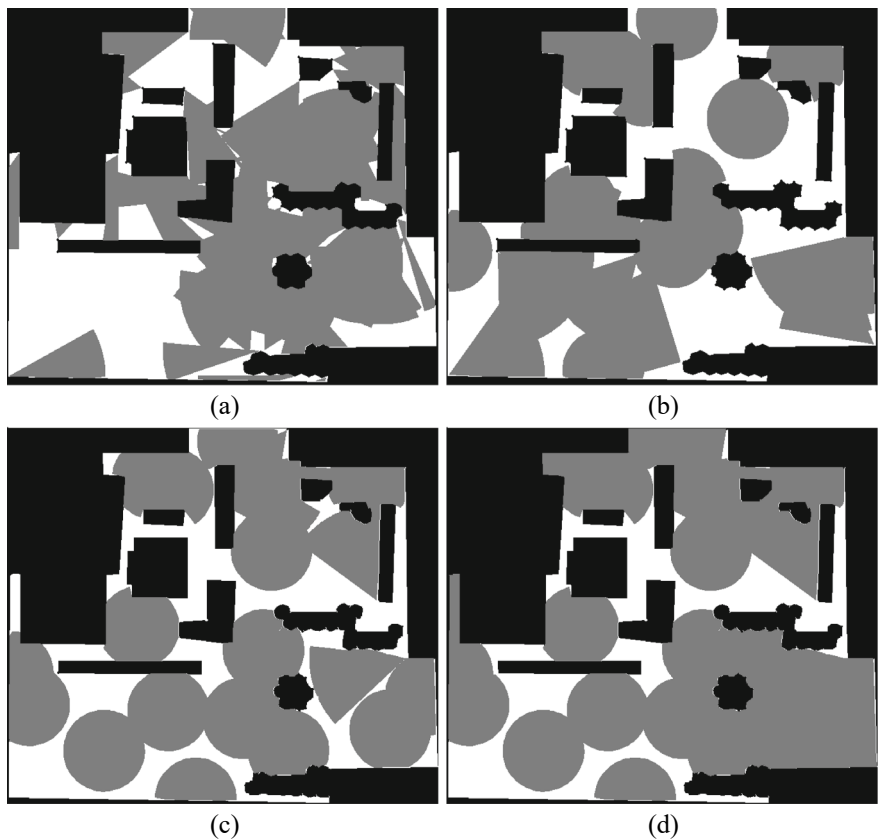


Fig. 4 Steps **a** to **d** graphically illustrate the placement of cameras in the interior of the polygon P performed by exploiting deterministic algorithms and ADBA, respectively

Table 1 The simulation results provided by proposed algorithms for a simple polygon P depicted in Fig. 3a

Algorithms	Area coverage	No. of cameras	Time (sec.)
<i>DACDT</i>	63.15	50	0.0468
<i>DASADT</i>	65.35	25	0.0161
EASADT	70.52	20	7.9378
ADBA	86.41	16	3.6184

The best-achieved results are marked in bold letters

Better coverage is achieved by the DASADT (see Fig. 4b), where this algorithm is superior in comparison to the previous algorithm by all metrics listed in Table 1. The EASADT method uses 20 cameras to realize polygon coverage of 70.52%, making it better than previous deterministic algorithms. Figure 4c shows a visualization of this approach’s coverage. Namely, the DASADT achieves a coverage of 65.35% using

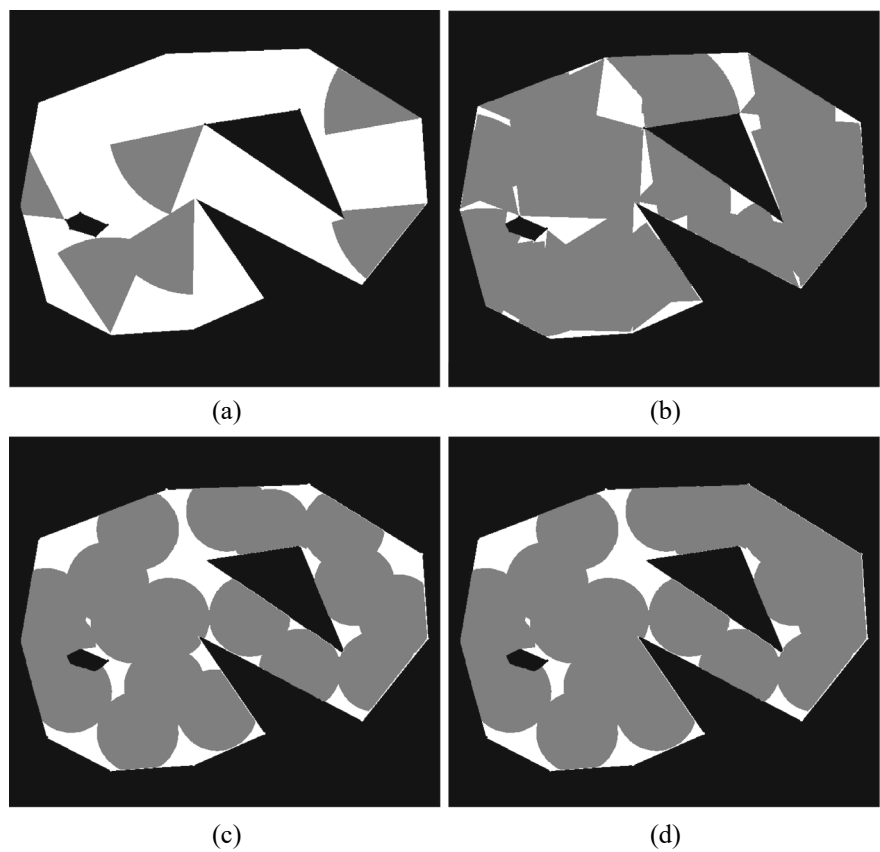


Fig. 5 Steps **a** to **d** visually demonstrate the placement of cameras inside polygon Q using algorithms 2, 3, 4, and ADBA, respectively

25 cameras, which is less compared to the EASADT. The best coverage is achieved with the ADBA approach, where using only 16 cameras, about 87% of the interior of polygon P is covered for only 3.6 s (see Fig. 4d). In addition to the polygon P , for testing the robustness of the mentioned algorithms, a noticeably simpler polygon Q with holes is used to observe the flaws and advantages of the developed algorithms (see Fig. 3c, d). The outcomes of testing the algorithms on this polygon are preserved in Table 2. Although the structure of the polygon Q is simple, the DACDT achieves area coverage of only 28.48% because after separating the polygon Q into Delaunay triangles, it is evident that only six colors are necessary to color the vertices of Delaunay triangles, which implies the usage of six cameras to cover the interior of Q (see Fig. 5a).

A drastically better effect was achieved by the DASADT, which managed to cover approximately 92% of the interior, using 36 cameras in a very short time, which can be seen in Fig. 5b. Although for a more complex environment (see the polygon P), the

Table 2 The simulation outcomes provided by proposed algorithms for a simple polygon Q depicted in Fig. 3c

Algorithms	Area coverage	No. of cameras	Time (sec.)
<i>DACDT</i>	28.48	6	0.0156
DASADT	91.16	36	0.0156
<i>EASADT</i>	84.63	20	5.9132
ADBA	95.76	16	2.1378

The best-achieved results are marked in bold letters

EASADT was superior to the DASADT, in the case of polygon Q , it gives somewhat weaker results in terms of coverage. However, it consumes a noticeably smaller number of cameras (see Fig. 5c). Finally, the ADBA method achieves an excellent coverage of almost 96% with only 16 cameras, making it superior to other methods. The optimal camera coverage of the polygon Q realized by the ADBA approach is shown in Fig. 5(d). Applying metaheuristics is evident as a potent and practical solution for tackling the camera placement issue. The optimal camera positioning is particularly significant in intelligent video surveillance, where these techniques play a crucial role in automatically pinpointing potential threats through the detection, localization, tracking, and identification of specific targets or events.

6 Conclusion and Future Work

Camera networks are sophisticated systems that capture comprehensive video data for intelligent processing tasks, such as target localization, identification, tracking, and more. This chapter delved into the formidable challenge of determining the optimal placement of cameras to ensure adequate coverage within interior areas characterized by simple concave polygons with obstacles. In response to this complex problem, four distinct algorithms were introduced and rigorously examined. The ADBA method as a swarm intelligence algorithm emerged as a standout solution through experimentation and analysis, offering exceptional overall performance in addressing the camera placement problem. Notably, the ADBA approach demonstrated remarkable robustness by effectively handling diverse instances from a broad spectrum of randomly generated scenarios. In future work, our research will extend to exploring authentic larger-scale instances, incorporating a more comprehensive array of metaheuristics, and investigating various polygon configurations to deepen our understanding and uncover further avenues for optimization and advancement.

References

1. Adarsh B, Raghunathan T, Jayabarathi T, Yang XS (2016) Economic dispatch using chaotic bat algorithm. *Energy* 96:666–675. <https://doi.org/10.1016/j.energy.2015.12.096>
2. Alihodzic A (2018) Training feed-forward neural networks employing improved bat algorithm for digital image compression. In: Lirkov I, Margenov S (eds) *Large-Scale Scientific Computing*. Springer International Publishing, Cham, pp 315–323
3. Alihodzic A (2022) Statistical Measurements of Metaheuristics for Solving Engineering Problems, pp. 1–26. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-82397-9_1
4. Alihodžić A, Chahin M, Čunjalo F (2022) New clustering techniques of node embeddings based on metaheuristic optimization algorithms. In: Lirkov I, Margenov S (eds) *Large-Scale Scientific Computing*. Springer International Publishing, Cham, pp 201–208
5. Alihodzic A, Delalic S, Gusic D (2020) An effective integrated metaheuristic algorithm for solving engineering problems. In: 2020 15th Conference on Computer Science and Information Systems (FedCSIS), pp. 207–214. <https://doi.org/10.15439/2020F81>
6. Alihodzic A, Delalic S, Hasic D (2020) An exact two-phase method for optimal camera placement in art gallery problem. In: 2020 15th Conference on Computer Science and Information Systems (FedCSIS), pp. 215–222. <https://doi.org/10.15439/2020F79>
7. Alihodzic A, Tuba E, Simian D, Tuba V, Tuba M (2018) Extreme learning machines for data classification tuning by improved bat algorithm. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. DOI <https://doi.org/10.1109/IJCNN.2018.8489546>
8. Alihodzic A, Tuba E, Tuba M (2017) An upgraded bat algorithm for tuning extreme learning machines for data classification. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '17*, p. 125–126. Association for Computing Machinery, New York, NY USA. <https://doi.org/10.1145/3067695.3076088>
9. Alihodzic A, Tuba E, Tuba M (2023) Optimal parameters selection of support vector machines using bat algorithm. In: 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), pp. 1–5. <https://doi.org/10.1109/ICECCME57830.2023.10252635>
10. Alihodzic A, Tuba M (2014) Improved bat algorithm applied to multilevel image thresholding. *Scientific World Journal* 2014(181767):1–16. <https://doi.org/10.1155/2014/176718>
11. Alihodzic A, Tuba M (2014) Improved hybridized bat algorithm for global numerical optimization. In: 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, pp. 57–62. <https://doi.org/10.1109/UKSim.2014.97>
12. Altahir AA, Asirvadani VS, Hamid NHB, Sebastian P, Saad NB, Ibrahim RB, Dass SC (2017) Optimizing visual surveillance sensor coverage using dynamic programming. *IEEE Sens J* 17(11):3398–3405
13. AApornak A, Raissi S, Keramati A, Khalili-Damghani K, (2021) Optimizing human resource cost of an emergency hospital using multi-objective bat algorithm. *Int J Healthc Manag* 14(3):873–879. <https://doi.org/10.1080/20479700.2019.1707415>
14. Arora A, Aggarwal K (2022) Detecting community structure in financial markets using the bat optimization algorithm. *Int J Inf Technol Proj Manag* 13(3):1–21. <https://doi.org/10.4018/IJITPM.313421>
15. Badr YA, Wassif KT, Othman M (2021) Automatic clustering of dna sequences with intelligent techniques. *IEEE Access* 9:140686–140699. <https://doi.org/10.1109/ACCESS.2021.3119560>
16. de Berg M, Cheong O, van Kreveld M, Overmars M (2008) *Computational geometry: algorithms and applications*, third edn. Springer, Berlin, Heidelberg. DOI <https://doi.org/10.1007/978-3-540-77974-2>
17. Bhuiyan MZA, Wang G, Cao J, Wu J (2014) Sensor placement with multiple objectives for structural health monitoring. *ACM Trans Sen Netw*. <https://doi.org/10.1145/2533669>
18. Bhuiyan MZA, Wang G, Cao J, Wu J (2015) Deploying wireless sensor networks with fault-tolerance for structural health monitoring. *IEEE Trans Comput* 64(2):382–395

19. Bjorling-Sachs I, Souvaine DL (1995) An efficient algorithm for guard placement in polygons with holes. *Discrete & Computational Geometry* 13:77–109. <https://doi.org/10.1007/BF02574029>
20. Bodor R, Drenner A, Schrater P, Papanikolopoulos N (2007) Optimal camera placement for automated surveillance tasks. *J Intell Rob Syst* 50(3):257–295. <https://doi.org/10.1007/s10846-007-9164-7>
21. Chowdhury A, Rakshit P, Konar A, Nagar AK (2014) A modified bat algorithm to predict protein-protein interaction network. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 1046–1053. <https://doi.org/10.1109/CEC.2014.6900518>
22. Chrysostomou D, Gasteratos A (2012) Optimum multi-camera arrangement using a bee colony algorithm. In: 2012 IEEE International Conference on Imaging Systems and Techniques Proceedings, pp. 387–392. <https://doi.org/10.1109/IST.2012.6295580>
23. Chvátal V (1975) A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory, Series B* 18(1):39–41. [https://doi.org/10.1016/0095-8956\(75\)90061-1](https://doi.org/10.1016/0095-8956(75)90061-1)
24. De Floriani L, Puppo E (1992) An on-line algorithm for constrained delaunay triangulation. *CVGIP: Graphical Models and Image Processing* 54(4):290–300. [https://doi.org/10.1016/1049-9652\(92\)90076-A](https://doi.org/10.1016/1049-9652(92)90076-A)
25. Delalić S, Alihodžić A, Tuba M, Selmanović E, Hasić D (2020) Discrete bat algorithm for event planning optimization. In: 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), pp. 1085–1090. <https://doi.org/10.23919/MIPRO48935.2020.9245276>
26. Delalić S, Žunić E, Alihodžić A, Selmanović E (2021) A discrete bat algorithm for the rich vehicle routing problem. In: 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), pp. 1058–1063. <https://doi.org/10.23919/MIPRO52101.2021.9597108>
27. dos Santos Coelho L, Askarzadeh A (2016) An enhanced bat algorithm approach for reducing electrical power consumption of air conditioning systems based on differential operator. *Applied Thermal Engineering* 99:834–840 <https://doi.org/10.1016/j.applthermaleng.2016.01.155>. URL <https://www.sciencedirect.com/science/article/pii/S1359431116301077>
28. Elnagar A, Lulu L (2005) An art gallery-based approach to autonomous robot motion planning in global environments. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2079–2084. <https://doi.org/10.1109/IROS.2005.1545170>
29. Feng G, Liu M, Guo X, Zhang J, Wang G (2011) Genetic algorithm based optimal placement of pir sensor arrays for human localization. In: 2011 IEEE International Conference on Mechatronics and Automation, pp. 1080–1084. <https://doi.org/10.1109/ICMA.2011.5985810>
30. Feng J, Kuang H, Zhang L (2022) Ebba: an enhanced binary bat algorithm integrated with chaos theory and lévy flight for feature selection. *Future Internet* 14(6). <https://doi.org/10.3390/fi14060178>
31. Fisk S (1978) A short proof of chvátal's watchman theorem. *J Comb Theory, Ser B* 24(3):374. [https://doi.org/10.1016/0095-8956\(78\)90059-X](https://doi.org/10.1016/0095-8956(78)90059-X)
32. Gandomi AH, Yang XS, Alavi AH, Talatahari S (2013) Bat algorithm for constrained optimization tasks. *Neural Comput Appl* 22:1239–1255. <https://doi.org/10.1007/s00521-012-1028-9>
33. Ghanem WAHM, Jantan A (2019) An enhanced bat algorithm with mutation operator for numerical optimization problems. *Neural Comput Appl* 31(1):617–651. <https://doi.org/10.1007/s00521-017-3021-9>
34. Gonzalez-Barbosa J, Garcia-Ramirez T, Salas J, Hurtado-Ramos J, Rico-Jimenez J (2009) Optimal camera placement for total coverage. In: 2009 IEEE International Conference on Robotics and Automation, pp. 844–848
35. Hoffmann F, Kaufmann M, Kriegel K (1991) The art gallery theorem for polygons with holes. In: [1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science, pp. 39–48. DOI <https://doi.org/10.1109/SFCS.1991.185346>
36. Islam SH, Vijayakumar P, Bhuiyan MZA, Amin R, Rajeev MV, Balusamy B (2018) A provably secure three-factor session initiation protocol for multimedia big data communications. *IEEE Int Things J* 5(5):3408–3418. <https://doi.org/10.1109/JIOT.2017.2739921>

37. Kamkar S, Ghezloo F, Moghaddam HA, Borji A, Lashgari R (2020) Multiple-target tracking in human and machine vision. *PLoS Comput Biol* 16(4):1–28. <https://doi.org/10.1371/journal.pcbi.1007698>
38. Katz MJ, Roisman GS (2008) On guarding the vertices of rectilinear domains. *Comput Geom* 39(3):219–228. <https://doi.org/10.1016/j.comgeo.2007.02.002>
39. Kora P, Krishna KSR (2016) Ecg based heart arrhythmia detection using wavelet coherence and bat algorithm. *Sens Imag* 17(1):1–16. <https://doi.org/10.1007/s11220-016-0136-5>
40. Lee D, Lin A (1986) Computational complexity of art gallery problems. *IEEE Trans Inf Theory* 32(2):276–282. <https://doi.org/10.1109/TIT.1986.1057165>
41. Li G, Xu H, Lin Y (2018) Application of bat algorithm based time optimal control in multi-robots formation reconfiguration. *J Bionic Eng* 15(1):126–138. <https://doi.org/10.1007/s42235-017-0010-8>
42. Liu J, Fookes C, Wark T, Sridharan S (2012) On the statistical determination of optimal camera configurations in large scale surveillance networks. In: *Computer Vision – ECCV 2012*, pp. 44–57. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33718-5_4
43. Lu S, Wang SH, Zhang YD (2021) Detection of abnormal brain in mri via improved alexnet and elm optimized by chaotic bat algorithm. *Neural Comput Appl* 33(17):10799–10811. <https://doi.org/10.1007/s00521-020-05082-4>
44. Nithya B, Jeyachidra J (2021) Optimized anchor based localization using bat optimization algorithm for heterogeneous wsn. In: *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICES)*, pp. 1–6. <https://doi.org/10.1109/ICES52305.2021.9633947>
45. Ochoa A, Margain L, Hernández A, Ponce J, De Luna A, Hernández A, Castillo O (2013) Bat algorithm to improve a financial trust forest. In: *2013 World Congress on Nature and Biologically Inspired Computing*, pp. 58–62. <https://doi.org/10.1109/NaBIC.2013.6617838>
46. O'Rourke J, Supowit K (1983) Some np-hard polygon decomposition problems. *IEEE Trans Inf Theory* 29(2):181–190. <https://doi.org/10.1109/TIT.1983.1056648>
47. Osaba E, Yang XS, Diaz F, Lopez-Garcia P, Carballedo R (2016) An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. *Eng Appl Artif Intell* 48:59–71. <https://doi.org/10.1016/j.engappai.2015.10.006>
48. Pourhadi A, Mahdavi-Nasab H (2020) A robust digital image watermarking scheme based on bat algorithm optimization and surf detector in swt domain. *Multimedia Tools Appl* 79(29–30):21653–21677. <https://doi.org/10.1007/s11042-020-08960-0>
49. Rauf HT, Gao J, Almadhor A, Arif M, Nafis MT (2021) Retracted article: enhanced bat algorithm for covid-19 short-term forecasting using optimized lstm. *Soft Comput* 17(1):12989–12999. <https://doi.org/10.1007/s00500-021-06075-8>
50. Rizk-Allah RM, Hassanien AE (2018) New binary bat algorithm for solving 0–1 knapsack problem. *Complex & Intelligent Systems* 4:31–53. <https://doi.org/10.1007/s40747-017-0050-z>
51. Satapathy SC, Raja NSM, Rajinikanth V, Ashour AS, Dey N (2018) Multi-level image thresholding using otsu and chaotic bat algorithm. *Neural Comput Appl* 29(2):1285–1307. <https://doi.org/10.1007/s00521-016-2645-5>
52. Schuchardt D, Hecker H (1995) Two np-hard art-gallery problems for ortho-polygons. *Math Log Q* 41(2):261–267. <https://doi.org/10.1002/malq.19950410212>
53. Senthilnath J, Kulkarni S, Benediktsson JA, Yang XS (2016) A novel approach for multispectral satellite image classification based on the bat algorithm. *IEEE Geosci Remote Sens Lett* 13(4):599–603. <https://doi.org/10.1109/LGRS.2016.2530724>
54. Shehab M, Abu-Hashem MA, Shambour MKY, Alsalihi AI, Alomari OA, Gupta JND, Alsoud AR, Abuhaija B, Abualigah L (2023) A comprehensive review of bat inspired algorithm: variants, applications, and hybridization. *Arch Comput Method Eng* 30(2):765–797. <https://doi.org/10.1007/s11831-022-09817-5>

55. Trivedi IN, Bhoje M, Jangir P, Parmar SA, Jangir N, Kumar A (2016) Voltage stability enhancement and voltage deviation minimization using bat optimization algorithm. In: 2016 3rd International Conference on Electrical Energy Systems (ICEES), pp. 112–116. <https://doi.org/10.1109/ICEES.2016.7510626>
56. Tuba E, Tuba M, Simian D (2016) Adjusted bat algorithm for tuning of support vector machine parameters. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 2225–2232. <https://doi.org/10.1109/CEC.2016.7744063>
57. Tuba M, Alihodzic A, Bacanin N (2015) Cuckoo Search and Bat Algorithm Applied to Training Feed-Forward Neural Networks, pp. 139–162. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-13826-8_8
58. Tuba M, Jordanski M, Arsic A (2016) Chapter 4 - improved weighted thresholded histogram equalization algorithm for digital image contrast enhancement using the bat algorithm. In: X.S. Yang, J.P. Papa (eds.) Bio-Inspired Computation and Applications in Image Processing, pp. 61–86. Academic Press. <https://doi.org/10.1016/B978-0-12-804536-7.00004-1>. URL <https://www.sciencedirect.com/science/article/pii/B9780128045367000041>
59. Wang T, Zeng J, Bhuiyan MZA, Chen Y, Cai Y, Tian H, Xie M (2018) Energy-efficient relay tracking with multiple mobile camera sensors. *Comput Netw* 133:130–140. <https://doi.org/10.1016/j.comnet.2018.01.002>
60. Wang X, Zhang H, Gu H (2020) Solving optimal camera placement problems in iot using lh-rpso. *IEEE Access* 8:40881–40891
61. Xie J, Zhou Y, Zheng H (2013) A hybrid metaheuristic for multiple runways aircraft landing problem based on bat algorithm. *J Appl Math* 2013:1–8. <https://doi.org/10.1155/2013/742653>
62. Yang XS (2010) A new metaheuristic bat-inspired algorithm, pp. 65–74. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-12538-6_6
63. Yang XS (2021) *Nature-Inspired Optimization Algorithms*. 2nd Edition. Academic Press
64. Yao Y, Chen C, Abidi B, Page D, Koschan A, Abidi M (2010) Can you see me now? sensor positioning for automated and persistent surveillance. *IEEE Trans Syst Man Cybern Part B (Cybernetics)* 40(1):101–115. <https://doi.org/10.1109/TSMCB.2009.2017507>
65. Zhao J, Cheung S, Nguyen T (2008) Optimal camera network configurations for visual tagging. *IEEE J Sel Top Sign Proc* 2(4):464–479
66. Zhou X, Gao F, Fang X, Lan Z (2021) Improved bat algorithm for uav path planning in three-dimensional space. *IEEE Access* 9:20100–20116. <https://doi.org/10.1109/ACCESS.2021.3054179>