Results in Engineering 21 (2024) 101859

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering

Shape and sizing optimisation of space truss structures using a new cooperative coevolutionary-based algorithm

Bahareh Etaati [a], Mehdi Neshat [b,c], Amin Abdollahi Dehkordi [d], Navid Salami Pargoo [e], Mohammed El-Abd [f], Ali Sadollah [g], Amir H. Gandomi [c,h,*]

[a] Heuristic and Evolutionary Algorithm Laboratory, University of Applied Sciences Upper Austria, Hagenberg, Austria
[b] Center for Artificial Intelligence Research and optimisation, Torrens University Australia, Brisbane, QLD, 4006, Australia
[c] Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, 2007, Australia
[d] Computer Engineering department, Najafabad Branch, Islamic Azad University, Najafabad, Iran
[e] Department of Civil, Chemical, Environmental and Materials Engineering, University of Bologna, Bologna, Italy
[f] College of Engineering and Applied Sciences, American University of Kuwait, Kuwait
[g] Department of Mechanical Engineering, University of Science and Culture, Tehran, Iran
[h] University Research and Innovation Center (EKIK), Obuda University, 1034 Budapest, Hungary

ARTICLE INFO

Keywords:
Real engineering problem
Truss optimisation
Optimal structural design
Bio-inspired optimisation algorithms
Cooperative coevolutionary algorithms
Greedy search

ABSTRACT

Optimising the shape and size of large-scale truss frames is challenging because there is a nonlinear interaction between cross-sectional and nodal coordinate forces of structures. Meanwhile, combining the shape and bar size variables creates a multi-modal search space with dynamic constraints, making an expensive optimisation engineering problem. Besides, most of the real truss problems are large-scale, and optimisation algorithms are faced with the issue of scalability by increasing the size of the problem. This paper proposed a novel Cooperative Coevolutionary marine predators algorithm combined with a greedy search (CCMPA-GS) for truss optimisation on shape and sizing. The proposed algorithm used the divide-and-conquer technique to optimise the shape and size separately. Therefore, in each iteration, the CCMPA-GS focuses on shape optimisation initially and then switches to the size of bars and tries to find the best cooperative combination of the solutions in the current population using a context vector (CV). A greedy search is embedded in the following to fix the remaining violations from the structure's stress and displacement. This novel alternative optimisation strategy (CCMPA-GS) compared with 13 established genetic, evolutionary, swarm, and memetic meta-heuristic optimisation algorithms. The comparison is based on optimising two large-scale truss structures consisting of 260-bar and 314-bar configurations. Experimental results demonstrate that the proposed CCMPA-GS method consistently outperforms the other meta-heuristic methods, delivering optimal designs for the 314-bar and 260-bar truss structures that are superior by 52 % and 63.4 %, respectively. This signifies a substantial enhancement in optimisation performance, highlighting the potential of CCMPA-GS as a powerful alternative in the field of structural optimisation.

1. Introduction

A truss structure embodies an architectural framework crafted from interconnected bars or members. These frameworks, commonly found in civil and mechanical engineering, serve to stabilise and bolster a variety of applications, including bridges, roofs, and towers. The hallmark of truss structures lies in their innate ability to deftly bear and distribute loads by capitalising on the inherent strength exuded by their triangular or polygonal formations [1]. The artistry of designing truss structures entails identifying the most optimal alignment for the bars as well as their dimensions, all while ensuring ample strength, rigidity, and stability while simultaneously minimising weight and cost. Engineers meticulously consider various factors, including material properties, geometric limitations, and the effects of loads, to fashion truss structures that withstand projected forces and retain structural integrity [2].

Truss structures boast a crucial advantage in their capacity to

* Corresponding author. Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, 2007, Australia.
E-mail addresses: bahareh.etaati@fh-hagenberg.at (B. Etaati), mehdi.neshat@uts.edu.au, mehdi.neshat@torrens.edu.au (M. Neshat), amin.abdollahi.dehkordi@gmail.com (A.A. Dehkordi), navid.salamipargoo@studio.unibo.it (N.S. Pargoo), melabd@auk.edu.kw (M. El-Abd), sadollah@usc.ac.ir (A. Sadollah), gandomi@uts.edu.au (A.H. Gandomi).

https://doi.org/10.1016/j.rineng.2024.101859
Received 5 September 2023; Received in revised form 10 January 2024; Accepted 28 January 2024
Available online 5 February 2024

disseminate and transfer loads efficiently. The triangular or polygonal configuration of bars in a truss serves to uniformly distribute forces, thereby curbing the presence of stress focal points and maximising structural stability. This unique attribute enables truss structures to achieve an exemplary strength-to-weight ratio, rendering them ideal for applications where weight reduction is of utmost importance [3]. In addition, truss structures offer unparalleled flexibility in both design and construction. These structures can be prefabricated and subsequently assembled on-site, facilitating efficient installation and minimising construction time. Furthermore, truss structures lend themselves to easy modification or expansion, allowing seamless adaptation to changing needs or load requirements [4]. Nevertheless, truss structures do present some inherent challenges. The design process entails meticulously analysing and forecasting the structure's behaviour under various loads and environmental conditions. Engineers must account for factors such as material selection, geometric constraints, and safety regulations to ensure that the truss structure adheres to the prescribed performance standards. In essence, truss structures are architectural frameworks that deliver stability and support to many applications. Their capacity for efficient load distribution, remarkable strength-to-weight ratio, and adaptability in design render them indispensable in civil and mechanical engineering endeavours. However, the design of truss structures necessitates a reasonable consideration of factors such as material properties, geometric limitations, and the impact of loads to ensure optimal performance and structural integrity [5]. Because truss optimisation under multiple frequency constraints has led to greater dynamic performance and less mass-producing cost, this field has attracted many experts over the past decades [6,7]. With this in mind, how to acquire the minimal structural weight regarding frequency constraints is the main issue discussed in these research studies. In this case, the optimisation of truss design variables plays a significant role in obtaining the minimal weight.

In a truss optimisation problem, truss sizing, shape, and topology are design variables. Considering this, there are three primary truss optimisation problems, including size optimisation, shape optimisation, and topology optimisation. Nevertheless, for a design variable, the optimal values depend on other design variables. For instance, the optimal truss shape depends on its size and topology and vice versa. Thus, the optimal structural weight is affected by more than one design variable. As a result, the simultaneous optimisation of truss shape and sizing variables under frequency constraints has attracted many scholars recently.

Coupling cross-sectional areas and nodal coordinate variables may cause such problems as mathematical difficulties, non-optimal solutions, and divergence problems. In addition, when considering design variables, frequency constraints are extremely non-linear, non-convex, and implicit [8]. Therefore, many researchers have proposed new optimisation algorithms to figure out truss sizing and shape optimisation with frequency constraints.

In recent years, various optimisation techniques have been developed to optimise truss problems [9,10]. Generally speaking, these techniques can be categorized into two groups, namely gradient-based and gradient-free optimisation methods, or meta-heuristics. While gradient-based methods are fast, they depend on the initial design variables to begin the process or are faced with premature convergence. In other words, they can be simply trapped in non-optimal areas if the starting points are not set appropriately. Furthermore, it requires complicated mathematical analysis to adjust proper derivatives of the objective function with respect to design variables and constraints, increasing the computational cost of the algorithm [6,11].

In comparison with gradient-based methods, meta-heuristics do not require any primary knowledge about the problem and are able to escape from local areas via their exploration and exploitation abilities. Although they do not usually guarantee to find the absolute global optimum, they can achieve desirable performance in a decent amount of time [8,12]. Additionally, in contrast to gradient-based methods, meta-heuristics are easy to implement, do not need complex mathematical analysis, and can support both discrete and continuous design variables.

For these reasons, meta-heuristics have attracted more attention in the field of truss optimisation in recent years. Gholizadeh et al. [13] suggested a modified sequential harmony search (SHS) algorithm to optimise structural size and shape with multiple natural frequencies. Lotfi et al. [14] proposed a hybrid Big Bang-Big Crunch method for optimising two-dimensional truss structures. Kaveh et al. [15] suggested a new hybrid particle swarm optimiser for truss sizing and shape optimisation with multiple frequency constraints. In order to enhance the efficiency of the particle swarm strategy, the authors embedded a ray optimiser and harmony search algorithm in the strategy.

In another study, Altay et al. [16] proposed a modified salp swarm algorithm (MSSA) for truss shape and sizing optimisation. The authors tried to address the convergence difficulties of the classic SSA by adjusting a parameter β in the movement of the leader's phase. Additionally, they modified the movement formula in the followers' phase to increase the exploration ability of SSA. The results showed that the MSSA outperformed SSA and remained competitive compared to the earlier results. Kaveh et al. [17] suggested an enhanced vibrating particles system (EVPS) for layout optimisation of truss structure problems. The proposed algorithm improved the performance of the VPS by incorporating a novel way to update the positions of agents. The algorithm was applied to four truss structure problems for optimisation of a variety of discrete and continuous design variables with constraints. In order to evaluate the effectiveness level of optimisation method with chaotic parameters, Kaveh et al. [18] applied three chaotic meta-heuristic algorithms, chaotic water evaporation optimisation (CWEO), chaotic tug-of-war optimisation (CTWO), and chaotic thermal exchange optimisation (CTEO) to optimise truss structure problems with frequency constraint. The authors have concluded that the combination of the chaos functions with meta-heuristics can improve.

In following the development of optimisation methods, Dede et al. [19] proposed a novel heuristic called teaching-learning-based optimisation (TLBO) was proposed for truss sizing and shape optimisation under multiple frequency constraints, such as allowable stress, displacement and the Euler buckling stress. A comparative study [20] consists of four popular optimisation methods: differential evolution (DE), improved differential evolution (IDE), LSHADE, and covariance matrix adaptation evolution strategy (CMAES) depicted that CMAES performed best in terms of convergence rate and quality of solutions proposed. Mortazavi et al. [21] proposed an integrated particle swarm optimiser to minimise the structural weight by simultaneous optimisation of truss size, shape, and topology. Also, in this study, an 'improved fly-back' strategy was suggested to deal with problem frequency constraints. Furthermore, Jawad et al. [22] proposed a new artificial bee colony algorithm (ABC) was proposed for combined optimisation of truss size and layout with multiple frequencies. The results indicated the efficiency of the proposed optimisation method with regard to the optimal weight and robustness. A higher-level version of TLBO was suggested by Farshchin et al. [23], a collaborative optimisation algorithm called school-based optimisation (SBO) was proposed for structural size and shape optimisation with frequency constraints. The proposed SBO enhanced the TLBO's abilities in terms of exploration and exploitation. In another effective study, Schwarz et al. [24] suggested sequential linear programming (SLP) problem formulation was proposed to simultaneously optimise truss shape and sizing under local buckling constraints. The proposed SLP included a post-processing phase to generate sub-solutions for the exact local buckling constraint boundary. A modified simulated annealing algorithm (MSAA) was developed for truss shape and size optimisation with frequency constraints. The suggested algorithm enhanced the simulated annealing algorithm's search abilities. One of the most effective heuristics to optimise various real engineering problems like truss structure is the semi-independent variable (SIV) method [25]. The SIV considers a collection of expected or preferred relationships among the decision variables to boost the process of search efficiency by cutting the search space and ignoring the infeasible solutions. The SIV could speed up the convergence rate considerably. Nguyen-Van et al. [6] proposed a new hybridization of differential evolution (DE) and symbiotic organisms search algorithm (SOS) named HDS for size and shape optimisation with frequency constraints. The proposed algorithm outshone other methods in terms of convergence speed and the problem solution quality. In the following, an improved version of SOS (MSOS) [26] was applied to boost the effectiveness of accuracy in the truss optimisation processes (exploitation and exploration) using proposing an adaptive benefit factor and modified parasitism vector.

In order to set the initial geometry parameters of the truss structure, Mandhyan et al. [27] proposed a predictor algorithm that is able to estimate the initial locations of joints, approximate cross-sectional areas and the connectivity of members. Recently, Tejani et al. [28] proposed the application of a modern multi-objective algorithm (Multi-objective heat transfer search algorithm) for optimising the conventional truss structures, which are discrete with two objectives, including truss weight minimisation and nodal displacement maximisation.

Solving truss structure problems using multi-objective insights have been extended during the last decade [29,30]. Most recently, Eid et al. [31] proposed a multi-objective water cycle algorithm (MOWCA) for optimising three truss problems. A spiral movement was incorporated into the proposed algorithm to enhance its local search capabilities. The results showed that the algorithm had an outstanding performance on various truss problems. Liu et al. [32] embedded a deep neural network is embedded into the genetic algorithm to improve its exploitation and exploration for truss sizing and layout optimisation. The algorithm performed best in terms of efficiency and applicability. Khodadi et al. [33] proposed a dynamic arithmetic optimisation algorithm (DAOA) for structural truss optimisation problems under natural frequencies. The proposed algorithm used two dynamic strategies in order to adjust exploitation and exploration efficiently. The results showed that the DAOA had excellent performance in terms of convergence speed and precision. In their other work [34], they applied generalized normal distribution optimisation (GNDO) to optimise structural problems. It presented that the GNDO performance was considered in terms of effectiveness and robustness. To deal with the high dimensionality of real engineering problems, recently, Gandomi et al. [35] proposed a concept-based technique (variable functioning Fx) consisting of PSO and various differential evolution algorithms regarding the relationships among variables and the effectiveness of the suggested optimisation method improved using Fx technique compared with an original idea.

Recently, a new technique for optimising the layout of truss structures was proposed [36] using isogeometric analysis (IGA) and stiffness spreading. The IGA-based stiffness spreading method simultaneously optimised truss elements' topology, geometry, and cross-sectional areas. The method employed energy conservation to obtain spreading stiffness matrices for truss elements embedded in weak IGA background grids. High-order continuous isogeometric elements were used to construct the IGA background grids, overcoming the discontinuous sensitivity limitation derived from traditional elements. The method could provide a continuous and smooth sensitivity field, making it easy to implement gradient-based optimisation algorithms. The effectiveness of the IGA-based stiffness spreading method was verified using illustrative examples, and the results show that the proposed method could provide a clear layout for different initial layouts. Associated with topology optimisation, an effective method proposed called quantile-based topology optimisation (QBTO) [37] as an alternative to reliability-based topology optimisation (RBTO) which faces issues with Monte Carlo simulation (MCS). The proposed QBTO method transforms the RBTO model into an equivalent quantile-based formulation, which addresses the problems with near-zero sensitivities of probabilistic constraint with respect to element densities and the computational cost of calculating sensitivities. In QBTO, the sensitivity calculation was limited to the sample corresponding to the quantile, which reduced the computational effort. Furthermore, a Kriging metamodel with a sequential update strategy was developed to efficiently calculate the quantile by evaluating the true constraint at fewer samples rather than all MCS samples. The effectiveness of the QBTO method was demonstrated through truss, beam, and bridge problems, which validated its high accuracy and efficiency.

One of the challenging aspects of large-scale structure optimisation is consuming considerable computational runtime. Lin et al. [38] proposed a parallel parameterised level set topology optimisation framework was proposed to handle this challenge with unstructured meshes that can handle arbitrary geometries and complex boundary conditions. The framework achieves full-scale optimisation through distributed memory parallel computing technology and utilizes unstructured meshes to handle complex geometries and boundary conditions. Several measures are employed to combine distributed memory parallel computing technology and parameterised level set topology optimisation with unstructured meshes. Firstly, shape functions in finite element analysis were used to parameterise the level set function. Secondly, a data structure called a directed acyclic graph was adopted to represent the unstructured mesh. Thirdly, passive domain and boundary conditions were imposed directly on the geometry entities of the structures. Finally, a multiple averaging filter was introduced to reduce tiny structural members in the optimised results for manufacturability. Several computing tests were presented to verify the framework's stability, efficiency, and scalability.

The marine predators algorithm (MPA) proposed by Faramarzi et al. [39] is a new and effective meta-heuristic technique inspired by marine predators' foraging behaviour. MPA's optimisation process contains three phases, imitating predator and prey behaviour through various velocity ratios. The algorithm adjusts exploration and exploitation depending on the phase where the optimisation takes place. Since 2020, many researchers have applied the MPA or its modified versions to several engineering problems, including forecasting COVID-19 positive cases [40], medical image segmentation [41], electrical modelling of photovoltaic (PV) [42–44], tackling the task scheduling in fog computing (TSFC) [45], feature selection [46], obtaining optimal reactive power dispatch (ORPD) problem [47], shape optimisation of developable ball surfaces [48], and product design [49] to name but a few. The research results showed that MPA had effective performance to solve engineering problems. Recently, Etaati et al. [50] conducted comparative research to apply 12 modern bio-inspired methods to two large-scale truss optimisation problems. The results showed that MPA outperformed other algorithms with regard to convergence rate and efficiency.

Among techniques applied to solve complex large-scale optimisation problems, decomposition-based, or divide-and-conquer methods have achieved widespread popularity in recent years [51–55]. In this method, a grouping strategy is used to split a large-dimensional search space into multiple smaller subspaces, each of which is optimised individually. Cooperative Coevolution (CC) was suggested by Potter and De Jong in 1994, and since then, many scholars have successfully applied the strategy to a wide range of real-world optimisation problems. Research on CC evolutionary algorithms (CCEA) has gained much attraction over the past two decades, evidenced by the continuous rise in annual publications and citations of Potter and De Jong's paper [56]. CCEA has four primary benefits over traditional EAs, mainly stemming from its divide-and-conquer decomposition technique [51]. Firstly, the problem decomposition enables parallelism, accelerating the optimisation process. Secondly, each subproblem is addressed with a distinct subpopulation, ensuring the maintenance of a diverse set of solutions [57]. Thirdly, breaking down a system into submodules not only enhances robustness against module errors and failures but also contributes to an overall increase in system robustness [58], consequently improving reusability in dynamic environments [59]. Lastly, proper problem decomposition can mitigate the" curse of dimensionality," addressing the decline in performance associated with increased decision variables [51]. This method is a decomposition-based technique and one of the subjects of interest in evolutionary computing to solve large-scale problems. In this approach, each subproblem is responsible for optimising particular components of a vector called Context Vector (CV) and evaluated in the context of the CV. In recent years, many CC algorithms have been developed for large-scale optimisation problems [52,55].

In this study, we integrated the CC strategy into the MPA to harness the advantages of both techniques. These results in three decomposition-based marine predator algorithms are proposed for the optimisation of two 260-bar and 314-bar truss problems on shape and sizing with multiple natural frequencies. All three proposed algorithms optimise structural shape and size independently. The first algorithm is called an Improved Marine Predators Algorithm (IMPA). In IMPA, the size and shaping are optimised separately for each individual. It helps the algorithm explore each decision variable space effectively and thus increases population diversity. In the second proposed algorithm, we adopted the cooperative coevolution strategy in the MPA to divide the whole population into several sub-populations with smaller dimensions. This decomposition-based method assists diversity maintenance in each subpopulation and global search in the entire population, preventing premature convergence [57]. Furthermore, the decision variables are assigned to different numbers of groups to find a more efficient group size to put more interactive variables in the same group. Thus, the algorithm is less likely to be trapped in a local optimum, leading to better exploitation.

The proposed framework is named Cooperative Coevolutionary Marine Predators Algorithm (CCMPA). The final algorithm is a combination of the CCMPA and a greedy search to boost the CCMPA exploitation abilities and convergence speed. In the proposed frameworks, the subpopulation of *n* sizing variables is grouped into *m* smaller cross-sectional areas with the same size, each of which is optimised independently. Therefore, instead of having *n* sizing variables for optimisation, there are only *m* decision variables where $m < n$. Moreover, a complete design vector (CV) is used to optimise each set of design variables independently. The CV holds the best-found configuration of the sizing and shape values. In each iteration, at first, the CCMPA optimises the subpopulation of shape variables, and then, it switches to optimise each subpopulation of sizing variables and tries to find the best cooperative combination of the solutions in the current population using the CV. This alternative optimisation strategy is compared with ten well-known meta-heuristics using two large-scale 260-bard and 314-bar truss structures.

To sum up, the primary contributions of this study are as follows.

1. Proposing two novel Cooperative Coevolutionary marine predators algorithms (IMPA and CCMPA) for largescale truss optimisation problems.
2. Combining the CCMPA with a greedy search to reduce the stress and displacement violations and improve the convergence rate.
3. Investigating to find the optimal number of clusters in improving the performance of the Cooperative Coevolutionary method.
4. Developing a comprehensive comparative truss optimisation framework, including ten modern meta-heuristic algorithms, Genetic algorithm (GA), particle swarm optimisation (PSO), Memetic algorithm (Shuffled Frog Leaping Algorithm, SFLA) [56] and three novel hybrid algorithms.

Our research endeavours represent the pioneering application of decomposition techniques to enhance the performance of the MPA (Modified Particle Algorithm). We introduce a novel approach by decomposing the sizing variables into distinct groups and evaluating each group independently. This decomposition strategy enables the algorithm to explore diverse regions within the sizing design variables more effectively. Notably, this approach proves advantageous when dealing with discrete design variables where finding the optimal vector becomes exceedingly challenging. By leveraging decomposition techniques, we aim to unlock new possibilities for optimising complex systems that involve discrete design variables, paving the way for improved algorithmic performance and enhanced search capabilities.

The paper's structure is as follows. Section 2 represents the formulation of two large-scale truss problems. Section 3 explains the basic MPA initially, and then it describes the proposed IMPA, CCMPA, and hybridization of the CCMPA with a greedy search in more detail. The numerical results are discussed in Section 4. Ultimately, Section 5 draws a conclusion from the research findings and results.

2. Truss problem formulation

The structural shape and sizing are formed through the optimisation process by finding the lightest structure, while the violation of stress and displacement of the truss structure is consequently minimized. The truss structural optimisation problem aims to identify optimal cross-sectional areas and nodal positions of the truss ground elements to attain minimum weight in the search domain as the objective function. Practicality, structural optimisation problems are subject to some structural constraints on compliance, including but not limited to element stress, joint deflection, critical buckling load, and natural frequencies. Furthermore, the objective function takes into account the nodal mass and the elemental mass.

Having checked the validity and kinematic stability of the structure, the general deterministic optimisation problem of truss structure can be formulated as the below objective function:

Find:

$$X = \{B_1, B_2, \ldots, B_m, A_1, A_2, \ldots, A_N, C_1, C_2, \ldots, C_M, \xi_1, \xi_2, \ldots, \xi_n\} \tag{1}$$

to minimise:

$$\begin{aligned} F(X) &= \sum_{i=1}^{m,N} B_i A_i \rho_i L_i + \sum_{j=1}^{n} b_j \\ \omega_j^{\min} &\le \omega_j \le \omega_j^{\max} \\ A_i^{\min} &\le A_i \le A_i^{\max} \\ \xi_i^{\min} &\le \xi_i \le \xi_i^{\max} \end{aligned}$$

where i and j are used to index truss elements and nodes, respectively. Hence, B_i is a binary element representing deletion or retention of the ith element (all $B_i = 1$ in this study), A and C are the cross-sectional area and nodal coordinate design variables, ξ_j is the positional value of the jth node, ρ_i is the mass density of the ith element, L_i is the length of the ith element and b_j is mass at the jth node. ω_j denotes the jth natural frequency constraint restricted to the lower and upper bounds ω_j^{min} and ω_j^{max}. Also, the cross-sectional area of each element and the position of each node are limited between the lower and upper bounds $[A_i^{min}, A_i^{max}]$ and $[\xi_i^{min}, \xi_i^{max}]$, respectively.

Additionally, there are four structural constraints that need to have complied. The stress constraints can be explained as:

$$|B_i \sigma_i| - \sigma_i^{max} \le 0 \tag{2}$$

where σ_i and σ_i^{max} are stress and maximum allowable stress in the ith element, respectively. Likewise, displacement constraints can be formulated as:

$$|\delta_j| - \delta_j^{max} \le 0 \tag{3}$$

where δ_j and δ_i^{max} represent the values of nodal displacement and maximum allowable nodal displacement, respectively. Another constraint that needs to be met is related to Euler buckling, which is expressed as:

$$|B_i \sigma_i^{comp}| - \sigma_i^{cr} \le 0; \sigma_i^{cr} = \frac{k_i A_i E_i}{L_i^2} \tag{4}$$

where σ_i^{comp} and σ_i^{cr} are compressive stress and critical buckling stress of the i th element, respectively while k_i, A_i, and E_i represent Euler's buckling coefficient, cross-section area, and Young modulus of elasticity of the i th element, respectively. Finally, the constraint relating to the first few natural frequencies is formulated as:

$$f_r^{min} \leq f_r \leq f_r^{max} \tag{5}$$

where f_r is rth natural frequency of the structure with f_r^{min} and f_r^{min} as its lower and upper bounds. A penalty function has been added to the objective function in 1 in order to transform the above-mentioned constrained optimisation problem into an unconstrained one. Therefore, the new objective function $F_{Penalty}(X)$ is recalculated as follows:

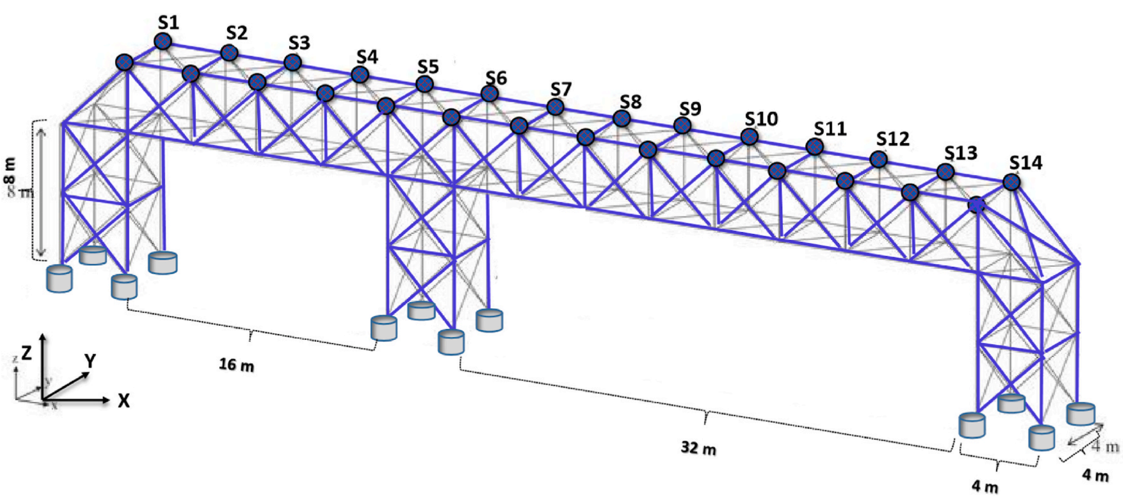
$$\text{to minimize}: F_{\text{Penalty}}(X) = F(X) + \Gamma \times PF \tag{6}$$

where: $\Gamma = \sum_{i=1}^{k} Vio_i$.

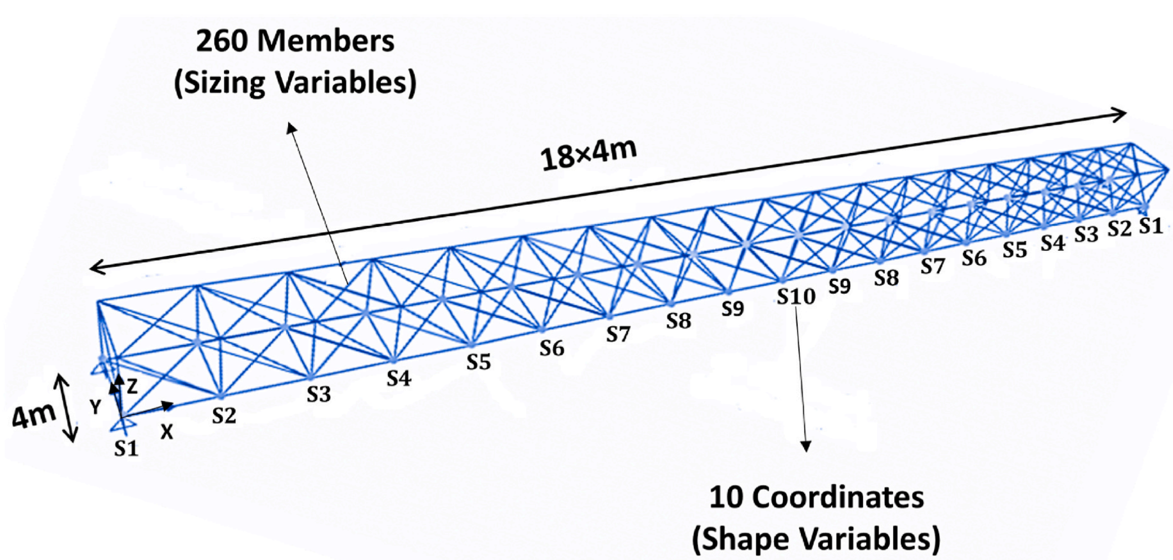
where Γ, PF, and Vio_i are the total value of constraint violations, penalty factor, and the amount of violation for the i th constraint, respectively. It is worth mentioning that the penalty factor is used to convert the constrained problem into an unconstrained one by penalising those design variables with a higher constraint violation. Here, the penalty factor is set to 1000.

2.1. Truss structure problem

In this section, we illustrate the technical details of the sizing and shape optimisation of the truss structure with fixed topology. Two case studies proposed by Bright Optimiser ISCSO 2018 and 2019 [60]. We have specifically chosen these two truss problems due to their unique characteristics and the limited amount of existing research [50,61] conducted on them. Both problems exhibit non-linearity, non-convexity, complexity, and multimodality, which distinguishes them from the commonly studied truss structures [13–15,21]. Furthermore, the design variables' discrete nature transforms the problems into NP-hard, introducing an additional layer of complexity compared to continuous variables [62]. Considering their role as essential components in bridges, these problems hold significant importance in the field of engineering optimisation. Moreover, the inherent heterogeneity in the parameters of these problems presents intriguing optimisation challenges, contributing to a more comprehensive exploration of this domain. Besides, the number of decision variables is, at most, the average count found in



(a)



(b)

Fig. 1. a) A landscape of the truss structure with 314-bar and 84-node. The number of shape variables is 14 (C), and 314 is the number of sizings (A) elements. C_i is shape variables, b) 260-bar truss structure with 76-node. The shape and sizing elements' variable numbers are 10 and 260, respectively.

traditional truss structure problems, posing a significant hurdle for the majority of optimisation methods. In these examples, all members have a unified density of the material, yield stress and elastic modulus equal to 7.85ton/m^3, 248.2MPa, and 200GPa, respectively. Since the design of these truss structures was conducted according to AISC-LRFD 1994, members are checked for the restriction states of tensile yielding and compressive buckling. The aim of the optimisation procedure is to discover design variable values that minimise structural weight while adhering to the constraints.

2.1.1. 314-Bar truss

The first case study is a three-dimensional steel truss structure consisting of 314 members and 84 nodes (Fig. 1 part a). The three independent load cases used to design this structure were similarly applied on all unsupported nodes and include two horizontal loads of 12 kN and 6 kN respectively in positive X and Y directions and a vertical load of 48 kN in the negative Z direction. All nodal displacements are capped at ± 50 mm. There are 328 design variables in total, with 314 sizing variables representing the cross-sections of the truss elements and 14 shape variables reflecting the z-coordinates of the top nodes of the structure. Each couple of upper nodes with a similar X-coordinate have identical heights. The sizing variable is chosen from a database of 37 pipe sections while the shape variable may take any integer value between $+9000$ mm and $+20000$ mm.

2.1.2. 260-Bar truss

A three-dimensional steel truss structure with 260 members and 76 nodes is the subject of the second case study (Fig. 1 part b). Three independent load cases applying on all unsupported nodes were considered to design the structure, which includes two horizontal loads of 5 kN and 1 kN, respectively, in positive X and Y directions and a vertical load of 5 kN in the negative Z direction. Furthermore, all nodal displacements in the X, Y, and Z directions are restricted to a maximum of ± 25 mm. There are 270 design variables in this problem, including 260 sizing variables representing the cross-sections of the truss members and 10 shape variables reflecting the z-coordinates of all bottom nodes of the structure. These bottom nodes are symmetrically grouped about the mid-span, with identical heights for each pair with the same X-coordinate. The sizing variable is selected from a database of 37 pipe sections, while the shape variable may take any integer value between -25000 mm to $+3500$ mm.

3. Methodology

Developing a well-established structure needs an iterative strategy using the experience and knowledge of designers to improve the structure's performance. The iterative strategies strive to evaluate a large number of experimental systems successively before proposing an optimal design. An optimal design should be cost-efficient, effective and more robust than other models. In truss optimisation proposing an optimal design result in saving the cost of material and construction considerably. Due to a high level of complexity in the truss optimisation problem, meta-heuristic optimisation algorithms [63] are widely used. This is mainly because these algorithms need no prior understanding of the technical context, such as search space characteristics or constraints, and follow just some heuristic and stochastic rules. This allows them to exploit and explore the search space more efficiently, being applicable for the optimisation of a wide range of problems, especially those optimisation challenges where detailed prior knowledge might be unavailable or impractical. The present study focuses on the shape and size optimisation of large-scale truss structures through an extensive number of state-of-the-art optimisation algorithms, which represent a high performance in real engineering problems [64]. Ten recently developed meta-heuristics (See Table 1) are systematically considered and compared in the first step. The optimisation methods' control parameters and population size are listed in Table 1.

In the following subsections, we initially give a brief explanation of other meta-heuristic algorithms listed in Table 1. Then, we explain the basic Marin Predators Algorithm (MPA) in more detail, and, consequently, we propose three MPA-based algorithms that enhance the efficiency of the basic MPA.

3.1. Applied meta-heuristic algorithms

Mirjalili et al. [71] proposed the Multi-Verse Optimiser (MVO) inspired by the parallel universes theory. It employs a population-based

Table 1
The setting details of meta-heuristic algorithms employed the truss shape and sizing problem. N_{pop} is the first population size.

	Abbreviation	full name	N_{pop}	Pre-defined Settings
1	MVO [65]	Multi Verse Optimiser	50	Wormhole existence probability maximum and minimum : $\text{WEP}_{\text{Max}} = 1, \text{WEP}_{\text{Min}} = 0.2,$ $\rho = 6.$
2	DA [60]	Dragonfly Algorithm	50	$w = 0.9 - 0.2, s = 0.1, a = 0.1,$ $c = 0.7, f = 1, e = 1.$
3	HGSO [61]	Henry Gas Solubility optimisation	50	$N_g = 5, l_1 = 0.05, l_2 = 100, l_3 = 0.01, \alpha = 1, \beta = 1, c_1 = 0.1, c_2 = 0.2$
4	AOA[63]	Arithmetic optimisation Algorithm	50	$MOP_{\text{Max}} = 1, MOP_{\text{Min}} = 0.2,$ $C_{\text{iter}} = 1, \alpha = 5, \mu = 0.499$
5	GNDO [63]	Generalized Normal Distribution	50	applied the default settings.
6	SSA [64]	Salp Swarm Algorithm	50	c_1 decreased from 2 to zero. $c_2 =$ rand and $c_3 =$ rand
7	MPA [39]	Marine Predators Algorithm	50	$p = 0.5, FAD = 0.2$
8	NNA [66]	Neural Network Algorithm	50	pre-defined settings
9	WCA [67]	Water Cycle Algorithm	50	$N_{sr} = 4, D_{\max} = 10^{-5}$
10	GTO [68]	Artificial Gorilla Troops Optimiser	50	$p = 0.03, \beta = 3, \omega = 0.8$
11	SFLA [56]	Shuffled frog-leaping algorithm	50	$\text{Memeplex}_{\text{pop}} = 10$, $\text{Memeplex}_{n} = 5, \alpha = 3, \beta = 5, \sigma = 1.2;$
12	PSO [69]	Particle Swarm Optimisation	50	$w = 1, w_{\text{damp}} = 0.99, c_1 = 2, c_2 = 2;$
13	GA [70]	Genetic Algorithm	50	$pc = 0.7, \gamma = 0.4, pm = 0.3, \mu = 0.1$, Roulette-Wheel selection;

approach where multiple solutions, representing different universes, evolve over iterations using quantum-inspired operators to explore and exploit the search space efficiently. Mirjalili et al. [67] innovated the Dragonfly Algorithm, a nature-inspired optimisation algorithm based on the collective behaviour of dragonflies. It mimics dragonfly swarms' swarming and hunting patterns to efficiently explore and converge on optimal solutions in complex search spaces. Hashim et al. [68] initiated Henry Gas Solubility Optimisation (HGSO) algorithm that mimics the behaviour governed by Henry's law to solve challenging optimisation problems. Abualigah et al. [69] proposed an Arithmetic Optimisation Algorithm (AOA) that utilizes the distribution behaviour of the main arithmetic operators in mathematics. Zhang et al. [70] introduced a Generalized Normal Distribution (GNDO) based on the Gaussian distribution model, where each individual makes use of this curve to improve their positions. Mirjalili et al. [72] suggested a Salp Swarm Algorithm inspired by the swarming behaviour of salps, marine invertebrates. SSA mimics the collective movement of salps in searching for optimal solutions by iteratively adjusting candidate solutions based on their fitness values. It combines exploration and exploitation strategies to efficiently navigate complex search spaces, making it applicable to various optimisation problems. Eskandar et al. [73] developed a Water Cycle Algorithm that mimics the flow of rivers and streams toward the sea and was derived by observing the water cycle process. Abdollahzadeh et al. [74] suggested an Artificial Gorilla Troops Optimiser (GTO) inspired by gorilla troops' social intelligence in nature. In this algorithm, gorillas' collective life is mathematically formulated, and new mechanisms are designed to perform exploration and exploitation. Eusuff et al. [65] implemented a Shuffled frogleaping algorithm (SFLA) combining the benefits of memetics with particle swarm optimisation. It has been used in various areas, especially engineering problems, owing to its easy implementation and limited variables. Kennedy et al. [75] presented a Particle Swarm Optimisation (PSO) inspired by the social behaviour of birds, where particles adjust their positions in a search space based on personal and global best solutions. Lastly, Holland et al. [76] proposed the well-known Genetic Algorithm (GA) that mimics natural selection processes, using crossover and mutation operations to evolve a population of potential solutions toward optimal ones.

3.2. Marine predators algorithm (MPA)

Faramarzi et al. [39] have just proposed the Marine Predator Method, a novel and effective meta-heuristic method. The algorithm simulates the behaviour of marine predators who pursue their prey using Brownian and Levy motion as their greatest foraging techniques. The predator's foraging strategy is separated into three stages based on the varying velocity rate between both the predator and the prey, with Brownian or Levy motion alternating in each stage. Whenever the velocity rate v among both predator and prey is negligible, less than 0.1, the best losing plan for a predator is Levy movement, irrespective of whether the prey is Levy or Brownian at the time. The algorithm's exploration phase is at this point. When v approaches 1, the predator's and prey's speeds are equivalent, and if the prey moves in a Levy step, the predator moves in a Brownian step. Where the velocity rate $v > 10$, the predator's best hunting approach is to stay still, irrespective of the step size of the prey. The following is a more in-depth explanation of MPA.

3.2.1. Initialization

The algorithm's first stage is to initialise the population. In most population-based meta-heuristic algorithms, this phase spreads the initial answer uniformly across the search space, and MPA is no exception. The following is the initialization formula:

$$\vec{X}_0 = \vec{X}_{min} + \vec{r}\left(\vec{X}_{max} - \vec{X}_{min}\right) \tag{7}$$

where $\vec{r}$ notifies a vector with random values between 0 to 1, $\vec{X}_{\min}$ and $\vec{X}_{\max}$ show the lower and upper boundaries, and the initial solution shows by $\vec{X}_0$.

According to the fittest principle of survival, the top predator is related to the most powerful individual in a species. The top predators are being employed to build an Elite matrix in which each array delivers present prey position details to each predator within the next hunting period.

$$\text{Elite} = \begin{bmatrix} X^I_{1,1} & X^I_{1,2} & \dots & X^I_{1,d} \\ X^I_{2,1} & X^I_{2,2} & \dots & X^I_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ X^I_{n,1} & X^I_{n,2} & \dots & X^I_{n,d} \end{bmatrix}_{n\times d} \tag{8}$$

where n denotes the size of the initial population and d denotes the variable's dimension, the best predator vector, X^I symbolizes the best predator and is replicated multiple times to build the matrix of *Elite*. If a predator with superior predation potential emerges throughout the iteration, it will replace the top predator, and the *Elite* matrix will be modified.

The second matrix is related to the *Prey* matrix indeed, which has a similar dimension as the matrix of *Elite* and is used by the predator to keep updating its position.

$$\text{Prey} = \begin{bmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,d} \\ X_{2,1} & X_{2,2} & \dots & X_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1} & .X_{n,2} & \dots & X_{n,d} \end{bmatrix}_{n\times d} \tag{9}$$

where $X_{i,j}$ indicates the jth dimension of the ith prey. The optimisation of MPA relies primarily upon these two equations.

3.2.2. Optimisation process

The MPA optimisation mechanism can also be separated into three phases depending on the predator and prey velocity ratios, including maximum velocity, unit velocity, and minimum velocity rate. The following is a synopsis of these three stages:

Phase1: In the case of a large velocity rate ($v > 10$), the predator proceeds slower than the prey, and the predator's optimum hunting technique is just not to move in whatsoever. This stage is in charge of algorithm exploration, and it is mathematically formulated as follows:

$$\begin{gathered} \text{Iter} < \frac{1}{3}\,\text{Max\_Iter} \\ \overrightarrow{\text{stepsize}}_i = \vec{R}_B \otimes \left(\overrightarrow{\text{Elite}}_i - \vec{R}_B \otimes \overrightarrow{\text{Pray}}_i\right) \\ \overrightarrow{\text{Pray}}_{ii} = \overrightarrow{\text{Pray}}_i + \text{P.R} \otimes \overrightarrow{\text{stepsize}}_i \end{gathered} \tag{10}$$

where *Iter* represents the current generation number and *Max-Iter* represents the maximum generation number, Brownian motion is represented as $\vec{R}_B$, a vector carrying a sequence of randomly generated numbers depending on a normal distribution. $\otimes$ symbolizes entry-wise multiplications, and $\overrightarrow{\text{stepsize}}_i$ is the step size vector that indicates the i th predator's next step. $\text{P} = 0.5$ is a fixed value. $\vec{R}$ is a vector of evenly

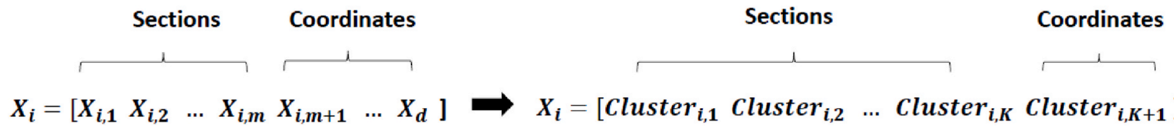


Fig. 2. The *i*th individual's structure in the IMPA.

produced random numbers ranging from 0 to 1.

Phase2: The prey and predator proceed simultaneously at the same velocity in the unit velocity ratio, and exploration progressively shifts to exploitation. The following is the formula:

$$\text{while } \frac{1}{3}\ \text{Max\_Iter} < \text{Iter} < \frac{2}{3}\ \text{Max\_Iter} \tag{11}$$

In the case of the first half of the population (exploration),

$$\overrightarrow{\text{stepsize}}_i = \overrightarrow{R}_L \otimes \left(\overrightarrow{\text{Elite}}_{i\_} \overrightarrow{R}_L \otimes \overrightarrow{\text{Pray}}_i \right) \longrightarrow \overrightarrow{\text{Pray}}_i = \overrightarrow{\text{Pray}}_{i\text{P.R}} \otimes \overrightarrow{\text{stepsize}}_i \tag{12}$$

In the case of the second half of the population (exploitation),

$$\overrightarrow{\text{stepsize}}_i = \overrightarrow{R}_B \otimes \left(\overrightarrow{R}_B \otimes \overrightarrow{\text{Elite}}_{i\_} \overrightarrow{\text{Prey}}_i \right) \longrightarrow \overrightarrow{\text{Pray}}_i = \overrightarrow{\text{Elite}}_{ii_i} + \text{P.CF} \otimes \overrightarrow{\text{stepsize}}_i \tag{13}$$

where $R{\rightarrow}_L$ is a Levy distribution-based random number vector that reflects Levy movement. *CF* is an adaptive coefficient that determines the predator's step size, and it is formulated as described in the following:

$$\text{CF} = \left(1 - \frac{\text{Iter}}{\text{Max\_Iter}}\right)^{\left(2\frac{\text{Iter}}{\text{Max Iter}}\right)} \tag{14}$$

In this step, the initial half of the population takes a Levy step (exploitation), while the second half takes a Brownian step (exploration). The transformation from exploration to exploitation using a combination of Levy and Brownian techniques.

Phase3: where the low-velocity rate is ($v = 0.1$), the predator outpaces the prey, and exploration gives way to exploitation. A predator's ideal hunting approach is a Levy movement, and the formula expression can be seen in the following:

While Iter $> \frac{2}{3}$ Max_iter (15)

$$\overrightarrow{\text{stepsize}}_i = \overrightarrow{R}_L \otimes \left(\overrightarrow{R}_L \otimes \overrightarrow{\text{Elite}}_{i\_} \overrightarrow{\text{Pray}}_i \right) \tag{16}$$

$$\overrightarrow{\text{Pray}}_i = \overrightarrow{\text{Elite}}_{ii} + \text{P.CF} \otimes \overrightarrow{\text{stepsize}}_i \tag{17}$$

The fish-gathering devices (FADs) in nature can indeed shape the behaviour of marine predators. FADs' impacts are regarded as local optimum traps in MPA, and their formula is mathematically developed as described in the following:

$$\overrightarrow{Pray}_i = \begin{cases} \overrightarrow{\text{Pray}}_i + \text{CF}\left[\overrightarrow{X}_{\min} + R \otimes \left(\overrightarrow{X}_{\max} - \overrightarrow{X}_{\min}\right)\right] \otimes \overrightarrow{U} & \text{if } r < \text{FADs} \\ \overrightarrow{Pray}_i + \left[\text{FADs}\,(1-r) + r\right]\left(\overrightarrow{Pray}_{r_1} - \overrightarrow{Pray}_{r_2}\right) & \text{if } r > \text{FADs} \end{cases} \tag{18}$$

where *FADS* is equal to 0.2 and indicates the likelihood of FADS influencing the optimisation phase. $\overrightarrow{X}_{\min}$ and $\overrightarrow{X}_{\max}$ are vectors that define the dimension's lower and upper boundaries. *r* is a random number vector produced in the range of 0 and 1. $\overrightarrow{U}$ is a vector of binary values, with its array assigned to 0 when *r* is greater than 0.2 and to 1 when *r* is less than 0.2. $\overrightarrow{\text{Prey}}_i$ and $\overrightarrow{\text{Prey}}_{r2}$ are two randomly generated variables of the prey matrix. $\overrightarrow{R}$ is a random number vector containing produced numbers ranging from 0 to 1.

r is less than 0.2. $Prey_i$ and $Prey_r2$ are two randomly generated variables of the prey matrix. $R^{\rightarrow}$ is a random number vector containing produced numbers ranging from 0 to 1.

Marine predators also have remarkable memory that enables them to recall the place of each successful predation. This is accomplished by storing the data in MPA. Each answer is assessed to the existing optimal solution every iteration, and if a superior solution arises, it is swapped with the actual best solution. MPA showed in several studies a competitive performance for optimising large-scale optimisation problems [61,77].

3.3. Improved marine predators algorithm (IMPA)

In this section, an improved marine predators algorithm named IMPA is proposed for the optimisation of shape and sizing variables of the large-scale truss structures with frequency constraints discussed in Section 2. The main idea of the IMPA is to evaluate cross-sectional areas and nodal coordinates independently. First, we decompose the entire n-dimensional design vector into two parts, namely sizing variables and shape variables, respectively. Second, we split the sizing variables into groups. Thus, each design vector has multiple smaller groups, including several sizing variable vectors and a shape variable vector as the last group. Then, we evaluate each group separately. The proposed algorithm is described in more detail in the following subsections.

3.3.1. Divide-and-conquer technique

Solving a large-scale optimisation problem is difficult because of the high dimensionality of the landscape. Indeed, the efficiency of an optimiser reduces dramatically as the dimensionality of the landscape increases. Because of this, many well-known state-of-the-art optimisation algorithms have not been able to do well on large-scale landscapes unless they have been modified to suit these problems. As a result, in recent years, new algorithms have been proposed to solve high-dimensional problems. One idea is a grouping strategy in which the large-scale problem is divided into multiple smaller sub-problems, which are solved separately in order to solve the entire problem. Firstly, suggested by Potter et al. [56], this idea called Cooperative Co-evolution (CC) was incorporated into the basic GA. The algorithm partitioned an n-dimensional individual into n one-dimensional sub-components and evaluated each component using a complete solution vector named context vector (*CV*). Thus, instead of having *n* variables, we have a smaller number of decision variables for optimisation. This resulted in a considerable improvement in the GA performance. Since then, many researchers have applied this idea to a variety of optimisation algorithms, including PSO [53,78,79], GA [80,81], DE [82,83], ABC [84], and AIS [85]. Recently, Cooperative Co-evolution (CC) techniques have been used in various engineering applications such as renewable energy optimisations [86,87], large water distribution networks [88], large-scale supply chain network design [89], etc.

In this paper, we incorporate this cooperative strategy of CC into the evaluation stage of the basic *MPA* for evaluating individuals. Algorithm 1 shows the framework of IMPA. The proposed IMPA is different from the basic *MPA* in the following ways. First, the algorithm partitions all individuals in the *Prey* matrix into two vectors, namely cross-sectional areas and nodal coordinates. Second, it decomposes the cross-sectional areas into multiple same-sized smaller areas named clusters, or groups. Fig. 2 shows *i*th individual's (X_i) structure in more detail. Third,

Algorithm 1 Improved Marine Predators Algorithm

```
FADs ← 0.2
P ← 0.5
Initialize a randomly generated design vector CV
Evaluate CV W(CV)
Initialize the population Prey of members X_i   i ∈ {1, 2, ..., n} according to Eq.(7) and Eq.(9)
Sections ← X_{1...m}
Coordinates ← X_{m+1...n}
for each X_i in Prey do
    Divide X_i into groups   X_i = {Section_1, Section_2, ..., Section_k, Coordinates}
end for
repeat
    for each X_i in Prey do
        for each cluster Cluster_j do
            Execute Algorithm 2
        end for
    end for
    Construct Elite matrix from CV according to Eq.(8)
    Calculate CF according to Eq.(14)
    for each X_i in Prey do
        if Iter < (1/3)Max_Iter then
            Update Prey according to equations Eq.(10)
        else if (1/3)Max_Iter < Iter < (2/3)Max_Iter then
            if i ≤ n/2 then
                Update Prey according to Eq.(12)
            else
                Update Prey according to Eq.(13)
            end if
        else
            Update Prey according to Eq.(15)
        end if
    end for
    for each X_i in Prey do
        for each cluster Cluster_j do
            Execute Algorithm 2
        end for
    end for
    Apply marine memory saving and update the best solution
    Update Prey using FADs according to Eq.(18)
    Apply marine memory saving and update the best solution
until stop criterion is not met
```

in each iteration, in the evaluation stage, for each individual, the algorithm evaluates each part of the sizing variables in the context of a complete design vector so-called *CV* at the first step. Then, it evaluates the last part of the individual, including shape variables. It is worth mentioning that, similar to cooperative co-evolution, at each step, if the current *CV* is better than the previous one, it replaces the previous CV. Otherwise, the algorithm dismisses the current *CV*. By decomposing the sizing variables into different groups and evaluating each one separately, the algorithm is better able to search various areas within the sizing design variables. It can especially be more effective when the design variables are discrete and finding the optimal vector is too difficult. Here, CV maintains the best cooperative combination of sizing and shape variables found so far. The evaluation process of IMPA is shown in Algorithm 2.

.

Additionally, when memory saving, IMPA considers the sizing and shape variables of each individual independently. It depends on whether the current variable vectors are better than the previous ones. For each individual, IMPA saves the best partitions. Thus, an individual can be

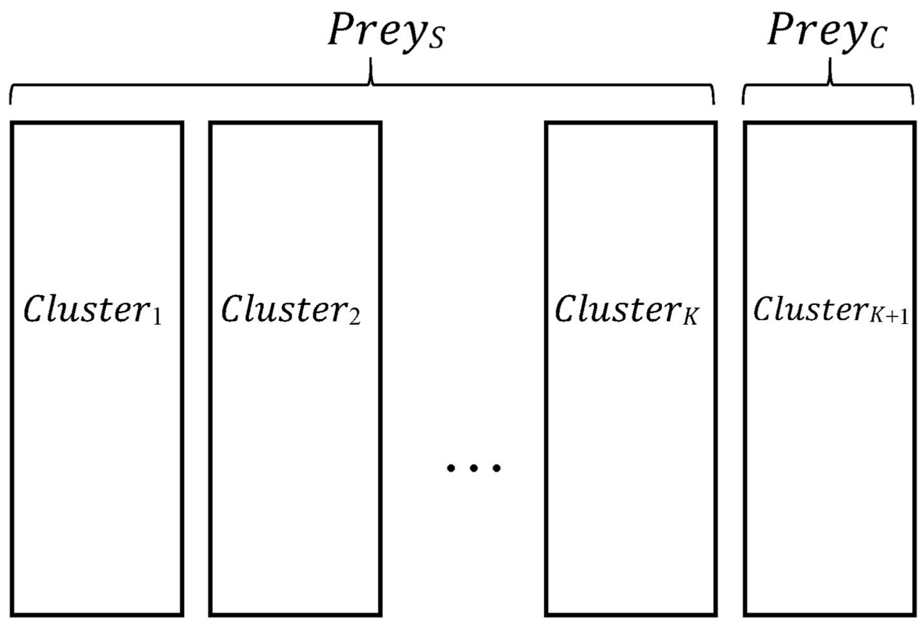


Fig. 3. The structure of the CCMPA.

modified in three ways its sizing variable vector, shape variable vector, or both can be modified.

Algorithm 2 Evaluation

$CV\_tmp \leftarrow CV$
$CV\_tmp_{Lb...Ub} \leftarrow X_{Lb...Ub}$
Calculate the weight of CV_tmp $W(CV\_tmp)$
if $W(CV\_tmp) < W(CV)$ **then**
    $CV \leftarrow CV\_tmp$
end if

3.4. Cooperative Co-evolutionary marine predators algorithm (CCMPA)

In this section, we propose a cooperative co-evolutionary MPA named CCMPA. Like the IMPA, the new proposed algorithm evaluates the sizing and shape variables independently. However, in the CCMPA, we incorporate the CC concept into the whole structure of the basic MPA. Indeed, while the IMPA applies the decomposition-based method to each individual only during the evaluation process, the CCMPA's framework is completely based on the cooperative co-evolutionary idea. While CC strategies maintain good solution diversity in sub-populations and thus prevent premature convergence [51,57], they are prone to being trapped in pseudo-minima, local minima created by inappropriate division of the search space [78]. With this in mind, in this paper, the objective is to combine the CC

and MPA exploration and exploitation abilities to create a more robust and powerful algorithm, as we explained how the basic MPA can create a good balance between exploration and exploitation through its three phases. Thus, this hybridisation can lead to better exploitation of CC and avoidance of premature convergence [90]. For this purpose, the whole population is divided into various sub-populations, each optimised independently.

The objective is to help MPA overcome its exploration and exploitation incapabilities by dividing the whole population into various sub-populations and optimising each sub-population independently. Additionally, this strategy helps MPA to escape from local optima, prevent premature convergence, and increase its global search abilities. As a result, in the CCMPA, we have two main populations, including a $l-$ *dimensional* population of c nodal coordinate individuals $Prey_C$ and a m $-$ *dimensional* population of section individuals $Preys_S$ $(c + s = n)$. Then, we split the $Preys_S$ population into several clusters of the same size. Fig. 3 shows the structure of the CCMPA in more detail. Also, because the population of coordinates has a lower number of dimensions, we evaluate and optimise $Prey_C$ before the section population in the CCMPA.

In this case, we assume that the complexity of the $Prey_C$ is much smaller than that of $Preys_S$ due to their lowdimensional space. Thus, by the early optimisation of coordinates, at each iteration, we can put aside a lot of fitness evaluations for the optimisation of sections, which can lead to better results. Algorithm 3 shows the pseudo-code of the CCMPA. First, the populations of sections and coordinates and the CV are initialized. Second, the population of sections is decomposed into several subpopulations or clusters. Thus, the procedure of clustering in the CCMPA is as follows.

Algorithm 3 Cooperative Coevolutionary Marine Predators Algorithm

$FADs \leftarrow 0.2$
$P \leftarrow 0.5$
Initialize a randomly generated design vector CV
Evaluate CV $W(CV)$
Initialize population $Prey_C$ of nodal coordinates C_j $\quad j \in \{m+1, m+2, \ldots, n\}$ according to Eq.(7) and Eq.(9)
Initialize population $Prey_S$ of cross-sectional areas S_i $\quad i \in \{1, 2, \ldots, m\}$ according to Eq.(7) and Eq.(9)
Divide population $Prey_S$ into K groups $\quad Prey = \{Prey_C, Group_1, Group_2, \ldots, Group_K\}$
repeat
  for each group $Group_i$ in $Prey$ **do**
    for each member X_j in $Group_i$ **do**
      Execute a generation of Algorithm 1
    end for
  end for
until stop criterion is not met

In the CCMPA, the procedure of grouping is as follows.

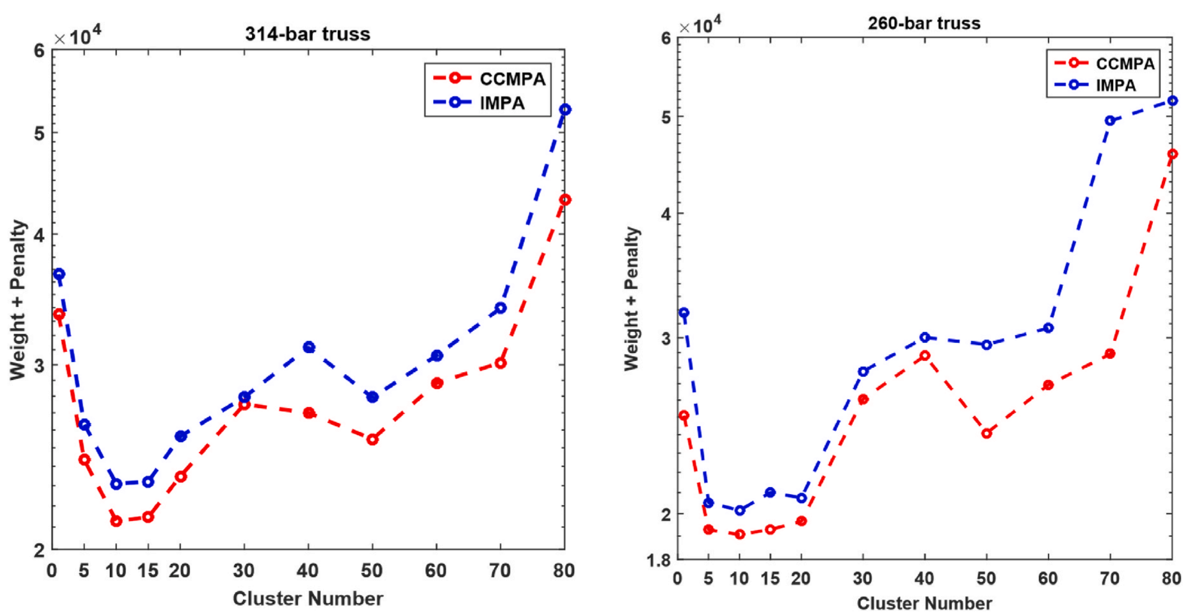
1. The population of coordinates ($Prey_C$) is considered as the first group.
2. The population of sections ($Prey_S$) is divided into K number of groups.

Hence, the total number of groups equals $K+1$. Also, because the sub-population of coordinates has a lower number of dimensions compared to that of sections, we evaluate and optimise the coordinates before the sections in the CCMPA. In this case, we assume that the complexity of the coordinates is much smaller than that of sections due to their low-dimensional space. Thus, by the early optimisation of coordinates, at each iteration, we can put aside a lot of fitness evaluations for the optimisation of sections, which can lead to better results.

3.5. Cooperative co-evolutionary marine predators algorithm + greedy search (CCMPA-GS)

Despite all the positive attributes of CCMPA, sometimes its suggested designs need modifications due to complex and nonlinear constraints of large-scale truss problems. In the meantime, the bar sizes yielded by optimal designs of CCMPA search are not guaranteed to provide the required minimum allowable stress and nodal displacement for each problem. Furthermore, in some cases, there is scope in the CCMPA results to reduce the sizes of some bars. To handle these matters, we propose a downward greedy search algorithm (Algorithm 4) combined with the CCMPA search (CCMPA-GS) to reinforce the exploitation ability and performance of the proposed truss optimisation framework.

The primary objective of the GS is smoothing the bar size concerning the constraints by decreasing the size of the bars one by one. In other words, GS is looking for modifications that give us the most negligible reduction in both stress and nodal displacement violations for the most significant decrease in the bar size. Therefore, the purpose is to maximize the improvement rate according to Equation (19).



(a) (b)

Fig. 4. Comparison between CCMPA and IMPA with different numbers of clusters a) on 314-bar truss problem b) on 260-bar truss problem.

$$
\begin{aligned}
&\text{Sum}_{PV_t} = \sum_{i=1}^{N}\left(\left|\sigma_i^{\max} - \sigma_i\right|\right) + \sum_{j=1}^{M}\left(\left|\delta_j^{\max} - \delta_j\right|\right)\\
&\text{Argmax} \rightarrow f(\Theta) = \left(\frac{\sum_{t=1}^{N} \Delta\, \text{barWeight}_t}{\Delta S\text{um}_{PV}}\right) \qquad (19)\\
&\text{Subject} - \text{to}: \Delta \text{Sum}_{PV} \geq 0, \Delta\, \text{barWeight}_t \geq 0
\end{aligned}
$$

As the sections' scale and nodal displacement are different, two search step size variables (*$Step_s$*, *$Step_c$*) are defined. Both step sizes are linearly decreased due to speeding up the convergence rate and improving exploitation capacity. Practically, GS is so fast and effective to attain the near best lowest weight solutions. The improvement rate is a coefficient of weight reduction over the violation smoothness. This rate shows which solution can produce a more competitive design compared with all solutions evaluated.

problems, the number of sizing variables is much larger than that of the shape variables. Thus, more fitness evaluations are needed to optimise sizing variables, and the optimal design vector is more dependent on finding the optimal sizing variable vector.

It is worth mentioning that in this study, our primary objective is not to directly compare the results obtained by our proposed algorithms with the most recent MPAs available. Instead, our focus is to advance beyond the achievements of our previous research [50] by introducing novel MPAs designed to excel in addressing the challenges posed by the two truss problems. Eventually, we compare the results obtained by the proposed MPAs with our recently proposed adaptive chaotic MPA [61].

In the first step, to find the best-performed approach, we developed a comparative optimisation study consisting of 10 popular recently published meta-heuristics (Table 1). According to the optimisation results

Algorithm 4 Downward Greedy Search

procedure Downward Greedy Search (GS)
Initialization
  Initialize σ_i^{max}, δ_j^{max} ▷ Initialize maximum allowable stress and nodal displacement in the ith element
  $Step_s = (Max_s - Min_s)/5$, $Step_c = (Max_c - Min_c)/20$ ▷ Compute the step size for GS
  $Truss_{iter}$={S, C} ▷ Read both sections (S) and coordinate (C) values
  $(Weight_b, \sigma_i, \delta_j)$=$Eval(Truss_{iter})$ ▷ Evaluate the design
  $Sum_{PV} = \sum_{i=1}^{N}(|\sigma_i^{max} - \sigma_i|) + \sum_{j=1}^{M}(|\delta_j^{max} - \delta_j|)$ ▷ Calculate sum of the violations
  while $iter \geq Max_{iter}$ **do**
    while $t \leq N + M$ **do**
      $Temp = Truss_{iter}$
      if $t \leq N$ **then**
        $Temp_t = Temp_t - Step_s$
      else
        $Temp_t = Temp_t - Step_c$
      end if
      $(Weight_t, \sigma_i, \delta_j)$=$Eval(Temp_t)$
      $Sum_{PV_t} = \sum_{i=1}^{N}(|\sigma_i^{max} - \sigma_i|) + \sum_{j=1}^{M}(|\delta_j^{max} - \delta_j|)$
      $Improvement_{rate}^{t} = \frac{Weight_b - Weight_t}{Sum_{PV} - Sum_{PV_t}}$ ▷ Compute the improvement rate
      $t = t + 1$
    end while
    $Truss_{iter}$=Max($Improvement_{rate}$) ▷ Select the best solution and update the truss
    $Step_s = Step_s - (\frac{iter}{Max_{iter}} Step_s) + 1$ ▷ $Step_c$ and $Step_s$ linearly decreased
    $iter = iter + 1$
  end while
  return $Truss_{iter}$
end procedure

4. Numerical results and discussions

In order to assess the performance of the proposed novel optimisation methods in this study and develop a proper comparative framework, we designed and applied the developed procedures to two different large-scale truss problems described in section 2.1. Meanwhile, we attempted to investigate the various pros and cons of these 15 optimisation methods to solve large-scale structures with nonlinear constraints. In order to provide a fair comparison for all optimisation methods, we set the same population size (50) for population-based methods and an equal evaluation number at 10^5. Additionally, in the CCMPA and CCMPA-GS, the population size for the *$Prey_C$* and *$Prey_S$* is set to 10 and 40, respectively. These values are obtained after a few initial experiments. The reason that the *$Prey_C$* is smaller in size is probably due to its smaller number of dimensions. Indeed, in the two truss structure

from both case studies, MPA performed best compared with the other nine meta-heuristics. However, we can see that MPA performance was reduced after consuming 25 % of the computational budget and encountering stagnation issues. A stagnation problem is when searching for an optimum process stagnates prior to discovering a globally optimal design. To address this issue and improve the convergence rate, three novel cooperative co-evolutionary MPAs were proposed, including IMPA, CCMPA and CCMPA + Greedy Search (CCMPA-GS).

4.1. Comparison between CCMPA and IMPA with different number of clusters

As in both novel cooperative optimisation methods, the cluster number plays a significant role in improving the performance of the proposed designs, we investigate this matter to find the optimal initial. Fig. 4 shows the comparison between CCMPA and IMPA with a different number of clusters on 314-bar and 260-bar truss problems. We can see that CCMPA outperforms the IMPA in both cases. This is because of the

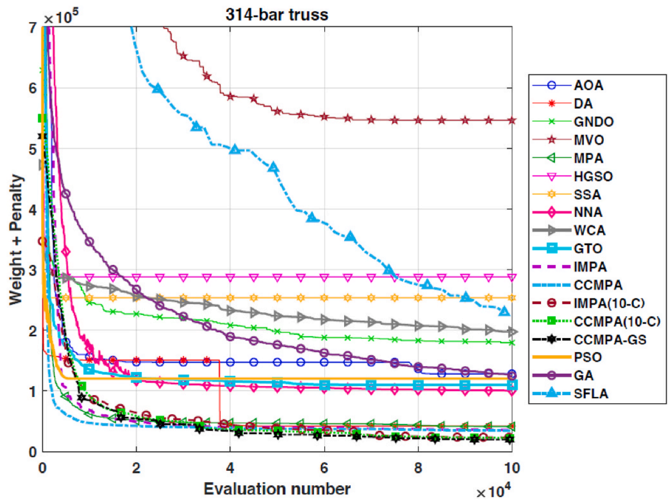


Fig. 5. Convergence behaviour comparison between 10 state-of-the-art meta-heuristic algorithms, two popular optimisation methods, GA and PSO, and one memetic algorithm (SLFA), and five proposed hybrid cooperative algorithms on the 314-bar truss problem. Each curve shows the average of best-found solutions of 10 independent runs.

higher convergence speed and lower computational cost of CCMPA. Also, both algorithms represent poor performance for the largest and smallest number of clusters. When clusters are too large, cooperation between clusters and diversity increases. Thus, both algorithms have stronger global searches. On the other hand, the local search ability within each cluster decreases due to the smaller number of fitness evaluations for each cluster. Hence, it leads to poor convergence speed and nonoptimal solutions. We can see that both algorithms have a poor performance when *Cluster no* $\geq$ 30 and the worst performance when *Cluster -no* = 80. While, when the number of clusters is too small (*Cluster no* = 1), the whole search space is considered a huge cluster. In this case, each individual within the cluster can have a larger number of fitness evaluations to improve itself, leading to a stronger local search.

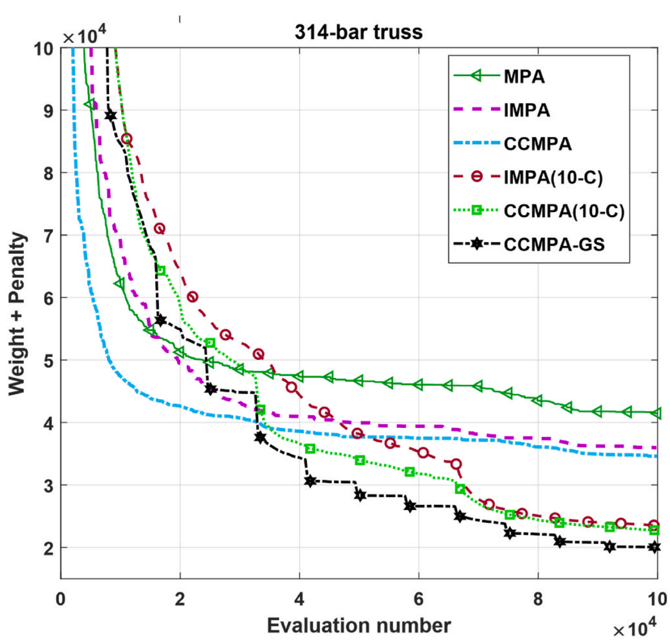


Fig. 6. Convergence behaviour comparison between the proposed cooperative optimisation algorithms (IMPA, CCMPA and CCMPA-GS) with one and ten clusters on the 314-bar truss problem. Each curve shows the average of best-found solutions of 10 independent runs.

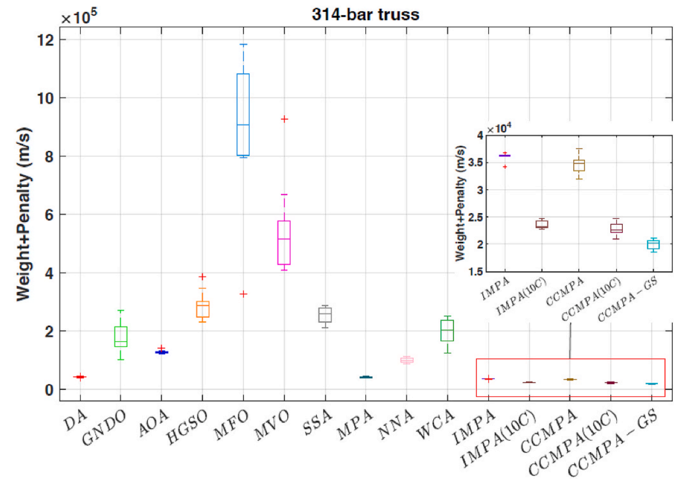


Fig. 7. Best-found designs of 314-bar truss per each experiment using 15 optimisation methods. The total evaluation number of is 10^5. The total number of decision variables is 328.

Nonetheless, the global search and population diversity deteriorate due to the lack of other clusters to cover the entire design variable. As a result, it will lead to being trapped in local optima. Overall, the algorithms have a good performance when 10 $\leq$ *Cluster no* $\leq$ 15 and 5 $\leq$ *Cluster no* $\leq$ 20 on 314-bar and 260-bar truss problems, respectively. Also, both algorithms perform best when *Cluster no* = 10 on both case studies.

4.2. 314-Bar truss optimisation results

In Fig. 5 each curve visualises the evolution of weight plus penalty factor for the 314-bar truss problem and is specified by the average of the best-found design per each iteration (population) for each meta-heuristic over 10^5 evaluation number. The control parameters applied by all methods are based on the original work recommended. In terms of convergence rate, the highest rank is related to CCMPA in the initial 20 % runtime. The reason is the high exploration and exploitation abilities of decomposition-based methods, where multiple sub-populations search simultaneously for different locations in the search space. In other words, while multiple sub-populations are responsible for

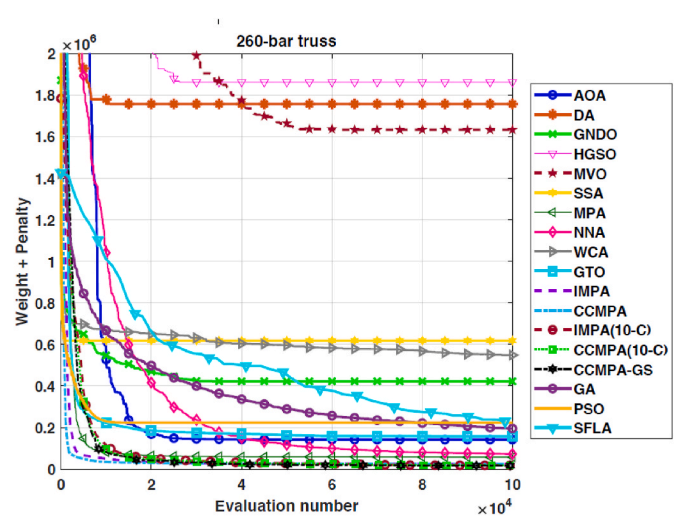


Fig. 8. Convergence behaviour comparison between 10 popular optimisation algorithms, GA, PSO, Memetic algorithm (SFLA), and five hybrid proposed methods on the 260-bar truss problem. Each curve shows the average of best-found solutions of 10 independent runs and the total evaluation number is 10^5.

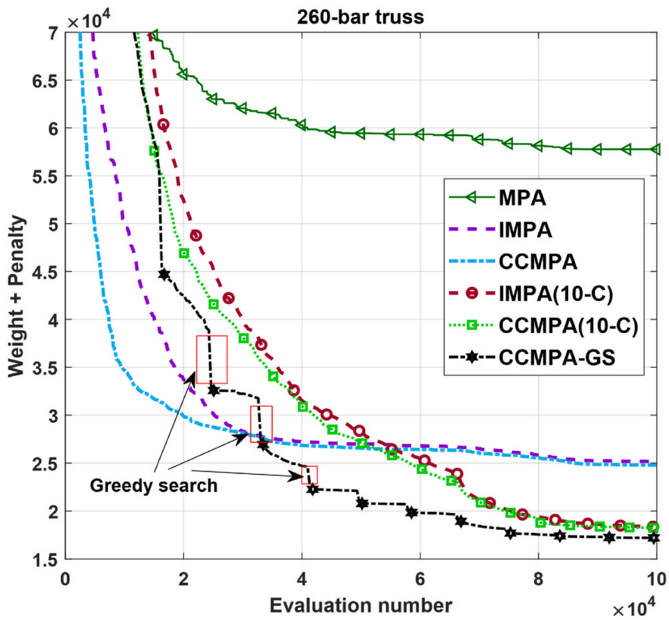


Fig. 9. Convergence behaviour comparison between 10 optimisation algorithms on the 260-bar truss problem. Each curve shows the average of best-found solutions of 10 independent runs and the total evaluation number of is 10^5.

searching multiple regions of search space, each sub-population escalates local search in its region. This searching ability also helps the algorithm to escape from local optima and prevent premature convergence. However, CCMPA-GS could considerably outperform CCMPA and other meta-heuristics in terms of convergence rate in a large-scale truss (314-bar+ 14 shape variables) with several natural frequency constraints. This is because the greedy search in later generations helps CCMPA-GS to exploit the promising regions effectively. Among the other optimisation methods in Fig. 5, it can be seen that NNA and GTO converged to reasonable solutions faster than others in a very similar time; however, DA could exploit a better local optimum finally.

Fig. 6 shows a close-up of the convergence rate for four meta-heuristics, including original MPA, two proposed cooperative MPAs and hybrid cooperative MPA with a local search (GS). This plot shows a sharp decline in minimising the weight of the 314-bar truss by CCMPA-GS and the best-found design is 18470 kg, which can be shown in Fig. 11 (a). The stepped shape of CCMPA-GS convergence is related to the successful performance of the greedy search (GS) that is adequately tuned to work with the CCMPA. In both CCMPA and IMPA with ten clusters, optimising the structure's weight performed better than with one cluster. However, CCMPA with ten clusters surpassed IMPA with the same cluster number in terms of quality of solutions and speed.

Fig. 7 illustrates a box-and-whiskers plot that includes the median, mean, minimum, maximum, first quartile, and third quartile of the best-achieved 314-bar designs for 15 optimisation search algorithms. The novel approaches expressed in this article are given in the last five columns of the Figure. These optimisation outcomes are reflected in the precise margin between the performance of the proposed methods and the other methods.

At first glance, it is crystal clear that the highest-ranked methods are related to the hybrid and cooperative MPA with ten clusters with a bit of variance for ten independent runs. Meanwhile, except for the proposed framework based on MPA, we can see that the DA can be an effective technique for optimising such large-scale problems. Furthermore, both IMPA and CCMPA with ten clusters could perform somewhat the same; however, the average performance of CCMPA was better than IMPA in terms of the best-found design per experiment. Finally, it can be observed that the highest execution was related to CCMPA-GS, which could propose the cheapest truss designs. This perhaps reflects the usefulness of a fast greedy decision among the feasible solutions (a trade-off between the weight of the structure and its total violations).

4.3. 260-Bar truss optimisation results

In order to evaluate the performance of the proposed optimisation framework, we applied the second large-scale truss structure with 260 bars and ten z-coordinates of the top nodes. Fig. 8 shows the convergence rate of 15 optimisation algorithms on the 260-bar truss problem. Each curve shows the average of the best-achieved designs of ten

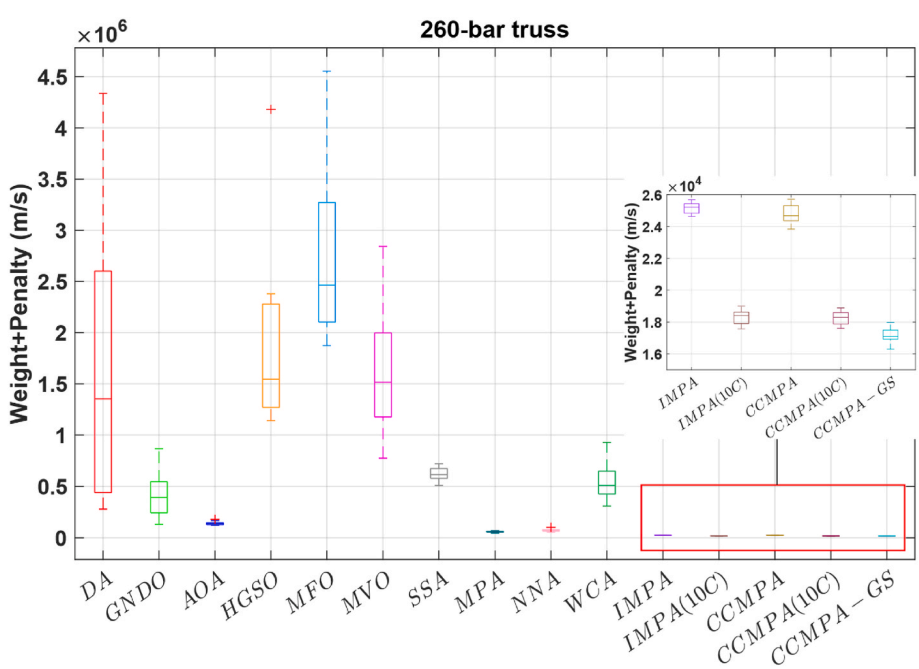


Fig. 10. Best-found designs of 260-bar truss per each experiment using 15 optimisation methods. The total evaluation number is 10^5. The total number of decision variables is 270.

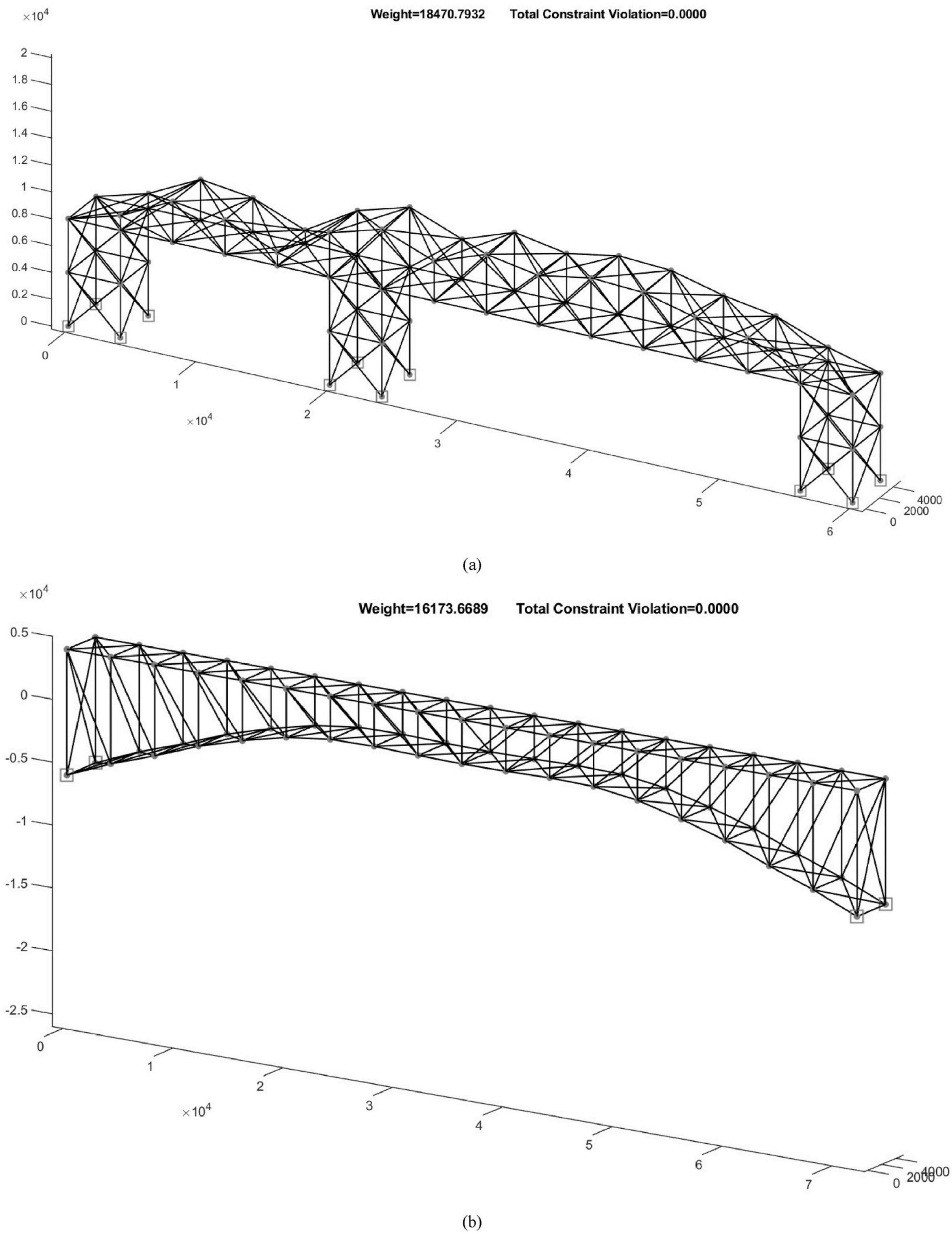


Fig. 11. Best-found feasible (a)314-bar truss design with a total weight at 18470.7932 (Kg), (b) 260-bar truss design with a total weight at 16173.6689 (Kg).

independent runs. From these results, we can observe that the most effective performing methods for truss design optimisation are the variants of MPA. These algorithms demonstrate robustness as they exhibit consistently strong performance across different problems. The best leading performance is related to CCMPA combined with greedy search with a minimum truss weight of 16173 kg (can be seen in Fig. 11(b)). The GTO was able to explore the search space appropriately with several local optimal configurations; however, it had a tendency to get stuck in local optima in the initial 20 % of the runtime, similar to WCA, SSA, and GNDO. Although the convergence speed of NNA was lower than MPA, GTO and AOA, it could effectively handle the nonlinear constraints, avoid falling into the traps (local minima) and obtain a fair balance between exploration and exploitation. It is noted that some of the popular meta-heuristics could not optimise this large structure with a complex search space such as HGSO, DA and MVO. The main reasons may come from the scalability attribute to handle the optimisation

Table 2
The statistical performance criteria for the three proposed optimisation frameworks compared with other 15 modern search algorithms for 314-bar truss.

Metric	DA	GNDO	AOA	HGSO	MFO	MVO	SSA	MPA	NNA
Min	39336	103096	122254	231501	326890	408869	210731	38885	87685
Max	45982	272333	140836	386397	1184149	928578	288499	44300	114471
Median	41924	165280	127870	287456	908014	516050	258967	41211	99055
Mean	42170	179515	128341	288184	892475	546289	253553	41446	100648
STD	1647	53755	5298	48962	238876	160152	27844	1697	8896
Friedman Test	6.6	12.4	10.7	15.3	17.8	17.1	14.9	6.4	8.3
Metric	WCA	PSO	GA	SFLA	IMPA	IMPA-10C	CCMPA	CCMPA-10c	CCMPA-GS
Min	125435	107225	88886	205554	34191	22714	32020	21047	18512
Max	251538	134067	167457	271075	36720	24749	37566	24685	21190
Median	204808	120603	123787	224063	36261	23251	34848	22570	20177
Mean	197520	121419	124658	229473	35936	23543	34603	22772	20014
STD	45330	8487	25372	18852	999	737	1750	1220	881
Friedman Test	12.5	9.9	10.1	14.0	4.8	2.7	4.2	2.3	1.0

Table 3
The statistical performance criteria for the three proposed optimisation frameworks compared with other 15 modern search algorithms for 260-bar truss.

Metric	DA	GNDO	AOA	HGSO	MFO	MVO	SSA	MPA	NNA
Min	279327	131850	120427	1141198	1873686	775263	509119	44524	55371
Max	4336301	866948	180133	4181235	4555126	2842608	721205	70314	102519
Median	1352414	392953	139926	1545486	2463511	1515654	614567	58363	69727
Mean	1756475	422380	142272	1861821	2843644	1632285	618163	57767	73054
STD	1482007	237157	18115	921701	1001899	652826	63015	7729	13059
Friedman Test	15.2	11.5	8.5	16.6	17.2	16.3	13.9	6.2	6.8
Metric	WCA	PSO	GA	SFLA	IMPA	IMPA-10C	CCMPA	CCMPA-10c	CCMPA-GS
Min	308566	140205	114496	278954	24643	17579	23842	17610	16302
Max	929709	330014	323663	451211	25678	19006	25717	18889	17981
Median	509055	214639	201222	371047	25216	18410	24665	18304	17230
Mean	548484	223832	195501	367637	25164	18321	24791	18303	17197
STD	178114	61887	63179	56838	388	494	606	451	509
Friedman Test	13.3	9.7	9.3	11.5	4.8	2.4	4.2	2.5	1.1

Table 4
A comparison between the three proposed optimisation algorithms with other state-of-the-art improved MPA algorithms.
314-bar Truss.

Metric	MPA [39]	HNMPA [61,91]	HNCMPA [61]	IMPA	IMPA-10c	CCMPA	CCMPA-10C	CCMPA-GS
Min	44524	19857	19398	24643	17579	23842	17610	16302
Max	70314	21589	29146	25678	19006	25717	18889	17981
Mean	57767	20559	21515	25164	18321	24791	18303	17197
Median	58363	20570	20619	25216	18410	24665	18303	17213
STD	7728.5	494.1	2839.3	388.2	493.7	605.8	398.2	479.5
Metric	MPA [39]	HNMPA [61,91]	HNCMPA [61]	IMPA	IMPA-10c	CCMPA	CCMPA-10C	CCMPA-GS
Min	44524	20639	20966	34191	22714	32020	21047	18512
Max	70314	24593	27423	36720	24749	37566	24685	21190
Mean	57767	22656	24129	35936	23543	34603	22772	20014
Median	58363	22394	24164	35936	23251	34848	22570	20177
STD	7728.5	1481.3	1982.7	666.1	737.0	1749.9	1220.4	881.3

260-bar Truss.

problems with many decision variables and apply smart strategies to face the constraints.

As Fig. 8 cannot provide a high-resolution comparison of the landscape for the proposed methods, especially for the last iterations, Fig. 9 is plotted and shows a clear understanding of MPA, cooperative MPA (IMPA and CCMPA) and hybrid CCMPA's performance. From this plot, it is indicated that the average optimisation result of CCMPA-GS was considerably above all other variants of MPA. Furthermore, using ten clusters for both IMPA and CCMPA performed better than one, similar to the 314-bar case study.

The second significant observation from Fig. 9 is that both IMPA and CCMPA convergence slop with one cluster is sharper than using ten clusters. Its primary reason is that applying several clusters decreases the exploitation ability and on other hand, it improves the exploration strategies for the initial iteration of the optimisation process. To reinforce the progress rate of the CCMPA, we proposed a greedy local search to assist the process of the global search.

Fig. 10 demonstrates the best-achieved solutions for the 260-bar truss case study received by ten popular meta-heuristics plus five proposed cooperative MPAs. Concerning these experiments, the hybrid cooperative MPA (CCMPA-GS) could beat the other competitors with the minimum structure weight.

Moreover, the searchability of AOA and NNA is considerable to propose acceptable designs with a high level of reliability. As can be seen in Fig. 10, DA, HGSO, MFO, and MVO proposed various design solutions with a wide variance and applied them for such large-scale structures with complex constraints that cannot be appropriate.

The statistical analysis metrics (Minimum, maximum, mean, median, and STD) of the optimisation methods performance are listed in Tables 2 and 3. In both large-scale case studies, the optimal designs were found

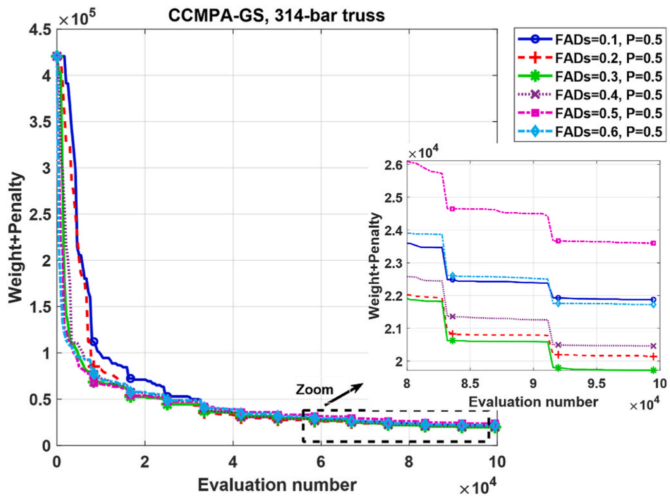


Fig. 12. A convergence rate comparison of various fish aggregating devices (FADs) effects on the proposed hybrid MPA (CCMPA-GS) applied for the large-scale 314-bar truss problem. The total evaluation number of is 10^5. The total number of decision variables is 328. All configurations are run with the same random seed.

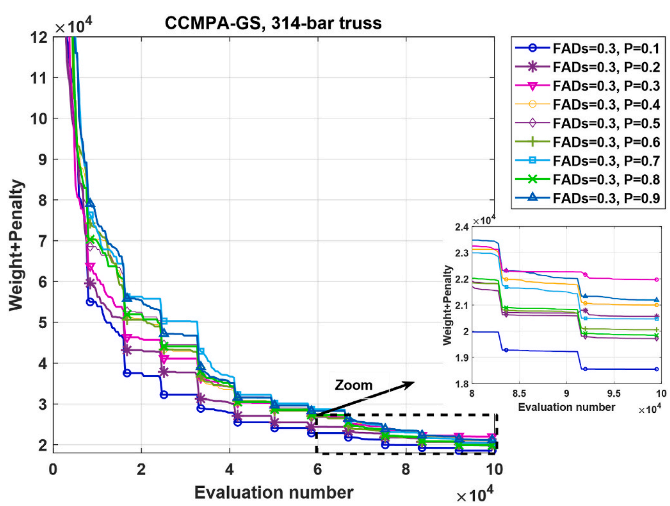


Fig. 13. A convergence rate comparison of various P (exploration coefficient) with the best-found value of FADs = 0.3 effects on the proposed hybrid MPA (CCMPA-GS) applied for the large-scale 314-bar truss problem. The total evaluation number is 10^5. The total number of decision variables is 328. All configurations are run with the same random seed.

by CCMPA-GS. Moreover, IMPA and CCMPA with ten clusters could propose considerable solutions for truss problems in this study as well. Undoubtedly, the most interesting outlook of these findings is the high level of robustness of the proposed truss optimisation framework. Furthermore, Table 4 shows a comparison between the Cooperative MPA with other modified MPA in order to minimise the weight of the truss.

4.4. Hyper-parameters analysis of CCMPA-GS

Some environmental parameters are considered in the simulation of marine predators (MPA), such as Eddy formation and Fish Aggregating Devices (FADs) effects. Fish are captivated by hovering objects and utilise them to keep sites for social interactions (mating movements and finding foods). For example, marine investigations [87] show that sharks would like mostly to concentrate near FADs (roughly 80 %) and set their motion with our other team members. However, sometimes they prefer exploring other areas to find prey groups (a random jump with a wider search step). The FADs effect is simulated (See Equation (18)) in MPA as a control parameter of 0.2 and plays the role of local optima to speed up the convergence rate or escape the optimisation process from stagnation issues.

In this study, we consider tuning the FADs control parameter and setting various values at [0.1,0.6]. Fig. 12 shows the CCMPA-GS convergence histories with different FADs parameters in order to optimise the 314-bar truss problem. The most interesting observation is that there is a direct relationship between the FADs coefficient and convergence rate. For example, we can see the highest convergence behaviour is related to $FADs = 0.6$ for the initial evaluations. However, the experiments with large FADs (such as 0.5 and 0.4) cannot escape the local optima effectively. From Fig. 12, the minimum structures' weight proposed by $FADs = 0.3$ which is performed better than the pre-defined value ($FADs = 0.2$) at 2.5 %.

The second control parameter is P which is a constant and assists in reducing or elaborating the search step sizes carried by predators. Thus, we kept FADs fixed and evaluated the effect of various P values at [0.1,0.9]. As illustrated in Fig. 13, The behaviour of the P value is complex, and there is not a linear relationship between the P value and CCMPA-GS performance. We can see that the best results were proposed by $P = 0.1$ with 6.3 % improvement compared with the recommended value ($P = 0.5$).

4.5. Shape variables analysis

As the shape variables play a significant role in optimising truss structures, we develop a more apparent observation of the shape optimisation process in both case studies. In this way, a parallel coordinated plot can be seen in Fig. 14. This parallel coordinated plot is used to visualise high dimensional shape data, where the series of its coordinate values describe each statement devised against their coordinate indices.

Fig. 14(a) shows the best experiment of CCMPA-GS for the 314-bar truss problem. Each line represents the configuration of 14 shape variables and total weight (weight + penalty). Furthermore, the colour line is related to the fitness level, where dark red shows the minimum weight. To have a wide landscape, we can see that the best configurations (dark red) have a complex combination of various 14-shape values; for instance, the best range of first and second shape variables ($C1$ and $C2$) are [9060–9080], and [11000–11100], respectively.

The behaviour of shape variables in the 260-bar truss problem differs from the previous case study and can be seen in Fig. 14(b). From this parallel plot, we can consider that the initial six optimal shape setups were concentrated on the range lower than zero; however, the last four shape coordination mostly focus on the positive values.

4.6. Technical benefits of CCMPA-GS

According to the results obtained from the rigorous experimental and statistical analysis it demonstrated that the optimiser proposed in this study, which encompasses a fusion of the Marine Predators Optimisation Algorithm (MPA) with cooperative coevolutionary (CC) algorithms and the incorporation of a greedy search mechanism, exhibits the potential to elicit substantial enhancements across a multitude of performance measures. Specifically, this amalgamation has been shown to yield significant improvements in terms of overall performance, exploration and exploitation capabilities, convergence rate, and mitigation of the occurrence of local optima. The rationale behind these advancements can be attributed to various factors as follows.

- Enhanced exploration: By incorporating the MPA concept with the Cooperative Coevolution (CC) strategy, which entails the utilisation of multiple subpopulations to optimise distinct segments of the problem's search space, the exploration process can be significantly

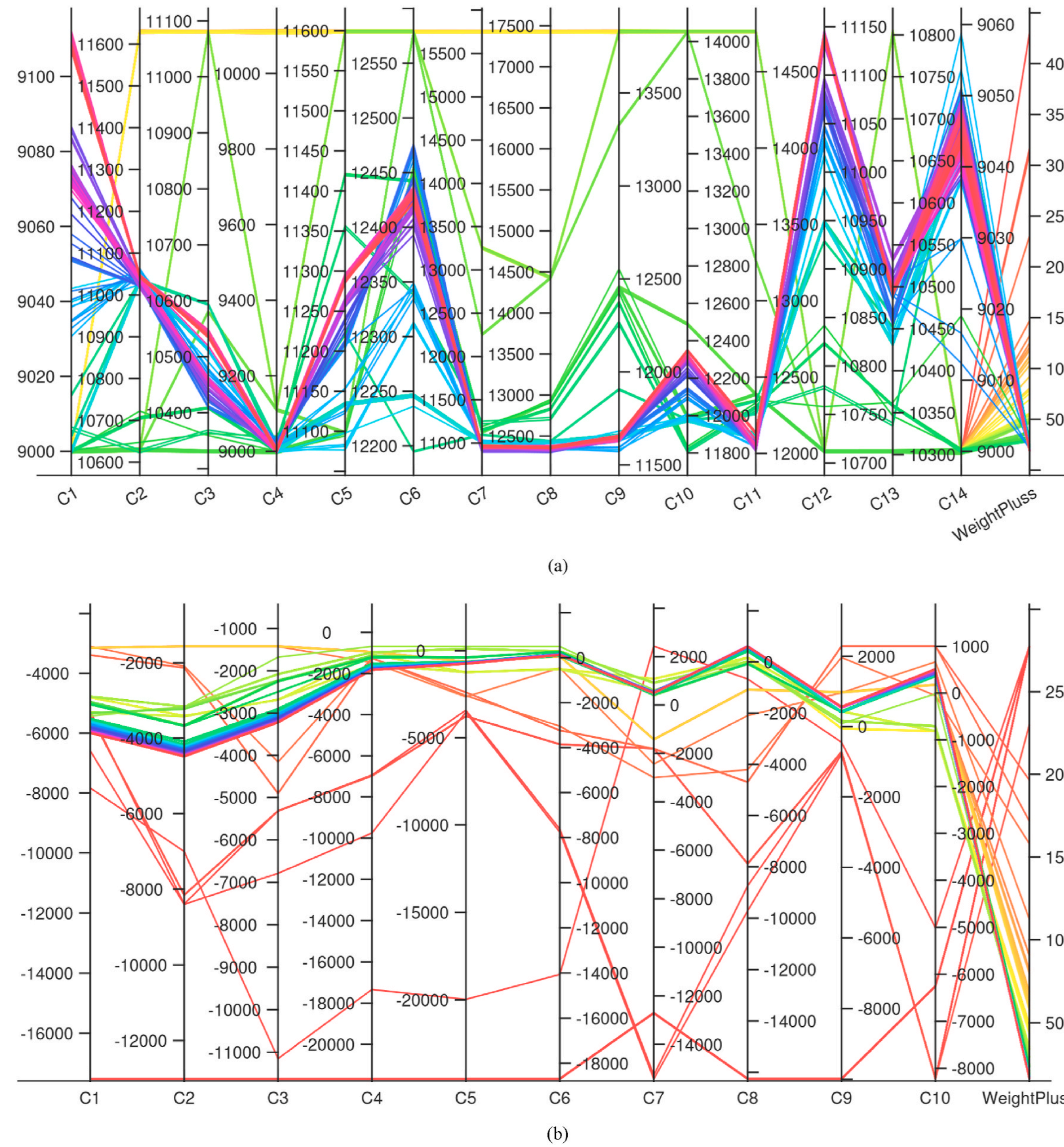


Fig. 14. Parallel plot for 14 shapes (*C*) variables (a) and 10 shape variables (b) of 314-bar and 260-bar truss problems obtained by CCMPA-GS. The dark pink colour shows the best configurations of shape variables. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

enhanced. The primary reason behind this assertion is that each subpopulation can effectively explore a specific region within the solution space, thereby emulating the varied hunting behaviours exhibited by marine predators. Consequently, this approach allows for a much more comprehensive exploration, ultimately leading to an increased likelihood of unearthing diverse and potentially superior solutions.

- Improved exploitation: Cooperative coevolutionary (CC) algorithms have shown remarkable proficiency in capitalising on subcomponents' intricate interplay and interconnectedness within a given problem. The algorithm's exploitation capabilities are significantly enhanced when the CC approach is augmented with MPA. With its unique ability to emulate predatory behaviour, such as the skilful manoeuvring and capture of prey, MPA continuously complements the CC framework. This combination of MPA, CC strategy and greedy search results in a synergistic effect that amplifies the algorithm's capacity to exploit the intricate interactions between subcomponents. Consequently, this heightened exploitation fosters more refined and optimally tailored solutions.
- Accelerated Convergence: The collaborative utilisation of the MPA in conjunction with cooperative coevolution has the potential to expedite the rate of convergence in the optimisation process. MPA's adeptness at efficiently exploring and adapting, inspired by the hunting behaviours of marine predators, when coupled with the CC approach's ability to decompose and coordinate subpopulations, can result in an accelerated convergence towards the most optimal solutions. These two methodologies harness the combined algorithm's parallel exploration and exploitation capabilities, enabling a more practical and resourceful search process and reducing the time required to find optimal solutions.
- Preventing local optima: MPA, like many other optimisation algorithms, can be vulnerable to local optima in various case studies. However, the integration of MPA with cooperative coevolution (CC) can significantly enhance its ability to overcome this limitation. By

leveraging CC, MPA gains the advantage of exploring diverse regions of the problem space simultaneously. This parallel exploration enables MPA to effectively navigate away from local optima that may impede its progress. Moreover, MPA's inherent hunting behaviours further strengthen its search capabilities by encouraging a comprehensive and exhaustive exploration for potential prey, or in this case, solutions. By combining these two approaches, the optimisation algorithm becomes more resilient against the trap of local optima, thereby facilitating the discovery of optimal or near-optimal solutions on a global scale. The synergy between MPA and cooperative coevolution not only mitigates the risk of stagnation but also amplifies the algorithm's potential to unearth the most favourable solutions within the given problem domain.

5. Conclusions and future scope

Shape and sizing optimisation of large-scale truss structures with considerable natural frequency constraints is a challenging problem due to a highly nonlinear interaction among cross-sectional and nodal coordinate forces, their various order of magnitude, and the natural frequencies' sensitivity to shape modifications.

To address the issues listed, this paper proposes two decomposition-based marine predators algorithms and a new hybrid Cooperative Coevolutionary marine predators algorithm combined with a greedy search (CCMPA-GS) framework to effectively figure out this large-scale and nonlinear optimisation problem.

The proposed algorithms used two different decomposition techniques (divide-and-conquer) to optimise the shape and size variables separately. An improved marine predators algorithm (IMPA) has been proposed that evaluates an individual's shape and sizing variable vectors independently. The results showed how it can enhance the global search of the basic MPA. Then, we incorporated cooperative coevolutionary strategy into the basic MPA that blends the exploration and exploitation abilities of CC and MPA to overcome the CC's premature convergence and stagnation. The proposed method is called CCMPA. Lastly, we showed that mixing a tuned local search (GS) with global optimisation can be helpful in terms of convergence speed and quality of the proposed designs.

A comprehensive comparative framework was developed, including ten popular meta-heuristic methods, GA, PSO, Memetic algorithm (SFLA) and three novel cooperative optimisation strategies. Comparing the optimisation results of two extensive case studies demonstrates that the proposed hybrid algorithm (CCMPA-GS) performed best in terms of computational cost and best-found designs' weight. It should be emphasised that the two proposed cooperative techniques reached better solutions with less computational cost than the other ten modern optimisation algorithms. Additionally, the proposed algorithms outperformed our recently proposed adaptive chaotic MPA [61]. Furthermore, the results showed that the CCMPAs performed well on the problems in terms of robustness and consistency.

The experimental findings demonstrated that the CCMPA-GS algorithm outperformed the conventional MPA method in the 314-bar and 260-bar problems, achieving a significant improvement of 52 % and 63 %, respectively. Furthermore, when compared to the latest variant, HNMPA, as proposed in Refs. [61,91], the CCMPA-GS algorithm exhibited even better performance with a superior improvement of 7 % and 21 % in the respective case studies.

In the future, this proposed optimisation framework can be extended in several ways. First, despite significant improvements in CCs, they still suffer from being trapped in pseudo-minima and search stagnation [51,78] caused by the improper decomposition of the decision variables. While numerous approaches to variable grouping have been suggested [92–100], effective strategies for variable grouping remain limited [51]. Regarding this, incorporating an adaptive variable grouping strategy putting interactive design variables in the same group could be investigated further. Second, according to the investigation results for adjusting control parameters, developing a self-adaptive strategy to tune the hyper-parameters would be the next step. In addition, considering more large-scale truss problems with various features and constraints will clarify the pros and cons of the proposed hybrid cooperative framework. Finally, evaluating the performance of more modern meta-heuristics and implementing novel cooperative techniques will help develop the optimisation results.

CRediT authorship contribution statement

Bahareh Etaati: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Mehdi Neshat:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Amin Abdollahi Dehkordi:** Writing – original draft, Software, Methodology, Investigation. **Navid Salami Pargoo:** Writing – original draft, Software, Investigation, Conceptualization. **Mohammed El-Abd:** Writing – review & editing, Investigation, Conceptualization. **Ali Sadollah:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Amir H. Gandomi:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] N. Khodadadi, A.O. Çiftçiğlu, S. Mirjalili, A. Nanni, A comparison performance analysis of eight meta-heuristic algorithms for optimal design of truss structures with static constraints, Dec. Anal. J. (2023) 100266.
[2] H.-A. Pham, T.-D. Tran, Optimal truss sizing by modified rao algorithm combined with feasible boundary search method, Expert Syst. Appl. 191 (2022) 116337.
[3] H.T. Oztürk, H.T. Kahraman, Meta-heuristic search algorithms in truss optimization: research on stability and complexity analyses, Applied Soft Computing 145 (2023) 110573.
[4] F.W. Zok, R.M. Latture, M.R. Begley, Periodic truss structures, J. Mech. Phys. Solid. 96 (2016) 184–203.
[5] J. Brutting, J. Desruelle, G. Senatore, C. Fivet, Design of truss structures through reuse, Structures 18 (2019) 128–137. Elsevier.
[6] S. Nguyen-Van, K.T. Nguyen, V.H. Luong, S. Lee, Q.X. Lieu, A novel hybrid differential evolution and symbiotic organisms search algorithm for size and shape optimization of truss structures under multiple frequency constraints, Expert Syst. Appl. 184 (2021) 115534.
[7] A.R. Kashani, C.V. Camp, M. Rostamian, K. Azizi, A.H. Gandomi, Population-based optimization in structural engineering: a review, Artif. Intell. Rev. (2021) 1–108.
[8] M. Khatibinia, S.S. Naseralavi, Truss optimization on shape and sizing with frequency constraints based on orthogonal multi-gravitational search algorithm, J. Sound Vib. 333 (24) (2014) 6349–6369.
[9] X. Zhang, Q. Lin, Three-learning strategy particle swarm algorithm for global optimization problems, Inf. Sci. 593 (2022) 289–313.
[10] H. Zhai, J. Zhang, Equilibrium reliability measure for structural design under twofold uncertainty, Inf. Sci. 477 (2019) 466–489.
[11] S. Kazemzadeh Azad, Enhanced hybrid metaheuristic algorithms for optimal sizing of steel truss structures with numerous discrete variables, Struct. Multidiscip. Optim. 55 (2017) 2159–2180.
[12] S. Pan, Y. Ma, Y. Wang, Z. Zhou, J. Ji, M. Yin, S. Hu, An improved master-apprentice evolutionary algorithm for minimum independent dominating set problem, Front. Comput. Sci. 17 (4) (2023) 174326.
[13] S. Gholizadeh, A. Barzegar, Shape optimization of structures for frequency constraints by sequential harmony search algorithm, Eng. Optim. 45 (6) (2013) 627–646.
[14] H. Lotfi, A. Ghoddosian, Size and shape optimization of two-dimensional trusses using hybrid big bang-big crunch algorithm, Int. J. Mechatron. Electr. Comput. Technol. 5 (14) (2015) 1987–1998.

[15] A. Kaveh, S. Javadi, Shape and size optimization of trusses with multiple frequency constraints using harmony search and ray optimizer for enhancing the particle swarm optimization algorithm, Acta Mech. 225 (6) (2014) 1595–1605.
[16] O. Altay, O. Cetindemir, I. Aydogdu, Size optimization of planar truss systems using the modified salp swarm algorithm, Eng. Optim. (2023) 1–17.
[17] A. Kaveh, M. Khosravian, Size/layout optimization of truss structures using vibrating particles system meta-heuristic algorithm and its improved version, Period. Polytech. Civ. Eng. 66 (1) (2022) 1–17.
[18] A. Kaveh, H. Yousefpour, Comparison of three chaotic meta-heuristic algorithms for the optimal design of truss structures with frequency constraints, Period. Polytech. Civ. Eng. 67 (4) (2023) 1130–1151.
[19] T. Dede, Y. Ayvaz, Combined size and shape optimization of structures with a new meta-heuristic algorithm, Appl. Soft Comput. 28 (2015) 250–258.
[20] H. Moosavian, P. Mesbahi, N. Moosavian, H. Daliri, Optimal design of truss structures with frequency constraints: a comparative study of de, ide, lshade, and cmaes algorithms, Eng. Comput. (2021) 1–19.
[21] A. Mortazavi, V. Togan, Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer, Struct. Multidiscipl. Optim. 54 (4) (2016) 715–736.
[22] F.K. Jawad, C. Ozturk, W. Dansheng, M. Mahmood, O. Al-Azzawi, A. Al-Jemely, Sizing and layout optimization of truss structures with artificial bee colony algorithm, Structures 30 (2021) 546–559. Elsevier.
[23] M. Farshchin, C.V. Camp, M. Maniat, Optimal design of truss structures for size and shape with frequency constraints using a collaborative optimization strategy, Expert Syst. Appl. 66 (2016) 203–218.
[24] J. Schwarz, T. Chen, K. Shea, T. Stankovic, Efficient size and shape optimization of truss structures subject to stress and local buckling constraints using sequential linear programming, Struct. Multidiscip. Optim. 58 (1) (2018) 171–184.
[25] A.H. Gandomi, K. Deb, R.C. Averill, S. Rahnamayan, M.N. Omidvar, Using semi-independent variables to enhance optimization search, Expert Syst. Appl. 120 (2019) 279–297.
[26] S. Kumar, G.G. Tejani, S. Mirjalili, Modified symbiotic organisms search for structural optimization, Eng. Comput. 35 (4) (2019) 1269–1296.
[27] A. Mandhyan, G. Srivastava, S. Krishnamoorthi, A novel method for prediction of truss geometry from topology optimization, Eng. Comput. 33 (1) (2017) 95–106.
[28] G.G. Tejani, S. Kumar, A.H. Gandomi, Multi-objective heat transfer search algorithm for truss optimization, Eng. Comput. 37 (1) (2021) 641–662.
[29] N. Pholdee, S. Bureerat, Hybridisation of real-code population-based incremental learning and differential evolution for multiobjective design of trusses, Inf. Sci. 223 (2013) 136–152.
[30] N. Panagant, S. Bureerat, K. Tai, A novel self-adaptive hybrid multi-objective meta-heuristic for reliability design of trusses with simultaneous topology, shape and sizing optimisation design variables, Struct. Multidiscip. Optim. 60 (5) (2019) 1937–1955.
[31] H.F. Eid, L. Garcia-Hernandez, A. Abraham, Spiral water cycle algorithm for solving multi-objective optimization and truss optimization problems, Eng. Comput. 38 (2) (2022) 963–973.
[32] J. Liu, Y. Xia, A hybrid intelligent genetic algorithm for truss optimization based on deep neutral network, Swarm Evol. Comput. (2022) 101120.
[33] N. Khodadadi, V. Snasel, S. Mirjalili, Dynamic arithmetic optimization algorithm for truss optimization under natural frequency constraints, IEEE Access 10 (2022) 16188–16208.
[34] N. Khodadadi, S. Mirjalili, Truss optimization with natural frequency constraints using generalized normal distribution optimization, Appl. Intell. (2022) 1–14.
[35] A.H. Gandomi, K. Deb, R.C. Averill, S. Rahnamayan, M.N. Omidvar, Variable functioning and its application to large scale steel frame design optimization, Struct. Multidiscipl. Optim. 66 (2023) 13.
[36] Y. Sun, Y. Zhou, Y. Shi, H. Li, K. Tian, B. Wang, Isogeometric-analysis-based stiffness spreading method for truss layout optimization, Comput. Methods Appl. Mech. Eng. 390 (2022) 114455.
[37] J. Zhang, M. Xiao, P. Li, L. Gao, Quantile-based topology optimization under uncertainty using kriging metamodel, Comput. Methods Appl. Mech. Eng. 393 (2022) 114690.
[38] H. Lin, H. Liu, P. Wei, A parallel parameterized level set topology optimization framework for large-scale structures with unstructured meshes, Comput. Methods Appl. Mech. Eng. 397 (2022) 115112.
[39] A. Faramarzi, M. Heidarinejad, S. Mirjalili, A.H. Gandomi, Marine predators algorithm: a nature-inspired metaheuristic, Expert Syst. Appl. 152 (2020) 113377.
[40] M.A. Al-Qaness, A.A. Ewees, H. Fan, L. Abualigah, M. Abd Elaziz, Marine predators algorithm for forecasting confirmed cases of covid-19 in Italy, USA, Iran and korea, Int. J. Environ. Res. Publ. Health 17 (10) (2020) 3520.
[41] M. Abd Elaziz, A.A. Ewees, D. Yousri, H.S.N. Alwerfali, Q.A. Awad, S. Lu, M.A. Al-Qaness, An improved marine predators algorithm with fuzzy entropy for multi-level thresholding: real world example of covid-19 ct image segmentation, IEEE Access 8 (2020) 125306–125330.
[42] M.A. Soliman, H.M. Hasanien, A. Alkuhayli, Marine predators algorithm for parameters identification of triple-diode photovoltaic models, IEEE Access 8 (2020) 155832–155842.
[43] M. Abdel-Basset, D. El-Shahat, R.K. Chakrabortty, M. Ryan, Parameter estimation of photovoltaic models using an improved marine predators algorithm, Energy Convers. Manag. 227 (2021) 113491.
[44] D. Yousri, T.S. Babu, E. Beshr, M.B. Eteiba, D. Allam, A robust strategy based on marine predators algorithm for large scale photovoltaic array reconfiguration to mitigate the partial shading effect on the performance of pv system, IEEE Access 8 (2020) 112407–112426.
[45] M. Abdel-Basset, R. Mohamed, M. Elhoseny, A.K. Bashir, A. Jolfaei, N. Kumar, Energy-aware marine predators algorithm for task scheduling in iot-based fog computing applications, IEEE Trans. Ind. Inf. 17 (7) (2020) 5068–5076.
[46] D.S. Abd Elminaam, A. Nabil, S.A. Ibraheem, E.H. Houssein, An efficient marine predators algorithm for feature selection, IEEE Access 9 (2021) 60136–60153.
[47] M.A. Shaheen, D. Yousri, A. Fathy, H.M. Hasanien, A. Alkuhayli, S. Muyeen, A novel application of improved marine predators algorithm and particle swarm optimization for solving the orpd problem, Energies 13 (21) (2020) 5679.
[48] G. Hu, X. Zhu, G. Wei, C.-T. Chang, An improved marine predators algorithm for shape optimization of developable ball surfaces, Eng. Appl. Artif. Intell. 105 (2021) 104417.
[49] B.S. Yıldız, V. Patel, N. Pholdee, S.M. Sait, S. Bureerat, A.R. Yıldız, Conceptual comparison of the ecogeography-based algorithm, equilibrium algorithm, marine predators algorithm and slime mold algorithm for optimal product design, Mater. Test. 63 (4) (2021) 336–340.
[50] B. Etaati, A.A. Dehkordi, A. Sadollah, M. El-Abd, M. Neshat, A comparative state-of-the-art constrained metaheuristics framework for truss optimisation on shape and sizing, Math. Probl Eng. 2022 (2022) 1–13.
[51] X. Ma, X. Li, Q. Zhang, K. Tang, Z. Liang, W. Xie, Z. Zhu, A survey on cooperative co-evolutionary algorithms, IEEE Trans. Evol. Comput. 23 (3) (2018) 421–441.
[52] M.N. Omidvar, M. Yang, Y. Mei, X. Li, X. Yao, Dg2: a faster and more accurate differential grouping for large-scale black-box optimization, IEEE Trans. Evol. Comput. 21 (6) (2017) 929–942.
[53] D. Yazdani, M.N. Omidvar, J. Branke, T.T. Nguyen, X. Yao, Scaling up dynamic optimization problems: a divide-and-conquer approach, IEEE Trans. Evol. Comput. 24 (1) (2019) 1–15.
[54] H. Liu, Y. Wang, N. Fan, A hybrid deep grouping algorithm for large scale global optimization, IEEE Trans. Evol. Comput. 24 (6) (2020) 1112–1124.
[55] R. Cheng, M.N. Omidvar, A.H. Gandomi, B. Sendhoff, S. Menzel, X. Yao, Solving incremental optimization problems via cooperative coevolution, IEEE Trans. Evol. Comput. 23 (5) (2018) 762–775.
[56] M.A. Potter, K.A. De Jong, A cooperative coevolutionary approach to function optimization, in: International Conference on Parallel Problem Solving from Nature, Springer, 1994, pp. 249–257.
[57] R. Chandra, M. Frean, M. Zhang, A memetic framework for cooperative coevolution of recurrent neural networks, in: The 2011 International Joint Conference on Neural Networks, IEEE, 2011, pp. 673–680.
[58] R.P. Wiegand, M.A. Potter, Robustness in cooperative coevolution, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, 2006, pp. 369–376.
[59] M.H. Nguyen, H.A. Abbass, R.I. McKay, Analysis of ccme: coevolutionary dynamics, automatic problem decomposition, and regularization, IEEE Trans. Sys. Man Cyber. Part C Appl. Rev. 38 (1) (2007) 100–109.
[60] B. Optimizer, Bright Optimizer: International Student Competition in Structural Optimization, 2021. https://www.brightoptimizer.com/. (Accessed 25 December 2021).
[61] A. A. Dehkordi, B. Etaati, M. Neshat, S. Mirjalili, Adaptive Chaotic Marine Predators Hill Climbing Algorithm for Large-Scale Design Optimisations, IEEE Access..
[62] M. Stolpe, Truss optimization with discrete design variables: a critical review, Struct. Multidiscip. Optim. 53 (2016) 349–374.
[63] C. Renkavieski, R.S. Parpinelli, Meta-heuristic algorithms to truss optimization: Literature mapping and application, Expert Syst. Appl. (2021) 115197.
[64] A. Slowik, H. Kwasnicka, Evolutionary algorithms and their applications to engineering problems, Neural Comput. Appl. 32 (2020) 12363–12379.
[65] M. Eusuff, K. Lansey, F. Pasha, Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization, Eng. Optim. 38 (2) (2006) 129–154.
[66] A. Sadollah, H. Sayyaadi, A. Yadav, A dynamic metaheuristic optimization model inspired by biological nervous systems: neural network algorithm, Appl. Soft Comput. 71 (2018) 747–782.
[67] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural, Comput. Appl. 27 (4) (2016) 1053–1073.
[68] F.A. Hashim, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, S. Mirjalili, Henry gas solubility optimization: a novel physics-based algorithm, Future Generat. Comput. Syst. 101 (2019) 646–667.
[69] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Eng. 376 (2021) 113609.
[70] Y. Zhang, Z. Jin, S. Mirjalili, Generalized normal distribution optimization and its applications in parameter extraction of photovoltaic models, Energy Convers. Manag. 224 (2020) 113301.
[71] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2) (2016) 495–513.
[72] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: a bio-inspired optimizer for engineering design problems, Adv. Eng. Software 114 (2017) 163–191.
[73] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110 (2012) 151–166.
[74] B. Abdollahzadeh, F. Soleimanian Gharehchopogh, S. Mirjalili, Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems, Int. J. Intell. Syst. 36 (10) (2021) 5887–5958.
[75] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-international Conference on Neural Networks vol. 4, IEEE, 1995, pp. 1942–1948.
[76] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73.

[77] H. Naseri, A. Golroo, M. Shokoohi, A.H. Gandomi, Sustainable pavement maintenance and rehabilitation planning using the marine predator optimization algorithm, Struct. Infrastruct. Eng. (2022) 1–13.
[78] F. Van den Bergh, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (2004) 225–239.
[79] X.-F. Song, Y. Zhang, Y.-N. Guo, X.-Y. Sun, Y.-L. Wang, Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data, IEEE Trans. Evol. Comput. 24 (5) (2020) 882–895.
[80] J.C.M. Diniz, F. Da Ros, E.P. da Silva, R.T. Jones, D. Zibar, Optimization of dp-m-qam transmitter using cooperative coevolutionary genetic algorithm, J. Lightwave Technol. 36 (12) (2018) 2450–2462.
[81] R. Sarkar, D. Barman, N. Chowdhury, A cooperative co-evolutionary genetic algorithm for multi-robot path planning having multiple targets, in: Comput. Intell. Pattern Recognit., Springer, 2020, pp. 727–740.
[82] Z.-Q. Cai, L. Lv, H. Huang, H. Hu, Y.-H. Liang, Improving sampling-based image matting with cooperative coevolution differential evolution algorithm, Soft Comput. 21 (15) (2017) 4417–4430.
[83] W. Deng, S. Shang, X. Cai, H. Zhao, Y. Zhou, H. Chen, W. Deng, Quantum differential evolution with cooperative coevolution framework and hybrid mutation strategy for large scale optimization, Knowl. Base Syst. 224 (2021) 107080.
[84] X. Zhou, A. Su, A. Liu, W. Cui, W. Liu, Cooperative approach to artificial bee colony algorithm for optimal power flow, Cluster Comput. 22 (4) (2019) 8059–8067.
[85] B. Etaati, Z. Ghorrati, M.M. Ebadzadeh, A full-featured cooperative coevolutionary memory-based artificial immune system for dynamic optimization, Appl. Soft Comput. (2021) 108389.
[86] M. Neshat, S. Mirjalili, N.Y. Sergiienko, S. Esmaeilzadeh, E. Amini, A. Heydari, D. A. Garcia, Layout optimisation of offshore wave energy converters using a novel multi-swarm cooperative algorithm with backtracking strategy: a case study from coasts of Australia, Energy 239 (2022) 122463.
[87] M. Neshat, B. Alexander, M. Wagner, A hybrid cooperative co-evolution algorithm framework for optimising power take off and placements of wave energy converters, Inf. Sci. 534 (2020) 218–244.
[88] W.-N. Chen, Y.-H. Jia, F. Zhao, X.-N. Luo, X.-D. Jia, J. Zhang, A cooperative co-evolutionary approach to large-scale multisource water distribution network optimization, IEEE Trans. Evol. Comput. 23 (5) (2019) 842–857.
[89] X. Zhang, K.-J. Du, Z.-H. Zhan, S. Kwong, T.-L. Gu, J. Zhang, Cooperative coevolutionary bare-bones particle swarm optimization with function independent decomposition for large-scale supply chain network design with uncertainties, IEEE Trans. Cybern. 50 (10) (2019) 4454–4468.
[90] R. Chandra, Competition and collaboration in cooperative coevolution of elman recurrent neural networks for time-series prediction, IEEE Transact. Neural Networks Learn. Syst. 26 (12) (2015) 3123–3136.
[91] A.S. Sadiq, A.A. Dehkordi, S. Mirjalili, Q.-V. Pham, Nonlinear marine predator algorithm: a cost-effective optimizer for fair power allocation in noma-vlc-b5g networks, Expert Syst. Appl. 203 (2022) 117395.
[92] L. Sun, S. Yoshida, X. Cheng, Y. Liang, A cooperative particle swarm optimizer with statistical variable interdependence learning, Inf. Sci. 186 (1) (2012) 20–39.
[93] S. Mahdavi, M.E. Shiri, S. Rahnamayan, Cooperative co-evolution with a new decomposition method for large-scale optimization, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014, pp. 1285–1292.
[94] F. Wei, Y. Wang, T. Zong, A novel cooperative coevolution for large scale global optimization, in: 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2014, pp. 738–741.
[95] F. Wei, Y. Wang, T. Zong, Variable grouping based differential evolution using an auxiliary function for large scale global optimization, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014, pp. 1293–1298.
[96] H. Ge, L. Sun, X. Yang, S. Yoshida, Y. Liang, Cooperative differential evolution with fast variable interdependence learning and cross-cluster mutation, Appl. Soft Comput. 36 (2015) 300–314.
[97] H. Liu, S. Guan, F. Liu, Y. Wang, Cooperative co-evolution with formula based grouping and cma for large scale optimization, in: 2015 11th International Conference on Computational Intelligence and Security (CIS), IEEE, 2015, pp. 282–285.
[98] X. Guan, X. Zhang, J. Wei, I. Hwang, Y. Zhu, K. Cai, A strategic conflict avoidance approach based on cooperative coevolutionary with the dynamic grouping strategy, Int. J. Syst. Sci. 47 (9) (2016) 1995–2008.
[99] F.B. de Oliveira, R. Enayatifar, H.J. Sadaei, F.G. Guimaraes, J.-Y. Potvin, A cooperative coevolutionary algorithm for the multi-depot vehicle routing problem, Expert Syst. Appl. 43 (2016) 117–130.
[100] S. Strasser, J. Sheppard, N. Fortier, R. Goodman, Factored evolutionary algorithms, IEEE Trans. Evol. Comput. 21 (2) (2016) 281–293.