Structures 83 (2026) 110858



Contents lists available at ScienceDirect

Structures

journal homepage: www.elsevier.com/locate/structures





A hyper-intelligent framework for large-scale structural optimisation

Shahin Jalili [a,*], Siamak Talatahari [b,c]

[a] Department of Civil and Environmental Engineering, Imperial College London, London SW7 2AZ, UK
[b] Data Science Institute, Faculty of Engineering & Information Technology, University of Technology Sydney, Ultimo 2007, Australia
[c] School of Computing, Macquarie University, Sydney, Australia

ARTICLE INFO

Keywords:
Hyper-intelligent
Optimal design
Algorithms
Large-scale structures
Optimisation

ABSTRACT

This study introduces a new algorithmic framework, termed the hyper-intelligent (HI) framework, designed to address diverse structural optimisation problems with improved statistical consistency. In this framework, a set of meta-heuristics in the low-level heuristic (LLH) space is applied concurrently by individuals during the optimisation process. Each individual is equipped with a high-level heuristic (HLH) subspace that provides an online, feedback-driven learning mechanism. Two variants are proposed based on the learning strategy: the hyper-intelligent algorithm with a choice function (HIA-CF) and the hyper-intelligent algorithm with a multi-armed bandit (HIA-MAB). The performance of the proposed approach is evaluated using three large-scale structural design examples with static and frequency constraints. The results demonstrate that the HI framework achieves more consistent performance than conventional meta-heuristics. This highlights the advantages of intelligently and adaptively integrating multiple search strategies simultaneously for complex structural design optimisation. The study emphasises the necessity of a paradigm shift in intelligent structural design optimisation.

1. Introduction

Recent advances in computational resources have led to an explosion in the development of artificial intelligence (AI) technology for various real-world applications over the past decades, from AI-driven medical imaging and diagnosis [1], to application in finance and banking [2], and energy [3] through to manufacturing [4] and supply chain [5]. The construction industry is no exception. Nowadays, data-driven techniques have been widely employed for decarbonisation [6], improving energy efficiency [7], enhancing visual comfort [8], advancing materials performance [9], construction management [10], and optimising structural design [11]. Structural design is one of earliest steps of the construction process. Structural elements are designed to ensure that both the individual elements and the entire structure perform as required under a set of combined loading conditions, while complying with the requirements specified by design codes. An intelligent structural design approach can not only reduce construction costs but also minimise the life-cycle environmental impacts of a structure throughout its service life.

When designing a structure, the structural designer adjusts a set of design parameters to achieve the desired performance and safety, while minimising material consumption and environmental impacts. However, due to the complexity of structures and the large number of design parameters involved, it is challenging for design engineers to reliably identify optimal designs through manual trial-and-error. The resulting vast design space makes it difficult to determine the most favourable designs. The structural optimal design process can be expressed as an optimisation problem, in which the design parameters such as layout, topology, and sections act as decision variables, while the cost, structural weight, or environmental impact associated with the design represent the objective function [12–17]. The requirements specified by the building codes act as constraints that the structural designs must comply with.

Structural optimisation often encounters challenges and constraints that make traditional gradient-based optimisation methods difficult to apply and sometimes inefficient. These challenges arise from factors such as the difficulty of explicitly formulating the optimal design problem in terms of design variables, the discrete nature of these variables, and the nonlinearity of design constraint functions. As a result, researchers have shifted their focus towards developing nature-inspired meta-heuristic algorithms to address various structural design optimisation problems. Depending on their source of inspiration, these algorithms are classified into various categories, such as evolutionary [18,19], swarm intelligence [20], physics-inspired [21–23], and

* Corresponding author.
E-mail address: s.jalili@imperial.ac.uk (S. Jalili).

https://doi.org/10.1016/j.istruc.2025.110858
Received 1 August 2025; Received in revised form 19 November 2025; Accepted 4 December 2025
Available online 11 December 2025
2352-0124/© 2025 The Authors. Published by Elsevier Ltd on behalf of Institution of Structural Engineers. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

socio-inspired [24,25] algorithms, all of which are collectively referred to as meta-heuristic algorithms in optimisation literature. The meta-heuristics are stochastic by nature and do not require gradient information from the objective and constraint functions, making them attractive alternatives to traditional approaches. However, according to the "no free lunch" theorem [26], these algorithms often exhibit varying performance in finding optimal solutions across different types of problems, excelling in some while underperforming in others.

Structural optimisation is a computationally expensive process that requires repeated structural analyses of modified designs, whether using traditional techniques or meta-heuristics [27–30]. However, the computational burden of structural optimisation is especially pronounced when using meta-heuristics. Although meta-heuristics do not require gradient information from the objective and constraint functions, which is a key advantage over traditional approaches, their application to structural optimisation requires thousands of repeated structural analyses. This makes them computationally intensive for the optimal design of large-scale structures [31,32]. This can pose significant challenges for the optimal design of large-scale structures with a high number of degrees of freedom [33].

The sensitivity of performance to problem type and the associated computational complexities highlight the need for caution when applying standard meta-heuristics in optimal structural design. Standard algorithms may exhibit statistically unstable convergence behaviour or premature convergence, often due to an imbalance between the exploration and exploitation capabilities. Additionally, meta-heuristics have a set of internal parameters that need to be well-tuned through a sensitivity analysis to achieve satisfactory performance [34,35]. This has led to the development of enhanced versions of meta-heuristic algorithms for various optimal structural optimisation problems. Some of these enhanced versions sometimes focus on improving a meta-heuristic algorithm by modifying its operators [36–42], while others involve hybridising two or more meta-heuristics to achieve better performance [43–48].

The successful yet cautious experience with meta-heuristics inspired the development of algorithms capable of handling a wider range of problem domains with relatively consistent performance. These algorithms are designed to minimise human intervention in internal parameters and methodology. This idea led to the development of hyper-heuristic (HH) algorithms in AI and computer science [49,50]. The algorithmic framework of the HHs is designed to provide flexibility, allowing the algorithm to select the most appropriate heuristic from a set of available search strategies. In some cases, it can even construct a suitable heuristic to solve a given problem. This paradigm shift aims to develop algorithms that can handle diverse problem domains with consistent performance [49,51].

The majority of the literature has focused on the meta-heuristics and their improved variants for structural optimisation. However, as noted, the performance of meta-heuristics is often problem-dependent. To address this challenge, the present study introduces a novel hyper-intelligent (HI) algorithmic framework, inspired by the principles of HH algorithms, for structural optimisation. Unlike conventional meta-heuristics that rely on a single search strategy, the framework integrates multiple meta-heuristic strategies, coordinated via high-level heuristic (HLH) subspaces. As shown in the simplified framework in Fig. 1, each individual in the proposed framework is equipped with an HLH subspace that has access to a global low-level heuristic (LLH) space. The global LLH space contains a pool of meta-heuristic search strategies. HLH subspaces function as decision-making tools that determine when to switch between these strategies to achieve the most favourable results. HLH subspaces receive feedback signals from associated individuals, including changes in objective function values and the success/failure history of each meta-heuristic applied in the LLH space. By leveraging this online feedback mechanism, HLH subspaces enable individuals to learn and adapt their behaviour dynamically, guiding the selection of the most appropriate strategies at different stages of the optimisation process.

Depending on the type of online learning mechanism employed in the HLH subspace, two algorithms are proposed based on the HI framework: the hyper-intelligent algorithm with choice function (HIA-

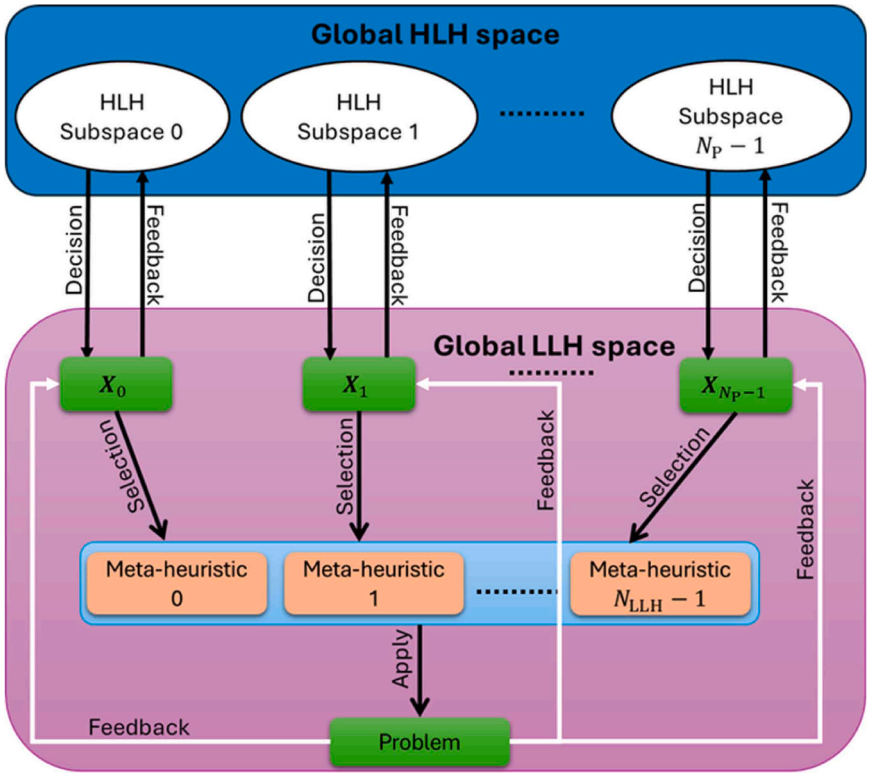


Fig. 1. Simplified framework of the HI approach proposed in this study.

CF) and the hyper-intelligent algorithm with multi-armed bandit (HIA-MAB) strategy. In HIA-CF, each HLH subspace operates based on a choice function model, whereas HIA-MAB incorporates a learning mechanism inspired by the multi-armed bandit problem from game theory. The performance of the proposed algorithms is evaluated against the standard meta-heuristic algorithms using three large-scale structural design examples under static and dynamic constraints. The results provide promising insights into the capabilities of the proposed algorithmic framework in handling different types of structural optimisation problems.

The remainder of this manuscript is structured as follows. Section 2 provides a brief overview of HH algorithms and their frameworks. Section 3 formulates the structural optimal design problems under static and dynamic constraints. Section 4 discusses the proposed HI framework in detail. Section 5 investigates three optimal structural design examples to test the performance of the proposed algorithms. Section 6 offers concluding remarks based on the findings of this study.

2. A brief overview of hyper-heuristics (HHs)

The idea behind hyper-heuristics (HHs) dates back to the 1960s [52,53]. At the time, it was hypothesised that the simultaneous application of various scheduling rules in production scheduling would yield better results than applying them separately [49]. The idea later evolved in automated planning in AI, internal parameter control in evolutionary algorithms, and automated learning of heuristics for constraint satisfaction problems [49]. Modern HHs aim to select or even construct the most appropriate heuristic from a set of available search strategies to solve a given problem. The aim is to develop an algorithm capable of constructing new heuristics or finding sequences of heuristics to solve a wider range of problem domains with relatively consistent performance [49,51]. Depending on their nature and feedback mechanism, HHs can be classified into different categories [54,55]. HHs that select the most suitable search strategy from a set of existing heuristics are called selection HHs, whereas those that generate new heuristics are termed constructive HHs. The way HHs learn also differs. Some HHs gradually learn based on the feedback received from the solution-finding process, which is referred to as the online feedback process. In contrast, in the offline feedback process, HHs learn from a set of training problems and generalise the method for future instances [51]. Interested readers are referred to Refs. [49,51,54,55] for various types of HHs. The focus of this study is on selection HHs with an online learning process.

In general, HH algorithms consist of two levels of search spaces: high-level heuristic (HLH) and low-level heuristic (LLH) spaces [55]. The LLH space directly tackles the problem, while the HLH indirectly controls it. The LLH comprises a set of heuristic search strategies applied to find a solution to the problem, whereas the HLH contains rules and mechanisms that record historical information based on the experience gained by the LLH space. The HLH determines which heuristic should be applied at each stage of the solution-finding process.

Technically speaking, there is a domain barrier between the LLH and HLH spaces in the sense that the HLH has no knowledge of the solution space [56]. Rather, it focuses on evaluating the performance experience of a given LLH in producing high-quality solutions. Over time, the HLH uses this experience to gradually learn and assess the performance of each LLH, determining which LLH should be applied at appropriate intervals to achieve better performance. This decision is typically based on probabilistic models built from the historical record of successes and failures of LLHs.

It is important to describe the overall idea of the HH framework, which forms the foundation for this study. Fig. 2 illustrates the overall framework of the selection HH, where the probabilistic HLH models reside in the HLH space, while a set of N_{LLH} heuristics is available in the LLH space. Based on feedback received from the LLH space, the parameters in the HLH space are updated. Selection of the appropriate LLH to apply is made using probabilistic models. The domain barrier signifies

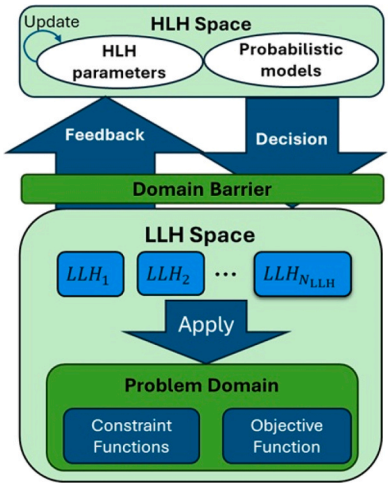


Fig. 2. Overall framework for the selection hyper-heuristic (HH) (adapted from Ref. [56]).

that no problem-specific knowledge is transferred between the LLH and HLH spaces. The HLH subspace only receives feedback on each LLH's performance, independent of the problem domain. The main aim of the domain barrier is to preserve the generality of the HH algorithm, allowing it to be applied to different problem types without modifying the framework [57]. The proposed approach in this study is primarily inspired by the overall framework in Fig. 2, a detailed description of which is provided in Section 4.

3. Problem formulation

In this study, two classes of structural optimisation problems are formulated to benchmark the proposed framework: frame design optimisation under static constraints and optimal truss design under dynamic constraints. In frame design optimisation under static constraints, the primary objective is to minimise structural weight while satisfying stress, displacement, and serviceability requirements. This problem involves complex constraint interactions, including combined axial-flexural behaviour, shear strength, and story drift limitations dictated by design codes [58]. The truss design problem under dynamic constraints represents a challenging scenario, as both cross-sectional sizes (discrete variables) and nodal coordinates (continuous shape variables) must be optimised simultaneously while ensuring that the structural natural frequencies remain within allowable ranges. This results in a mixed discrete-continuous problem with a highly nonlinear search landscape. Together, these problem classes provide a comprehensive evaluation of the HI framework across search domains with varying complexity and constraint characteristics.

3.1. Optimal design of frame structures under static constraints

The objective of this problem is to minimise the structural weight while satisfying combined constraints on axial-flexural behaviour, shear strength, and inter-story drifts based on the AISC-LRFD [59] code. The problem can be mathematically expressed as:

Find: $\boldsymbol{X} = [x_1, x_2, \ldots, x_{n_{\mathrm{eg}}}]$

To minimise: $f(\boldsymbol{X}) = W(\boldsymbol{X}) \quad \psi(\boldsymbol{X})^{\alpha}$

$$W(\boldsymbol{X}) = \sum_{i=1}^{m} \gamma A_i l_i$$

$$x_i \in \boldsymbol{S_i}, \quad \boldsymbol{S_i} = [S_{i1}, S_{i2}, \ldots, S_{ir_i}], \quad i = 1, 2, \ldots, n_{\mathrm{eg}}$$

Subjected to:

$$C_{IEL}(\boldsymbol{X}) = \begin{cases} \left[\frac{P_{uJ}}{\varphi P_n}\right]_{IEL} + \frac{8}{9}\left(\frac{M_{uxJ}}{\varphi_b M_{nx}} + \frac{M_{uyJ}}{\varphi_b M_{ny}}\right)_{IEL} - 1 \le 0 for \left[\frac{P_{uJ}}{\varphi P_n}\right]_{IEL} \ge 0.2 \\ \left[\frac{P_{uJ}}{2\varphi P_n}\right]_{IEL} + \left(\frac{M_{uxJ}}{\varphi_b M_{nx}} + \frac{M_{uyJ}}{\varphi_b M_{ny}}\right)_{IEL} - 1 \le 0 for \left[\frac{P_{uJ}}{\varphi P_n}\right]_{IEL} < 0.2 \end{cases}$$

$$C^{v}_{IEL}(\boldsymbol{X}) = \frac{(V_{uJ})_{IEL}}{(\varphi_v V_n)_{IEL}} - 1 \le 0$$

$$C^{t}_{D}(\boldsymbol{X}) = \Delta_{MaxJ} - \Delta^{a}_{Max} \le 0$$

$$C^{d}_{F}(\boldsymbol{X}) = [\delta_J]_F - [\delta^a]_F \le 0 \quad (1)$$

Here, $\boldsymbol{X}$ represents the vector containing cross-sectional areas for element design groups. n_{eg} indicates the number of element design groups. $f(.)$ denotes the objective function. $W(.)$ is the structural weight. m is the number of structural elements. The material density is indicated by γ. The penalty function defined to handle the constraints is denoted by $\psi(.)$. α represents a penalty constant parameter controlling the impact of constraints violations on the objective function. The cross-section and length for each element are represented by A_i and l_i, respectively. S_i is the discrete set of available sections. The number of available sections for the ith element design group is indicated by r_i. $C_{IEL}(.)$, $C^{v}_{IEL}(.)$, $C^{t}_{D}(\boldsymbol{X})$, and $C^{d}_{F}(.)$ are the stress ratio, shear stress, lateral displacement, and inter-story drift constraints, respectively. *IEL* denotes the element number. J represents the load combination index. P_{uJ} is the required axial strength for the Jth load case; M_{uxJ} and M_{uyJ} are the required flexural strengths about the x (strong) and y (weak) axes, respectively. P_n, M_{nx}, and M_{ny} are the nominal axial and flexural strengths of the *IEL*th element. The axial resistance factor φ is 0.85 for compression and 0.9 for tension. The flexural resistance factor φ_b is 0.9. V_{uJ} and V_n are the required and nominal shear strengths, respectively. The shear resistance factor φ_v set to 0.9. $C^{t}_{D}(\boldsymbol{X})$ compares the maximum lateral displacement of the structure in the Dth direction under the Jth load combination (Δ_{MaxJ}) with the maximum allowable displacement (Δ^{a}_{Max}). Similarly, $C^{d}_{F}(\boldsymbol{X})$ evaluates the inter-story drift for the Fth story under the Jth load combination ($[\delta_J]_F$) against its allowable limit ($[\delta^a]_F$). The penalty constant, α, is set to 2.

The penalty function is evaluated for each design candidate $\boldsymbol{X}$ as:

$$\psi(\boldsymbol{X}) = 1 + \sum_{IEL} \max(C_{IEL}(\boldsymbol{X}), 0) + \sum_{IEL} \max\left(C^{v}_{IEL}, 0\right) + \sum_{J}\sum_{D} \max\left(C^{t}_{D}(\boldsymbol{X}), 0\right) + \sum_{F} \max\left(C^{d}_{F}(\boldsymbol{X}), 0\right) \quad (2)$$

3.2. Optimal truss design under dynamic constraints

Unlike the static cases, the dynamic design optimisation problem involves both discrete cross-sections and continuous shape variables, subject to natural frequency constraints. The objective remains to minimise weight while satisfying the frequency bounds:

Find: $\boldsymbol{X} = \{x_1, x_2, \ldots, x_{n_{eg}}\}$

$\boldsymbol{Y} = \{y_1, y_2, \ldots, y_{n_s}\}$

To minimise: $f(\boldsymbol{X}, \boldsymbol{Y}) = W(\boldsymbol{X}, \boldsymbol{Y}) \quad \psi(\boldsymbol{X}, \boldsymbol{Y})^{\alpha}$

$$W(\boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{m} \gamma A_i l_i(\boldsymbol{Y})$$

$$x_i \in \boldsymbol{S_i}, \quad \boldsymbol{S_i} = \{S_{i1}, S_{i2}, \ldots, S_{ir_i}\}, \quad i = 1, 2, \ldots, n_{eg}$$

$$y_i^{min} \le y_i \le y_i^{max} \quad , i = 1, 2, \ldots, n_s$$

Subject to:

$$C^{U}_{k}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\omega_k}{\omega^{u}_{k}} - 1 \le 0, \quad k = 1, \quad 2, \quad \ldots, \quad n_{\omega}$$

$$C^{L}_{k}(\boldsymbol{X}, \boldsymbol{Y}) = 1 - \frac{\omega_k}{\omega^{l}_{k}} \le 0, \quad k = 1, \quad 2, \quad \ldots, \quad n_{\omega} \quad (3)$$

where, $\boldsymbol{Y}$ represents a vector containing shape variables. n_s denotes the number of shape variables. y_i^{min} and y_i^{max} are the lower and upper bounds for the ith shape variable, respectively. $C^{U}_{k}(.)$ and $C^{L}_{k}(.)$ indicate the upper and lower bound constraints for the kth frequency, respectively. ω_k is the kth frequency of the structure, ω^{u}_{k} and ω^{l}_{k} are the upper and lower bounds for the kth frequency, respectively, and n_{ω} is the number of constrained frequencies. Due to its nonlinear nature, this problem is highly sensitive to the penalty parameter α.

In the early stages of the optimisation process, most individuals explore the entire design space. A small penalty parameter is appropriate at this stage, allowing them to explore diverse regions. As the search progresses, individuals concentrate near the boundaries between infeasible and feasible designs, where optimal solutions are likely to be located. At this stage, some individuals may slightly violate design constraints, yet their objective function values remain close to those of individuals with feasible designs. Hence, the penalty parameter must be sufficiently large to enable the algorithm to effectively distinguish between feasible and infeasible designs. Sensitivity to the penalty parameter is more pronounced in structural optimisation with frequency constraints than with static constraints, owing to the higher nonlinearity. Consequently, an incremental approach is commonly adopted in the literature for adjusting the penalty parameter in structural optimisation under frequency constraints [45,60]. For the truss design optimisation problems in this study, α is initially set to 2 and linearly increased to 8 throughout the optimisation process [61]. The penalty function $\psi(\boldsymbol{X}, \boldsymbol{Y})$ is calculated by considering the frequency constraints as:

$$\psi(\boldsymbol{X}, \boldsymbol{Y}) = 1 + \sum_{k=1}^{n_{\omega}} \left(\max\left(C^{U}_{k}(\boldsymbol{X}, \boldsymbol{Y}), 0\right) + \max\left(C^{L}_{k}(\boldsymbol{X}, \boldsymbol{Y}), 0\right)\right) \quad (4)$$

4. Hyper-intelligent approach

The main idea of this study is to develop a hyper-intelligent (HI) framework in which individual solutions learn and adapt their behaviour based on an online feedback mechanism. The overall framework of the proposed algorithms is inspired by the principles of HH algorithms. This approach aims to design algorithms that can be applied to various structural optimisation problems, offering the potential for more consistent performance.

The proposed HI framework equips each individual with a high-level heuristic (HLH) subspace that acts as memory. This enables continuous learning and behavioural adaptation throughout the search process via an online feedback mechanism. A shared global low-level heuristic (LLH) space serves as a repository of meta-heuristic strategies that individuals explore and use to record experiences within their HLH subspaces. These recorded experiences guide the selection of appropriate meta-heuristics at different stages of optimisation. Based on the type of online learning mechanism implemented in the HLH subspaces, the two algorithmic variants are proposed based on the framework: HI with choice function (HI-CF), where each HLH subspace uses a choice function model, and HI with multi-armed bandit (HI-MAB), which employs a learning strategy inspired by the multi-armed bandit problem from game theory.

As discussed in Section 2, the LLH space in HH algorithms consists of a set of heuristics that are directly responsible for finding solutions to the problem. Most of selection HHs developed in the literature employ a set of so-called 'single-point' heuristics within the LLH space [54,55,62], where a single solution is generated in each iteration. In contrast, meta-heuristics are typically population-based algorithms in which a set of individual solution candidates explores the search space. The proposed approach aims to leverage the strengths of both HH and

meta-heuristic algorithms by integrating their concepts into a single framework.

The overall framework of the HI is graphically illustrated in Fig. 3. Let n-dimensional vectors $\boldsymbol{X}_0^t$, $\boldsymbol{X}_1^t$, $\cdots, \boldsymbol{X}_{\mathrm{N_p}-1}^t$ represent the individual solutions in a meta-heuristic algorithm. In the proposed framework, the LLH space consists of a set of meta-heuristic algorithms, each of which could potentially update the individuals and produce $\boldsymbol{X}_0^{t+1}$, $\boldsymbol{X}_1^{t+1}$, $\cdots$, $\boldsymbol{X}_{\mathrm{N_p}-1}^{t+1}$ vectors. Each individual has its own sub-HLH space, specifically defined to guide its behaviour during the search process. Hence, the HLH space in the HI framework is denoted by a tuple of size N_P as:

$$\mathscr{H}^t = \left(\mathscr{h}_0^t, \quad \mathscr{h}_1^t, \quad \ldots, \quad \mathscr{h}_{N_\mathrm{P}-1}^t\right) \tag{5}$$

where $\mathscr{h}_0^t$, $\mathscr{h}_1^t$, …, $\mathscr{h}_{N_\mathrm{P}-1}^t$ represent the HLH subspaces of the individuals at iteration t. Each HLH subspace, $\mathscr{h}_i^t$, operates independently of other HLH subspaces and plays multiple learning roles throughout the search process. This results in an independent learning experience for each individual. First, it tracks how frequently each meta-heuristic strategy in the LLH space is applied. Second, it evaluates each individual's history of success or failure with each strategy. Third, it uses the individual's historical experience to determine which meta-heuristic strategy should be applied in the next iteration. Each HLH subspace can be viewed as an individual's memory, helping it adjust its behaviour during the search process. As individuals accumulate different experiences, this is expected to lead to diverse behavioural patterns during the optimisation process.

In the following subsections, the meta-heuristics employed in the LLH space are first introduced, followed by a detailed discussion of the mechanisms used for the HLH subspaces.

4.1. LLHs

In the proposed algorithms, meta-heuristic algorithms are embedded within the LLH space, rather than using single-point methods. Although any set of meta-heuristics can be considered at this level, this study employs a selection of widely used algorithms, specifically, five variants of the cultural algorithm (CA) and three variants of the differential evolution (DE) algorithm, within the LLH space. Depending on the HLH strategy, one of the meta-heuristic search strategies is applied to update the position of an individual in the search space. A brief overview of the employed meta-heuristics is presented below.

4.1.1. Cultural algorithms (CAs)

CAs, originally developed by Reynolds [25,63–65], are population-based meta-heuristics inspired by bio-cultural evolutionary theory, in which genes and culture are two interacting forms of inheritance [24]. The CAs perform based on the population and belief spaces. The population space consists of a set of N_P individuals, while the belief space stores a set of knowledge components containing information about the individuals and the search space. The situational and normative are two key knowledge components that have been widely used in the literature for CAs [66]. Hence, the belief space at iteration t is represented as a tuple as $\mathscr{B}^t = (\mathscr{S}^t, \mathscr{N}^t)$, in which $\mathscr{S}^t$ and $\mathscr{N}^t$ represent the situational and normative knowledge components at iteration t, respectively.

The situational knowledge stores the individual with the best objective function value at iteration t, which is indicated by vector $\mathscr{S}^t = \left[\mathscr{s}_0^t, \mathscr{s}_1^t, \ldots, \mathscr{s}_{n-1}^t\right]$. The normative knowledge stores a set of behavioural standards recorded from the evolution process of individuals, which can be expressed as [24]:

$$\mathscr{N}^t = \left(\mathscr{n}_0^t, \quad \mathscr{n}_1^t, \quad \ldots, \quad \mathscr{n}_{n-1}^t\right) \tag{6}$$

in which:

$$\mathscr{n}_j^t = \left(I_j^t, L_j^t, U_j^t\right), \quad j = 0, \quad 1, \quad \ldots, \quad n-1 \tag{7}$$

In Eqs. (6) and (7), $\mathscr{n}_j^t$ represents the normative knowledge specific to the jth dimension. The interval $I_j^t = \left[x_{\min,j}^t, x_{\max,j}^t\right]$ defines the behavioural interval, where $x_{\min,j}^t$ and $x_{\max,j}^t$ are the lower and upper bounds, respectively. Similarly, L_j^t and U_j^t are the objective function values corresponding to these bounds. The subscript j indicates the jth dimension of the problem, while the superscript t denotes the iteration number.

The positions of the individuals within the search space are updated based on the belief space knowledge components and influence functions. Different versions of CAs employ different influence functions. In this study, the different versions of CAs are categorised in terms of the type of influence functions they use as follows [24,67]:

- **CA-I:**

$$x_{ij}^{t+1} = x_{ij}^t + \mathrm{size}\left(I_j^t\right) N_{ij}(0,1) \tag{8}$$

- **CA-II:**

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t + \left|\sigma_{ij}^t N_{ij}(0,1)\right|, & \text{if } x_{ij}^t < \mathscr{s}_j^t \\ x_{ij}^t - \left|\sigma_{ij}^t N_{ij}(0,1)\right|, & \text{if } x_{ij}^t > \mathscr{s}_j^t \\ x_{ij}^t + \sigma_{ij}^t N_{ij}(0,1), & \text{otherwise} \end{cases} \tag{9}$$

- **CA-III:**

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t + \left|\mathrm{size}\left(I_j^t\right) N_{ij}(0,1)\right|, & \text{if } x_{ij}^t < s_j^t \\ x_{ij}^t - \left|\mathrm{size}\left(I_j^t\right) N_{ij}(0,1)\right|, & \text{if } x_{ij}^t > s_j^t \\ x_{ij}^t + \mathrm{size}\left(I_j^t\right) N_{ij}(0,1), & \text{otherwise} \end{cases} \tag{10}$$

- **CA-IV:**

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t + \left|\mathrm{size}\left(I_j^t\right) N_{ij}(0,1)\right|, & \text{if } x_{ij}^t < x_{\min,j}^t \\ x_{ij}^t - \left|\mathrm{size}\left(I_j^t\right) N_{ij}(0,1)\right|, & \text{if } x_{ij}^t > x_{\max,j}^t \\ x_{ij}^t + \beta \mathrm{size}\left(I_j^t\right) N_{ij}(0,1), & \text{otherwise} \end{cases} \tag{11}$$

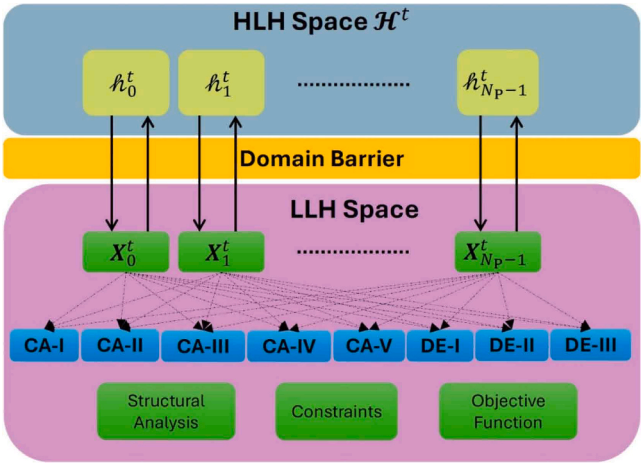


Fig. 3. Overall framework of the proposed hyper-intelligent (HI) framework.

- **CA-V** [68]**:**

$$x_{i,j}^{t+1} = \begin{cases} x_{\min,j}^{t} + rand\left(x_{\min,j}^{t} - x_{i,j}^{t}\right) + rand\left(s_j^{t} - x_{i,j}^{t}\right), & \text{if } x_{i,j}^{t} < x_{\min,j}^{t} \\ x_{\max,j}^{t} + rand\left(x_{\max,j}^{t} - x_{i,j}^{t}\right) + rand\left(s_j^{t} - x_{i,j}^{t}\right), & \text{if } x_{i,j}^{t} > x_{\max,j}^{t} \\ s_j^{t} + rand\left(x_{i,j}^{t} - s_j^{t}\right), & \text{otherwise} \end{cases} \tag{12}$$

where $N_{i,j}(0,1)$ represents the normal distribution with a mean value of 0 and a standard deviation of 1. β is a positive constant parameter. $\sigma_{i,j}^{t} = \alpha\left(x_j^{\max} - x_j^{\min}\right)$ is a strategy parameter in which $x_j^{\min}$ and $x_j^{\max}$ represent the lower and upper bounds of the search space in the jth dimension, respectively, and α is a constant parameter.

In each iteration, the normative and situational knowledge components are updated by using a set of top N_{accept} selected individuals as follows [25]:

$$\mathscr{S}^{t+1} = \begin{cases} \boldsymbol{X}_l^{t}, & \text{if } f(\boldsymbol{X}_l^{t}) < f(\mathscr{S}^{t}) \\ \mathscr{S}^{t}, & \text{otherwise} \end{cases} \tag{13}$$

$$x_{\min,j}^{t+1} = \begin{cases} x_{l,j}^{t}, & \text{if } x_{l,j}^{t} \le x_{\min,j}^{t} \text{ or } f(\boldsymbol{X}_l^{t}) < L_j^{t} \\ x_{\min,j}^{t}, & \text{otherwise} \end{cases} \tag{14}$$

$$x_{\max,j}^{t+1} = \begin{cases} x_{l,j}^{t}, & \text{if } x_{l,j}^{t} \ge x_{\max,j}^{t} \text{ or } f(\boldsymbol{X}_l^{t}) < U_j^{t} \\ x_{\max,j}^{t}, & \text{otherwise} \end{cases} \tag{15}$$

$$L_j^{t+1} = \begin{cases} f(\boldsymbol{X}_l^{t}), & \text{if } x_{l,j}^{t} \le x_{\min,j}^{t} \text{or } f(\boldsymbol{X}_l^{t}) < L_j^{t} \\ L_j^{t}, & \text{otherwise} \end{cases} \tag{16}$$

$$U_j^{t+1} = \begin{cases} f(\boldsymbol{X}_l^{t}), & \text{if } x_{l,j}^{t} \ge x_{\max,j}^{t} \text{ or } f(\boldsymbol{X}_l^{t}) < U_j^{t} \\ U_j^{t}, & \text{otherwise} \end{cases} \tag{17}$$

In the above equations, $x_{l,j}^{t}$ represents the jth variable of the lth accepted individual at iteration t.

Algorithm 1 presents the pseudo-code of CAs. In this study, the number of accepted individuals is set to $N_{\text{accept}} = 0.20N_{\text{P}}$, β is taken to 0.10, and α is assumed to be 0.10 [69].

Algorithm 1. Pseudo-code of the CAs (slightly modified from Ref. [24])

Set $t = 0$;
Initialise the population space with a set N_{P} of random individuals within the search space;
Initialise the belief space $\mathscr{B}^0$ with suitable initial assumptions for cultural variants;
while *stopping criteria are not met* **do**
  Set $t = t + 1$;
  Calculate the objective function values of individuals;
  Select a set of N_{accept} individuals from the population;
  Employ the selected individuals to adjust the cultural variants and obtain $\mathscr{B}^{t+1}$ using equations (13-17)
  Update the positions of individuals using one of influence functions in equations (8-12);
end

4.1.2. Differential evolution (DE)

As its name suggests, the DE algorithm is a population-based evolutionary algorithm proposed by Storn and Price [70,71]. The DE algorithm consists of mutation and cross-over operators. For an individual solution $\boldsymbol{X}_i^{t}$, the mutation operator creates a mutant vector $\boldsymbol{V}_i^{t+1}$. There are several versions of mutation operators proposed for DE algorithm in the literature. In this study, the following three mutation formulas are considered in the LLHs to generate new individuals:

- *DE-I* [72]

$$\boldsymbol{V}_i^{t+1} = \boldsymbol{X}_{r_1}^{t} + F\left(\boldsymbol{X}_{\text{best}}^{t} - \boldsymbol{X}_{r_2}^{t}\right) + F(\boldsymbol{X}_{r_3}^{t} - \boldsymbol{X}_{r_4}^{t}) \tag{18}$$

- DE-II [73]

$$\boldsymbol{V}_i^{t+1} = \boldsymbol{X}_{r_1}^{t} + F\left(\boldsymbol{X}_{r_2}^{t} - \boldsymbol{X}_i^{t}\right) + F(\boldsymbol{X}_{r_3}^{t} - \boldsymbol{X}_{r_4}^{t}) \tag{19}$$

- DE-III [73]

$$\boldsymbol{V}_i^{t+1} = \boldsymbol{X}_{r_1}^{t} + F(\boldsymbol{X}_{\text{best}}^{t} - \boldsymbol{X}_i^{t}) + F(\boldsymbol{X}_{r_3}^{t} - \boldsymbol{X}_{r_4}^{t}) \tag{20}$$

where F is a scale factor.

The cross-over operator is then applied to construct the updated position of individuals as:

$$\boldsymbol{X}_{ik}^{t+1} = \begin{cases} \boldsymbol{V}_{ik}^{t+1}, & \text{if } rand < CR \text{ or } k = randi(0, n-1) \\ \boldsymbol{X}_{ik}^{t}, & \text{otherwise} \end{cases}, \; k = 0, 1, \ldots, n-1 \tag{21}$$

where *rand* is a random number with uniform distribution between 0 and 1. $randi(a,b)$ is a random integer number between the a and b, and CR is a crossover rate control parameter. The parameters F and CR are set to 0.5 and 0.4, respectively, based on the consistent performance of the DE in structural optimisation problems reported in [47].

4.2. High-level heuristics (HLHs)

The meta-heuristics in the LLH space discussed in the previous subsection need to be properly managed through effective HLH strategies. In the proposed HI framework, each individual has its own HLH subspace, designed for two strategies: the choice function (CF) and the multi-armed bandit with failure reward rate (FRRMAB) methods. These HLH subspaces guide the selection of the most promising meta-heuristic in the LLH space. In the following subsections, the CF and MAB methods

are first described, and then their definitions are extended to the population level.

4.2.1. Choice function (CF)

The CF method was originally proposed by Cowling et al. [74] to score LLHs using various strategies for solving the personal scheduling problem. This approach has since been further developed and extended by several studies in the literature [75–78]. The CF ranks the LLHs based on historic performance data collected during the solution-finding process. It records the information on objective function improvements or deteriorations resulting from the application of each LLH, the elapsed time since each LLH was last applied, and the impact of consecutive applications of different pairs of LLHs on solution quality.

Let us assume $\mathscr{C}^t = \left[c_0^t,\ c_1^t,\ \ldots,\ c_{N_{\text{LLH}}-1}^t\right]$ be the credit vector in which c_j^t shows the credit value for the jth LLH. Let also λ^t be the index of the LLH selected at iteration t. The equation for credit assignment to each LLH in the CF method is expressed as follows [74,79]:

$$c_j^{t+1} = \varphi \mathscr{I}_j^t + \varphi \mathscr{I}_{j\lambda^t}^t \quad + (1-\varphi)\mathscr{I}_j^t, \quad j = 0, 1, \ldots, N_{\text{LLH}} - 1 \tag{22}$$

where c_j^{t+1} represents the choice function for the jth LLH at iteration $t+1$. $\mathscr{I}_j^t$ indicates the most recent objective function improvement or deterioration recorded from the application of the jth LLH until iteration t. $\mathscr{I}_{j\lambda^t}^t$ denotes the most recent objective function improvement or deterioration when the jth LLH was applied immediately after λ^tth LLH. $\mathscr{I}_j^t$ symbolises the elapsed time since the jth LLH was last applied, and φ is the exploitation factor that controls the learning process. The value of φ depends on the achieved performance from the application of a given LLH.

The objective function improvement resulted from the application of the λ^tth LLH at iteration t is expressed as:

$$\mathscr{I}_{\lambda^t}^t = \frac{f(\boldsymbol{X}^t) - f(\boldsymbol{X}^{t+1})}{f(\boldsymbol{X}^t)} \tag{23}$$

where $f(\boldsymbol{X}^{t+1})$ and $f(\boldsymbol{X}^t)$ represent the current and previous objective function values of the solution $\boldsymbol{X}$, respectively.

When the application of a given LLH results in an improvement in the objective function value, it may indicate that this LLH is an effective strategy at the current solution-finding stage, and its probability of being selected in the future should be increased. Conversely, when the currently selected LLH does not yield better results, its selection probability should be reduced, and priority should be given to other LLHs that have not been applied recently. Castro et al. [80] suggested setting the parameter φ to 0.99 when the application of the jth LLH results in an improvement in the objective function; otherwise, it is set to $\varphi = 0.01$. Similar approaches are also proposed by other Refs. [81–83]. The approach recommended by Castro et al. [80] is adopted in this study.

4.2.2. Failure reward rate of multi-armed bandit (FRRMAB)

The multi-armed bandit (MAB) is a classical problem in game theory in which a player or learner faces a set of arms or actions, each yielding different rewards upon selection [84]. The player is only aware of the reward obtained from the chosen arm, while the probabilities of receiving rewards from the other arms remain unknown [85]. The main objective of the player is to maximise the accumulated rewards over a sequence of selections [85]. A wide range of algorithms has been proposed in the literature to solve the MAB problem [86].

Similar to the MAB problem, the HLH in HH algorithms plays the role of a player, selecting the most promising LLH to achieve the desired outcomes, while the success probabilities of other LLHs remain unknown. This similarity between the role of the HLH in HH algorithms and the player in the MAB problem has motivated the adoption of MAB algorithms within the HLH space [87,88]. The upper confidence bound (UCB) algorithm, originally developed for the MAB problem [89], was adapted by Li et al. [90] for adaptive operator selection in evolutionary algorithms and has since been widely employed by HH algorithms in the literature [54]. This study also focuses on the failure reward rate of multi-armed bandit (FRRMAB) approach proposed by Li et al. [90].

The FRRMAB method records the historical fitness improvement rate (FIR) resulting from the application of various LLHs. These FIR values are used to evaluate the performance of each LLH strategy and to select the most appropriate one for future iterations. Since the performance of a given LLH strategy is often more dependent on its recent performance, the FRRMAB method retains a fixed number of FIR values while discarding older ones, which also improves computational efficiency [90]. Let us assume the λ^tth LLH is applied at iteration t, and let define the FIR at a given iteration as:

$$FIR_{\lambda^t}^t = \frac{f(\boldsymbol{X}^t) - f(\boldsymbol{X}^{t+1})}{f(\boldsymbol{X}^t)} \tag{24}$$

where $\boldsymbol{X}^{t+1}$ and $\boldsymbol{X}^t$ are the current and previous solutions, respectively. The $FIR_{\lambda^t}^t$ is positive if the application of the λ^tth LLH is successful, and negative if the application is unsuccessful, resulting in a deterioration of the fitness function value. The $FIR_{\lambda^t}^t$ values obtained from the application of various LLHs are stored within a matrix as follows:

$$\mathscr{T}^t = \begin{bmatrix} \mathscr{I}_{00}^t & \cdots & \mathscr{I}_{0(W-1)}^t \\ \mathscr{I}_{10}^t & \cdots & \mathscr{I}_{1(W-1)}^t \end{bmatrix} = \begin{bmatrix} \cdots & \cdots & \lambda^t \\ \cdots & \cdots & FIR_{\lambda^t}^t \end{bmatrix}, \quad \mathscr{T}^t \in \mathbb{R}^{2\times W} \tag{25}$$

where $\mathscr{T}^t$ represents the sliding window matrix at iteration t and $\mathscr{I}_{ij}^t$ indicates its components. W is the sliding window size, which is defined to control memory usage. The first row of matrix $\mathscr{T}^t$ represents the indices of the applied LLHs, while the second row shows the FIR values corresponding to each selected LLH. When the number of stored FIR values exceeds the sliding window size W, the oldest FIR value and its associated index are discarded, and a new FIR and its associated index are added to the matrix $\mathscr{T}^t$.

Let also $\mathscr{G}^t = \left[\mathscr{g}_0^t, \mathscr{g}_1^t, \mathscr{g}_2^t, \ldots, \mathscr{g}_{N_{\text{LLH}}-1}^t\right] \in \mathbb{R}^{N_{\text{LLH}}}$ be the accumulated reward vector, where $\mathscr{g}_i^t$ indicates the accumulated FIR value obtained from the ith LLH at iteration t, is calculated as:

$$\mathscr{g}_j^t = \sum_{\substack{l=0 \\ \mathscr{I}_{0l}=j}}^{W-1} \mathscr{I}_{1l}^t, \quad j = 0, 1, \ldots, N_{\text{LLH}} - 1 \tag{26}$$

The LLHs are ranked based on their accumulated FIR values, $\mathscr{g}_j^t$. Then, the decay parameter is obtained for each LLH as:

$$\mathscr{d}_j^t = D^{\text{Rank}_j} \mathscr{g}_j^t, \quad j = 0, 1, \ldots, N_{\text{LLH}} - 1 \tag{27}$$

where D represents the decaying factor and Rank_j denotes the rank of the jth LLH. The credit value for each LLH, represented by $\mathscr{I}_j^t$, is obtained as:

$$\mathscr{I}_j^t = \frac{\mathscr{d}_j^t}{\sum_{k=0}^{N_{\text{LLH}}-1} \mathscr{d}_k^t}, \quad j = 0,\ 1,\ \ldots,\ N_{\text{LLH}} - 1 \tag{28}$$

Let $\mathscr{M}^{t+1} = \left[m_0^{t+1},\ m_1^{t+1},\ \ldots,\ m_{N_{\text{LLH}}-1}^{t+1}\right] \in \mathbb{R}^{N_{\text{LLH}}}$ be the estimated quality

vector, in which m_j^{t+1} indicates the estimated quality of the jth LLH at iteration $t+1$. The estimated qualities of LLHs are calculated as:

$$m_j^{t+1} = \ell_j^t + C\sqrt{\frac{2\ln\sum_{k=0}^{N_{\text{LLH}}-1} a_k^t}{a_j^t}}, \quad j = 0,\ 1,\ \ldots,\ N_{\text{LLH}} - 1 \tag{29}$$

where C denotes a scaling factor that controls the exploration and exploitation capabilities, and a_j^t represents the number of times that the jth LLH has been applied, which is obtained from the sliding window matrix $\mathcal{T}^t$ in Eq. (25). Based on the estimated quality vector $\mathcal{M}^t$, the FRRMAB method probabilistically selects LLH to generate new solutions for the problem.

4.3. Expansion

As mentioned earlier, each individual in the proposed HI framework has its own HLH subspace. To model this, the formulations presented in Subsection 4.2 are defined and expanded for each individual. In this subsection, the CF and FRRMAB formulations are extended to the population level, and the pseudo-codes for the two proposed HI algorithms are presented.

4.3.1. Expanded CF

Let us assume that λ_i^t represents the index for the selected LLH at iteration t to be used by the ith individual to update its position. To define the credit vector for each individual, the vector in Eq. (22) needs to be created for each individual as:

$$\mathcal{C}_i^t = \left[c_{i0}^t,\ c_{i1}^t,\ \ldots,\ c_{i(N_{\text{LLH}}-1)}^t\right], \quad i = 0,\ 1,\ \ldots,\ N_{\text{P}} - 1 \tag{30}$$

where c_{ij}^t represents the credit value of the jth LLH of the ith individual, that is defined similar to Eq. (22) as:

$$c_{ij}^{t+1} = \varphi\ell_{ij}^t + \varphi r_{ij\lambda_i^t}^t + (1-\varphi)\ell_{ij}^t,\ i = 0,\ 1,\ \ldots,\quad N_{\text{P}} - 1,\quad j = 0,\ 1,\ \ldots,\ N_{\text{LLH}} - 1 \tag{31}$$

where ℓ_{ij}^t represents the improvement or deterioration in the objective function of the ith individual resulting from the latest application of the jth LLH since the beginning of the optimisation process. $r_{ij\lambda_i^t}^t$ indicates the improvement or deterioration in the objective function of the ith individual when the jth LLH is immediately applied after the λ_i^tth LLH, and ℓ_{ij}^t denotes the elapsed time since the jth LLH was last applied. ℓ_{ij}^t is obtained as:

$$\ell_{ij}^t = \frac{f(\boldsymbol{X}_i^t) - f(\boldsymbol{X}_i^{t+1})}{f(\boldsymbol{X}_i^t)}, \quad i = 0,\ 1,\ \ldots,\ N_{\text{P}} - 1 \tag{32}$$

It should be noted that the credit vector is initialised as an unit vector at the beginning, i.e., $\mathcal{C}_i^0 = \left[c_{i0}^0,\ c_{i1}^0,\ \ldots,\ c_{i(N_{\text{LLH}}-1)}^0\right] = [1\ 1\ 1\ \ldots\ 1]$.

4.3.2. Expanded FRRMAB

Let again λ_i^t be the index of the LLH selected by the ith individual at iteration t, which results in the following improvement in the objective function:

$$FIR_{\lambda_i^t}^t = \frac{f(\boldsymbol{X}_i^t) - f(\boldsymbol{X}_i^{t+1})}{f(\boldsymbol{X}_i^t)}, \quad i = 0,\ 1,\ \ldots,\ N_{\text{P}} - 1 \tag{33}$$

where $\boldsymbol{X}_i^{t+1}$ and $\boldsymbol{X}_i^t$ are the current and previous position of the individual i. This FIR value is then stored within the sliding window matrix defined for the ith individual, $\mathcal{T}_i^t$, as follows:

$$\mathcal{T}_i^t = \begin{bmatrix} \ell_{i00}^t & \cdots & \ell_{i0(W-1)}^t \\ \ell_{i10}^t & \cdots & \ell_{i1(W-1)}^t \end{bmatrix} = \begin{bmatrix} \cdots & \cdots & \lambda_i^t \\ \cdots & \cdots & FIR_{\lambda_i^t}^t \end{bmatrix}, \quad \mathcal{T}_i^t \in \mathbb{R}^{2\times W},\ i = 0,\ 1,\ \ldots,\ N_{\text{P}} - 1 \tag{34}$$

The accumulated reward vector is defined for the ith individuals as $\mathcal{G}_i^t = \left[g_{i0}^t, g_{i1}^t, g_{i2}^t,\ \ldots, g_{i(N_{\text{LLH}}-1)}^t\right] \in \mathbb{R}^{N_{\text{LLH}}}$, where the components are obtained as:

$$g_{ij}^t = \sum_{\substack{l=0 \\ \ell_{i0l}=j}}^{W-1} \ell_{i1l}, \quad j = 0,\ 1,\ \ldots,\ N_{\text{LLH}} - 1, \quad i = 0,\ 1,\ \ldots,\ N_{\text{P}} - 1 \tag{35}$$

in which g_{ij}^t represents the accumulated FIR value obtained by the ith individual for the jth LLH at iteration t.

The LLHs are ranked for each individual based on their g_{ij}^t values. Thereafter, the decay parameter of the ith individual for the jth LLH, represented by d_{ij}^t, is calculated as:

$$d_{ij}^t = D^{\text{Rank}_j} g_{ij}^t, \quad i = 0,\quad 1,\quad \ldots,\quad N_{\text{P}} - 1,\quad j = 0,\quad 1,\quad \ldots,\quad N_{\text{LLH}} - 1 \tag{36}$$

Then, the credit value for each LLH from the ith individual's viewpoint is obtained as follows:

$$\ell_{ij}^t = \frac{d_{ij}^t}{\sum_{k=0}^{N_{\text{LLH}}-1} d_{ik}^t}, \quad i = 0,\ 1,\ \ldots,\ N_{\text{P}} - 1, \quad j = 0,\ 1,\ \ldots,\ N_{\text{LLH}} - 1 \tag{37}$$

Finally, the estimated quality of each LLH for each individual is obtained by the following equation:

$$m_{ij}^{t+1} = \ell_{ij}^t + C\sqrt{\frac{2\ln\sum_{k=0}^{N_{\text{LLH}}-1} a_{ik}^t}{a_{ij}^t}}, \quad i = 0,\ 1,\ \ldots,\ N_{\text{P}} - 1, \quad j = 0,\ 1,\ \ldots,\ N_{\text{LLH}} - 1 \tag{38}$$

where a_{ij}^t represents the number of times that the jth LLH has been applied by the ith individual in the past, which is obtained from the sliding window matrix $\mathcal{T}_i^t$ in Eq. (34). The parameters of the FRRMAB are taken as $W = 20$, $D = 1$, and $C = 0.5$, partly based on Ref. [90].

4.4. Algorithmic representation

Depending on the type of HLH subspaces, two HI algorithms are proposed: the hyper-intelligent algorithm with the choice function (HIA-CF) and the hyper-intelligent algorithm with the multi-armed bandit (HIA-MAB). The HIA-CF utilises the extended version of CF method explained in Subsection 4.3.1 as its HLH subspaces, while the HIA-MAB uses the MAB method presented in Subsection 4.3.2 to define HLH subspaces for individuals. Algorithm 2 and Algorithm 3 present the pseudo-codes for the HIA-CF and HIA-MAB approaches, respectively.

Algorithm 2. Pseudo-code of the proposed hyper-intelligent algorithm with choice function (HIA-CF)

Begin

Set $t = 0$

Set the internal parameters of meta-heuristics

Initialise the CF parameters of HLH space for each individual as:

$$\boldsymbol{\mathcal{J}}_i^0 = \left[\jmath_{i0}^0, \jmath_{i1}^0, \jmath_{i2}^0, \dots, \jmath_{i(N_{\mathrm{LLH}}-1)}^0\right] = [0\ 0\ 0\ \dots\ 0] \in \mathbb{R}^{N_{\mathrm{LLH}}}$$

$$\boldsymbol{\mathcal{R}}_i^0 = \begin{bmatrix} r_{i00}^0 & \cdots & r_{i0(N_{\mathrm{LLH}}-1)}^0 \\ \vdots & \ddots & \vdots \\ r_{i(N_{\mathrm{LLH}}-1)0}^0 & \cdots & r_{i(N_{\mathrm{LLH}}-1)(N_{\mathrm{LLH}}-1)}^0 \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{N_{\mathrm{LLH}} \times N_{\mathrm{LLH}}}$$

$$\boldsymbol{\mathcal{T}}_i^0 = \left[t_{i0}^0, t_{i1}^0, t_{i2}^0, \dots, t_{i(N_{\mathrm{LLH}}-1)}^0\right] = [0\ 0\ 0\ \dots\ 0] \in \mathbb{R}^{N_{\mathrm{LLH}}}$$

$$\boldsymbol{\mathcal{C}}_i^0 = \left[c_{i0}^0, c_{i1}^0, \dots, c_{i(N_{\mathrm{LLH}}-1)}^0\right] = [1\ 1\ 1\ \dots\ 1] \in \mathbb{R}^{N_{\mathrm{LLH}}}$$

Initialise the population space $\boldsymbol{\mathcal{P}}^0$ randomly within the search space, represented by $\boldsymbol{X}_0^0, \boldsymbol{X}_1^0, \dots, \boldsymbol{X}_{N_{\mathrm{P}}-1}^0$ vectors

Evaluate the objective functions of individuals denoted by $f(\boldsymbol{X}_i^0)$

Initialise and update the belief space of CAs as: $\boldsymbol{\mathcal{B}}^0 = (\boldsymbol{\mathcal{S}}^0, \boldsymbol{\mathcal{N}}^0)$

While** the termination criterion is not satisfied **do

***for** $\boldsymbol{i = 0}$ **to** $\boldsymbol{N_{\mathrm{P}} - 1}$*

Select the λ_i^tth LLH to be applied to the ith individual with the probabilities of $\boldsymbol{\mathcal{C}}_i^t = \left[c_{i,0}^t, c_{i,1}^t, \dots, c_{i,N_{\mathrm{LLH}}-1}^t\right]$

Update the position of the ith individual using the selected λ_i^tth LLH and obtain $\boldsymbol{X}_i^{t+1}$

Evaluate the objective function of the ith individual and obtain $f(\boldsymbol{X}_i^{t+1})$

Update the HLH space of the ith individual as follows:

$\jmath_{i\lambda_i^t}^t = \frac{f(\boldsymbol{X}_i^t) - f(\boldsymbol{X}_i^{t+1})}{f(\boldsymbol{X}_i^t)}$

Update the vector $\boldsymbol{\mathcal{J}}_i^t$ as:

$\boldsymbol{\mathcal{J}}_i^t = \left[\jmath_{i0}^t, \jmath_{i1}^t, \jmath_{i2}^t, \dots, \jmath_{i\lambda_i^t}^t, \dots, \jmath_{i(N_{\mathrm{LLH}}-1)}^t\right]$

Update the matrix $\boldsymbol{\mathcal{R}}_i^t$ and obtain $\boldsymbol{\mathcal{R}}_i^{t+1}$ as:

$r_{i\lambda_i^t\lambda_i^{t-1}}^t = \jmath_{i\lambda_i^t}^t$

$$\boldsymbol{\mathcal{R}}_i^t = \begin{bmatrix} r_{i00}^t & \cdots & \cdots & \cdots & r_{i0(N_{\mathrm{LLH}}-1)}^t \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \vdots & \ddots & r_{i\lambda_i^t\lambda_i^{t-1}}^t & \ddots & \vdots \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{i(N_{\mathrm{LLH}}-1)0}^t & \cdots & \cdots & \cdots & r_{i(N_{\mathrm{LLH}}-1)(N_{\mathrm{LLH}}-1)}^t \end{bmatrix}$$

if $\jmath_{i,\lambda_i^t}^t > 0$

$\varphi = 0.99$

else

$\varphi = 0.01$

endif

***for** $j = \boldsymbol{0}$ **to** $\boldsymbol{N_{\mathrm{LLH}} - 1}$*

***if** $\boldsymbol{t} == \boldsymbol{0}$*

$c_{ij}^{t+1} = \varphi \jmath_{ij}^t + (1 - \varphi) t_{ij}^t$

else

$c_{ij}^{t+1} = \varphi \jmath_{ij}^t + \varphi\, r_{ij\lambda_i^t}^t + (1 - \varphi) t_{ij}^t$

endif

end for

$t_{i\lambda_i^t}^t = 0$

$\boldsymbol{\mathcal{T}}_i^{t+1} = \boldsymbol{\mathcal{T}}_i^t + 1$

$\boldsymbol{\mathcal{R}}_i^{t+1} = \boldsymbol{\mathcal{R}}_i^t$

$\boldsymbol{\mathcal{J}}_i^{t+1} = \boldsymbol{\mathcal{J}}_i^t$

***if** $\boldsymbol{X}_i^{t+1}$ **is not better than** $\boldsymbol{X}_i^t$*

$\boldsymbol{X}_i^{t+1} = \boldsymbol{X}_i^t$

end for

Update the belief space and obtain $\boldsymbol{\mathcal{B}}^{t+1} = (\boldsymbol{\mathcal{S}}^{t+1}, \boldsymbol{\mathcal{N}}^{t+1})$

Set $t = t + 1$

end while

end

Algorithm 3. Pseudo-code of the proposed hyper-intelligent algorithm with multiarmed bandit (HIA-MAB)

Begin
Set $t = 0$
Set the internal parameters of meta-heuristics
Initialise the MAB parameters of HLH space for each individual as:
Set the sliding window size W
$\boldsymbol{\mathcal{F}}_i^0 = \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{2\times W}$
$\boldsymbol{\mathcal{G}}_i^0 = \left[\mathcal{g}_{i0}^0, \mathcal{g}_{i1}^0, \mathcal{g}_{i2}^0, \cdots, \mathcal{g}_{i(N_{\mathrm{LLH}}-1)}^0\right] = [0\ 0\ 0\ \ldots\ 0] \in \mathbb{R}^{N_{\mathrm{LLH}}}$
$\boldsymbol{\mathcal{M}}_i^0 = \left[m_{i0}^0, m_{i1}^0, \ldots, m_{i(N_{\mathrm{LLH}}-1)}^0\right] = [1\ 1\ 1\ \ldots\ 1] \in \mathbb{R}^{N_{\mathrm{LLH}}}$
Initialise the population space $\boldsymbol{\mathcal{P}}^0$ randomly within the search space, represented by $\boldsymbol{X}_0^0, \boldsymbol{X}_1^0, \ldots, \boldsymbol{X}_{N_{\mathrm{P}}-1}^0$ vectors
Evaluate the objective functions of individuals denoted by $f(\boldsymbol{X}_i^0)$
Initialise and update the belief space of CAs as: $\boldsymbol{\mathcal{B}}^0 = (\boldsymbol{\mathcal{S}}^0, \boldsymbol{\mathcal{N}}^0)$
While *the termination criterion is not satisfied* ***do***
  for $\boldsymbol{i = 0}$ ***to*** $\boldsymbol{N_{\mathrm{P}} - 1}$
    Select the λ_i^t th LLH to be applied to the ith individual with the probabilities of $\boldsymbol{\mathcal{M}}_i^t = \left[m_{i0}^t, m_{i1}^t, \ldots, m_{i(N_{\mathrm{LLH}}-1)}^t\right]$
    Update the position of the ith individual using the selected λ_i^tth LLH and obtain $\boldsymbol{X}_i^{t+1}$
    Evaluate the objective function of the ith individual and obtain $f(\boldsymbol{X}_i^{t+1})$
    Update the HLH space of the ith individual as follows:
    $FIR_{\lambda_i^t}^t = \frac{f(X_i^t) - f(X_i^{t+1})}{f(X_i^t)}$
    Store the $FIR_{j_t^*}^t$ within the sliding window as:
    $\boldsymbol{\mathcal{F}}_i^t = \begin{bmatrix} \mathcal{f}_{i00}^t & \cdots & \mathcal{f}_{i0(W-1)}^t \\ \mathcal{f}_{i10}^t & \cdots & \mathcal{f}_{i1(W-1)}^t \end{bmatrix} = \begin{bmatrix} \cdots & \cdots & \lambda_i^t \\ \cdots & \cdots & FIR_{\lambda_i^t}^t \end{bmatrix}$
    for $\boldsymbol{j = 0}$ ***to*** $\boldsymbol{N_{\mathrm{LLH}} - 1}$
      $\mathcal{g}_{ij}^t = \sum_{\substack{l=0 \\ \mathcal{f}_{i0l}=j}}^{W-1} \mathcal{f}_{i1l}$
    end for
    Rank the LLH based on their gain values $\mathcal{g}_{ik}^t$
    for $\boldsymbol{j = 0}$ ***to*** $\boldsymbol{N_{\mathrm{LLH}} - 1}$
      $d_{ij}^t = D^{\mathrm{Rank}_k} \mathcal{g}_{ij}^t$
    end for
    for $\boldsymbol{j = 0}$ ***to*** $\boldsymbol{N_{\mathrm{LLH}} - 1}$
      $\ell_{ij}^t = \frac{d_{ij}^t}{\sum_{k=0}^{N_{\mathrm{LLH}}-1} d_{ij}^t}$
    end for
    Form the FFR vector denoted by $\boldsymbol{\mathcal{L}}_i^t = \left[\ell_{i0}^t, \ell_{i1}^t, \ldots, \ell_{i(N_{\mathrm{LLH}}-1)}^t\right]$
    Define vector $\boldsymbol{A}_i^t = \left[a_{i0}^t, a_{i1}^t, \ldots, a_{i(N_{\mathrm{LLH}}-1)}^t\right] = [0, 0, \ldots, 0] \in \mathbb{R}^{N_{\mathrm{LLH}}}$
    for $\boldsymbol{k = 0}$ ***to*** $\boldsymbol{W - 1}$
      $a_{i(\mathcal{f}_{0,k}^t)}^t \mathrel{+}= 1$
    end for
    if 0 is in $\boldsymbol{A_i^t}$
      $\boldsymbol{\mathcal{M}}_i^{t+1} = \left[m_{i0}^{t+1}, m_{i1}^{t+1}, \ldots, m_{i(N_{\mathrm{LLH}}-1)}^{t+1}\right] = [1\ 1\ 1\ \ldots\ 1] \in \mathbb{R}^{N_{\mathrm{LLH}}}$
    else
      for $\boldsymbol{j = 0}$ ***to*** $\boldsymbol{N_{\mathrm{LLH}} - 1}$
        $m_{ij}^{t+1} = \ell_{ij}^t + C\sqrt{\frac{2ln\sum_{k=0}^{N_{\mathrm{LLH}}-1} a_{ik}^t}{a_{ij}^t}}$
      end for
    endif
    if $\boldsymbol{X_i^{t+1}}$ ***is not better than*** $\boldsymbol{X_i^t}$
      $\boldsymbol{X}_i^{t+1} = \boldsymbol{X}_i^t$
    endif
  end for
  Update the belief space and obtain $\boldsymbol{\mathcal{B}}^{t+1} = (\boldsymbol{\mathcal{S}}^{t+1}, \boldsymbol{\mathcal{N}}^{t+1})$
  Set $t = t + 1$
end while
end

5. Case studies

To evaluate the performance of the proposed algorithms, three optimal structural design case studies are conducted. As the primary aim of this study is to demonstrate how the proposed algorithms handle different classes of problems, the selected structural optimisation cases exhibit varying levels of complexity and pose distinct challenges in terms of problem scale, as well as the number and types of design var-

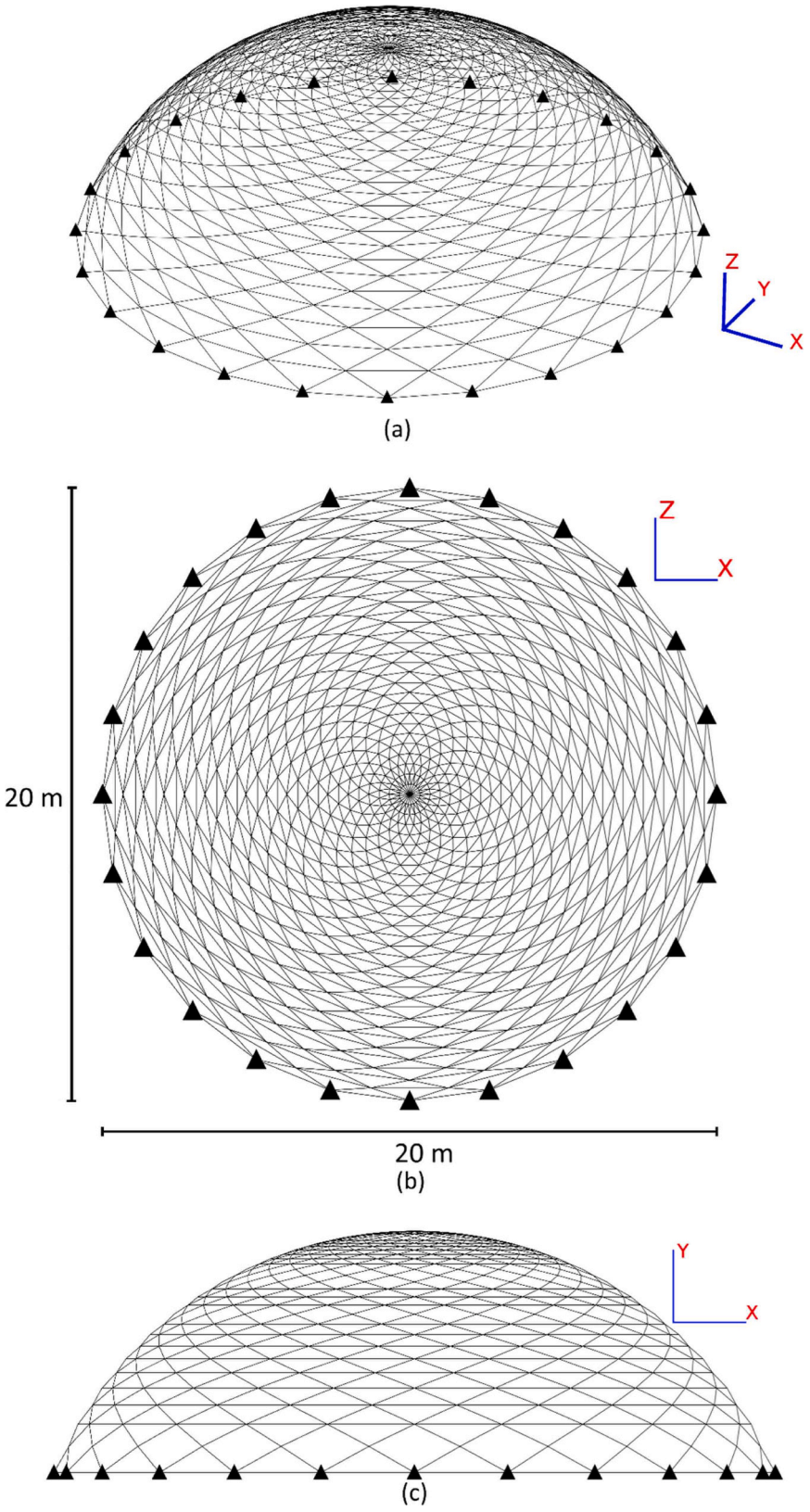


Fig. 4. The 2136-bar lamella dome structure: a) 3D view, b) top view, and c) side view.

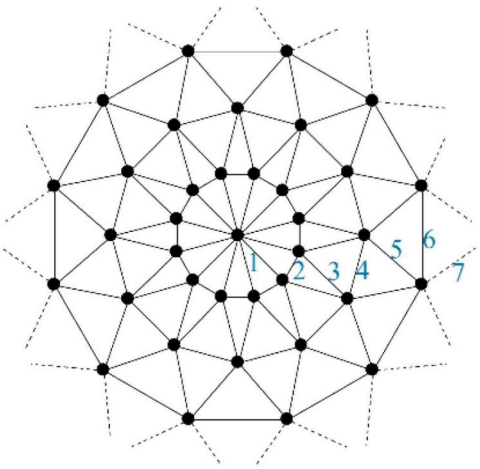


Fig. 5. Elements grouping pattern in the 2136-bar lamella dome structure.

iables and constraints. The design examples include a lamella dome, a double-layer barrel vault, and a 19-story frame structure, comprising 2136, 3341, and 1935 elements, respectively. To reflect realistic design conditions, the design variables are restricted to a predefined set of available sections. The statistical results for each algorithm are based on 30 independent runs, with the maximum number of structural analyses set as the termination criterion. The number of individuals N_P is taken as 50 for all algorithms.

5.1. 2136-bar lamella dome structure

The first design example focuses on the size and shape optimisation of a 2136-bar lamella dome structure depicted in Fig. 4, subject to frequency constraints. The dome is composed of 30 rings, each containing 24 nodes. Considering the structure's symmetric geometry, the elements are grouped into 60 design categories. The pattern of design group numbering is shown in Fig. 5. The dome height is treated as shape variable, which brings the total number of design variables to 61. Cross-sections for these groups are selected from pipe sections listed in Table A.1 in Appendix A, based on the AISC database. The height of the dome is assumed to be between 1 m and 20 m. Material properties include a Young's modulus of 210 GPa and a steel density of 7971.81 kg/m^3. The dome has a diameter of 20 m (see Fig. 4). The nodal coordinates can be expressed in terms of dome height or the coordinate of the top node. For further geometric details, see [68]. A lumped mass of 100 kg is assumed to be attached to each free node of the structure. The design constraints require the first four natural frequencies to fall within specific ranges as follows: $2\,Hz \le f_1 \le 4$ Hz, $5\,Hz \le f_2 \le 6\,Hz$, $5\,Hz \le f_3 \le 6\,Hz$, and $7.50\,Hz \le f_4 \le 10\,Hz$.

The optimum cross-sections and dome heights obtained using various algorithms for the lamella dome structure are summarised in Table B. 1. Table 1 also presents optimal weights and statistical results obtained by each algorithm. Among the standard algorithms, the CA-II algorithm delivers the best performance, achieving an optimum weight of 24.72 tonnes. In contrast, the designs obtained by the HIA-CF and HIA-MAB weigh approximately 21.41 tonnes and 20.69 tonnes, respectively, representing reductions of over 13 % and 16 % compared to the best design by the CA-II. The statistical results from HIA-CF and HIA-MAB demonstrate their consistency in producing high-quality designs across independent runs, outperforming the standard algorithms. Notably, HIA-MAB shows slightly better statistical performance than HIA-CF. The heaviest design from HIA-MAB is still lighter than the best design achieved by standard algorithms. It should also be noted that the HIA-MAB obtained a design approximately 3.50 % lighter than that of the HIA-CF, demonstrating the relative effectiveness of the MAB approach over the CF method in this design example.

To verify the feasibility of the optimal designs produced by the various algorithms with respect to the design constraints, the first four natural frequencies of the lamella dome structure at the optimum designs are listed in Table 2. The results indicate that the first natural frequency in all optimal designs is close to its upper bound, while the second, third, and fourth frequencies are near their respective lower bounds. It is evident that the optimal designs generated by most algorithms satisfy the design constraints, with the exception of the CA-IV algorithm. This suggests that CA-IV failed to effectively explore the feasible regions of the search space.

Another important aspect of algorithm evaluation is the investigation of how individuals evolve in terms of frequency values throughout the optimisation process. This can provide additional insights into the exploration and exploitation capabilities of the algorithms. To this end, the evolution of natural frequencies associated with individuals in the CA-II, HIA-CF, and HIA-MAB algorithms is illustrated in Fig. 6. As shown in Fig. 6, individuals in the HIA-CF and HIA-MAB algorithms rapidly converge towards the frequencies corresponding to the optimal designs, whereas those in CA-II remain scattered, often falling outside the feasible frequency range. HIA-MAB exhibits slightly better convergence towards the optimal frequencies compared to HIA-CF. Overall, these results confirm that both HIA-CF and HIA-MAB can identify the feasible design space and converge effectively to optimal frequency values. It should be noted that the frequency evolution for the third mode is not shown in Fig. 6, as it is identical to that of the second mode due to the repeated frequencies in this structure.

The average convergence behaviour of the proposed algorithms is compared with that of the standard algorithms in Fig. 7. Both HIA-CF and HIA-MAB demonstrate significantly superior convergence rates relative to all standard algorithms, highlighting their consistency in

Table 1
Comparison of optimal weights and statistical results obtained using various algorithms for the 2136-bar lamella structure (tonnes).

	CA-I	CA-II	CA-III	CA-IV	CA-V	DE-I	DE-II	DE-III	HIA-CF	HIA-MAB
Best weight	32.45	24.72	25.84	50.48	35.81	34.69	36.69	38.49	21.41	20.69
Mean	36.91	26.08	27.78	57.40	37.72	39.02	41.19	38.80	22.87	21.83
Standard deviation	2.83	1.23	1.30	3.97	3.33	2.63	2.74	3.25	1.06	0.66
Worst	44.44	30.54	30.97	65.11	45.80	45.20	46.49	46.16	25.63	23.34

Table 2
Comparison of the natural frequencies for various modes at the optimal designs obtained using different algorithms for the 2136-bar lamella structure.

Frequencies (Hz)	CA-I	CA-II	CA-III	CA-IV*	CA-V	DE-I	DE-II	DE-III	HIA-CF	HIA-MAB
f_1	3.90094	3.99617	3.97310	3.56365	3.94502	3.98953	3.99845	3.96930	3.92503	3.92268
f_2	5.11289	5.26073	5.08025	5.01806	5.47599	5.32257	5.24471	5.16526	5.03137	5.02380
f_3	5.11289	5.26073	5.08025	5.01806	5.47599	5.32257	7.67679	5.16526	5.03137	5.02380
f_4	7.61219	7.50832	7.50004	6.10640	7.50040	7.52128	7.67679	7.57383	7.50296	7.50003

* The algorithm provided an infeasible design

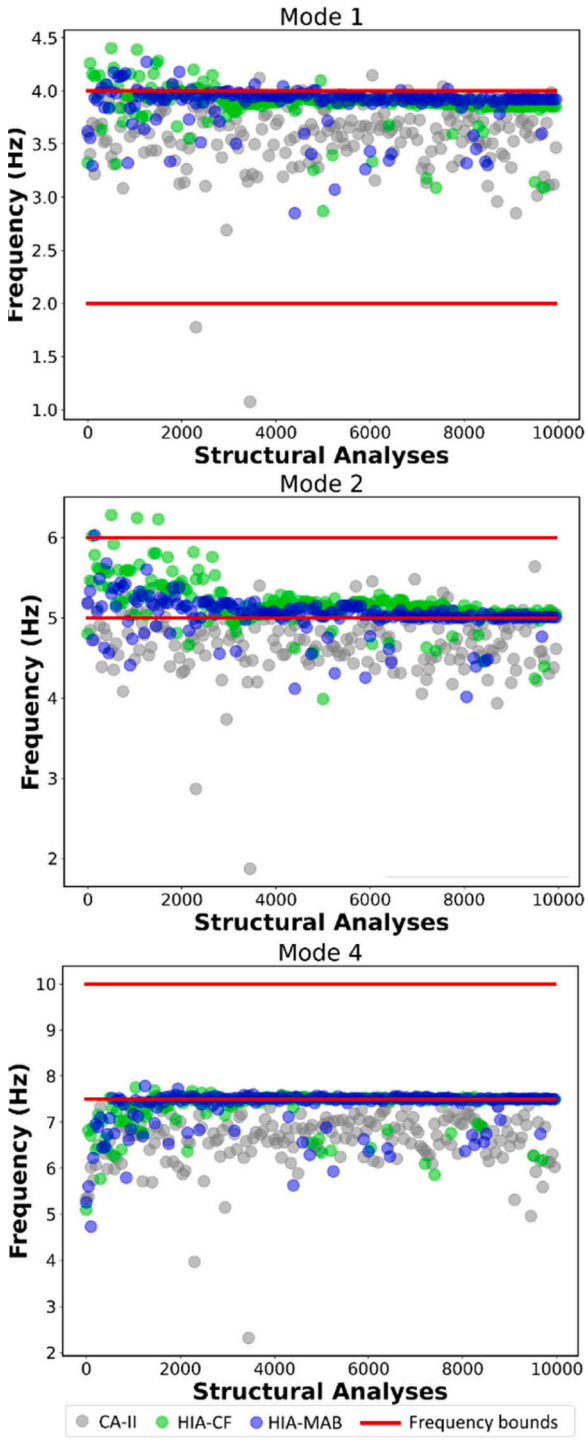


Fig. 6. Evolution of modal frequencies in the CA-II, HIA-CF, and HIA-MAB algorithms during the design optimisation of the 2136-bar lamella dome structure.

achieving high-quality optimal designs across independent runs. HIA-MAB exhibits slightly better performance than HIA-CF in exploiting the design space during the later stages.

One of the key parameters that can affect performance of algorithms in handling the structural optimisation under frequency constraints is the penalty parameter α. As discussed in Section 3.2, this study adopts an incremental approach for the penalty parameter in the truss design problems under frequency constraints, defines as:

$$\alpha^t = \alpha_0 + \left(\frac{t}{t_{\max}}\right)\alpha_1 \tag{39}$$

where ^{t}max is the maximum number of iterations. To investigate the influence of different penalty parameter values on the quality of the final designs, a sensitivity analysis was conducted for the HIA-MAB algorithm under various static and incremental penalty parameter settings, as summarised in Table 3. The results show that using a constant penalty parameter of $\alpha_o = 2$ leads to infeasible final designs, as indicated by penalty function values greater than one in Table 3 (also see Eq. (4)). When the parameter was increased to $\alpha_o = 8$, the algorithm produced a design that was nearly feasible but had a structural weight approximately 10 % higher than that obtained using the incremental approach. This indicates that simply increasing the penalty parameter does not guarantee both feasibility and efficiency. In contrast, adopting an incremental penalty parameter (linearly increased from 2 to 8 using Eq. (39)) resulted in a fully feasible and more efficient design. These results demonstrate that the incremental approach enables the algorithm to gradually shift its focus from the infeasible design space to the feasible region while avoiding unnecessary increases in structural weight.

5.2. 3341-bar double-layer barrel vault structure

Fig. 8 shows a double-layer barrel vault structure consisting of 3341 elements. The internal radius of the bottom layer is 20.0 m, while the radius of the upper arch is 25.0 m. Due to the structure's symmetric configuration, its elements are categorised into 55 design groups, as illustrated in Fig. 8. The design groups 1–25 correspond to the elements on the lower arch, while groups 26–50 include the elements on the upper arch. The diagonal elements connecting the lower and upper arches are assigned to design group 51. The diagonal and longitudinal elements on the top-roof layer are categorised into design groups 52 and 53, respectively, and the corresponding elements on the bottom-roof layer are grouped into design groups 54 and 55. The Young's modulus is taken as 210 GPa, and the material density of steel is assumed to be 7971.81 kg/m^3. It is assumed that a lumped mass of 100 kg is attached to all free nodes of the structure. The cross-sections for the design groups must be selected from the pipe sections listed in Table A.1 in Appendix A.

In addition to the large number of design variables, the presence of repeated frequencies in symmetric structures adds further complexity to the optimal design problem. These challenges can impact the effectiveness of optimisation algorithms and reveal their practical strengths and weaknesses. To this end, the design constraints are defined to isolate the first four natural frequencies within specific target ranges, as follows:1 Hz $\leq f_1 \leq$ 1.5 Hz, 2 Hz $\leq f_2 \leq$ 2.5 *Hz*, 3 Hz $\leq f_3 \leq$ 3.5 Hz, and 4 Hz $\leq f_4 \leq$ 6 Hz.

The optimal weights and statistical results obtained by various algorithms are summarised in Table 4. The results clearly demonstrate the superiority of the proposed algorithms over the standard ones. The optimum weights achieved by the HIA-CF and HIA-MAB are about 92.14 and 89.50 tonnes, respectively, which are significantly lighter than the best result from the standard algorithms, 99.97 tonnes, obtained by the CA-III. This represents weight reductions of about 10 % and 8 % for the HIA-CF and HIA-MAB, respectively. Another notable aspect is that the worst weight obtained by the HIA-MAB is still better than the best weight yielded by any standard algorithm. Among the standard algorithms, CA-II can be considered the best-performing in terms of statistical results, with an average weight of 102.89 tonnes. However, this is still approximately 8 % and 10 % heavier than the average weights produced by the HIA-CF and HIA-MAB, respectively. Comparison of the optimum weights reveals that the HIA-MAB produced a design approximately 2.9 % lighter than the HIA-CF. The optimal sections produced by various algorithms are listed in Table B. 2 in Appendix B.

The first four natural frequencies of the structure at the optimum

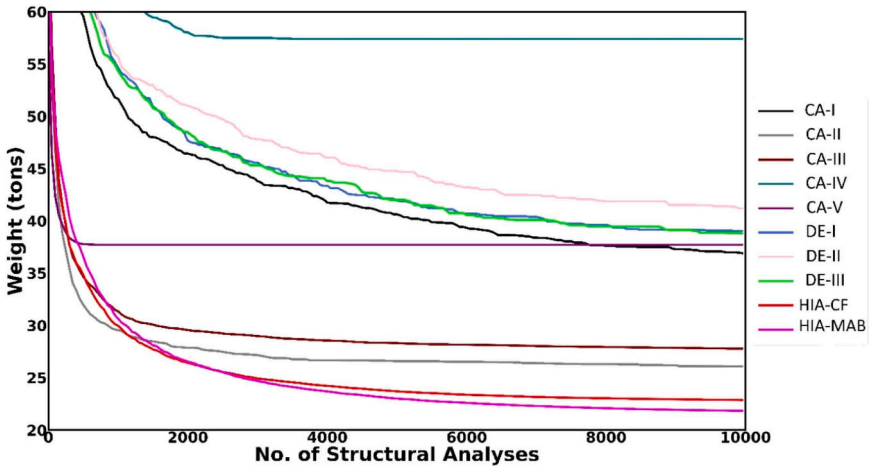


Fig. 7. Average convergence behaviour of the different algorithms for the 2136-bar lamella dome structure.

Table 3
Sensitivity analysis of the HIA-MAB algorithm with static and incremental penalty parameters for the 2136-bar lamella dome structure problem.

	Static ($\alpha_1 = 0$)		Incremental
	$\alpha_0 = 2$	$\alpha_0 = 8$	$\alpha_0 = 2, \alpha_1 = 6$
Final design weight (tonnes)	21.01	22.65	20.69
Penalty function value ($\psi(\boldsymbol{X}, \boldsymbol{Y})$)	1.0014*	1.0000068*	1.0000

*Infeasible designs

designs obtained by various algorithms are presented in Table 5. A quick examination of these frequencies reveals that the optimum designs reach the upper frequency bound of the first mode and the lower frequency bounds of the second, third and fourth modes. It can also be observed that the designs obtained by HIA-CF and HIA-MAB fully satisfy the frequency constraints. In contrast, CA-IV and CA-V failed to find feasible designs, which is not surprising given the complexity of this design problem.

Fig. 9 illustrates how the frequencies obtained by individuals evolve during the optimisation process using HIA-CF, HIA-MAB, and CA-III algorithms. As shown in Fig. 9, individuals in the HIA-CF and HIA-MAB converge towards the optimal frequencies after about 2000 structural analyses. It can also be observed that the frequency disparities among individuals in the HIA-CF and HIA-MAB gradually decrease, indicating a transition from exploration of the search space to exploitation of the near-optimal frequency design space. Furthermore, the proposed algorithms exhibit slightly lower frequency disparities compared to CA-III, suggesting more consistent convergence behaviour. The disparities also reflect the sensitivity of the structure to variations in the design space.

The average convergence behaviours of the various algorithms are illustrated in Fig. 10. The HIA-CF and HIA-MAB converge quickly and consistently to designs weighing under 100 tonnes within approximately 2000 structural analyses. With continued optimisation, HIA-MAB is able to find slightly better designs than HIA-CF. While the clear superiority of the HIA-CF and HIA-MAB over the standard algorithms is evident from the smoother convergence curves, the premature convergence of the standard algorithms, particularly CA-IV and CA-V, cannot be overlooked. The best-performing standard algorithm in terms of convergence behaviour is CA-II, which typically converges to designs weighing over 100 tonnes, still significantly heavier than those found by HIA-CF and HIA-MAB.

5.3. 19-story steel frame with 1935 members

The last example involves the optimisation of a 19-story steel moment-resisting frame [91] shown in Fig. 11 and Fig. 12, studied here using the proposed algorithms. The structural model comprises 59 design variable groups. In this grouping scheme, columns located at the corners, along the building's perimeter (exterior), and within the core (interior) are each assigned to separate groups to reflect their distinct load-bearing roles. Similarly, interior and exterior beams are grouped independently to capture differences in their structural behaviour. The bracing elements are assigned to a dedicated group, recognising their unique contribution to lateral resistance. For modelling consistency, the structural elements on the first story are treated uniformly, while for the upper stories, groupings are updated every three floors. This grouping strategy reflects both structural regularity and design flexibility, aiming to balance realism with computational tractability. Cross-sections for this problem are selected from 268 w-shaped sections based on the AISC [59] database. Detailed information on material properties, loading conditions, element effective lengths, and deflection constraints for this structure is presented in Table 6.

Table 7 presents a comparison of ten optimisation algorithms applied to the 19-story structure, reporting statistical results in terms of total structural weight. The corresponding optimum sections are also provided in Table B. 3 in Appendix B. The proposed intelligent algorithms, HIA-CF and HIA-MAB, clearly outperform all classical methods across all performance metrics. Specifically, HIA-CF achieved the best solution with a total weight of 2172 tonnes, followed closely by HIA-MAB at 2256 tonnes. These values represent a substantial improvement over the best results achieved by conventional approaches, most notably CA-II at 2598 tonnes, DE-I at 2637 tonnes, and DE-III at 2768 tonnes. The CA-IV and CA-V algorithms produced significantly heavier designs, with best weights of 6333 tonnes and 4929 tonnes, respectively, highlighting their inefficiency in converging toward optimal solutions. Due to their consistently poor performance across all statistical measures, their results are omitted from Table 7 for clarity and conciseness.

In addition to their superior best-case performance, the proposed hyper-intelligent algorithms demonstrated remarkable robustness and consistency across independent runs. HIA-CF achieved a mean weight of 2390 tonnes and a worst-case value of 2512 tonnes, while HIA-MAB followed closely with a mean of 2365 tonnes and a worst-case of 2520 tonnes. These confirm that even in less favourable scenarios, both methods yield designs that are substantially more efficient than the best results from many classical methods. For instance, CA-I and CA-III produced average weights of 4566 tonnes and 4577 tonnes, and their worst cases rose to 4967 tonnes and 5124 tonnes, respectively. Among the DE-based algorithms, DE-I delivered a relatively strong performance

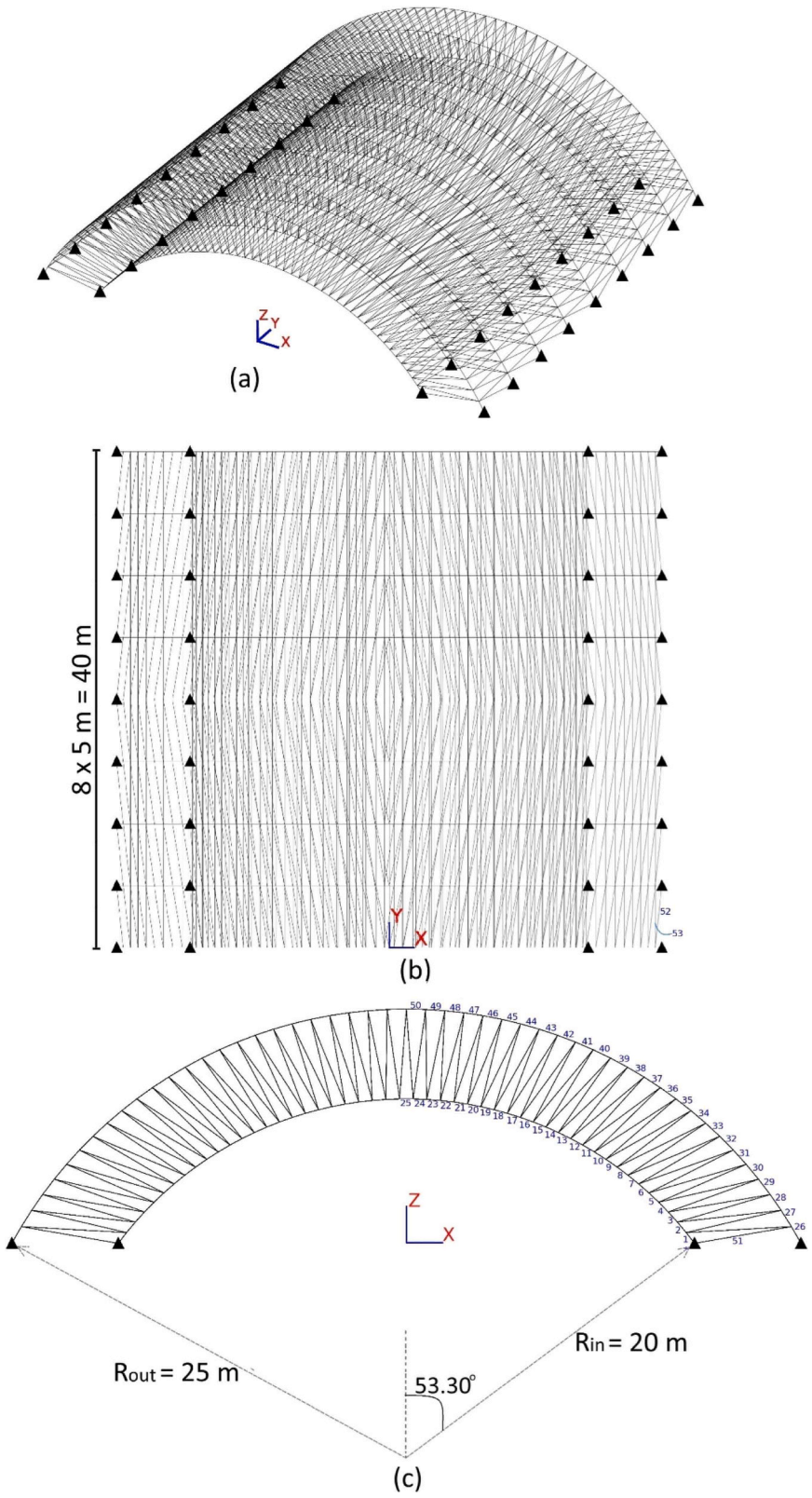


Fig. 8. A 3341-bar double-layer barrel vault structure: a) 3D view, b) top view, and c) side view.

with a mean of 2745 tonnes and a worst-case of 2846 tonnes, while DE-II and DE-III yielded mean values of 2773 tonnes and 2767 tonnes, and worst cases around 3039 tonnes and 2890 tonnes. Although the DE methods performed noticeably better than CA-based algorithms, they still exhibited higher variability and heavier worst-case solutions compared to HIA-CF and HIA-MAB.

Another important dimension of performance is algorithm stability, as reflected by the standard deviation of the results. The proposed algorithms consistently yielded low variability, with standard deviations of 86.55 tonnes (HIA-CF) and 65.21 tonnes (HIA-MAB) indicating strong

Table 4
Comparison of optimal weights and statistical results obtained using various algorithms for the 3341-bar double-layer barrel vault structure (tonnes).

	CA-I	CA-II	CA-III	CA-IV	CA-V	DE-I	DE-II	DE-III	HIA-CF	HIA-MAB
Best weight	112.47	106.09	99.97	131.77	106.41	106.64	107.67	104.06	92.14	89.50
Mean	120.77	102.89	109.28	204.28	160.10	108.53	110.37	109.32	95.26	93.91
Standard deviation	6.59	3.34	21.85	39.33	33.86	1.92	1.81	2.43	2.92	2.49
Worst	137.02	113.20	207.27	284.66	249.23	113.03	113.56	115.45	103.93	98.16

Table 5
Comparison of the natural frequencies for various modes at the optimal designs obtained using different algorithms for the 3341-bar double-layer barrel vault structure.

Frequencies (Hz)	CA-I	CA-II	CA-III	CA-IV*	CA-V*	DE-I	DE-II	DE-III	HIA-CF	HIA-MAB
f_1	1.49929	1.49604	1.49982	1.64295	1.52490	1.49515	1.49821	1.48973	1.49910	1.48666
f_2	2.06864	2.05503	2.08664	2.19597	2.10842	2.07810	2.06245	2.05153	2.07286	2.04074
f_3	3.04622	3.01720	3.01109	3.08012	3.07318	3.00320	3.00522	3.01216	3.04525	3.00183
f_4	4.00134	4.06208	4.0001	3.08227	3.94718	4.07393	4.05239	4.02064	4.00076	4.00155

* The algorithm provided an infeasible design

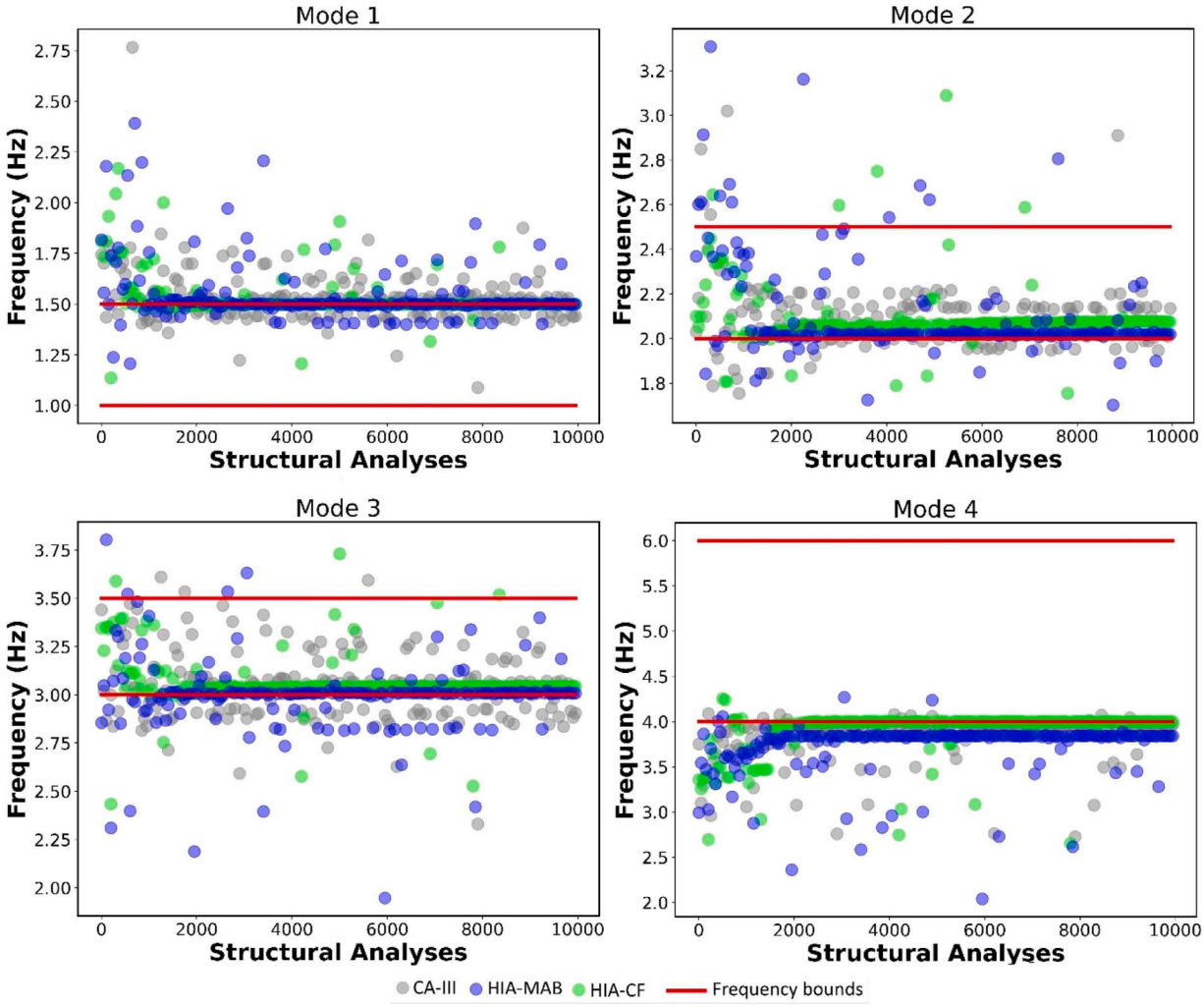


Fig. 9. Evolution of modal frequencies in the CA-III, HIA-CF, and HIA-MAB algorithms during the optimal design of the 3341-bar double-layer barrel vault structure.

convergence reliability. In contrast, several classical methods such as CA-V and CA-IV exhibited severe fluctuations, with standard deviations exceeding 600 tonnes. Even DE-based algorithms, while more stable than CA methods, had higher deviations (e.g., 116 tonnes for DE-II).

Fig. 13 illustrates the convergence history of all methods over the course of the iterations. Three distinct performance trends emerge from the curves. The first group consists of our proposed methods, HIA-CF and HIA-MAB, which start strongly and continuously improve their solutions throughout the optimisation process. Notably, these two methods consistently maintain lower objective values than all others at nearly every iteration, demonstrating a clear and substantial advantage in minimising structural weight. The second group includes CA-I, CA-II,

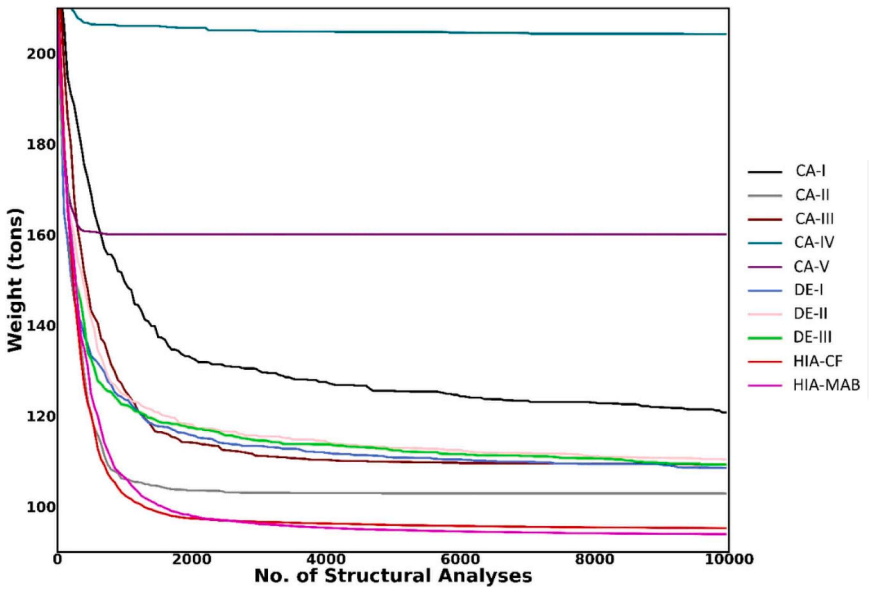


Fig. 10. Average convergence behaviour of various algorithms in the 3341-bar double-layer barrel vault structure.

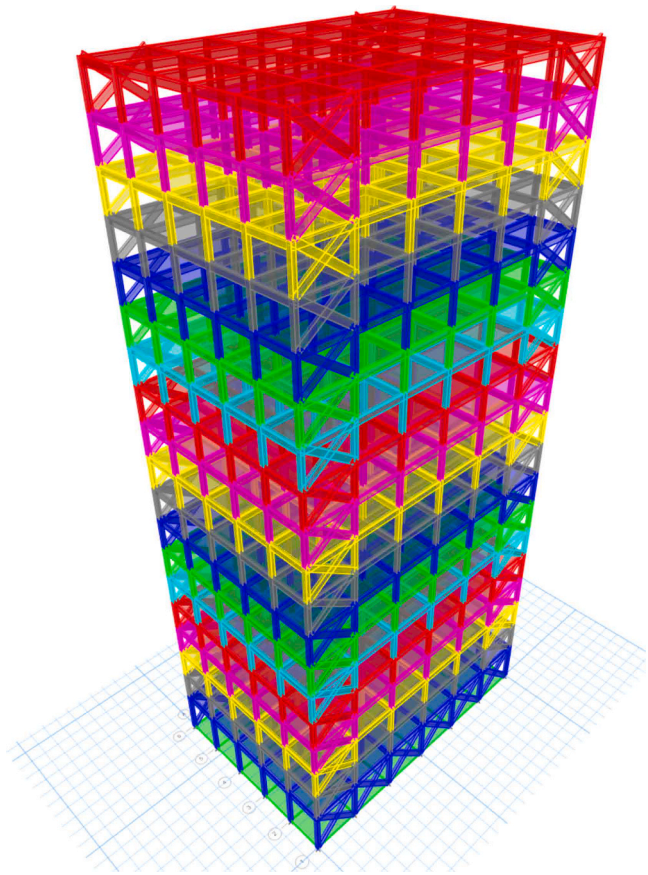


Fig. 11. The 19-story steel frame with 1935 members (from Ref. [91]): 3D view.

DE-I, DE-II, and DE-III. While some of these algorithms, such as CA-II, start with a promising convergence rate but eventually stagnates and finishes as the weakest performer within this group, unable to close the gap with the proposed algorithms. The third group comprises the remaining classical methods, which display comparatively poor performance. These algorithms converge slowly and remain far from the best solutions, finishing with significantly higher objective function values. Their weak convergence behaviour underscores their limited suitability for complex, large-scale structural optimisation problems.

Table 7 also presents a comparative analysis between the results obtained in this study and those reported in a previous study [91]. The number of analyses for each method was held constant at 60,000 to ensure a fair comparison. Despite this, both the HIA-MAB and HIA-CF methods achieved superior results relative to the advanced method proposed in [91]. Specifically, HIA-MAB and HIA-CF demonstrated improvements of 10 % and 15 %, respectively, over the advanced CSS method. When compared with the standard CSS method, the improvements are approximately 5 % and 9 %, respectively. These results underscore the superior performance of the proposed HIA-MAB and HIA-CF algorithms in achieving more optimal solutions within the same computational budget.

To check whether the optimum design obtained from each algorithm is feasible, the ratios of inter-story drifts to their allowable values are compared in Fig. 14. As can be seen, the designs yielded by the proposed algorithms fully satisfy the inter-story drift constraint, while the designs produced by CA-I, CA-II, and DE-I slightly violate this constraint.

In the truss design benchmarks, the natural frequencies of the structures are treated as design constraints to capture seismic and dynamic performance requirements, such as avoiding excessively close or overlapping frequencies. These constraints are particularly relevant in the design of special buildings where vibration characteristics are critical. For the 19-story steel frame, stress and inter-story drift ratio constraints were considered in accordance with the AISC-LRFD [59] design code. Together, these examples introduce different levels of complexity for the optimisation algorithms. The results demonstrate that the designs obtained using the proposed algorithms satisfy both the frequency requirements and the AISC-LRFD [59] provisions (see Fig. 6 and Fig. 14).

5.4. Discussion on the LLHs

The observations in previous sub-sections indicate that the performance of the proposed HIAs consistently outperforms individual algorithms across the investigated design examples. In the global LLH space of the HIA-CF and HIA-MAB methods, various variants of the DE and CA methods are employed. However, a natural question arises: if additional algorithms were added to the meta-heuristic pool in the LLH, would the proposed HI framework still demonstrate superior statistical performance compared to all individual algorithms? To investigate this

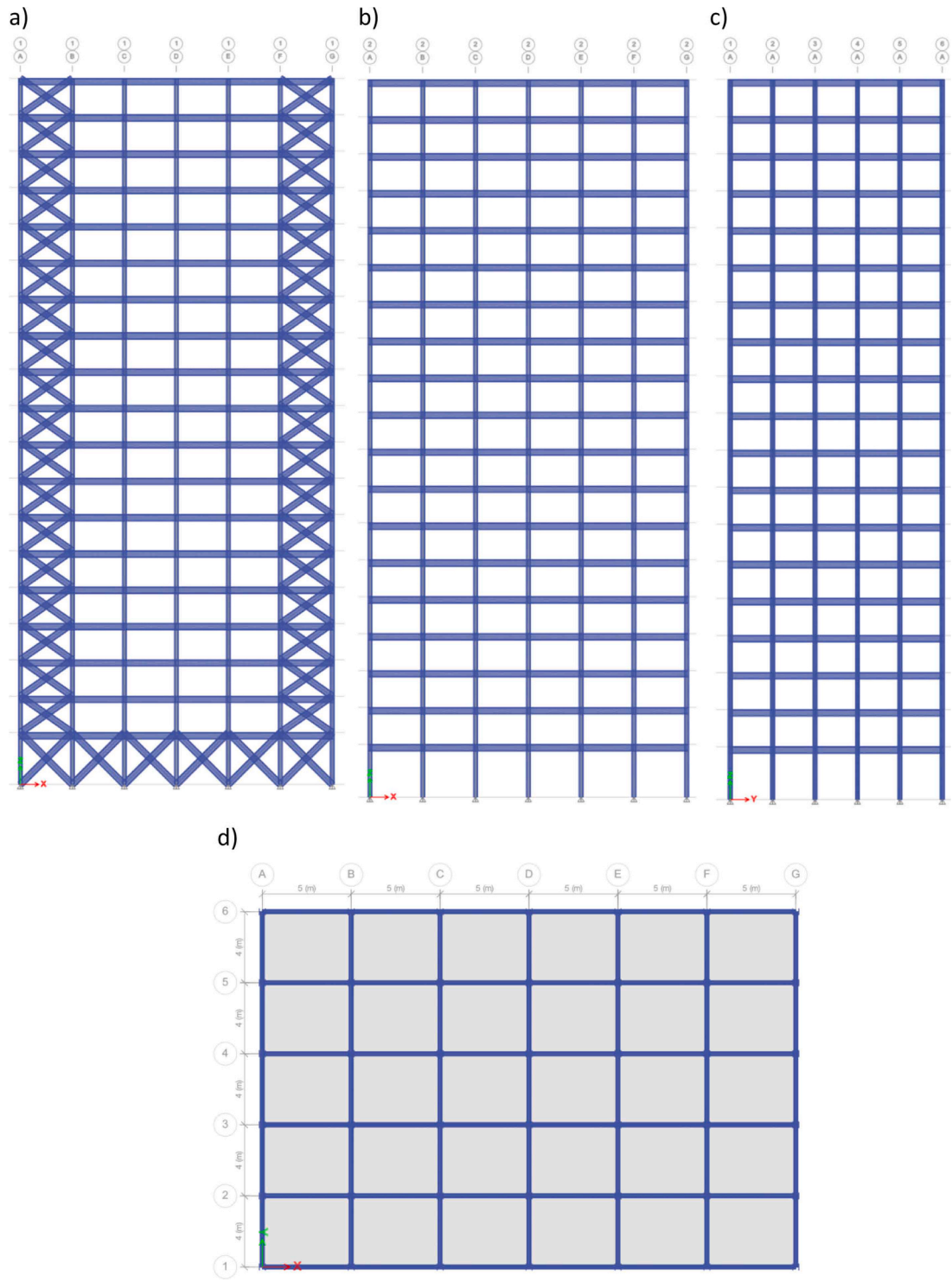


Fig. 12. Side and plan views of the 19-story steel frame: a) side view of frames 2, 3 and 4, b) side view of frames 1 and 5, c) side view of frames A, B, C, D, E, F and G, d) plan view.

scenario, two additional well-known meta-heuristics, namely genetic algorithm (GA) and particle swarm optimisation (PSO), are incorporated into the LLH space of the HIA-CF and HIA-MAB algorithms. Specifically, the steady state GA [92] and accelerated PSO [93] methods are used.

The average convergence behaviour of the HIA-CF and HIA-MAB algorithms, incorporating GA, APSO, and along with various variants of DE and CA in their LLH space, is compared with the standard GA and APSO methods for the 2136-bar lamella dome structure in Fig. 15. As observed, the HIA-CF and HIA-MAB algorithms continue to demonstrate significantly better performance than the GA and APSO methods. By analysing Fig. 15 alongside the performance of different DE and CA variants for the same problem in Fig. 7, it is evident that adding additional meta-heuristics to the LLH space does not compromise the efficiency of the HIA-CF and HIA-MAB algorithms. Expanding the LLH space with more meta-heuristics could even be advantageous, as it provides the algorithms with a broader set of options to select the most suitable meta-heuristic at the right time to potentially improving results. Moreover, even if some added meta-heuristics prove inefficient for a

Table 6
Detailed information on material properties, loading conditions, element effective lengths, and deflection constraints for the design of a 19-story steel frame.

Material Properties		
	Modulus of elasticity	200 GPa
	Yield stress	248.2 MPa
	Unit weight of the steel	7.85 ton/m^3
Load Values		
	Live load	
		12 kN/m (on floors beam)
		7 kN/m (on roof beam)
	Dead load	Structural weight +
		20 kN/m (on floors beam)
		15 kN/m (on roof beam)
	Earthquake loads	
	Base shear	1. * *Ws*
		(Ws is seismic load and here is dead load)
	C_T	0.0853
	H_n	68 m
	T	2.019 s
	k	1.7595
	Load Combinations	
		1) 1.4*D*
		2) 1.2*D* + 1.0*Eex*+ 0.5 *L*
		3) 0.9*D* + 1.0*Ex*
		4) 0.9*D* + 1.0*Eey*
		5) 1.2*D* + 1.6 *L*
		6) 1.2*D* + 1.0*Ey* + 0.5 *L*
		7) 0.9*D* + 1.0*Eex*
		8) 1.2*D* + 1.0*Ex* + 0.5 *L*
		9) 1.2*D* + 1.0*Eey* + 0.5 *L*
		10) 0.9*D* + 1.0*Ey*
Unbraced lengths		
	Beams	One-fifth of its length
	Columns	Unbraced along its lengths
Effective length factor		
	Beams and braces	1
	columns around minor direction	1
	columns around main direction	Jackson and Moreland frame buckling monograph [91]
Lateral movement		
	the top level is restricted to 0.1 m, and the upper drift is considered as *ht*/400, where *ht* is the story's height.	10 cm
	drift	h_t/400 (h_t is the story's height)

particular problem at any stage of the optimisation process, the HIA-CF and HIA-MAB algorithms are expected to effectively abandon those inefficient meta-heuristics and focus on the more efficient ones during solution search.

5.5. Statistical study

Table 8 provides statistical results for all examples. The non-parametric Friedman test ($p = 6.04 \times 10^{-47}$) confirmed the presence of statistically significant differences among the ten compared algorithms for the 2,136-bar lamella dome structure, indicating that the central tendencies of their performance distributions are not equivalent. Given that at least one population violated normality (DE-II, $p = 0.001$), the Friedman test was selected as the omnibus test, followed by the Nemenyi post-hoc procedure (critical distance = 2.473). According to the mean rank (MR) and median performance values, HIA-MAB (MR = 9.87) and HIA-CF (MR = 9.10) achieved the highest overall rankings among all compared algorithms. Both algorithms consistently exhibited lower variability (MAD≈0.53) and narrow confidence intervals, indicating robust and stable optimisation behaviour. The Nemenyi post-hoc results showed that the performance of the developed HIA-MAB and HIA-CF algorithms (and to some extent CA-I) was significantly better than that of the remaining algorithms included in the comparison.

For the second example, the Friedman test detected statistically significant differences among the ten optimisation algorithms ($p = 1.02 \times 10^{-45}$), indicating that their median performances are not equivalent. Because at least one population (CF) violated the normality assumption ($p < 0.005$), the non-parametric Friedman test followed by the Nemenyi post-hoc comparison was applied. According to the mean ranks, HIA-MAB (MR=9.57) and HIA-CF (MR=9.23) achieved the highest rankings across all algorithms. Both algorithms also exhibited low MAD values, suggesting stable optimisation behaviour. The Nemenyi post-hoc test further showed that HIA-MAB and HIA-CF consistently lie within the top statistical performance group, with no significant differences between them. Importantly, both algorithms outperform the majority of the remaining meta-heuristic methods, with significant differences relative to the CA and DE variants.

For the frame example, Normality testing (minimum p-value= 0.007) indicated that all populations can be treated as normally distributed under the adopted significance level. Bartlett's test, however, rejected homogeneity of variances ($p < 0.001$), indicating heteroscedasticity. Therefore, the non-parametric Friedman test was used as the omnibus test, followed by the Nemenyi post-hoc procedure. The Friedman test showed a highly significant difference among the algorithms ($p = 1.15 \times 10^{-35}$). Examination of the mean ranks reveals that HIA-CF (MR=7.47) and HIA-MAB (MR=7.47) achieved the highest rankings, indicating the most favourable optimisation performance for this structure. Both algorithms also exhibited relatively low standard deviations (SD ≈70–90), highlighting the stability of the proposed hyper-intelligent approach across multiple independent runs. The post-hoc Nemenyi analysis further showed that HIA-CF and HIA-MAB are statistically indistinguishable and form the top performance group, clearly separated from the remaining algorithms. The next performance group consists of DE-I, DE-II, DE-III, and CA-II, while CA-III and CA-I form the lowest-ranking group with significantly poorer performance.

In summary, the statistical evidence suggests that the proposed hyper-intelligent algorithms (HIA-MAB and HIA-CF) provide reliable and high-quality solutions across repeated runs, ranking consistently among the top performers. Their close performance and high mean ranks highlight the benefit of the hyper-intelligent framework in achieving stable convergence behaviour compared to conventional meta-heuristics.

Fig. 16 provides critical distance (CD) diagrams for the three

Table 7
Comparison of optimal weights obtained using various algorithms for the 19-story steel frame structure (tonnes).

	CA-I	CA-II	CA-III	DE-I	DE-II	DE-III	HIA-CF	HIA-MAB	CSS (Kaveh et al. [91])	ACSS (Kaveh et al. [91])
Best weight	3831.3	2598.7	4185.2	2637.3	2768.6	2708.5	2172.1	2256.6	2493.659	2362.733
Mean	4566.280	3101.657	4576.842	2745.487	2772.690	2766.849	2389.798	2365.079	N/A	N/A
Standard deviation	364.089	436.525	277.288	62.512	116.032	62.472	86.551	65.207	N/A	N/A
Worst	4966.547	4087.303	5124.354	2845.802	3039.346	2889.831	2512.420	2520.425	N/A	N/A

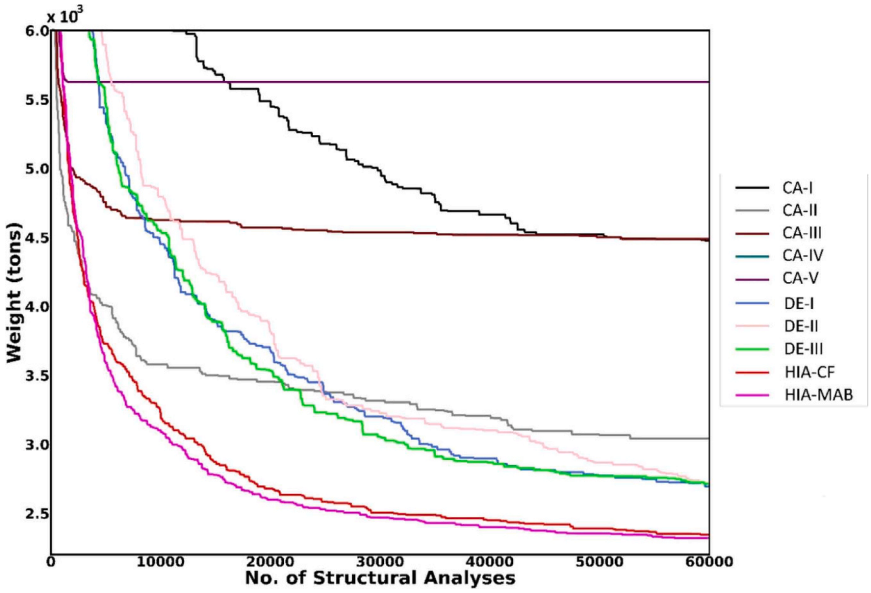


Fig. 13. Average convergence behaviour of various algorithms in the 19-story frame structure.

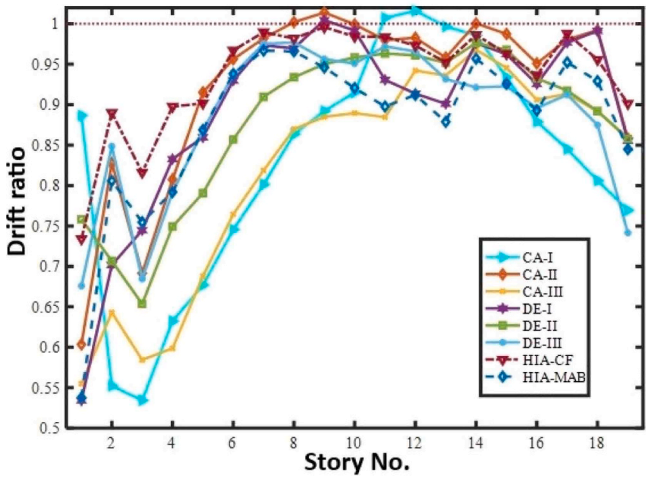


Fig. 14. Comparison of the inter-story drift ratios obtained from various algorithms at the optimal designs for the 19-story steel frame.

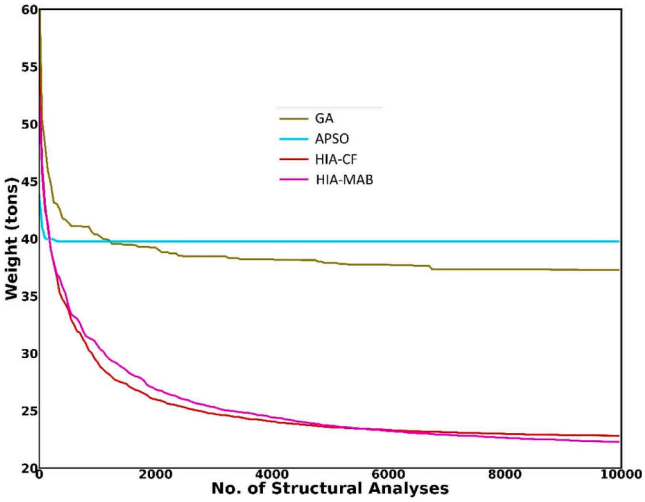


Fig. 15. Comparison of the average convergence behaviour of standard GA and APSO methods with those obtained from the HI framework employing a diverse low-level heuristic (LLH) space that integrates multiple variants of CAs, DE, GA, and PSO methods, applied to the 2136-bar lamella dome structure.

benchmark examples. The CD values are CD = 2.473 for the first and second examples, and CD = 1.917 for the third example. Algorithm pairs whose difference in mean rank exceeds the corresponding CD value are considered significantly different at the $\alpha = 0.05$ significance level.

6. Concluding remarks

This study introduced a new algorithmic framework, termed the hyper-intelligent (HI) framework, aimed at developing algorithms capable of solving various types of structural optimisation problems with a more statistically consistent performance. The framework is based on the hypothesis that the simultaneous, yet intelligent, application of multiple population-based meta-heuristics yields better results than applying them individually. Unlike conventional meta-heuristics that rely on a single search strategy, the novelty of the framework lies in the integration of multiple meta-heuristic strategies, coordinated through a set of high-level heuristic (HLH) subspaces. In the proposed HI framework, each individual has access to a HLH subspace that provides them with an online, feedback-based mechanism for learning and adapting their behaviour throughout the search process. By interacting with the low-level heuristic (LLH) space, individuals record experiences in their HLH subspaces, which then guide the selection of the most appropriate meta-heuristics at different stages of optimisation.

Based on the proposed framework, two algorithms were developed: the hyper-intelligent algorithm with choice function (HIA-CF) and the hyper-intelligent algorithm with multi-armed bandit (HIA-MAB). The two algorithms differ in the type of HLH subspace employed. The HIA-CF incorporates a choice function within each HLH subspace, whereas the HIA-MAB employs a learning mechanism inspired by the multi-armed bandit problem from game theory. The effectiveness of the proposed algorithms was validated on three large-scale structural design examples with static and frequency constraints: a lamella dome, a double-layer barrel vault, and a 19-story frame structure. The results provided strong evidence that the proposed algorithms achieved more reliable and effective performance compared to standard meta-heuristics. The concluding remarks of this study can be summarised as follows:

Table 8
Statistical results for all design examples.

	Meanrank	Median	Mad	Ci_lower	Ci_upper	Effect_size	Magnitude	Sffect_size_above	Magnitude_above
2136-bar lamella dome structure									
HIA-MAB	9.866667	21.84422	0.537247	21.13931	22.66733	0	negligible	0	negligible
HIA-CF	9.1	22.68359	0.530847	21.85414	24.62055	-1.0601	large	-1.0601	large
CA-II	7.9	25.71595	0.629611	24.96088	27.64986	-4.46208	large	-3.51228	large
CA-III	7.133333	27.61106	0.883096	26.25099	29.7755	-5.32161	large	-1.66675	large
CA-I	4.8	36.84794	2.050506	33.55843	40.20702	-6.75168	large	-3.94647	large
CA-V	4.366667	38.14813	1.845264	33.90356	42.28567	-8.09201	large	-0.44959	small
DE-III	3.966667	38.80049	2.196012	35.46627	44.78589	-7.15425	large	-0.21694	small
DE-I	3.9	38.75308	1.177061	36.53346	44.71467	-12.4656	large	0.018153	negligible
DE-II	2.966667	41.70188	2.357178	38.21856	45.04744	-7.83484	large	-1.06759	large
CA-IV	1	57.10408	2.995494	53.11101	63.86924	-11.0517	large	-3.85436	large
3341-bar double-layer barrel vault structure									
HIA-MAB	9.566667	93.65731	1.840261	91.58391	97.41962	0	negligible	0	negligible
HIA-CF	9.233333	95.07223	1.632009	92.24404	98.29868	-0.54871	medium	-0.54871	medium
CA-II	7.366667	103.4239	2.292232	99.28843	106.2511	-3.16924	large	-2.83115	large
CA-III	6.866667	102.6824	2.847832	98.13122	128.7846	-2.53897	large	0.193467	negligible
DE-I	5.533333	108.3869	1.267394	106.664	111.0151	-6.28794	large	-1.74564	large
DE-III	5.233333	109.4321	1.799198	107.3495	112.7013	-5.84662	large	-0.453	small
DE-II	4.766667	110.6179	1.230144	107.8906	112.9222	-7.30875	large	-0.51898	medium
CA-I	3.133333	120.1988	4.02537	113.9269	130.3581	-5.72001	large	-2.17121	large
CA-V	2.066667	153.3562	17.47154	127.3724	217.4594	-3.24138	large	-1.76405	large
CA-IV	1.233333	200.6558	28.42754	153.7819	252.4303	-3.58279	large	-1.35216	large
19-story steel frame									
HIA-CF	7.466667	2395.865	90.86005	2346.952	2444.777	0	negligible	0	negligible
HIA-MAB	7.466667	2371.098	71.72102	2332.488	2409.707	0.302584	small	0.302584	small
DE-I	5	2742.228	66.4092	2706.479	2777.978	-4.35244	large	-5.36967	large
DE-II	4.733333	2776.037	123.021	2709.811	2842.262	-3.51547	large	-0.342	small
DE-III	4.6	2765.427	65.6056	2730.11	2800.744	-4.66352	large	0.10762	negligible
CA-II	3.666667	3114.399	478.1434	2857.001	3371.796	-2.08786	large	-1.02258	large
CA-III	1.566667	4599.584	294.1683	4441.225	4757.943	-10.1225	large	-3.74139	large
CA-I	1.5	4529.42	380.8299	4324.409	4734.431	-7.70666	large	0.206203	small
HIA-CF	7.466667	2395.865	90.86005	2346.952	2444.777	0	negligible	0	negligible
HIA-MAB	7.466667	2371.098	71.72102	2332.488	2409.707	0.302584	small	0.302584	small

- The novelty of this study lies in development of an HI framework for designing algorithms that can be applied to various structural optimisation problems, offering the potential for more consistent performance.
- The performance of meta-heuristics is problem-dependent, as demonstrated in the design examples investigated in this study, where the ranking of standard meta-heuristics varied across different structural optimisation problems.
- Standard meta-heuristics are prone to fail in effectively exploring the feasible regions of the search space in highly non-convex problems, as observed in structural optimisation with frequency constraints, where convergence to infeasible designs occurred.
- The results revealed that the optimum designs obtained by the proposed HIA-CF and HIA-MAB are significantly better than those yielded by the standard meta-heuristics, resulting in reductions in structural weights ranging from 8 % to 16 % for the truss structure, and from 20 % to more than 100 % for the frame structure.
- The HIA-MAB method achieved optimum designs that were approximately 3 % lighter than those obtained by HIA-CF across truss design examples, although in the frame example it was nearly 4 % heavier. This means that the results produced by both HIA-MAB and HIA-CF are quite close to each other and demonstrate a significant improvement over the standard algorithms.
- Investigation of the evolution process of the natural frequencies of structures validated the ability of both HIA-CF and HIA-MAB to identify the feasible design space and converge effectively to the optimal frequency values, indicating a successful transition from exploration of the search space to exploitation of the near-optimal frequency design space
- The key finding of this study is that the simultaneous application of multiple meta-heuristics, in an intelligent manner, yields better results than applying them individually.
- In this study, a selection of widely used algorithms, including five variants of the cultural algorithm (CA) and three variants of the differential evolution (DE) algorithm, was employed within the LLH space. However, other standard meta-heuristics can also be used within the LLH space.
- Investigating the addition of other standard algorithms to the LLH meta-heuristic pool showed that expanding the LLH space can be beneficial. It provides the HIA-CF and HIA-MAB algorithms with a broader range of options to select the most suitable meta-heuristic at the right time, potentially improving performance.
- Instead of developing entirely new meta-heuristics based on similar metaphors from nature, the focus should be on efficiently utilising existing algorithms to improve the chances of discovering optimal solutions for complex problems, a paradigm shift that is more critical now than ever before.

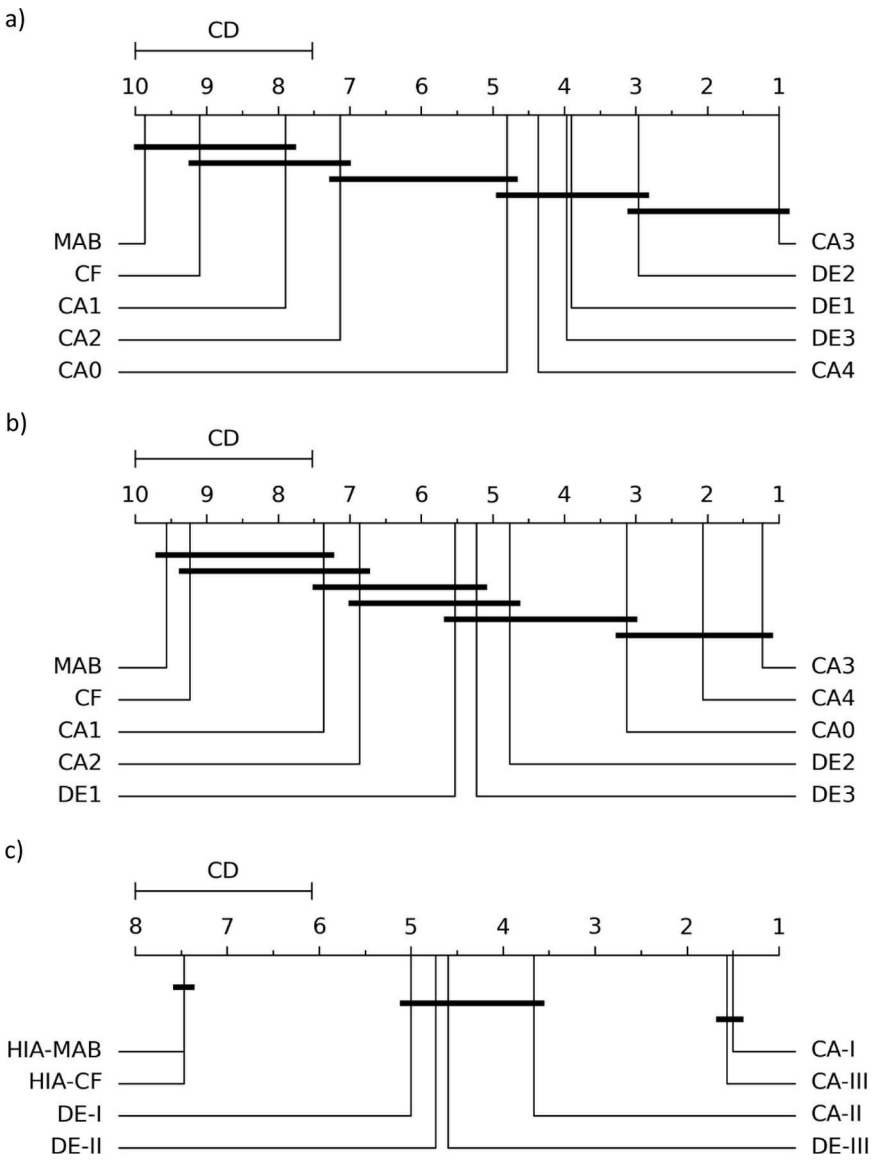


Fig. 16. Nemenyi critical distance (CD) diagrams for all examples.

CRediT authorship contribution statement

Shahin Jalili: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Siamak Talatahari:** Writing – review & editing, Visualization, Validation, Investigation, Data curation.

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.

Appendix A. Section list

Table A.1
A set of available steel pipe sections and their specifications from the AISC historic shapes database v15.0H [94]

No.	Name	Area (in^2)	Radius of gyration (in)	Unit weight (lb/ft)
1	PIPE1/2SCH40	0.250	0.261	0.852
2	PIPE1/2SCH80	0.320	0.250	1.090
3	PIPE3/4SCH40	0.333	0.334	1.130
4	PIPE3/4SCH80	0.433	0.321	1.480
5	PIPE1SCH40	0.494	0.421	1.680
6	PIPE1SCH80	0.639	0.407	2.170

(*continued on next page*)

Table A.1 (*continued*)

No.	Name	Area (in^2)	Radius of gyration (in)	Unit weight (lb/ft)
7	PIPE1–1/4SCH40	0.669	0.540	2.270
8	PIPE1–1/2SCH40	0.799	0.623	2.720
9	PIPE1–1/4SCH80	0.881	0.524	3.000
10	PIPE2SCH40	1.070	0.787	3.660
11	PIPE1–1/2SCH80	1.070	0.605	3.630
12	PIPE2SCH80	1.480	0.766	5.030
13	PIPE2–1/2SCH40	1.700	0.947	5.800
14	PIPE3SCH40	2.230	1.160	7.580
15	PIPE2–1/2SCH80	2.250	0.924	7.670
16	PIPE2XXS	2.660	0.703	9.040
17	PIPE3–1/2SCH40	2.680	1.340	9.120
18	PIPE3SCH80	3.020	1.140	10.300
19	PIPE4SCH40	3.170	1.510	10.800
20	PIPE3–1/2SCH80	3.680	1.310	12.500
21	PIPE2–1/2XXS	4.030	0.844	13.700
22	PIPE5SCH40	4.300	1.880	14.600
23	PIPE4SCH80	4.410	1.480	15.000
24	PIPE3XXS	5.470	1.050	18.600
25	PIPE6SCH40	5.580	2.250	19.000
26	PIPE5SCH80	6.110	1.840	20.800
27	PIPE4XXS	8.100	1.370	27.600
28	PIPE8SCH40	8.400	2.940	28.600
29	PIPE6SCH80	8.400	2.190	28.600
30	PIPE5XXS	11.300	1.720	38.600
31	PIPE10SCH40	11.900	3.670	40.500
32	PIPE8SCH80	12.800	2.880	43.400
33	PIPE12STD	14.600	4.380	49.600
34	PIPE6XXS	15.600	2.060	53.200
35	PIPE10SCH60	16.100	3.630	54.800
36	PIPE12XS	19.200	4.330	65.500
37	PIPE8XXS	21.300	2.760	72.500

Appendix B. Optimal design variables

Table B. 1
Comparison of optimal sections and dome heights obtained using various algorithms for the 2136-bar lamella structure

Design variables	CA-I	CA-II	CA-III	CA-IV	CA-V	DE-I	DE-II	DE-III	HIA-CF	HIA-MAB
A_1	PIPE2XXS	PIPE2SCH80	PIPE4SCH80	PIPE2–1/2SCH40	PIPE2SCH40	PIPE2XXS	PIPE12XS	PIPE4SCH80	PIPE1–1/4SCH80	PIPE1–1/4SCH40
A_2	PIPE12STD	PIPE5XXS	PIPE3SCH80	PIPE2–1/2SCH40	PIPE10SCH40	PIPE4SCH40	PIPE8SCH80	PIPE3SCH40	PIPE4SCH80	PIPE5SCH40
A_3	PIPE6SCH40	PIPE4XXS	PIPE3–1/2SCH80	PIPE2XXS	PIPE2–1/2XXS	PIPE12XS	PIPE3XXS	PIPE10SCH60	PIPE4SCH40	PIPE5SCH40
A_4	PIPE3SCH80	PIPE2XXS	PIPE3SCH40	PIPE8SCH80	PIPE5SCH80	PIPE8SCH40	PIPE10SCH60	PIPE4SCH40	PIPE3–1/2SCH40	PIPE4SCH80
A_5	PIPE2XXS	PIPE3–1/2SCH40	PIPE4SCH80	PIPE5SCH80	PIPE5SCH40	PIPE6SCH80	PIPE2–1/2XXS	PIPE6SCH80	PIPE2SCH40	PIPE1–1/2SCH80
A_6	PIPE6SCH80	PIPE5SCH80	PIPE4SCH80	PIPE2–1/2SCH80	PIPE2–1/2XXS	PIPE5SCH40	PIPE2–1/2SCH80	PIPE8SCH40	PIPE2SCH80	PIPE2SCH40
A_7	PIPE2SCH40	PIPE2–1/2XXS	PIPE3SCH80	PIPE3SCH80	PIPE4SCH40	PIPE6XXS	PIPE2XXS	PIPE2–1/2XXS	PIPE3–1/2SCH40	PIPE1–1/2SCH40
A_8	PIPE2XXS	PIPE6SCH80	PIPE3SCH80	PIPE3–1/2SCH80	PIPE3XXS	PIPE6SCH80	PIPE2–1/2SCH40	PIPE6SCH80	PIPE4SCH80	PIPE1–1/2SCH80
A_9	PIPE4SCH80	PIPE3SCH40	PIPE1SCH40	PIPE2SCH40	PIPE1–1/2SCH40	PIPE1SCH40	PIPE5SCH80	PIPE1–1/2SCH80	PIPE2XXS	PIPE1–1/4SCH80
A_{10}	PIPE6SCH40	PIPE1SCH80	PIPE2XXS	PIPE5SCH40	PIPE1–1/4SCH80	PIPE3–1/2SCH80	PIPE8SCH80	PIPE4SCH40	PIPE4SCH40	PIPE4SCH40
A_{11}	PIPE4SCH40	PIPE1–1/4SCH40	PIPE2SCH40	PIPE2–1/2SCH40	PIPE2SCH80	PIPE5SCH40	PIPE4SCH80	PIPE1–1/2SCH40	PIPE1–1/2SCH40	PIPE1–1/4SCH80
A_{12}	PIPE2–1/2SCH40	PIPE2SCH80	PIPE3SCH80	PIPE4SCH40	PIPE3–1/2SCH80	PIPE2–1/2SCH80	PIPE1–1/4SCH80	PIPE12XS	PIPE2–1/2SCH40	PIPE1–1/2SCH80
A_{13}	PIPE5SCH80	PIPE1–1/2SCH40	PIPE3–1/2SCH40	PIPE3SCH80	PIPE3SCH80	PIPE6SCH80	PIPE3SCH80	PIPE6SCH80	PIPE2SCH80	PIPE1–1/2SCH40
A_{14}	PIPE3XXS	PIPE2–1/2SCH40	PIPE2–1/2SCH40	PIPE1SCH40	PIPE2SCH80	PIPE6SCH40	PIPE3SCH80	PIPE3SCH40	PIPE1–1/4SCH80	PIPE3–1/2SCH40
A_{15}	PIPE2SCH40	PIPE4SCH40	PIPE3–1/2SCH40	PIPE2–1/2SCH80	PIPE6SCH40	PIPE4SCH40	PIPE4SCH40	PIPE6SCH40	PIPE2–1/2SCH40	PIPE2SCH80
A_{16}	PIPE3–1/2SCH40	PIPE1–1/2SCH80	PIPE2XXS	PIPE4SCH80	PIPE2–1/2SCH80	PIPE1–1/4SCH80	PIPE1–1/4SCH80	PIPE12XS	PIPE1–1/4SCH80	PIPE4SCH40

(*continued on next page*)

Table B. 1 (*continued*)

Design variables	CA-I	CA-II	CA-III	CA-IV	CA-V	DE-I	DE-II	DE-III	HIA-CF	HIA-MAB
A_{17}	PIPE2SCH40	PIPE3SCH40	PIPE2SCH80	PIPE2XXS	PIPE3–1/2SCH40	PIPE1–1/2SCH40	PIPE2XXS	PIPE2SCH80	PIPE2–1/2SCH80	PIPE1–1/2SCH80
A_{18}	PIPE4SCH80	PIPE3SCH40	PIPE3SCH80	PIPE5SCH40	PIPE4SCH80	PIPE4SCH80	PIPE3SCH40	PIPE2–1/2SCH80	PIPE3–1/2SCH40	PIPE2SCH80
A_{19}	PIPE10SCH40	PIPE2–1/2XXS	PIPE2–1/2XXS	PIPE8SCH40	PIPE4SCH40	PIPE3–1/2SCH80	PIPE1–1/2SCH80	PIPE6XXS	PIPE3–1/2SCH80	PIPE2–1/2SCH40
A_{20}	PIPE3–1/2SCH80	PIPE2–1/2SCH40	PIPE3SCH80	PIPE2–1/2XXS	PIPE3–1/2SCH40	PIPE1–1/4SCH40	PIPE8SCH40	PIPE2XXS	PIPE3–1/2SCH40	PIPE2–1/2SCH40
A_{21}	PIPE2XXS	PIPE2SCH80	PIPE4SCH40	PIPE1–1/2SCH40	PIPE5XXS	PIPE1–1/4SCH80	PIPE2XXS	PIPE3SCH80	PIPE5SCH40	PIPE2–1/2SCH40
A_{22}	PIPE3SCH40	PIPE3–1/2SCH40	PIPE2SCH80	PIPE2–1/2SCH40	PIPE2–1/2XXS	PIPE1–1/4SCH80	PIPE2–1/2SCH40	PIPE2XXS	PIPE2–1/2SCH80	PIPE3SCH40
A_{23}	PIPE3–1/2SCH40	PIPE2SCH80	PIPE3SCH80	PIPE2SCH40	PIPE4SCH40	PIPE6SCH40	PIPE2–1/2SCH40	PIPE1–1/4SCH80	PIPE2–1/2SCH40	PIPE2–1/2SCH80
A_{24}	PIPE3XXS	PIPE2–1/2SCH80	PIPE3SCH80	PIPE2–1/2XXS	PIPE2–1/2SCH80	PIPE3SCH40	PIPE5SCH40	PIPE6SCH80	PIPE3–1/2SCH40	PIPE2SCH80
A_{25}	PIPE3SCH80	PIPE2XXS	PIPE3SCH80	PIPE1–1/2SCH80	PIPE2–1/2SCH40	PIPE6SCH40	PIPE3SCH40	PIPE3–1/2SCH80	PIPE3SCH80	PIPE2XXS
A_{26}	PIPE2XXS	PIPE3SCH80	PIPE4SCH40	PIPE2SCH80	PIPE3XXS	PIPE3SCH80	PIPE3–1/2SCH80	PIPE5XXS	PIPE2SCH80	PIPE3–1/2SCH40
A_{27}	PIPE2XXS	PIPE3XXS	PIPE2SCH80	PIPE5SCH80	PIPE3–1/2SCH40	PIPE4SCH40	PIPE3–1/2SCH40	PIPE3SCH80	PIPE2–1/2SCH80	PIPE2XXS
A_{28}	PIPE6SCH40	PIPE2XXS	PIPE3SCH80	PIPE2–1/2SCH80	PIPE2–1/2XXS	PIPE3XXS	PIPE2–1/2XXS	PIPE3–1/2SCH80	PIPE2–1/2SCH80	PIPE3–1/2SCH40
A_{29}	PIPE6SCH40	PIPE4SCH40	PIPE3SCH40	PIPE1–1/2SCH40	PIPE3SCH80	PIPE3SCH80	PIPE4SCH80	PIPE3SCH80	PIPE2–1/2SCH80	PIPE2XXS
A_{30}	PIPE2–1/2XXS	PIPE3SCH40	PIPE4SCH40	PIPE1–1/4SCH40	PIPE4SCH80	PIPE3SCH80	PIPE6SCH40	PIPE3SCH80	PIPE2–1/2SCH80	PIPE4SCH40
A_{31}	PIPE4SCH40	PIPE2XXS	PIPE3–1/2SCH40	PIPE1–1/2SCH80	PIPE4SCH80	PIPE5SCH40	PIPE3SCH80	PIPE2–1/2SCH40	PIPE3–1/2SCH40	PIPE3–1/2SCH40
A_{32}	PIPE3SCH80	PIPE2SCH80	PIPE2–1/2SCH80	PIPE1–1/4SCH80	PIPE5SCH40	PIPE2–1/2SCH40	PIPE4SCH40	PIPE4SCH40	PIPE3–1/2SCH40	PIPE3SCH40
A_{33}	PIPE4SCH40	PIPE2SCH80	PIPE3–1/2SCH40	PIPE1–1/2SCH80	PIPE5SCH80	PIPE2–1/2SCH80	PIPE3SCH40	PIPE2XXS	PIPE4SCH40	PIPE3–1/2SCH40
A_{34}	PIPE1–1/2SCH80	PIPE3–1/2SCH40	PIPE2–1/2SCH80	PIPE2–1/2SCH80	PIPE2–1/2XXS	PIPE5SCH40	PIPE2XXS	PIPE2–1/2SCH80	PIPE3SCH40	PIPE2–1/2SCH80
A_{35}	PIPE3XXS	PIPE3–1/2SCH40	PIPE2XXS	PIPE6SCH80	PIPE3SCH80	PIPE3–1/2SCH40	PIPE2XXS	PIPE4SCH40	PIPE3–1/2SCH40	PIPE3–1/2SCH40
A_{36}	PIPE5SCH40	PIPE2–1/2SCH40	PIPE2–1/2SCH80	PIPE3SCH40	PIPE2–1/2XXS	PIPE4SCH40	PIPE3SCH40	PIPE4SCH40	PIPE2–1/2SCH80	PIPE3–1/2SCH40
A_{37}	PIPE4SCH40	PIPE3–1/2SCH80	PIPE2XXS	PIPE2–1/2SCH40	PIPE3–1/2SCH80	PIPE2–1/2SCH40	PIPE2XXS	PIPE2XXS	PIPE2–1/2SCH80	PIPE3–1/2SCH80
A_{38}	PIPE2XXS	PIPE3–1/2SCH40	PIPE3SCH40	PIPE8SCH40	PIPE3–1/2SCH80	PIPE3XXS	PIPE1–1/2SCH80	PIPE4SCH80	PIPE3SCH40	PIPE4SCH40
A_{39}	PIPE3–1/2SCH40	PIPE2–1/2SCH40	PIPE2–1/2XXS	PIPE2SCH40	PIPE2XXS	PIPE2–1/2XXS	PIPE2–1/2SCH80	PIPE2–1/2SCH80	PIPE2XXS	PIPE3–1/2SCH40
A_{40}	PIPE3–1/2SCH80	PIPE3SCH80	PIPE2XXS	PIPE2SCH80	PIPE3–1/2SCH40	PIPE2SCH80	PIPE2–1/2SCH40	PIPE5SCH80	PIPE3–1/2SCH40	PIPE3SCH40
A_{41}	PIPE3–1/2SCH80	PIPE3SCH80	PIPE3–1/2SCH40	PIPE2–1/2SCH40	PIPE3SCH80	PIPE3–1/2SCH40	PIPE2SCH80	PIPE3–1/2SCH40	PIPE2–1/2SCH80	PIPE3SCH80
A_{42}	PIPE2SCH40	PIPE3–1/2SCH80	PIPE3SCH40	PIPE2–1/2SCH40	PIPE4SCH40	PIPE3–1/2SCH80	PIPE5SCH80	PIPE3SCH80	PIPE4SCH40	PIPE2XXS
A_{43}	PIPE2–1/2SCH40	PIPE2–1/2SCH80	PIPE2SCH80	PIPE6SCH80	PIPE1–1/2SCH80	PIPE2–1/2SCH40	PIPE3–1/2SCH40	PIPE6SCH40	PIPE2–1/2SCH80	PIPE2–1/2SCH80
A_{44}	PIPE4SCH80	PIPE3SCH40	PIPE3–1/2SCH80	PIPE3SCH80	PIPE5SCH40	PIPE8SCH40	PIPE3SCH80	PIPE5SCH80	PIPE2SCH80	PIPE2–1/2SCH80
A_{45}	PIPE2–1/2XXS	PIPE2XXS	PIPE3SCH80	PIPE3–1/2SCH80	PIPE5SCH40	PIPE3SCH80	PIPE3XXS	PIPE3–1/2SCH40	PIPE3SCH40	PIPE2–1/2SCH80
A_{46}	PIPE3SCH40	PIPE3SCH40	PIPE3–1/2SCH40	PIPE3SCH40	PIPE3–1/2SCH40	PIPE3–1/2SCH40	PIPE4SCH80	PIPE5SCH40	PIPE2–1/2SCH40	PIPE2SCH80
A_{47}	PIPE2–1/2SCH40	PIPE3SCH40	PIPE3–1/2SCH40	PIPE2–1/2SCH40	PIPE3–1/2SCH40	PIPE2–1/2XXS	PIPE4SCH80	PIPE3–1/2SCH80	PIPE2XXS	PIPE2–1/2SCH40
A_{48}	PIPE1–1/2SCH80	PIPE2–1/2XXS	PIPE2SCH80	PIPE6SCH80	PIPE3SCH40	PIPE3–1/2SCH40	PIPE4SCH40	PIPE3–1/2SCH40	PIPE2–1/2SCH40	PIPE2–1/2SCH40
A_{49}	PIPE3SCH80	PIPE2–1/2SCH40	PIPE3–1/2SCH40	PIPE3SCH40	PIPE4SCH40	PIPE2SCH40	PIPE3SCH80	PIPE1–1/2SCH80	PIPE2–1/2SCH40	PIPE2SCH80
A_{50}	PIPE1–1/2SCH80	PIPE1–1/2SCH80	PIPE1SCH80	PIPE2–1/2SCH40	PIPE2–1/2XXS	PIPE2XXS	PIPE8SCH40	PIPE1–1/4SCH40	PIPE1–1/4SCH80	PIPE1–1/4SCH80
A_{51}	PIPE3SCH40	PIPE2SCH80	PIPE1–1/2SCH80	PIPE1–1/4SCH80	PIPE2–1/2XXS	PIPE4SCH40	PIPE2SCH80	PIPE1–1/4SCH80	PIPE2–1/2SCH40	PIPE1–1/2SCH80
A_{52}	PIPE1SCH40	PIPE2–1/2SCH40	PIPE1–1/2SCH40	PIPE2SCH80	PIPE1–1/2SCH80	PIPE1–1/2SCH80	PIPE1–1/4SCH80	PIPE3SCH80	PIPE1–1/2SCH80	PIPE2SCH40
A_{53}	PIPE2–1/2SCH80	PIPE2–1/2SCH40	PIPE4SCH40	PIPE1–1/2SCH80	PIPE2SCH40	PIPE1–1/2SCH80	PIPE5SCH40	PIPE1SCH80	PIPE1–1/2SCH80	PIPE3–1/2SCH40

(*continued on next page*)

Table B. 1 (*continued*)

Design variables	CA-I	CA-II	CA-III	CA-IV	CA-V	DE-I	DE-II	DE-III	HIA-CF	HIA-MAB
A_{54}	PIPE1–1/ 4SCH40	PIPE1–1/ 2SCH80	PIPE1–1/ 2SCH40	PIPE2SCH80	PIPE1–1/ 4SCH80	PIPE2SCH80	PIPE1–1/ 2SCH40	PIPE2SCH80	PIPE1–1/ 2SCH40	PIPE1–1/ 4SCH80
A_{55}	PIPE1–1/ 2SCH80	PIPE1–1/ 4SCH80	PIPE2XXS	PIPE3–1/ 2SCH40	PIPE1SCH80	PIPE2SCH80	PIPE3SCH40	PIPE4SCH40	PIPE2SCH40	PIPE1–1/ 2SCH80
A_{56}	PIPE3–1/ 2SCH40	PIPE1–1/ 4SCH40	PIPE1SCH80	PIPE1–1/ 4SCH40	PIPE3–1/ 2SCH80	PIPE2–1/ 2SCH40	PIPE3/4SCH40	PIPE4SCH80	PIPE1–1/ 4SCH40	PIPE1–1/ 2SCH40
A_{57}	PIPE2–1/ 2SCH40	PIPE3–1/ 2SCH40	PIPE3–1/ 2SCH40	PIPE1–1/ 4SCH80	PIPE2–1/2XXS	PIPE1–1/ 4SCH40	PIPE3SCH80	PIPE3–1/ 2SCH40	PIPE1–1/ 2SCH80	PIPE1–1/ 2SCH80
A_{58}	PIPE5SCH80	PIPE1–1/ 4SCH80	PIPE3–1/ 2SCH40	PIPE2SCH40	PIPE5SCH80	PIPE2–1/ 2SCH80	PIPE3SCH80	PIPE2SCH40	PIPE2SCH80	PIPE1–1/ 4SCH80
A_{59}	PIPE4SCH40	PIPE2XXS	PIPE2SCH40	PIPE5SCH80	PIPE5SCH40	PIPE2XXS	PIPE6SCH80	PIPE3SCH40	PIPE2–1/ 2SCH80	PIPE2–1/ 2SCH40
A_{60}	PIPE2XXS	PIPE1–1/ 4SCH80	PIPE4SCH40	PIPE2–1/ 2SCH40	PIPE3–1/ 2SCH80	PIPE3XXS	PIPE1–1/ 4SCH80	PIPE1–1/ 4SCH40	PIPE1SCH80	PIPE1SCH40
h (m)	8.41028	8.65110	11.31842	8.00026	14.67116	11.17885	11.78097	8.90607	11.27247	11.33181

Table B. 2
Comparison of optimal sections obtained using various algorithms for the 3341-bar double-layer barrel vault structure

Design variables	CA-I	CA-II	CA-III	CA-IV	CA-V	DE-I	DE-II	DE-III	HIA-CF	HIA-MAB
A_1	PIPE3/ 4SCH80	PIPE4SCH80	PIPE6SCH40	PIPE5XXS	PIPE1–1/ 2SCH40	PIPE2–1/ 2SCH80	PIPE3–1/ 2SCH80	PIPE3–1/ 2SCH80	PIPE1–1/ 2SCH40	PIPE2–1/ 2SCH80
A_2	PIPE3–1/ 2SCH40	PIPE3SCH80	PIPE2XXS	PIPE1–1/ 4SCH80	PIPE2–1/ 2SCH40	PIPE1–1/ 2SCH40	PIPE1/ 2SCH80	PIPE4SCH80	PIPE3SCH40	PIPE3–1/ 2SCH40
A_3	PIPE4SCH40	PIPE3SCH40	PIPE1–1/ 2SCH80	PIPE2–1/ 2SCH80	PIPE6SCH40	PIPE2XXS	PIPE2SCH40	PIPE4SCH80	PIPE1–1/ 2SCH80	PIPE3SCH40
A_4	PIPE3–1/ 2SCH40	PIPE3–1/ 2SCH80	PIPE3–1/ 2SCH80	PIPE6XXS	PIPE3SCH80	PIPE3/ 4SCH40	PIPE3–1/ 2SCH80	PIPE3–1/ 2SCH40	PIPE1SCH80	PIPE1–1/ 2SCH40
A_5	PIPE5SCH80	PIPE1–1/ 2SCH40	PIPE2–1/ 2SCH80	PIPE8XXS	PIPE3–1/ 2SCH40	PIPE3XXS	PIPE2SCH80	PIPE2SCH80	PIPE2SCH40	PIPE4SCH40
A_6	PIPE1–1/ 4SCH40	PIPE4SCH80	PIPE3–1/ 2SCH80	PIPE3–1/ 2SCH80	PIPE2–1/ 2SCH80	PIPE3SCH80	PIPE2–1/ 2SCH80	PIPE1–1/ 4SCH40	PIPE2–1/ 2SCH40	PIPE3–1/ 2SCH40
A_7	PIPE1–1/ 2SCH40	PIPE2XXS	PIPE3XXS	PIPE1/ 2SCH80	PIPE2SCH40	PIPE2–1/ 2XXS	PIPE1–1/ 2SCH40	PIPE5SCH80	PIPE4SCH40	PIPE2–1/ 2SCH80
A_8	PIPE2–1/ 2XXS	PIPE3SCH80	PIPE4SCH40	PIPE2–1/ 2XXS	PIPE3–1/ 2SCH40	PIPE2XXS	PIPE3–1/ 2SCH80	PIPE3XXS	PIPE2–1/ 2SCH80	PIPE2XXS
A_9	PIPE4SCH40	PIPE2SCH80	PIPE5SCH40	PIPE8SCH40	PIPE2SCH40	PIPE3SCH80	PIPE3SCH40	PIPE1–1/ 2SCH80	PIPE3–1/ 2SCH80	PIPE2XXS
A_{10}	PIPE3–1/ 2SCH80	PIPE3–1/ 2SCH40	PIPE10SCH40	PIPE2–1/ 2XXS	PIPE8SCH40	PIPE3XXS	PIPE6XXS	PIPE2–1/ 2XXS	PIPE3–1/ 2SCH80	PIPE2–1/ 2SCH80
A_{11}	PIPE2XXS	PIPE4SCH80	PIPE3XXS	PIPE3–1/ 2SCH80	PIPE5SCH40	PIPE2–1/ 2XXS	PIPE5SCH80	PIPE6SCH40	PIPE4SCH80	PIPE5SCH40
A_{12}	PIPE3–1/ 2SCH80	PIPE6SCH40	PIPE5SCH40	PIPE3SCH40	PIPE3SCH80	PIPE5SCH40	PIPE2–1/ 2SCH80	PIPE3–1/ 2SCH40	PIPE6SCH40	PIPE4SCH40
A_{13}	PIPE3SCH80	PIPE4XXS	PIPE3–1/ 2SCH40	PIPE6XXS	PIPE3SCH40	PIPE3SCH40	PIPE2–1/ 2XXS	PIPE6SCH40	PIPE3–1/ 2SCH80	PIPE2–1/ 2XXS
A_{14}	PIPE2–1/ 2XXS	PIPE1–1/ 4SCH80	PIPE6SCH40	PIPE1/ 2SCH80	PIPE3–1/ 2SCH80	PIPE4SCH40	PIPE3SCH40	PIPE1–1/ 2SCH80	PIPE5SCH40	PIPE4SCH40
A_{15}	PIPE8SCH40	PIPE3–1/ 2SCH80	PIPE6SCH40	PIPE1–1/ 4SCH40	PIPE5SCH40	PIPE4SCH40	PIPE3XXS	PIPE5XXS	PIPE3–1/ 2SCH80	PIPE4SCH40
A_{16}	PIPE2SCH80	PIPE3XXS	PIPE5SCH40	PIPE1SCH40	PIPE3–1/ 2SCH80	PIPE3XXS	PIPE2–1/ 2XXS	PIPE5SCH40	PIPE2XXS	PIPE3SCH80
A_{17}	PIPE2SCH40	PIPE4SCH40	PIPE5SCH40	PIPE3SCH40	PIPE3SCH40	PIPE5SCH40	PIPE4SCH40	PIPE3XXS	PIPE2–1/ 2XXS	PIPE4SCH40
A_{18}	PIPE1–1/ 2SCH80	PIPE3–1/ 2SCH40	PIPE4SCH40	PIPE3–1/ 2SCH80	PIPE4SCH40	PIPE5SCH40	PIPE1–1/ 4SCH80	PIPE3XXS	PIPE6SCH40	PIPE2–1/ 2SCH80
A_{19}	PIPE4SCH80	PIPE5SCH80	PIPE4XXS	PIPE6XXS	PIPE5SCH40	PIPE1SCH40	PIPE2–1/ 2XXS	PIPE3–1/ 2SCH80	PIPE5SCH40	PIPE5SCH40
A_{20}	PIPE3SCH40	PIPE8SCH40	PIPE4SCH80	PIPE5XXS	PIPE2XXS	PIPE2SCH80	PIPE3SCH80	PIPE2SCH80	PIPE4SCH80	PIPE2–1/ 2SCH80
A_{21}	PIPE5XXS	PIPE2–1/ 2SCH40	PIPE3–1/ 2SCH40	PIPE8SCH40	PIPE2XXS	PIPE2XXS	PIPE2SCH40	PIPE2SCH40	PIPE2–1/ 2SCH40	PIPE3SCH80
A_{22}	PIPE6SCH80	PIPE3XXS	PIPE2–1/ 2SCH40	PIPE3/ 4SCH40	PIPE5SCH80	PIPE4XXS	PIPE2–1/ 2XXS	PIPE5XXS	PIPE6SCH80	PIPE4SCH40
A_{23}	PIPE1–1/ 4SCH80	PIPE3–1/ 2SCH40	PIPE5SCH40	PIPE3–1/ 2SCH40	PIPE4XXS	PIPE1SCH40	PIPE3SCH80	PIPE1–1/ 4SCH80	PIPE4XXS	PIPE3–1/ 2SCH40
A_{24}	PIPE5SCH40	PIPE1–1/ 2SCH40	PIPE6SCH80	PIPE3/ 4SCH40	PIPE4SCH40	PIPE1–1/ 4SCH80	PIPE1SCH80	PIPE3/ 4SCH80	PIPE3–1/ 2SCH40	PIPE1–1/ 2SCH80
A_{25}	PIPE1/ 2SCH40	PIPE5SCH40	PIPE3–1/ 2SCH40	PIPE1–1/ 4SCH40	PIPE2SCH40	PIPE4SCH40	PIPE8SCH80	PIPE4SCH40	PIPE3XXS	PIPE2XXS

(*continued on next page*)

Table B. 2 (*continued*)

Design variables	CA-I	CA-II	CA-III	CA-IV	CA-V	DE-I	DE-II	DE-III	HIA-CF	HIA-MAB
A_{26}	PIPE5SCH40	PIPE2SCH40	PIPE3–1/ 2SCH40	PIPE1–1/ 4SCH80	PIPE3–1/ 2SCH80	PIPE3SCH40	PIPE4XXS	PIPE4SCH40	PIPE4SCH80	PIPE4SCH40
A_{27}	PIPE1–1/ 2SCH40	PIPE2–1/ 2XXS	PIPE6SCH40	PIPE5SCH40	PIPE3SCH40	PIPE3SCH80	PIPE3/ 4SCH40	PIPE3SCH40	PIPE1–1/ 4SCH40	PIPE4SCH40
A_{28}	PIPE4XXS	PIPE6SCH40	PIPE2SCH40	PIPE2–1/ 2SCH80	PIPE3XXS	PIPE3–1/ 2SCH40	PIPE1–1/ 4SCH40	PIPE2–1/ 2XXS	PIPE1SCH80	PIPE1–1/ 4SCH40
A_{29}	PIPE2–1/ 2SCH80	PIPE4SCH80	PIPE2–1/ 2XXS	PIPE5XXS	PIPE5SCH40	PIPE3/ 4SCH80	PIPE4SCH40	PIPE1–1/ 2SCH80	PIPE1/ 2SCH80	PIPE1/ 2SCH40
A_{30}	PIPE8SCH40	PIPE1–1/ 4SCH80	PIPE2XXS	PIPE6XXS	PIPE3–1/ 2SCH80	PIPE1–1/ 2SCH40	PIPE1–1/ 2SCH40	PIPE3SCH80	PIPE4SCH40	PIPE1–1/ 4SCH80
A_{31}	PIPE8SCH40	PIPE5SCH40	PIPE4XXS	PIPE1SCH40	PIPE4SCH80	PIPE1–1/ 4SCH80	PIPE1SCH80	PIPE1–1/ 4SCH40	PIPE2–1/ 2SCH80	PIPE1–1/ 2SCH80
A_{32}	PIPE1SCH40	PIPE2–1/ 2SCH40	PIPE2–1/ 2SCH80	PIPE1SCH80	PIPE3SCH80	PIPE2SCH40	PIPE3SCH80	PIPE2SCH40	PIPE1–1/ 2SCH80	PIPE1–1/ 2SCH80
A_{33}	PIPE5SCH40	PIPE8SCH40	PIPE2–1/ 2XXS	PIPE3/ 4SCH80	PIPE5SCH40	PIPE1–1/ 4SCH40	PIPE3/ 4SCH40	PIPE3SCH40	PIPE2–1/ 2SCH40	PIPE2–1/ 2SCH40
A_{34}	PIPE2SCH40	PIPE2SCH80	PIPE3–1/ 2SCH40	PIPE2–1/ 2SCH40	PIPE6SCH80	PIPE2SCH40	PIPE2–1/ 2SCH40	PIPE5SCH40	PIPE2SCH80	PIPE2SCH80
A_{35}	PIPE3SCH40	PIPE1–1/ 2SCH80	PIPE6SCH40	PIPE3–1/ 2SCH40	PIPE3–1/ 2SCH40	PIPE3–1/ 2SCH40	PIPE2XXS	PIPE1–1/ 2SCH80	PIPE2–1/ 2SCH40	PIPE1–1/ 2SCH80
A_{36}	PIPE5SCH80	PIPE4SCH80	PIPE1–1/ 2SCH80	PIPE1/ 2SCH40	PIPE4SCH40	PIPE1–1/ 4SCH80	PIPE3SCH40	PIPE6SCH40	PIPE2–1/ 2SCH80	PIPE1–1/ 4SCH80
A_{37}	PIPE1–1/ 4SCH40	PIPE3SCH40	PIPE2–1/ 2SCH40	PIPE1–1/ 2SCH80	PIPE5SCH40	PIPE2SCH80	PIPE2XXS	PIPE3SCH40	PIPE2SCH40	PIPE2SCH80
A_{38}	PIPE2–1/ 2SCH80	PIPE3SCH40	PIPE4SCH80	PIPE6SCH80	PIPE3XXS	PIPE3–1/ 2SCH80	PIPE3SCH40	PIPE4SCH40	PIPE2–1/ 2SCH40	PIPE2SCH80
A_{39}	PIPE5SCH80	PIPE2–1/ 2SCH80	PIPE3XXS	PIPE3/ 4SCH40	PIPE3–1/ 2SCH80	PIPE1–1/ 2SCH40	PIPE3–1/ 2SCH40	PIPE3–1/ 2SCH80	PIPE3–1/ 2SCH40	PIPE4SCH40
A_{40}	PIPE3–1/ 2SCH40	PIPE3SCH40	PIPE4XXS	PIPE1–1/ 4SCH40	PIPE5SCH40	PIPE2–1/ 2SCH80	PIPE6SCH40	PIPE2–1/ 2SCH80	PIPE2–1/ 2SCH40	PIPE2SCH80
A_{41}	PIPE2SCH40	PIPE4SCH80	PIPE3–1/ 2SCH80	PIPE5SCH80	PIPE6SCH40	PIPE2SCH40	PIPE3SCH40	PIPE5SCH40	PIPE2SCH80	PIPE4SCH40
A_{42}	PIPE2–1/ 2SCH80	PIPE3–1/ 2SCH40	PIPE3SCH40	PIPE1–1/ 4SCH80	PIPE1–1/ 4SCH80	PIPE4SCH40	PIPE2–1/ 2SCH80	PIPE2–1/ 2SCH40	PIPE2XXS	PIPE2SCH80
A_{43}	PIPE4SCH40	PIPE5SCH40	PIPE4SCH80	PIPE3SCH40	PIPE5SCH40	PIPE2XXS	PIPE4XXS	PIPE1–1/ 2SCH40	PIPE2–1/ 2SCH80	PIPE2SCH80
A_{44}	PIPE4XXS	PIPE3–1/ 2SCH80	PIPE2–1/ 2SCH80	PIPE4SCH80	PIPE2–1/ 2SCH80	PIPE4SCH80	PIPE2SCH40	PIPE3SCH80	PIPE2SCH80	PIPE5SCH40
A_{45}	PIPE2–1/ 2XXS	PIPE3SCH40	PIPE2SCH40	PIPE5XXS	PIPE3SCH40	PIPE3–1/ 2SCH80	PIPE2XXS	PIPE1–1/ 4SCH80	PIPE2SCH80	PIPE2–1/ 2SCH40
A_{46}	PIPE3–1/ 2SCH80	PIPE1–1/ 2SCH80	PIPE5SCH40	PIPE6SCH80	PIPE1–1/ 2SCH80	PIPE1SCH40	PIPE1–1/ 4SCH80	PIPE4SCH40	PIPE2SCH80	PIPE2XXS
A_{47}	PIPE5XXS	PIPE4XXS	PIPE2XXS	PIPE2XXS	PIPE6SCH80	PIPE3SCH80	PIPE3–1/ 2SCH80	PIPE1/ 2SCH40	PIPE2XXS	PIPE2–1/ 2SCH40
A_{48}	PIPE12STD	PIPE4SCH80	PIPE4SCH40	PIPE1/ 2SCH80	PIPE2SCH80	PIPE6XXS	PIPE1–1/ 4SCH40	PIPE1SCH40	PIPE1–1/ 4SCH40	PIPE2–1/ 2XXS
A_{49}	PIPE1–1/ 2SCH80	PIPE6SCH80	PIPE3–1/ 2SCH40	PIPE2–1/ 2SCH40	PIPE5SCH40	PIPE1SCH40	PIPE3–1/ 2SCH40	PIPE1/ 2SCH40	PIPE3/ 4SCH80	PIPE1–1/ 4SCH40
A_{50}	PIPE3–1/ 2SCH40	PIPE2SCH40	PIPE1SCH80	PIPE3/ 4SCH80	PIPE4SCH80	PIPE1–1/ 2SCH80	PIPE4SCH40	PIPE4XXS	PIPE3–1/ 2SCH40	PIPE1–1/ 2SCH80
A_{51}	PIPE4SCH40	PIPE3SCH80	PIPE1–1/ 2SCH40	PIPE2SCH80	PIPE3–1/ 2SCH40	PIPE4SCH40	PIPE4SCH40	PIPE3SCH80	PIPE2XXS	PIPE2XXS
A_{52}	PIPE3/ 4SCH80	PIPE1/ 2SCH80	PIPE1/ 2SCH80	PIPE1–1/ 2SCH40	PIPE1–1/ 4SCH40	PIPE1–1/ 2SCH40	PIPE1SCH80	PIPE1–1/ 4SCH40	PIPE1–1/ 4SCH40	PIPE1/ 2SCH80
A_{53}	PIPE3/ 4SCH40	PIPE3/ 4SCH80	PIPE1/ 2SCH80	PIPE1SCH80	PIPE1–1/ 2SCH40	PIPE1SCH80	PIPE1–1/ 2SCH40	PIPE1–1/ 4SCH40	PIPE1SCH80	PIPE3/ 4SCH40
A_{54}	PIPE1–1/ 4SCH80	PIPE1–1/ 2SCH40	PIPE2–1/ 2XXS	PIPE1SCH40	PIPE1/ 2SCH80	PIPE3/ 4SCH40	PIPE3/ 4SCH40	PIPE3/ 4SCH80	PIPE1/ 2SCH80	PIPE1–1/ 4SCH40
A_{55}	PIPE1SCH80	PIPE1SCH80	PIPE2XXS	PIPE1/ 2SCH40	PIPE3/ 4SCH80	PIPE3/ 4SCH40	PIPE3/ 4SCH80	PIPE1/ 2SCH40	PIPE3/ 4SCH40	PIPE1SCH40

Table B. 3
Comparison of optimal sections obtained using various algorithms for the 19-story steel frame structure

Design variables	CA-I	CA-II	CA-III	DE-I	DE-II	DE-III	HIA-CF	HIA-MAB	CSS (Kaveh et al. [91])	ACSS (Kaveh et al. [91])
A_1	W24X492	W40X466	W36X650	W40X466	W14X311	W36X260	W30X477	W30X477	W36X848	W14X500
A_2	W14X193	W14X193	W18X175	W14X193	W12X336	W30X191	W27X217	W14X193	W27X258	W24X492
A_3	W18X311	W12X279	W18X311	W12X279	W40X331	W36X527	W14X257	W36X256	W33X318	W30X211
A_4	W18X234	W40X503	W36X328	W40X503	W14X426	W24X408	W44X230	W36X232	W40X331	W40X278
A_5	W14X257	W18X211	W24X335	W40X215	W40X321	W40X321	W30X211	W18X211	W12X279	W40X249

(*continued on next page*)

Table B. 3 (*continued*)

Design variables	CA-I	CA-II	CA-III	DE-I	DE-II	DE-III	HIA-CF	HIA-MAB	CSS (Kaveh et al. [91])	ACSS (Kaveh et al. [91])
A_6	W14X398	W18X35	W21X132	W18X35	W8X35	W10X30	W24X62	W12X53	W33X169	W10X45
A_7	W33X263	W40X183	W33X169	W40X183	W33X141	W21X111	W12X50	W30X124	W36X439	W36X256
A_8	W33X318	W33X263	W27X539	W33X263	W30X211	W27X368	W33X263	W33X263	W14X145	W36X160
A_9	W36X359	W24X408	W14X605	W14X398	W36X280	W33X291	W14X550	W36X393	W14X398	W24X492
A_{10}	W27X368	W14X550	W27X368	W14X550	W24X408	W36X359	W14X550	W14X550	W33X241	W36X527
A_{11}	W12X305	W12X336	W30X292	W36X328	W30X477	W44X290	W36X328	W44X335	W36X328	W30X391
A_{12}	W33X263	W40X211	W36X260	W14X233	W36X280	W14X311	W36X194	W40X235	W36X359	W33X263
A_{13}	W27X368	W36X194	W40X249	W12X190	W40X277	W12X230	W24X176	W27X194	W24X279	W40X211
A_{14}	W33X263	W33X169	W14X605	W33X169	W36X150	W30X173	W40X149	W40X149	W30X90	W30X90
A_{15}	W36X527	W14X283	W14X311	W36X280	W44X230	W36X300	W30X108	W30X108	W40X278	W40X174
A_{16}	W27X448	W36X245	W14X605	W12X72	W18X175	W27X161	W18X86	W18X86	W36X160	W24X131
A_{17}	W33X241	W14X605	W24X176	W14X605	W33X201	W40X264	W40X431	W14X665	W24X408	W36X359
A_{18}	W14X500	W33X263	W40X321	W33X263	W36X359	W40X199	W33X263	W33X263	W14X500	W36X393
A_{19}	W36X328	W14X342	W40X431	W14X342	W40X503	W40X331	W12X305	W12X336	W18X258	W27X235
A_{20}	W30X477	W30X191	W40X431	W36X160	W14X342	W24X408	W14X193	W14X193	W36X280	W27X258
A_{21}	W18X258	W14X342	W14X500	W14X342	W40X372	W33X241	W14X176	W14X176	W27X258	W14X342
A_{22}	W27X258	W30X132	W33X354	W30X132	W33X152	W18X158	W30X132	W30X132	W33X118	W36X135
A_{23}	W40X466	W44X290	W40X372	W44X290	W40X297	W40X431	W40X215	W40X278	W40X215	W44X290
A_{24}	W40X593	W14X193	W40X431	W14X193	W40X199	W36X232	W14X193	W14X193	W40X277	W40X149
A_{25}	W14X283	W33X201	W24X335	W33X201	W14X398	W14X500	W36X328	W36X300	W33X263	W36X359
A_{26}	W24X279	W12X170	W30X391	W12X170	W14X370	W33X291	W40X199	W40X199	W36X280	W33X291
A_{27}	W40X593	W14X426	W24X146	W14X426	W36X182	W33X354	W24X408	W14X426	W30X477	W40X167
A_{28}	W40X372	W30X326	W14X550	W30X326	W27X448	W14X398	W14X311	W14X311	W14X257	W44X262
A_{29}	W14X605	W27X368	W10X88	W27X368	W44X262	W40X211	W36X359	W24X408	W36X280	W27X539
A_{30}	W44X230	W40X249	W33X354	W27X161	W40X235	W44X230	W30X116	W30X116	W36X170	W24X76
A_{31}	W40X431	W40X211	W36X300	W40X211	W36X194	W44X230	W40X211	W40X211	W40X199	W40X199
A_{32}	W14X500	W18X192	W14X257	W36X280	W18X258	W14X176	W30X261	W30X261	W33X152	W30X235
A_{33}	W40X235	W24X162	W21X182	W21X166	W30X292	W24X279	W12X230	W36X210	W14X176	W40X199
A_{34}	W18X311	W14X455	W14X808	W14X455	W40X297	W21X132	W14X455	W14X455	W14X257	W14X455
A_{35}	W36X280	W30X108	W36X393	W30X108	W18X258	W12X87	W30X108	W24X103	W36X393	W36X439
A_{36}	W40X331	W24X131	W27X114	W30X191	W14X311	W33X318	W30X148	W33X130	W14X257	W33X354
A_{37}	W44X290	W14X426	W40X199	W14X426	W33X354	W14X665	W14X370	W14X426	W33X291	W14X455
A_{38}	W36X256	W21X201	W21X83	W40X264	W33X201	W40X297	W36X150	W24X146	W36X150	W40X149
A_{39}	W24X335	W40X249	W40X183	W33X263	W33X263	W12X45	W40X174	W40X174	W36X194	W40X264
A_{40}	W27X235	W16X100	W40X392	W27X258	W33X318	W24X103	W21X132	W21X132	W30X124	W14X74
A_{41}	W36X160	W18X46	W24X103	W18X46	W12X65	W33X130	W24X55	W24X55	W14X120	W36X245
A_{42}	W14X426	W30X391	W14X550	W33X354	W40X277	W36X527	W33X354	W33X354	W14X730	W14X176
A_{43}	W24X408	W36X160	W36X439	W40X167	W27X84	W18X86	W33X263	W40X167	W12X190	W27X194
A_{44}	W18X106	W14X145	W12X96	W14X145	W36X359	W14X283	W14X145	W14X145	W14X193	W40X249
A_{45}	W18X311	W33X141	W24X408	W40X278	W40X372	W36X256	W14X120	W33X118	W14X370	W27X146
A_{46}	W40X235	W24X84	W27X539	W24X84	W27X178	W30X90	W24X84	W24X84	W40X199	W30X124
A_{47}	W36X245	W40X297	W14X342	W40X297	W27X178	W40X199	W44X290	W44X290	W30X90	W40X277
A_{48}	W18X106	W18X158	W27X161	W18X158	W36X182	W30X211	W18X143	W14X211	W24X117	W30X116
A_{49}	W40X211	W40X331	W27X448	W40X331	W30X261	W14X233	W21X182	W40X149	W12X120	W14X159
A_{50}	W27X217	W27X146	W36X194	W27X146	W33X263	W14X398	W27X146	W27X146	W18X311	W21X201
A_{51}	W30X261	W12X87	W14X426	W12X87	W36X256	W24X207	W27X84	W24X94	W10X45	W14X193
A_{52}	W10X77	W14X109	W12X170	W14X109	W10X112	W12X96	W30X124	W14X109	W40X278	W40X249
A_{53}	W40X235	W21X182	W24X335	W21X182	W14X211	W36X150	W27X146	W16X67	W24X68	W8X35
A_{54}	W44X230	W12X58	W14X176	W12X58	W21X73	W18X40	W16X31	W14X30	W24X76	W21X50
A_{55}	W36X160	W40X211	W21X147	W33X263	W21X111	W27X161	W33X118	W40X211	W30X124	W33X241
A_{56}	W40X331	W21X122	W12X279	W30X116	W44X262	W18X55	W14X109	W21X122	W30X124	W12X58

References

[1] Suman A, Suman P, Padhy S, Kumar N, Singh A. Healthcare revolution: Advances in AI-driven medical imaging and diagnosis. Responsible and Explainable Artificial Intelligence in Healthcare. Elsevier; 2025. p. 155–82. https://doi.org/10.1016/B978-0-443-24788-0.00007-8.

[2] Dasilas A, Rigani A. Machine learning techniques in bankruptcy prediction: a systematic literature review. Expert Syst Appl 2024;255:124761. https://doi.org/10.1016/j.eswa.2024.124761.

[3] Hosseini E, Al-Ghaili AM, Kadir DH, Gunasekaran SS, Ahmed AN, Jamil N, et al. Meta-heuristics and deep learning for energy applications: review and open research challenges (2018–2023). Energy Strategy Rev 2024;53:101409. https://doi.org/10.1016/j.esr.2024.101409.

[4] Inayathullah S, Buddala R. Review of machine learning applications in additive manufacturing. Results Eng 2025;25:103676. https://doi.org/10.1016/j.rineng.2024.103676.

[5] Momenitabar M, Dehdari Ebrahimi Z, Ghasemi P. Designing a sustainable bioethanol supply chain network: a combination of machine learning and meta-heuristic algorithms. Ind Crops Prod 2022;189:115848. https://doi.org/10.1016/j.indcrop.2022.115848.

[6] Farahzadi L, Kioumarsi M. Application of machine learning initiatives and intelligent perspectives for CO2 emissions reduction in construction. J Clean Prod 2023;384:135504. https://doi.org/10.1016/j.jclepro.2022.135504.

[7] Kaloop MR, Ahmad F, Samui P, Elbeltagi E, Hu J-W, Wefki H. Predicting energy consumption of residential buildings using metaheuristic-optimized artificial neural network technique in early design stage. Build Environ 2025;274:112749. https://doi.org/10.1016/j.buildenv.2025.112749.

[8] Takhmasib M, Lee HJ, Yi H. Machine-learned kinetic Façade: construction and artificial intelligence enabled predictive control for visual comfort. Autom Constr 2023;156:105093. https://doi.org/10.1016/j.autcon.2023.105093.

[9] Wang XQ, Chen P, Chow CL, Lau D. Artificial-intelligence-led revolution of construction materials: From molecules to Industry 4.0. Matter 2023;6:1831–59. https://doi.org/10.1016/j.matt.2023.04.016.

[10] Wu C, Ding W, Jin Q, Jiang J, Jiang R, Xiao Q, et al. Retrieval augmented generation-driven information retrieval and question answering in construction management. Adv Eng Inform 2025;65:103158. https://doi.org/10.1016/j.aei.2025.103158.

[11] Manmatharasan P, Bitsuamlak G, Grolinger K. AI-driven design optimization for sustainable buildings: a systematic review. Energy Build 2025;332:115440. https://doi.org/10.1016/j.enbuild.2025.115440.

[12] Oval R, Orr J, Shepherd P. Structural design and fabrication of concrete reinforcement with layout optimisation and robotic filament winding. Autom Constr 2025;170:105952. https://doi.org/10.1016/j.autcon.2024.105952.
[13] Zhang L, Ma J, Wang J, Sun Q, Yang H. Multi-objective optimization for flexible design of aerial building machine under various wind conditions. Autom Constr 2025;171:105956. https://doi.org/10.1016/j.autcon.2024.105956.
[14] Huang M, Wang C, Lin W, Xiao Z. Topology and element sizing optimization for performance-based design of wind sensitive tall buildings. Eng Struct 2025;328:119726. https://doi.org/10.1016/j.engstruct.2025.119726.
[15] Zhou Y, Zhang J, Long K, Saeed A, Tao T, Chen J. Topology optimization on a jacket structure for offshore wind turbines by altering structural design domain. Appl Ocean Res 2025;154:104421. https://doi.org/10.1016/j.apor.2025.104421.
[16] Hosseinzadeh Y, Jalili S, Khani R. Investigating the effects of flax fibers application on multi-objective optimization of laminated composite plates for simultaneous cost minimization and frequency gap maximization. J Build Eng 2020;32:101477. https://doi.org/10.1016/j.jobe.2020.101477.
[17] Jalili S, Khani R, Hosseinzadeh Y. On the performance of flax fibres in multi-objective design of laminated composite plates for buckling and cost. Structures 2021;33. https://doi.org/10.1016/j.istruc.2021.06.026.
[18] Knobloch R, Mlýnek J, Srb R. The classic differential evolution algorithm and its convergence properties. Appl Math 2017;62:197–208. https://doi.org/10.21136/AM.2017.0274-16.
[19] Simon D. Biogeography-based optimization. IEEE Trans Evolut Comput 2008;12:702–13. https://doi.org/10.1109/TEVC.2008.919004.
[20] Kennedy J., Eberhart R. Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE; n.d., p. 1942–8. https://doi.org/10.1109/ICNN.1995.488968.
[21] Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: A Gravitational Search Algorithm. Inf Sci 2009;179:2232–48. https://doi.org/10.1016/j.ins.2009.03.004.
[22] Jalili S, Husseinzadeh Kashan A. Optimum discrete design of steel tower structures using optics inspired optimization method. Struct Des Tall Spec Build 2018;27. https://doi.org/10.1002/tal.1466.
[23] Jalili S, Husseinzadeh Kashan A. An optics inspired optimization method for optimal design of truss structures. Struct Des Tall Spec Build 2019;28. https://doi.org/10.1002/tal.1598.
[24] Jalili S. Cultural Algorithms. Singapore: Springer Nature Singapore; 2022. https://doi.org/10.1007/978-981-19-4633-2.
[25] Reynolds RG. Cultural algorithms: Theory and applications. N ideas Optim 1999:367–78.
[26] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evolut Comput 1997;1:67–82. https://doi.org/10.1109/4235.585893.
[27] Kirsch U. A unified reanalysis approach for structural analysis, design, and optimization. Struct Multidiscip Optim 2003;25:67–85. https://doi.org/10.1007/s00158-002-0269-0.
[28] Kirsch U, Bogomolni M, Sheinman I. Efficient structural optimization using reanalysis and sensitivity reanalysis. Eng Comput 2007;23:229–39. https://doi.org/10.1007/s00366-007-0062-1.
[29] Hosseinzadeh Y, Jalili S. Structural sensitivity reanalysis formulations based on the polynomial-type extrapolation methods. Struct Multidiscip Optim 2020;61:1027–50. https://doi.org/10.1007/s00158-019-02401-9.
[30] Hosseinzadeh Y, Taghizadieh N, Jalili S. A new structural reanalysis approach based on the polynomial-type extrapolation methods. Struct Multidiscip Optim 2018;58:1033–49. https://doi.org/10.1007/s00158-018-1945-z.
[31] Hasançebi O. Adaptive evolution strategies in structural optimization: enhancing their computational performance with applications to large-scale structures. Comput Struct 2008;86:119–32. https://doi.org/10.1016/j.compstruc.2007.05.012.
[32] Cao H, Li H, Wang M, Huang B, Sun Y. A structural reanalysis assisted harmony search for the optimal design of structures. Comput Struct 2022;270:106844. https://doi.org/10.1016/j.compstruc.2022.106844.
[33] Jalili S. Structural vibration and sensitivity reanalyses via polynomial-type extrapolation. Comput Struct 2025;319:108000. https://doi.org/10.1016/j.compstruc.2025.108000.
[34] Degertekin SO, Hayalioglu MS. Sizing truss structures using teaching-learning-based optimization. Comput Struct 2013;119:177–88. https://doi.org/10.1016/j.compstruc.2012.12.011.
[35] Jalili S, Kashan AH, Hosseinzadeh Y. League championship algorithms for optimum design of pin-jointed structures. J Comput Civ Eng 2017;31. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000617.
[36] Degertekin SO. Improved harmony search algorithms for sizing optimization of truss structures. Comput Struct 2012;92–93:229–41. https://doi.org/10.1016/j.compstruc.2011.10.022.
[37] Salama A, Farag AA, El-Sisi A, Eraky A, Samir R. An enhanced meta-heuristic algorithm for optimizing the roof slope of tapered steel frames. Eng Struct 2023;291:116393. https://doi.org/10.1016/j.engstruct.2023.116393.
[38] Hasançebi O, Erdal F, Saka MP. Adaptive harmony search method for structural optimization. J Struct Eng 2010;136:419–31. https://doi.org/10.1061/(ASCE)ST.1943-541X.0000128.
[39] Jalili S, Hosseinzadeh Y, Taghizadieh N. A biogeography-based optimization for optimum discrete design of skeletal structures. Eng Optim 2016;48:1491–514. https://doi.org/10.1080/0305215X.2015.1115028.
[40] Hasançebi O, Çarbaş S, Saka MP. Improving the performance of simulated annealing in structural optimization. Struct Multidiscip Optim 2010;41:189–203. https://doi.org/10.1007/s00158-009-0418-9.
[41] Khatibinia M, Yazdani H. Accelerated multi-gravitational search algorithm for size optimization of truss structures. Swarm Evol Comput 2018;38:109–19. https://doi.org/10.1016/j.swevo.2017.07.001.
[42] Wang SY, Tai K, Wang MY. An enhanced genetic algorithm for structural topology optimization. Int J Numer Methods Eng 2006;65:18–44. https://doi.org/10.1002/nme.1435.
[43] Ficarella E, Lamberti L, Degertekin SO. Comparison of three novel hybrid metaheuristic algorithms for structural optimization problems. Comput Struct 2021;244:106395. https://doi.org/10.1016/j.compstruc.2020.106395.
[44] Djerouni S, Bekdaş G, Nigdeli SM. Optimization and performance assessment of Multi-Tuned Mass Dampers (MTMD) to mitigate seismic pounding of adjacent buildings via a novel hybrid algorithm. J Build Eng 2025;103:112168. https://doi.org/10.1016/j.jobe.2025.112168.
[45] Hosseinzadeh Y, Taghizadieh N, Jalili S. Hybridizing electromagnetism-like mechanism algorithm with migration strategy for layout and size optimization of truss structures with frequency constraints. Neural Comput Appl 2016;27:953–71. https://doi.org/10.1007/s00521-015-1912-1.
[46] Degertekin SO, Tutar H. Optimized seismic design of planar and spatial steel frames using the hybrid learning based jaya algorithm. Adv Eng Softw 2022;171:103172. https://doi.org/10.1016/j.advengsoft.2022.103172.
[47] Jalili S, Hosseinzadeh Y. Design optimization of truss structures with continuous and discrete variables by hybrid of biogeography-based optimization and differential evolution methods. Struct Des Tall Spec Build 2018;27. https://doi.org/10.1002/tal.1495.
[48] Jalili S, Hosseinzadeh Y. Combining migration and differential evolution strategies for optimum design of truss structures with dynamic constraints. Iran J Sci Technol Trans Civ Eng 2019;43:289–312. https://doi.org/10.1007/s40996-018-0165-5.
[49] Burke EK, Gendreau M, Hyde M, Kendall G, Ochoa G, Özcan E, et al. Hyper-heuristics: a survey of the state of the art. J Oper Res Soc 2013;64:1695–724. https://doi.org/10.1057/jors.2013.71.
[50] Basgalupp MP, Barros RC, da Silva TS, de Carvalho ACPLF. Software effort prediction: a hyper-heuristic decision-tree based approach. Proceedings of the 28th Annual ACM Symposium on Applied Computing. New York, NY, USA: ACM; 2013. p. 1109–16. https://doi.org/10.1145/2480362.2480571.
[51] Burke E.K., Hyde M., Kendall G., Ochoa G., Özcan E., Woodward J.R. A Classification of Hyper-heuristic Approaches, 2010, p. 449–468. https://doi.org/10.1007/978-1-4419-1665-5_15.
[52] Crowston W.B.S. Probabilistic and parametric learning combinations of local job shop scheduling rules. Defense Technical Information Center; 1963.
[53] Thompson G.L. Industrial scheduling. Prentice-Hall; 1963.
[54] Drake JH, Kheiri A, Özcan E, Burke EK. Recent advances in selection hyper-heuristics. Eur J Oper Res 2020;285:405–28. https://doi.org/10.1016/j.ejor.2019.07.073.
[55] Dokeroglu T, Kucukyilmaz T, Talbi E-G. Hyper-heuristics: a survey and taxonomy. Comput Ind Eng 2024;187:109815. https://doi.org/10.1016/j.cie.2023.109815.
[56] Burke E., Kendall G., Newall J., Hart E., Ross P., Schulenburg S. Hyper-Heuristics: An Emerging Direction in Modern Search Technology. Handbook of Metaheuristics, Boston: Kluwer Academic Publishers; n.d., p. 457–74. https://doi.org/10.1007/0-306-48056-5_16.
[57] Maashi M, Özcan E, Kendall G. A multi-objective hyper-heuristic based on choice function. Expert Syst Appl 2014;41:4475–93. https://doi.org/10.1016/j.eswa.2013.12.050.
[58] Talatahari S, Jalili S, Azizi M. Optimum design of steel building structures using migration-based vibrating particles system. Structures 2021;33:1394–413. https://doi.org/10.1016/j.istruc.2021.05.028.
[59] American Institute of Steel Construction. Manual of steel construction. 1989.
[60] Zakian P. Meta-heuristic design optimization of steel moment resisting frames subjected to natural frequency constraints. Adv Eng Softw 2019;135:102686. https://doi.org/10.1016/j.advengsoft.2019.102686.
[61] Jalili S, Talatahari S. Optimum design of truss structures under frequency constraints using hybrid CSS-MBLS algorithm. KSCE J Civ Eng 2018;22:1840–53. https://doi.org/10.1007/s12205-017-1407-y.
[62] Chakhlevitch K., Cowling P. Hyperheuristics: Recent Developments. Adaptive and Multilevel Metaheuristics, Berlin, Heidelberg: Springer Berlin Heidelberg; n.d., p. 3–29. https://doi.org/10.1007/978-3-540-79438-7_1.
[63] Reynolds R.G. An adaptive computer model of the evolution of agriculture for hunter-gatherers in the valley of Oaxaca, Mexico. University of Michigan, 1979.
[64] Chung C-J, Reynolds RG. A testbed for solving optimization problems using cultural algorithms. Evolut Program 1996:225–36.
[65] Reynolds R.G. The Cultural Algorithm: Culture on the Edge of Chaos, 2018, p. 1–11. https://doi.org/10.1007/978-3-319-74171-0_1.
[66] Maheri A, Jalili S, Hosseinzadeh Y, Khani R, Miryahyavi M. A comprehensive survey on cultural algorithms. Swarm Evol Comput 2021;62:100846. https://doi.org/10.1016/j.swevo.2021.100846.
[67] Reynolds R., Al-Shehri H. The use of cultural algorithms with evolutionary programming to control the data mining of large-scale spatio-temporal databases. 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, IEEE; n.d., p. 4098–103. https://doi.org/10.1109/ICSMC.1997.637338.
[68] Jalili S., Hosseinzadeh Y., Rabczuk T. Simultaneous Size and Shape Optimization of Dome-Shaped Structures Using Improved Cultural Algorithm, 2019, p. 93–120. https://doi.org/10.1007/978-981-13-6569-0_5.
[69] Jalili S, Khani R, Maheri A, Hosseinzadeh Y. Performance assessment of meta-heuristics for composite layup optimisation. Neural Comput Appl 2022;34. https://doi.org/10.1007/s00521-021-06519-0.

[70] Storn R. System design by constraint adaptation and differential evolution. IEEE Trans Evolut Comput 1999;3:22–34. https://doi.org/10.1109/4235.752918.
[71] Storn R, Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 1997;11:341–59. https://doi.org/10.1023/A:1008202821328.
[72] Qin AK, Huang VL, Suganthan PN. Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evolut Comput 2009;13:398–417. https://doi.org/10.1109/TEVC.2008.927706.
[73] Tan Z, Tang Y, Li K, Huang H, Luo S. Differential evolution with hybrid parameters and mutation strategies based on reinforcement learning. Swarm Evol Comput 2022;75:101194. https://doi.org/10.1016/j.swevo.2022.101194.
[74] Cowling P., Kendall G., Soubeiga E. A Hyperheuristic Approach to Scheduling a Sales Summit, 2001, p. 176–190. https://doi.org/10.1007/3-540-44629-X_11.
[75] Maashi M, Kendall G, Özcan E. Choice function based hyper-heuristics for multi-objective optimization. Appl Soft Comput 2015;28:312–26. https://doi.org/10.1016/j.asoc.2014.12.012.
[76] Drake JH, Ozcan E, Burke EK. A modified choice function hyper-heuristic controlling unary and binary operators. IEEE Congress on Evolutionary Computation (CEC), 2015. IEEE; 2015. p. 3389–96. https://doi.org/10.1109/CEC.2015.7257315.
[77] Din F, Alsewari ARA, Zamli KZ. A Parameter Free Choice Function Based Hyper-Heuristic Strategy for Pairwise Test Generation. 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C. IEEE; 2017. p. 85–91. https://doi.org/10.1109/QRS-C.2017.22.
[78] Drake J.H., Özcan E., Burke E.K. Modified Choice Function Heuristic Selection for the Multidimensional Knapsack Problem, 2015, p. 225–234. https://doi.org/10.1007/978-3-319-12286-1_23.
[79] Kendall G., Cowling P., Soubeiga E. 2002. Choice function and random hyperheuristics. Proceedings of the 4th Asia-pacific conference on simulated evolution and learning, 2002, p. 667–671.
[80] Castro OR, Fritsche GM, Pozo A. Evaluating selection methods on hyper-heuristic multi-objective particle swarm optimization. J Heuristics 2018;24:581–616. https://doi.org/10.1007/s10732-018-9369-x.
[81] Drake J.H., Özcan E., Burke E.K. Modified Choice Function Heuristic Selection for the Multidimensional Knapsack Problem, 2015, p. 225–234. https://doi.org/10.1007/978-3-319-12286-1_23.
[82] Drake JH, Ozcan E, Burke EK. A modified choice function hyper-heuristic controlling unary and binary operators. 2015 IEEE Congress on Evolutionary Computation (CEC). IEEE; 2015. p. 3389–96. https://doi.org/10.1109/CEC.2015.7257315.
[83] Drake J.H., Özcan E., Burke E.K. An Improved Choice Function Heuristic Selection for Cross Domain Heuristic Search, 2012, p. 307–16. https://doi.org/10.1007/978-3-642-32964-7_31.
[84] Bouneffouf D, Féraud R. Multi-armed bandit problem with known trend. Neurocomputing 2016;205:16–21. https://doi.org/10.1016/j.neucom.2016.02.052.
[85] Shahrampour S, Rakhlin A, Jadbabaie A. Multi-armed bandits in multi-agent networks. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2017. p. 2786–90. https://doi.org/10.1109/ICASSP.2017.7952664.
[86] Mui J., Lin F., Dewan M.A.A. Multi-armed Bandit Algorithms for Adaptive Learning: A Survey, 2021, p. 273–278. https://doi.org/10.1007/978-3-030-78270-2_49.
[87] Goncalves RA, de Almeida CP, Venske SMGS, Delgado MR de B da S, Pozo ATR. A New Hyper-Heuristic Based on a Restless Multi-armed Bandit for Multi-objective Optimization. 2017 Brazilian Conference on Intelligent Systems (BRACIS). IEEE; 2017. p. 390–5. https://doi.org/10.1109/BRACIS.2017.67.
[88] Goncalves R, Almeida C, Luders R, Delgado M. A New Hyper-Heuristic Based on a Contextual Multi-Armed Bandit for Many-Objective Optimization. 2018 IEEE Congress on Evolutionary Computation (CEC). IEEE; 2018. p. 1–8. https://doi.org/10.1109/CEC.2018.8477930.
[89] Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem. Mach Learn 2002;47:235–56. https://doi.org/10.1023/A:1013689704352.
[90] Li K, Fialho A, Kwong S, Zhang Q. Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evolut Comput 2014;18:114–30. https://doi.org/10.1109/TEVC.2013.2239648.
[91] Kaveh A, Khodadadi N, Azar BF, Talatahari S. Optimal design of large-scale frames with an advanced charged system search algorithm using box-shaped sections. Eng Comput 2021;37:2521–41. https://doi.org/10.1007/s00366-020-00955-7.
[92] Goldberg D.E., Deb K. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms, 1991, p. 69–93. https://doi.org/10.1016/B978-0-08-050684-5.50008-2.
[93] Talatahari S, Khalili E, Alavizadeh SM. Accelerated particle swarm for optimum design of frame structures. Math Probl Eng 2013;2013:1–6. https://doi.org/10.1155/2013/649857.
[94] AISC. AISC Historic Shapes Database v15.0H n.d. ⟨https://www.aisc.org/publications/steel-construction-manual-resources/15th-ed-steel-construction-manual/shapes-database-v15.0/⟩ (accessed April 3, 2025).