Contents lists available at ScienceDirect

Expert Systems With Applications

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

A hyper-heuristic algorithm via proximal policy optimization for multi-objective truss problems

Shihong Yin, Zhengrong Xiang *

School of Automation, Nanjing University of Science and Technology, Nanjing 210094, Jiangsu, China

ARTICLE INFO

Keywords:
Multi-objective optimization
Truss optimization
Proximal policy optimization
Hyper-heuristic algorithm
Dynamic crowding distance

ABSTRACT

This paper proposes a hyper-heuristic evolutionary algorithm via proximal policy optimization, named HHEA-PPO, for solving multi-objective truss optimization problems. HHEA-PPO has a two-layer structure: a high-level strategy and low-level heuristics. The high-level strategy consists of proximal policy optimization, while the low-level heuristics consist of ten predefined heuristic operators. During the iteration process, the high-level strategy selects the most promising low-level heuristic according to the state of the individuals and the population. To maintain the convergence and distribution of the external Pareto archive, a dynamic crowding distance mechanism is employed. HHEA-PPO is applied to eight multi-objective truss optimization problems and compared with thirteen state-of-the-art optimization algorithms in terms of success rate, average computation duration, and average fitness evaluations to evaluate its performance. The results show that HHEA-PPO has higher search efficiency and greater stability, demonstrating its ability to solve large-scale engineering design problems.

1. Introduction

Truss optimization is an important topic in the fields of aerospace, mechanical manufacturing, and civil engineering. The purpose of truss optimization is to improve structural performance, reduce mass, enhance load-bearing capacity, and meet other engineering requirements by adjusting the design variables of the truss structure. Truss optimization problems are mainly classified into topology optimization (Fairclough & Gilbert, 2020; Poulsen et al., 2020; Tejani, Savsani, et al., 2019), shape optimization (Azizi et al., 2022; Kaveh & Khayatazad, 2013; Nguyen-Van et al., 2021), size optimization (Lieu et al., 2018; Pham & Tran, 2022; Wu et al., 2023), and the joint optimization of these three variables (Anosri et al., 2022; Panagant & Bureerat, 2018). Topology variables determine the basic connectivity of the truss, shape variables adjust the geometric shape of the truss members, and size variables dictate the specific physical sizes of the members (Panagant et al., 2021). Size optimization helps to adjust the actual construction of the truss structure to better meet the design requirements. However, current research on size optimization is relatively limited, so this paper primarily focuses on the optimization of size variables.

As research and applications advance, truss size optimization has become increasingly complex, involving multiple objectives, constraints, and discrete variables (Panagant et al., 2019). Traditional truss optimization methods, such as linear programming (Lamberti & Pappalettere, 2004), are no longer sufficient to meet practical requirements. To overcome these challenges, researchers have proposed various improved metaheuristic algorithms. These algorithms can effectively help engineers find optimal solutions that satisfy constraints, thereby minimizing material cost and truss mass while improving safety and stability.

In previous research, a variety of truss optimization algorithms have been proposed, which can be broadly classified into single-objective algorithms and multi-objective algorithms. Some successful single-objective truss optimization algorithms include: artificial bee colony (Jawad et al., 2021; Sonmez, 2011), particle swarm optimization (PSO) (Gomes, 2011), cuckoo search algorithm (Gandomi et al., 2013), hybrid intelligent genetic algorithm (GA) (Liu & Xia, 2022), improved differential evolution (DE) (Ho-Huu et al., 2016; Renkavieski & Parpinelli, 2021), improved symbiotic organism search (Kumar et al., 2019; Tejani et al., 2016), improved dolphin echolocation algorithm (Gholizadeh & Poorhoseini, 2016), improved harmony search algorithm (Degertekin, 2012), improved coyote optimization algorithm (Pierezan et al., 2021), improved slime mould algorithm (Kaveh et al., 2022), improved whale optimization algorithm (Jiang et al., 2021), and generalized normal

* Corresponding author.
E-mail addresses: yinshihong@njust.edu.cn (S. Yin), xiangzr@njust.edu.cn (Z. Xiang).

https://doi.org/10.1016/j.eswa.2024.124929
Received 4 January 2024; Received in revised form 21 June 2024; Accepted 26 July 2024
Available online 30 July 2024

distribution optimization (Khodadadi & Mirjalili, 2022). In the above studies, the structural mass is typically considered as the optimization objective.

Some additional objective functions, such as compliance (Kumar et al., 2022; Panagant et al., 2021), reliability (Ho-Huu, Duong-Gia, et al., 2018; Techasen et al., 2019), or displacement (Ho-Huu, Hartjes, et al., 2018; Kumar, Tejani, Pholdee, & Bureerat, 2021), have been considered by researchers in multi-objective truss problems in order to obtain a more realistic design solution. Unlike the single-objective algorithm, the challenge of the multi-objective algorithm lies in balancing various conflicting objectives. To solve the multi-objective truss problem, various multi-objective optimization algorithms have been introduced by researchers, some of which include: NSGA-II (Ho-Huu, Duong-Gia, et al., 2018), MOEA/D (Ho-Huu, Hartjes, et al., 2018), and multi-objective symbiotic organism search (Tejani, Pholdee, et al., 2019).

Metaheuristic algorithms have shown excellent performance in multi-objective truss optimization problems, but they are often limited by specific problem instances and the experience of human experts. To address this issue, hyper-heuristic evolutionary algorithms (HHEAs) have received considerable attention (Escalante et al., 2021). A review of the existing literature shows that almost no HHEA has been developed specifically for truss problems, especially multi-objective optimization problems (Öztürk & Kahraman, 2023). Therefore, developing an HHEA for solving multi-objective truss problems will fill a research gap in this area.

Due to the “no free lunch” theorem (Wolpert & Macready, 1997), it is difficult for researchers to determine in advance which heuristic algorithm is best suited to solve a particular problem. HHEA can alleviate this problem by selecting, generating, and arranging the best heuristic algorithms (Dokeroglu et al., 2024). HHEA includes high-level strategies (HLS) and low-level heuristic74510s (LLH). Currently, many HHEAs select LLHs based solely on their performance, such as the multi-armed bandit-based HHEA (Almeida et al., 2020) and the Thompson sampling-based HHEA (Sun & Li, 2020). These algorithms learn only the values of the LLHs and ignore the state information of the search agents. Other HHEAs focus on using tabular reinforcement learning methods, such as SARSA-based HHEAs (Cao et al., 2021) and Q-learning-based HHEAs (Zhang, Wu, et al., 2023; Zhao et al., 2023). However, these algorithms require discretization of the state space when dealing with continuous state spaces, leading to unreliable results (Yi et al., 2023). There is limited research on using advanced reinforcement learning methods to handle continuous state spaces in the design of HHEAs. Tian et al. (Tian et al., 2023) designed an adaptive multi-objective algorithm using deep Q-network (DQN), which achieved higher convergence accuracy. However, DQN has a long training time and does not consider the constraint handling mechanism. Yi et al. (Yi et al., 2023) designed a general search framework using both DQN and proximal policy optimization (PPO), and the results showed that PPO-based adaptive algorithms outperformed DQN-based adaptive algorithms. Although this study demonstrated the potential of PPO, it did not consider multi-objective and constraint handling mechanisms and used a time-consuming offline training method. Due to the presence of multiple conflicting objective functions and stress constraints in truss problems, existing HHEAs are not suitable for solving this problem.

Based on the above discussion, a PPO-based HHEA (HHEA-PPO) is proposed to address constrained multi-objective truss problems. The main innovations of this paper include the PPO-based HLS, the integration of ten different LLHs, and an external Pareto archive maintenance method using dynamic crowding distance (DCD). These innovations aim to improve the adaptability, convergence performance, and ability to handle large-scale problems of the algorithm.

The main contributions of this paper can be summarized as follows:

- A PPO-based online learning mechanism is designed to combine the training process of neural networks with the population evolution.
- Ten low-level heuristics are integrated to provide diverse choices for high-level strategies.
- A new state space and reward mechanism are designed for constrained multi-objective optimization problems.
- An external Pareto archive using dynamic crowding distances is integrated to maintain the convergence and diversity of the Pareto set.
- The performance of HHEA-PPO is evaluated on eight multi-objective truss problems and compared with thirteen state-of-the-art algorithms in terms of success rate, average computation duration, and average number of fitness evaluations.

The rest of this paper is organized as follows. The background and related work on hyper-heuristic algorithms and truss optimization problems are presented in Section 2. The design and implementation of HHEA-PPO is described in detail in Section 3. Simulation results, comparisons and discussions are presented in Section 4. Finally, the conclusion is given in Section 5.

2. Background and related works

2.1. Hyper-heuristic algorithms for multi-objective optimization

Recent studies have explored various hyper-heuristic algorithms to improve the performance of multi-objective optimization problems. The main contributions and unique aspects of these studies are summarized in Table 1.

Table 1
Recent studies of multi-objective hyper-heuristic algorithms.

Research	Algorithm	Main Contributions	Unique Aspects
(Almeida et al., 2020)	HH-LinUCB	Used multi-armed bandit models as a hyper-heuristic framework to dynamically select the optimal operators	Proposes an upper confidence bound algorithm and a dynamic Thompson sampling algorithm
(Venske et al., 2022)	HHMOEA/DD	Proposed a selection hyper-heuristic algorithm for multi-objective and many-objective quadratic assignment problems	Integrates multiple selection strategies, improving the adaptability of the algorithm to different problems
(Tian et al., 2023)	MOEA/D-DQN	Proposed an adaptive operator selection method using deep reinforcement learning	Uses deep neural networks to learn the Q-values of operators, dynamically selecting the optimal operator
(Ahmed & Babu, 2024)	HHOPSO	Proposed a hyper-heuristic multi-objective online optimization algorithm for cyber security problems in big data	Dynamically adjusts algorithm parameters through online learning mechanism, improving problem-solving performance for cyber security problems
(Li et al., 2024)	COHH	Proposed a compass-based hyper-heuristic method for multi-objective optimization problems	Uses a directional guidance mechanism to dynamically select operators, improving search efficiency and solution quality
(Yin & Xiang, 2024)	AdaW-DDQN	Proposed an adaptive operator selection paradigm with dueling double deep Q-network (DDQN)	Decomposes the Q-network into state value network and action advantage network, improving learning efficiency and precision of strategy selection
(Xu et al., 2024)	NSGA-II-DQN	Proposed a hyper-heuristic method via deep Q-network for multi-objective optimization of unmanned surface vehicle scheduling problems	Deep Q-network learns the utility values of different operators and dynamically selects the optimal operator

Compared with the current research, this paper proposes the HHEA-PPO algorithm, which is the first algorithm to use PPO for multi-objective truss optimization. It integrates ten different LLHs and a DCD mechanism to maintain convergence and diversity. Unlike previous studies that rely on simpler bandit models or Q-learning, HHEA-PPO uses PPO, a more advanced policy optimization method, which ensures better adaptability and stability. The use of DCD further enhances its efficiency in maintaining Pareto optimality.

2.2. Multi-objective truss optimization problems

Truss optimization remains a critical area in structural engineering, with recent studies focusing on different optimization strategies. Table 2 provides an overview of significant contributions in this area.

In this paper, HHEA-PPO is introduced for multi-objective truss optimization, which provides an online learning mechanism and a diverse set of LLHs. The proposed algorithm uses PPO for policy

Table 2
Recent studies of multi-objective truss optimization problems.

Research	Algorithm	Application	Main Contributions	Unique Aspects
(Carvalho et al., 2021)	MM-IPDE	Multi-objective truss design optimization	Proposed a multi-objective truss design optimization method using differential evolution algorithms, demonstrating superior performance in various optimization problems	Uses differential evolution algorithms, improving the performance in multi-objective optimization
(Kumar, Tejani, Pholdee, & Bureerat, 2021)	MOMHTS	Multi-objective truss optimization	Proposed a modified heat transfer search algorithm for multi-objective truss optimization, demonstrating high efficiency and robustness in dealing with multi-objective problems	Combines heat transfer optimization strategies, improving the diversity and quality of optimization results
(Kumar, Tejani, Pholdee, Bureerat, et al., 2021)	HHTS-PVS	Multi-objective structural optimization	Proposed a hybrid heat transfer search and passing vehicle search algorithm for multi-objective structural optimization, demonstrating high efficiency and solution quality	Combines heat transfer and passing vehicle search strategies, improving the efficiency and quality of multi-objective optimization
(Lemonge et al., 2021)	GDE3	Truss structural optimization	Proposed a multi-objective truss structural optimization method considering natural frequencies of vibration and global stability, demonstrating high efficiency and solution quality	Combines natural frequencies of vibration and global stability, improving the stability and quality of optimization results
(Panagant et al., 2021)	SHAMODE-WO	Constrained truss optimization problems	Compared recent multi-objective metaheuristics for solving constrained truss optimization problems, providing a comprehensive performance evaluation and comparison	Provides a comparative study of various recent algorithms, demonstrating the advantages and disadvantages of each under different constraint conditions
(Tejani et al., 2021)	MOHTS	Truss optimization	Proposed a multi-objective heat transfer search algorithm for truss optimization, demonstrating high efficiency and solution quality	Combines heat transfer optimization strategies, improving the efficiency and quality of multi-objective optimization
(Anosri et al., 2022)	iSHAMODE-WO	Simultaneous topology, shape, and size optimization of trusses	Proposed a success history-based adaptive multi-objective differential evolution algorithm for simultaneous topology, shape, and size optimization of trusses, improving reliability	Combines adaptive strategy with differential evolution algorithm, improving the stability and adaptability of the algorithm
(Eid et al., 2022)	MOSWCA	Multi-objective truss optimization	Proposed a multi-objective spiral water cycle algorithm, incorporating hyperbolic spiral motion to guide the search process, demonstrating superior optimization performance	Incorporates hyperbolic spiral motion into the water cycle algorithm, enhancing the utilization ability and convergence performance
(Kumar et al., 2022)	MOTEO	Truss design optimization	Proposed a physics-based multi-objective thermal exchange optimization algorithm and applied it to truss design optimization, demonstrating high efficiency and solution quality	Introduces a thermal exchange physical model, enhancing the effectiveness and accuracy of multi-objective optimization
(David et al., 2023)	EvoDN2	Truss topology optimization of additively manufactured components	Proposed a multi-objective evolutionary algorithm for truss topology optimization of additively manufactured components, optimizing structural stiffness and thermal conductivity to enhance design quality	Combines additive manufacturing technology with efficient topology optimization
(Kumar et al., 2023)	MOMVO2arc	Multi-objective truss design optimization	Proposed a two-archive multi-objective multi-verse optimizer for truss design, demonstrating high efficiency and robustness	Combines two-archive and multi-verse strategies, enhancing the adaptability of the algorithm
(Yin et al., 2023)	IBMSMA	Multi-objective truss optimization	Proposed an indicator-based multi-swarm slime mould algorithm for solving multi-objective truss optimization problems, demonstrating high efficiency and stability in dealing with complex optimization problems	Combines multi-swarm strategies with indicator methods, improving the quality and diversity of optimization results
(Zhong et al., 2023)	MO-SHADE-MRFO	Multi-objective structural optimization	Proposed a multi-objective SHADE with manta ray foraging optimizer for structural design problems, demonstrating high efficiency and robustness	Combines SHADE and manta ray foraging strategies, enhancing the adaptability of the algorithm
(Carvalho et al., 2024)	CMOPSO	Truss structural optimization	Proposed a multi-objective truss structural optimization method considering natural frequencies of vibration and automatic member grouping, demonstrating high efficiency and robustness	Combines natural frequencies of vibration with automatic member grouping strategies, enhancing the stability and adaptability of optimization results
(Carvalho et al., 2024)	MOEAs	Truss structural optimization	Proposed a multi-objective truss structural optimization method combining evolutionary algorithms with automatic member grouping, demonstrating high efficiency and robustness	Combines evolutionary algorithms with automatic member grouping strategies, improving the quality and adaptability of optimization results
(Kupwiwat et al., 2024)	MADDPG	Truss structural optimization	Proposed a multi-objective optimization method for truss structures using multi-agent reinforcement learning and graph representation, demonstrating high performance and solution quality	Combines multi-agent reinforcement learning strategies, enhancing optimization performance
(Vo et al., 2024)	MOGWOCS	Spatial truss design optimization	Proposed a multi-objective grey wolf-cuckoo search algorithm for spatial truss design optimization, demonstrating high efficiency and solution quality	Combines grey wolf and cuckoo search strategies, enhancing the robustness of the algorithm

optimization and incorporates DCD to maintain the diversity of Pareto frontiers, which provides a significant improvement in terms of stability and convergence speed compared to existing methods. By integrating the PPO-based HLS and a diverse set of LLHs, HHEA-PPO stands out in both hyper-heuristic algorithm research and truss optimization. The DCD mechanism further ensures the efficiency and robustness of the algorithm in solving large-scale complex multi-objective optimization problems.

2.3. Definition of multi-objective truss optimization problem

Without loss of generality, a constrained multi-objective optimization problem can be represented as

$$\begin{aligned} &\min F(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})], \\ &s.t. g_i(\boldsymbol{x}) \leqslant 0, i = 1, 2, \ldots, p, \\ &h_j(\boldsymbol{x}) = 0, j = 1, 2, \ldots, q, \\ &\boldsymbol{x} \in [\boldsymbol{lb}, \boldsymbol{ub}] \subseteq \mathbb{R}^n, \end{aligned} \tag{1}$$

where $F(\boldsymbol{x})$ is a vector of objectives, $g_i(\boldsymbol{x})$ and $h_j(\boldsymbol{x})$ are constraints, $\boldsymbol{x}$ is a vector of variables, $\boldsymbol{lb}$ and $\boldsymbol{ub}$ are the lower and upper limits of the variables.

The relevant definitions of multi-objective optimization are as follows:

(1) Pareto dominance: In the objective space, if solution $\boldsymbol{x}_1$ is superior to or equal to solution $\boldsymbol{x}_2$ in all objectives and is superior to $\boldsymbol{x}_2$ in at least one objective, then $\boldsymbol{x}_1$ dominates $\boldsymbol{x}_2$.

(2) Non-dominated solution: A solution $\boldsymbol{x}$ is called a non-dominated solution if there is no solution $\boldsymbol{y} \in \mathbb{R}^n$ that dominates $\boldsymbol{x}$.

(3) Pareto set (PS): PS refers to the set of all non-dominated solutions in a multi-objective optimization problem. These solutions cannot be surpassed by any other solution in all objectives.

(4) Pareto front (PF): PF refers to the mapping of the PS in the objective space.

The multi-objective truss optimization problem is a major challenge in the field of structural engineering. Trusses are structural systems composed of members and nodes that are commonly used to support and distribute loads. In structural engineering design, optimization of truss variables such as topology, shape, material, and size, is crucial to meet specific requirements in terms of mass, compliance, reliability, and cost. Truss optimization decision variables are divided into topology, shape, and size variables. Topology variables determine the overall connectivity and layout of the structure. Shape variables correspond to joint locations that play a significant role in determining the overall geometry and configuration of the truss. Size variables are used to define the cross-sectional area of the members, which directly affects the mechanical properties of the structure such as strength and stiffness. A comprehensive truss design starts with topology optimization, then shape optimization, and finally size optimization (Panagant et al., 2019). Many researches related to topology and shape optimization are well established (Panagant et al., 2021). In this work, truss size optimization is investigated with the goal of finding the appropriate cross-sectional area of the members to reduce the structural mass and compliance. The mathematical model of multi-objective truss optimization can be described as

$$\begin{aligned} &\min F(\boldsymbol{A}) = [M, C], \\ &M = \sum_{i=1}^{n} \rho_i \cdot A_i \cdot L_i, \\ &C = \boldsymbol{D}^{\mathrm{T}} \boldsymbol{F}, \\ &\boldsymbol{F} = \sum_{i=1}^{n} \frac{E_i \cdot A_i}{L_i} \cdot K_i, \\ &\text{s.t. } g_i(\boldsymbol{A}) = |\sigma_{\max}| - \sigma_{\text{allow}} \leqslant 0, \\ &\text{where } \boldsymbol{A} = [A_1, A_2, \ldots, A_n], \\ &A_i^{\min} \leqslant A_i \leqslant A_i^{\max}, i = 1, 2 \ldots, \boldsymbol{n}, \end{aligned} \tag{2}$$

where M is the total mass of the truss, C is the compliance of the truss, n is the number of truss members, ρ_i is the material density of the i-th truss member, A_i is the cross-sectional area of the i-th truss member, L_i is the length of the i-th truss member, $\boldsymbol{D}$ denotes the nodal displacement matrix, $\boldsymbol{F}$ denotes the external load matrix, E_i is the elastic modulus of the i-th truss member, K_i is the stiffness matrix related to the i-th truss member, $\sigma_{\max}$ is the maximum stress, and σ_{allow} is the allowable maximum stress.

The structural mass of the truss is determined by the cross-sectional area and material density of the members. The stiffness of the structure is calculated using a linear finite element approach. As a result, the compliance value, which is the product of the nodal displacement and external force vectors, is a measure of the global stiffness matrix (Eid et al., 2022). During the optimization process, the constraint handling method is utilized to transform the constrained optimization problem into the corresponding unconstrained optimization problem. The fitness value after processing the constraint penalty can be defined as

$$F(\boldsymbol{A}) = \begin{cases} [M, C], & \sum_{i=1}^{m} \max\{g_i(\boldsymbol{A}), 0\} = 0, \\ [\max(M), \max(C)] + \sum_{i=1}^{m} \max\{g_i(\boldsymbol{A}), 0\}, & \text{otherwise}, \end{cases} \tag{3}$$

where m denotes the number of constraints.

3. Proposed hyper-heuristic algorithm

3.1. Design of the guidance mechanism

In the proposed HHEA-PPO algorithm, the guidance mechanism is realized by the dynamic selection of high-level strategy (HLS) and low-level heuristics (LLHs). In the following, the design of the guidance mechanism is explained in detail.

3.1.1. Design of high-level strategy

The HLS is realized by a PPO-based stochastic strategy, which includes the following aspects: state space construction, action space design, and reward mechanism design.

(1) The state space consists of three parts: the position difference between the current individual and the optimal individual, the population diversity in the decision space, and the population diversity in the objective space. By taking this information into account, the HLS can more accurately determine whether an individual should be explored or exploited.

(2) The action space consists of ten predefined LLHs. These LLHs have shown excellent performance in various optimization problems. By dynamically selecting these LLHs, the HLS can expand the search space

and improve the search efficiency of the algorithm.

(3) The reward mechanism is used to evaluate the effectiveness of the actions (selected LLHs) performed in each state. Specifically, if the LLH improves the fitness of the individual, it receives a positive reward; if it does not improve, it receives no reward or a negative reward. This design makes the algorithm focus more on whether there is an improvement rather than the magnitude of the improvement, thereby increasing the stability of strategy learning.

3.1.2. *Selection of low-level heuristics*

The HLS dynamically selects the most appropriate LLH through the PPO algorithm for optimization at different stages of evolution. The specific process is as follows:

Step 1: Initialize the policy and value networks. Using the state information of the population, the policy network generates the probability distribution of actions, and the value network evaluates the value of the current state.

Step 2: Action selection and execution. According to the probability distribution generated by the policy network, a LLH is selected by random sampling and applied to the current population to generate offspring.

Step 3: Reward evaluation and policy update. The reward value of the selected LLH is evaluated by the fitness changes of the offspring, and the policy network and value network are updated by the PPO algorithm.

With the design of the above guidance mechanisms, HHEA-PPO is able to effectively select the most suitable LLH dynamically in various optimization problems, which ensures efficient global search capability and improves convergence accuracy and stability. The effective combination of these mechanisms enables HHEA-PPO to demonstrate strong performance in solving multi-objective truss optimization problems.

3.2. *PPO-based high-level strategy*

In this section, a PPO-based HLS is proposed to dynamically select appropriate LLHs. The HHEA-PPO is mainly divided into two parts: HLS and LLH. The PPO algorithm is used to design the HLS, and the LLH is adaptively selected at different evolutionary stages through feedback signals. The framework of HHEA-PPO is shown in Fig. 1.

3.2.1. *Proximal policy optimization*

Selection-based hyper-heuristic algorithms adapt to different problem instances and optimization objectives by dynamically selecting LLHs. In this paper, an online learning mechanism is designed to use the PPO algorithm for HLS training. The online learning mechanism combines the neural network training process with the population evolution process, so that the HLS can guide the algorithm to select suitable LLHs in real time during the iteration process. This online learning mechanism improves the adaptability and convergence performance of the algorithm, and avoids the high time cost and inefficiency associated with traditional offline training methods. The PPO (Schulman et al., 2017) is considered an advanced policy-based deep reinforcement learning algorithm that can be used to train an agent (i.e., HLS) to select LLHs for superior performance. Compared to trust region policy optimization (TRPO) (Schulman et al., 2015), PPO is simpler, more versatile, and has better sampling efficiency. The core idea of PPO is to optimize the policy iteratively, ensuring that changes in the policy space correspond to performance changes within a controllable range to ensure training stability. In the design of adaptive algorithms, PPO outperforms DQN (Yi et al., 2023). The objective function of PPO is

$$\max L(\theta) = \widehat{\mathrm{E}}_t[\min(r_t(\theta)\widehat{A}_t, \mathrm{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\widehat{A}_t)], \tag{4}$$

where $L(\theta)$ denotes the objective function of the policy, θ denotes the parameters of the policy, $r_t(\theta) = \pi_\theta(a|s)/\pi_{\theta_{old}}(a|s)$ is the probability ratio between the current policy and the old one, $\widehat{A}_t$ is the estimated value of the advantage function at timestep t, $clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon) \triangleq \max(\min(r_t(\theta), 1+\varepsilon), 1-\varepsilon)$ denotes the motivation to trim actions outside the range of $1-\varepsilon$ and $1+\varepsilon$, and ε is the extent of the clipping (Schulman et al., 2017). If $\widehat{A}_t > 0$, it indicates that the value of this action is higher than the average level, maximizing $L(\theta)$ will increase $r_t(\theta)$, but it will not exceed $1+\varepsilon$. Conversely, if $\widehat{A}_t < 0$, maximizing $L(\theta)$

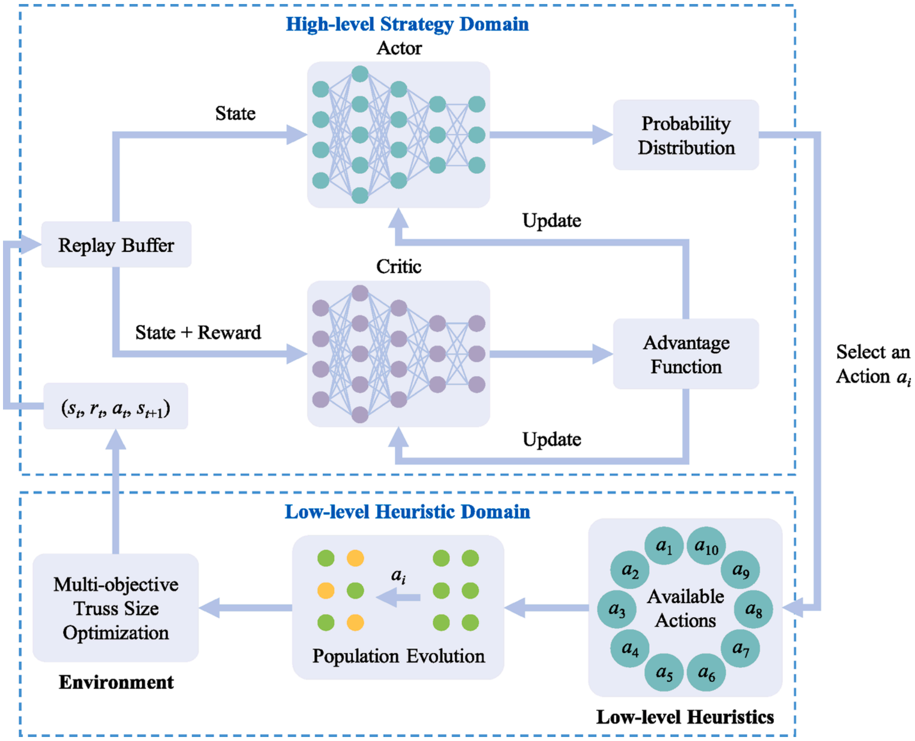


Fig. 1. The framework of HHEA-PPO.

will decrease $r_t(\theta)$, but it will not bring it below $1-\varepsilon$.

The policy with T timesteps is run and the policy is updated using the collected samples (Schulman et al., 2017). A truncated version of the generalized advantage estimation method is utilized to estimate the advantage function $\widehat{A}_t$. It can be expressed as

$$\begin{gathered}\widehat{A}_t = \sum_{l=0}^{T-t+1}(\gamma\lambda)^l\delta_{t+l}, \\ \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t),\end{gathered} \tag{5}$$

where $\gamma \in [0,1]$ is the discount factor, which determines the significance of future rewards. $\lambda \in [0,1]$ is the decay rate of the advantage function, which is used to attenuate future rewards when calculating the empirical advantage estimate. If $\lambda = 0$, then $\widehat{A}_t = \delta_t$, only considers the advantage obtained from a one-step difference is considered; if $\lambda = 1$, then $\widehat{A}_t = \sum_{l=0}^{T-t+1}\gamma^l r_{t+l} - V(s_t)$, the fully averaged advantage obtained from each step difference is taken into account.

By interacting with problem instances, PPO trains the agent online to learn an HLS. During the training process, the agent adjusts the policy parameters using the current state and the feedback from the objective function to achieve the goals of generating a high-performance HLS and selecting the most appropriate LLH. With the PPO algorithm, HHEA is able to learn adaptive strategies across different problem instances, improving convergence accuracy and stability in multi-objective truss optimization problems. The pseudocode for the PPO algorithm is presented in **Algorithm 1**.

```
Algorithm 1: Proximal policy optimization
Input: $A^{\pi_\theta}$ (actor network), $V^\phi$ (critic network), $R_B$ (replay buffer), $\gamma$ (discount factor),
  $\lambda$ (parameter of bias-variance tradeoff), $\varepsilon$ (clip range)
Extract $\{\langle s_i, op_i, \widetilde{r}(op_i), s_{i+1}\rangle\}, i = 1, \ldots, N$ from $R_B$;
Calculate the temporal difference error $\delta_t$;
Compute the advantage function $\widehat{A}_t$ using Eq. (5);
Compute the old logarithmic probability;
for $epoch = 1 : epochs$
    Evaluate the loss and gradient for actor $A^{\pi_\theta}$ and critic $V^\phi$;
    Update the $\theta$ and $\phi$ using the Adam algorithm;
end for
Output: trained $A^{\pi_{\theta'}}$ and $V^{\phi'}$
```

3.2.2. State space design

In reinforcement learning, states typically represent features of the external environment. The state space in this paper consists of three parts: the positional difference between the current individual and the optimal individual, the diversity of the population in the decision space, and the diversity of the population in the objective space. In designing the state space, existing research often directly uses the individual's position as the state (Tian et al., 2023). Although the individual's position is fundamental information for offspring generation, considering only the current position is insufficient to provide effective information for HLS to select LLH. An improved approach is to incorporate the positional difference between the current individual and the optimal individual as part of the state. This allows a better judgment of whether the individual's position is more suitable for exploration or exploitation, thereby determining the update strategy for each decision variable. Furthermore, the diversity of the population in the decision space reflects the distribution of individuals in the decision variables, guiding the HLS to explore and exploit the solution space more effectively. The information about the diversity of the population in the objective space reflects the distribution of individuals in terms of objective function values, which is crucial for enhancing the breadth and uniformity of the search space. Therefore, the state of the agent can be represented by

$$s = [\boldsymbol{x}_i - \boldsymbol{g}_b, \mathrm{div}(X), \mathrm{div}(F(X))], \tag{6}$$

where $\boldsymbol{x}_i$ is the individual's current position, and $\boldsymbol{g}_b$ is the optimal position found so far. $\mathrm{div}(X)$ and $\mathrm{div}(F(X))$ represent the population diversity in the decision space and objective space, respectively. Diversity can be measured by the distance between individuals, as indicated by

$$\mathrm{div}(X) = \frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{7}$$

$$\mathrm{div}(F(X)) = \frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{dist}(F(\boldsymbol{x}_i), F(\boldsymbol{x}_j)), \tag{8}$$

where $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y})$ represents the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{y}$, and N is the size of the population. $F(\boldsymbol{x})$ is the fitness value evaluated according to **Eq.** (3), which includes the penalty for the constraint violations.

3.2.3. Action space design

The LLH in HHEA-PPO corresponds to the action in reinforcement learning, i.e., $a \in \{\mathrm{LLH}_1, \mathrm{LLH}_2, \ldots, \mathrm{LLH}_n\}$. The design of the LLH, as a critical component of HHEA, has a significant impact on the efficiency of the algorithm. There are ten low-level heuristic operations integrated into HHEA-PPO to enhance the search capability and adaptability of the algorithm. These LLHs have demonstrated superior performance in various optimization problems (Tian et al., 2023). Combining them for adaptive co-evolution can enhance the exploration capability of the HHEA. A detailed description of these ten LLHs is given below.

(1) LLH_1 is adapted from the genetic algorithm (GA) designed for optimizing continuous variables, as shown in **Eq.** (9) (Deb et al., 2007). LLH_1 is suitable for large-scale search spaces, exhibiting strong global search capability.

$$\begin{gathered}\boldsymbol{x} = 0.5\left[(\boldsymbol{p}^1 + \boldsymbol{p}^2) + \beta(\boldsymbol{p}^1 - \boldsymbol{p}^2)\right], \\ \beta = \begin{cases}(2\mu)^{\frac{1}{\eta+1}}, & \mu \leqslant 0.5, \\ (2(1-\mu))^{-\frac{1}{\eta+1}}, & \mu > 0.5,\end{cases}\end{gathered} \tag{9}$$

where μ is a random number taken from (0,1), and $\eta = 20$ is an adjustable parameter.

(2) LLH_2 is derived from the PSO, as illustrated in **Eq.** (10) (Kennedy & Eberhart, 1995). LLH_2 has strong local search capability and fast convergence speed.

$$\boldsymbol{x}' = \boldsymbol{x} + w\cdot\boldsymbol{v} + r_1(\boldsymbol{p}_b^i - \boldsymbol{x}) + r_2(\boldsymbol{g}_b - \boldsymbol{x}), \tag{10}$$

where $\boldsymbol{v}$ is the velocity of the particle, r_1 and r_2 are random numbers taken from (0,1), $\boldsymbol{p}_b^i$ is the historical optimal position of individual i, and $\boldsymbol{g}_b$ is the historical optimal position of the population.

(3) $\mathrm{LLH}_3 \sim \mathrm{LLH}_8$ are adapted from the differential evolution (DE), as shown in **Eq.** (11) (Storn & Price, 1997). These operators exhibit good adaptability for high-dimensional, non-smooth, and non-convex problems, allowing them to jump out of local optima in complex landscapes.

$$\begin{gathered}\boldsymbol{x}' = \boldsymbol{x} + F(\boldsymbol{p}^1 - \boldsymbol{p}^2), \\ \boldsymbol{x}' = \boldsymbol{g}_b + F(\boldsymbol{p}^1 - \boldsymbol{p}^2), \\ \boldsymbol{x}' = \boldsymbol{x} + F(\boldsymbol{p}^1 - \boldsymbol{p}^2) + F(\boldsymbol{p}^3 - \boldsymbol{p}^4), \\ \boldsymbol{x}' = \boldsymbol{g}_b + F(\boldsymbol{p}^1 - \boldsymbol{p}^2) + F(\boldsymbol{p}^3 - \boldsymbol{p}^4), \\ \boldsymbol{x}' = \boldsymbol{x} + K(\boldsymbol{x} - \boldsymbol{p}^1) + F(\boldsymbol{p}^2 - \boldsymbol{p}^3), \\ \boldsymbol{x}' = \boldsymbol{x} + K(\boldsymbol{x} - \boldsymbol{p}^1) + F(\boldsymbol{p}^2 - \boldsymbol{p}^3) + F(\boldsymbol{p}^4 - \boldsymbol{p}^5),\end{gathered} \tag{11}$$

where $F = 0.5$ and $K = 0.5$ are adjustable parameters, $\boldsymbol{g}_b$ is the best position found so far.

(4) LLH_9 is taken from the fast evolutionary programming (FEP), as shown in **Eq.** (12) (Yao et al., 1999). LLH_9 provides better performance on coverage optimization tasks by striking an excellent balance between exploration and exploitation.

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{v}\cdot e^{\tau_1 r_{i1}^g + \tau_2 r_{i2}^g}\cdot\boldsymbol{r}_i^c, \tag{12}$$

where $\tau_1 = 1/\sqrt{2n}$, $\tau_2 = 1/\sqrt{2\sqrt{n}}$, n is the dimension of the decision variable, r_i^c and r_i^g denote random numbers with Cauchy and Gaussian distributions, respectively.

(5) LLH_{10} is the uniform mutation (UM) operator, as shown in **Eq.** (13) (Deb, 2011). LLH_{10} can effectively maintain the population diversity and is well-suited for complex search spaces.

$$\boldsymbol{x}' = \boldsymbol{x} + r(\boldsymbol{ub} - \boldsymbol{lb}), \tag{13}$$

where r_i is a random number taken from (0,1), u_i and l_i are the upper and lower bounds of the i-th decision variable.

These ten LLH operators provide HHEA-PPO with a variety of search strategies, allowing it to explore the solution space of multi-objective truss optimization problems more comprehensively. The process of selecting actions for the PPO algorithm involves using a policy network A^{π_θ} to infer the probability distribution of taking each action over a given state $\boldsymbol{s}$. Unlike DQN, PPO is a stochastic policy-based algorithm where the output $\pi_\theta(\cdot|s)$ is a probability distribution rather than a deterministic action. Therefore, each agent also needs to sample an action, i.e., $a \sim \text{Samp}(\pi_\theta(\cdot|s))$. Subsequently, the sampled action is executed, and the agent interacts with the environment to obtain a new state and reward. The stochastic policy ensures that each state has a certain probability of selecting different actions, allowing the algorithm to achieve a good balance between exploration and exploitation.

3.2.4. *Reward mechanism*

The reward mechanism is a key component of the online learning mechanism, used to evaluate the effectiveness of actions performed in each state. A new state space and reward mechanism are designed to enable the agent to effectively learn the mapping from states to actions. By updating the policy parameters online in real time, the algorithm can continuously optimize itself during the search process, which enhances its adaptability to multi-objective optimization problems.

In HHEA-PPO, the reward value is used as feedback to evaluate the effectiveness of actions performed in each state. The reward reflects whether the execution of an action contributes to improve the optimal solution found so far. In swarm intelligence algorithms, lower individual fitness implies a closer proximity to the global optima. The improvement in population quality means a potential movement of the population towards promising regions. Therefore, previous research has typically considered the ratio of fitness improvement as the reward for the selected LLH (Li et al., 2019; Zhang, Tang, et al., 2023). However, as the algorithm iterates, the ratio of fitness improvement may decrease gradually or sharply, leading to instability of the learned policy. An improved approach is to focus only on whether the LLH brings improvement, irrespective of the rate of improvement. The learning objective for the agent is the expected improvement probability of the LLH over its parent generation. In multi-objective optimization, the goal is to minimize all objective functions. Thus, if the agent improves a particular objective function, the reward is 1; if it worsens a particular objective function, the reward is −1; otherwise, it is 0. For all objective functions, if the overall fitness becomes better after executing the LLH, the reward is 1; otherwise, it is 0. Consequently, the reward for the agent executing the selected LLH in the current state is

$$\widehat{r}(\text{LLH}_i) = \begin{cases} 1, & \sum_{i=1}^{m} r_i > 0, \\ 0, & \text{otherwise}, \end{cases}$$
$$r_i = \begin{cases} 1, & f_{\text{LLH}_i}^{(t)} < f_i^{(t)}, \\ -1, & f_{\text{LLH}_i}^{(t)} > f_i^{(t)}, \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

where $f_i^{(t)}$ denotes the fitness of the i-th individual at the t-th iteration, and $f_{\text{LLH}_i}^{(t)}$ denotes the fitness of the individual generated by the selected LLH. After each credit assignment, the quadruple $\left\langle s_i^{(t)}, a_i^{(t)}, \widehat{r}(a_i^{(t)}), s_i^{(t+1)} \right\rangle$ is saved as a sample in the replay buffer, providing sample data for the training of the policy network and value network.

3.3. *Archive with dynamic crowding distance*

Many crowding distance methods have been proposed, as shown in Table 3. Most of these methods focus on the diversity of the decision space and the objective space, such as SCD, CPD, CSCD, DCCD, and ICD. In the DSC method, non-dominated solutions are represented in three different reference spaces: the decision space, the objective space, and the unified space. By dynamically switching between these three spaces, the calculation of crowding distances is improved to maintain diversity and balance in both the decision and objective spaces (Kahraman, Akbel, Duman, et al., 2022). These methods are aimed at solving multimodal multi-objective optimization problems.

For general multi-objective problems, the focus is mainly on the diversity of the objective space. The DECD method calculates crowding distances dynamically, solving the problem of traditional crowding distance methods that may lead to overcrowded or sparse regions. In addition, the SDE method incorporates convergence into the crowding distance calculation. By evaluating the shifted diversity of the objective

Table 3
Innovative methods proposed to improve the crowding distance calculation.

Research	Method	Characteristics
(Yue et al., 2018)	Special crowding distance (SCD)	An index-based ring topology is used to find more Pareto optimal solutions, and a special crowding distance and environmental selection criteria are used to obtain a good distribution in decision and objective spaces
(Liu et al., 2019)	Convergence-penalized density (CPD)	A double *k*-nearest neighbor method is used to ensure diversity in decision and objective spaces
(Liang et al., 2021)	Clustering-based special crowding distance (CSCD)	A clustering-based special crowding distance method and elite selection mechanism are used to improve diversity in multimodal multi-objective optimization problems
(Lin et al., 2021)	Dual clustering-based crowding distance (DCCD)	The initial clustering is performed in the decision space, and non-dominated solutions in each local cluster are selected to maintain local Pareto clustering, and the second clustering is performed in the objective space
(Yue et al., 2021)	Improved crowding distance (ICD)	A weighted sum of Euclidean distances is used to replace the crowding distance in the decision space
(Kahraman, Akbel, Duman, et al., 2022)	Dynamic switched crowding (DSC)	Improve search performance of multi-objective algorithms by dynamically switching between different reference spaces (decision space, objective space, and a unified space combining both)
(Zhao et al., 2022)	Dynamic elimination-based crowding distance (DECD)	The archive is maintained by a dynamic elimination mechanism, while a non-dominated sorting strategy is incorporated to build a solution update mechanism
(Yin et al., 2023)	Shift-based density estimation (SDE)	Crowding distance calculation is improved by a shift-based density estimation method, which increases the crowding degree of poorly converging individuals to increase environmental selection pressure

space, it allows solutions with better convergence to have more competitive crowding distances, thereby accelerating the convergence speed.

To maintain the diversity and convergence of the Pareto set, a dynamic crowding distance (DCD) method is introduced and applied to the maintenance of the external Pareto archive. Compared with traditional crowding distance methods, DCD mainly focuses on the diversity of the objective space by incorporating constraint violation information. It only dynamically calculates the crowding distance of the neighbors of the removed individual, resulting in lower computational complexity. Previous DCD methods require recalculating the crowding distance for all remaining solutions when one solution is removed, leading to higher computational complexity, especially when there are a large number of solutions (Luo et al., 2023; Yin et al., 2023). Fig. 2 illustrates the change in crowding distance before and after removing a solution $\boldsymbol{x}_i$ in a two-objective scenario.

According to Fig. 2**(a)**, before removing the current solution $\boldsymbol{x}_i$, the crowding distance of the previous solution $\boldsymbol{x}_{i-1}$ and the next solution $\boldsymbol{x}_{i+1}$ adjacent to the i-th solution is

$$D(\boldsymbol{x}_{i-1}) = \sum_{k=1}^{m} \frac{|f_k(\boldsymbol{x}_i) - f_k(\boldsymbol{x}_{i-2})|}{\max(f_k) - \min(f_k)}, \tag{15}$$

$$D(\boldsymbol{x}_{i+1}) = \sum_{k=1}^{m} \frac{|f_k(\boldsymbol{x}_i) - f_k(\boldsymbol{x}_{i+2})|}{\max(f_k) - \min(f_k)}, \tag{16}$$

where m is the number of objectives. According to Fig. 2**(b)**, after removing the current solution $\boldsymbol{x}_i$ with the minimum crowding distance, the crowding distances of the previous solution $\boldsymbol{x}_{i-1}$ and the next solution $\boldsymbol{x}_{i+1}$ adjacent to the i-th solution are changed to

$$D(\boldsymbol{x}_{i-1}) = \sum_{k=1}^{m} \frac{|f_k(\boldsymbol{x}_{i+1}) - f_k(\boldsymbol{x}_{i-2})|}{\max(f_k) - \min(f_k)}, \tag{17}$$

$$D(\boldsymbol{x}_{i+1}) = \sum_{k=1}^{m} \frac{|f_k(\boldsymbol{x}_{i-1}) - f_k(\boldsymbol{x}_{i+2})|}{\max(f_k) - \min(f_k)}. \tag{18}$$

The interaction between the population and the archive is crucial to ensure the convergence and diversity of solutions. In HHEA-PPO, the archive serves as an external memory repository to store the Pareto optimal solutions found during the optimization process. Specifically, after the population generates offspring, non-dominated sorting and DCD are used to compare the offspring with the parents, and the better solutions are selected to update the population. The updated population is then merged with the archive to form a combined solution set, which is re-evaluated using non-dominated sorting and DCD. Through this interaction, the external Pareto archive is periodically updated to ensure that it contains only well-distributed Pareto solutions in the objective space, thereby maintaining the diversity and convergence of the solutions. **Algorithm 2** provides the pseudocode for updating the external Pareto archive using DCD.

```
Algorithm 2: Update archive with dynamic crowding distance
Input: $A_e$ (external Pareto archive), $P_t$ (current population), $N$ (capacity of the
  archive)
// Population-Archive Interaction
Combine the archive with the population $A_e \leftarrow \{A_e \cup P_t\}$;
Sort individuals by fitness values;
Calculate the crowding distance of individuals;
if size($A_e$) > $N$
  // Dynamic Crowding Distance
  for $i = 1 : (\mathrm{size}(A_e) - N)$
    Remove solution $\boldsymbol{x}_i$ with the minimum crowding distance from $A_e$;
    Update the crowding distance of $\boldsymbol{x}_{i-1}$ using Eq. (17);
    Update the crowding distance of $\boldsymbol{x}_{i+1}$ using Eq. (18);
  end for
end if
Output: updated Pareto archive $A_e$
```

After removing a solution using the DCD method, only the crowding distances of neighboring solutions need to be recalculated, thus reducing the computational complexity from $O(TMN^2)$ to $O(TMN)$, where N is the population size, M is the number of objective functions, and T is the maximum number of iterations. The DCD method demonstrates obvious advantages when dealing with a large number of non-dominated solutions. Assuming there are 100 non-dominated solutions and the task is to remove 50 of them, the effectiveness of the DCD method is illustrated in Fig. 3. The initial scenario is depicted in Fig. 3**(a)**, while **(b) and (c)** present the processing results of the crowding distance-based and DCD-based external archives, respectively. The results indicate that the DCD-based external archive has better diversity of solutions compared to the traditional crowding distance-based one.

3.4. Procedure of HHEA-PPO

The pseudocode of the HHEA-PPO is presented in **Algorithm 3**. The execution steps of the HHEA-PPO can be decomposed into the following stages.

(1) Initialization (steps 1–7): Initialize the position of the population, the parameters of the policy network and the value network.

(2) Parent selection (step 9): Select the parents that will be used to generate offspring.

(3) LLH selection (steps 10–14): According to the probabilities inferred by the policy network, select a set of actions for each individual, i.e., select a predefined LLH to generate offspring.

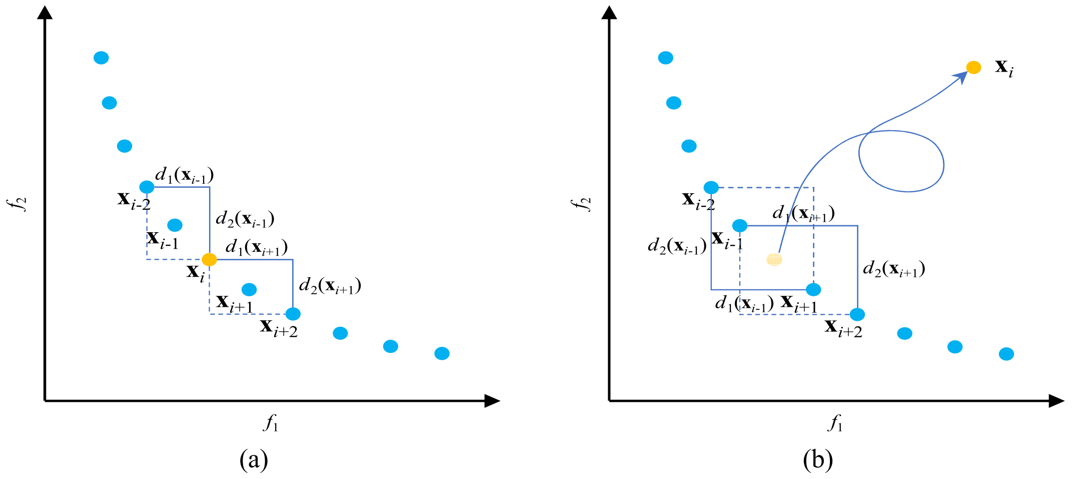


Fig. 2. Changes in crowding distance after removing a solution $\boldsymbol{x}_i$ using DCD. (a) Before the solution is removed. (b) After the solution is removed.

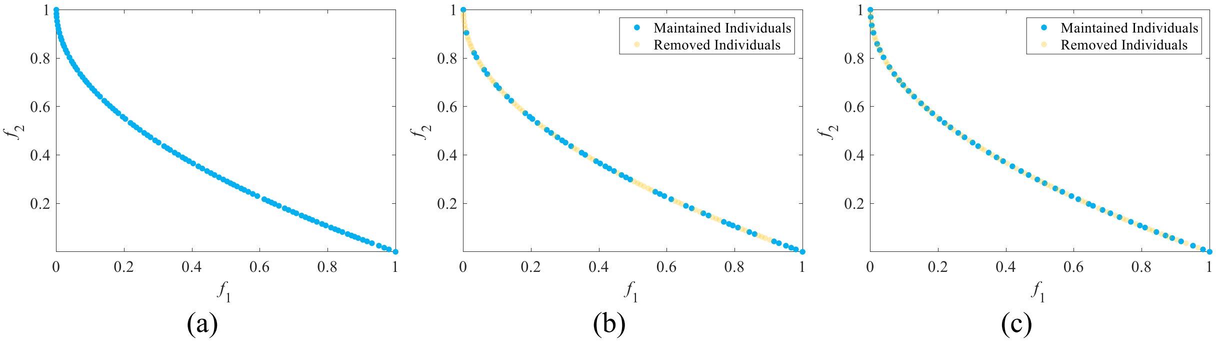


Fig. 3. Comparison of archive updating methods. (a) Initial archive. (b) Archive updated by traditional crowding distance. (c) Archive updated by DCD.

(4) Reward evaluation (steps 15–17): Evaluate whether the offspring improves upon the parents to determine whether the selected LLH receives a reward.

(5) Position update (steps 18–19): Update the position of the optimal individual found so far, which is used to guide the evolution of the population and the selection of the LLH.

(6) Data collection and training (steps 20–23): Build the replay buffer of sample data for training the policy network and value network of PPO. In each iteration, PPO is trained for a specified number of epochs (see **Algorithm 1**).

(7) Maintain the Pareto archive (steps 24–27): Use the dynamic crowding distance to maintain the diversity of the Pareto archive.

```
Algorithm 3: Procedure of the HHEA-PPO
Input: $N$ (size of the population), $Q_A$ (architectural parameters of the network), $\gamma$
    (discount factor), $\lambda$ (parameter of bias-variance tradeoff), epochs (iteration of policy
    updates), $\varepsilon$ (clip range), $lr_a$ (learning rate of the actor), $lr_c$ (learning rate of the
    critic), $T_{\max}$ (maximum number of iterations)
1.  Initialize the population $X$ and evaluate fitness $F$;
2.  Execute the non-dominated sorting and evaluate crowding distance $D$;
3.  $\mathbf{g}_b \leftarrow$ The individual with the maximum crowding distance in the PF;
4.  Initialize the policy network $A^{\pi_{\theta_0}}$ and value network $V^{\phi_0}$ using $Q_A$;
5.  Initialize the external archive $A_e$;
6.  Evaluate the diversity of the population in decision and objective space;
7.  Extract the initial state $s^0$;
8.  for $t = 1 : T_{\max}$
9.    Select the parent $P$ from population $X$ via 2-tournament selection;
10.     for $i = 1 : N$
11.       Extract the state $s_i^t = [\mathbf{p}_i - \mathbf{g}_b, \mathrm{div}(X), \mathrm{div}(F(X))]$;
12.       Select a LLH according to the policy network $A^{\pi_{\theta_t}}(s_i^t, a_i^t)$;
13.       Apply the selected LLH to generate the offspring $\mathbf{x}_i$;
14.     end for
15.     Evaluate the fitness of offspring population $X$;
16.     Evaluate the reward of the selected LLH using Eq. (14);
17.     Compare offspring to parent population and retain better individuals;
18.     Execute the non-dominated sorting and evaluate crowding distance $D$;
19.     $\mathbf{g}_b \leftarrow$ The individual with the maximum crowding distance in the PF;
20.     Evaluate the diversity of the population in decision and objective space;
21.     Extract the state $s^t = [X - \mathbf{g}_b, \mathrm{div}(X), \mathrm{div}(F(X))]$;
22.     Replay buffer $R_b \leftarrow$ The tuple $\{\langle s_i^t, \mathrm{LLH}_i, \hat{r}(\mathrm{LLH}_i), s_i^{t+1} \rangle\}, i = 1, 2, \ldots, N$;
23.     Update the parameters of the actor and critic network via Algorithm 1;
24.     Update the Pareto archive $A_e$ using Algorithm 2;
25.   end for
Output: external Pareto archive $A_e$
```

4. Experimental studies

4.1. Experimental settings

Eight multi-objective truss optimization problems, comprising trusses with 10, 25, 37, 60, 72, 120, 200, and 942 bars, were employed to evaluate the algorithm's performance (Panagant et al., 2021). The material characteristics and allowable stresses are the same for all of these trusses. The material density, elastic modulus and permissible stresses are set to $7850\mathrm{kg/m^3}$, $200 \times 10^9\mathrm{Pa}$, and $400 \times 10^6\mathrm{Pa}$, respectively. Considering that conventional size of the bars is usually restricted, the size of each structural unit is planned as a discrete variable. For all trusses except the 942-bar, the minimum cross-sectional area of the bars is $10^{-3}\mathrm{m}^2$, the maximum cross-sectional area is $2.1 \times 10^{-2}\mathrm{m}^2$, and the interval between variables is $5 \times 10^{-4}\mathrm{m}^2$. In the 942-bar truss problem, the minimum cross-sectional area is $10^{-3}\mathrm{m}^2$, the maximum cross-sectional area is $10^{-1}\mathrm{m}^2$, and the interval between variables is $10^{-3}\mathrm{m}^2$. Fig. 4 illustrates the topologies and shapes of these truss problems, including planar trusses with 10, 37, 60, and 200 bars, as well as spatial trusses with 25, 72, 120, and 942 bars. Since the symmetry of the truss structure, design variables are typically grouped in advance. Therefore, during the optimization process, the number of design variables is usually less than the number of truss members. The number of design variables in this paper is 10, 8, 15, 25, 16, 7, 29, and 59 corresponding to the 10, 25, 37, 60, 72, 120, 200, and 942 bars truss problems, respectively.

To evaluate the performance of HHEA-PPO, it was compared with thirteen algorithms, including classical NSGA-II (Deb et al., 2002), multi-objective cuckoo search (MOCS) (Yang & Deb, 2013), dual-population algorithm based on alternate evolution and degeneration (CAEAD) (Zou et al., 2021), helper-problem-assisted constrained multi-objective evolutionary algorithm (MSCEA) (Zhang, Tian, et al., 2023), IBMSMA (Yin et al., 2023), RPBILDE (Anosri et al., 2022), SHAMODE-WO (Carvalho et al., 2021; Panagant et al., 2019), dynamic switched crowding based multi-objective symbiotic organism search algorithm (DSC-MOSOS) (Ozkaya, Kahraman, et al., 2024), multi-objective PSO using ring topology and special crowding distance (MO-Ring-PSO-SCD) (Yue et al., 2018), improved multi-objective manta ray foraging optimization (IMOMRFO) (Kahraman, Akbel, & Duman, 2022), decomposition-based multi-objective evolutionary algorithm with deep Q-network (MOEA/D-DQN) (Tian et al., 2023), hyper-heuristic evolutionary algorithm with deep Q-network (HHEA-DQN), and HHEA-PPO with dynamically switched crowding distance (HHEA-PPO-DSC). Among them, NSGA-II and MOCS are classical evolutionary and swarm intelligence algorithms. CAEAD and IBMSMA utilize multiple swarm co-evolutionary frameworks. MSCEA helps the agent to better explore the solution space by introducing an auxiliary problem. RPBILDE and SHAMODE-WO are regarded as state-of-the-art algorithms for most multi-objective truss problems (Panagant et al., 2021). DSC-MOSOS, MO-Ring-PSO-SCD, and IMOMRFO are powerful and recent multi-objective optimization algorithms that integrate advanced methods for crowding distance estimation. MOEA/D-DQN is an adaptive operator selection algorithm using a deep Q-network, similar to hyper-heuristic algorithms. HHEA-DQN and HHEA-PPO have the same LLH set, differing only in the HLS, and is used to verify whether the PPO-based HLS outperforms DQN. HHEA-PPO-DSC is used to validate the efficiency of DCD. To ensure a fair comparison, the parameters of all algorithms were set according to their original literature, as detailed in Table 4. All codes were run on MATLAB R2022b with Win 11 OS, Intel® Core™ i7-13700H CPU (2.40 GHz) and 32 GB RAM. Experimental data were collected using PlatEMO 4.5 (Tian et al., 2017), and the default parameters of the platforms were used unless otherwise stated.

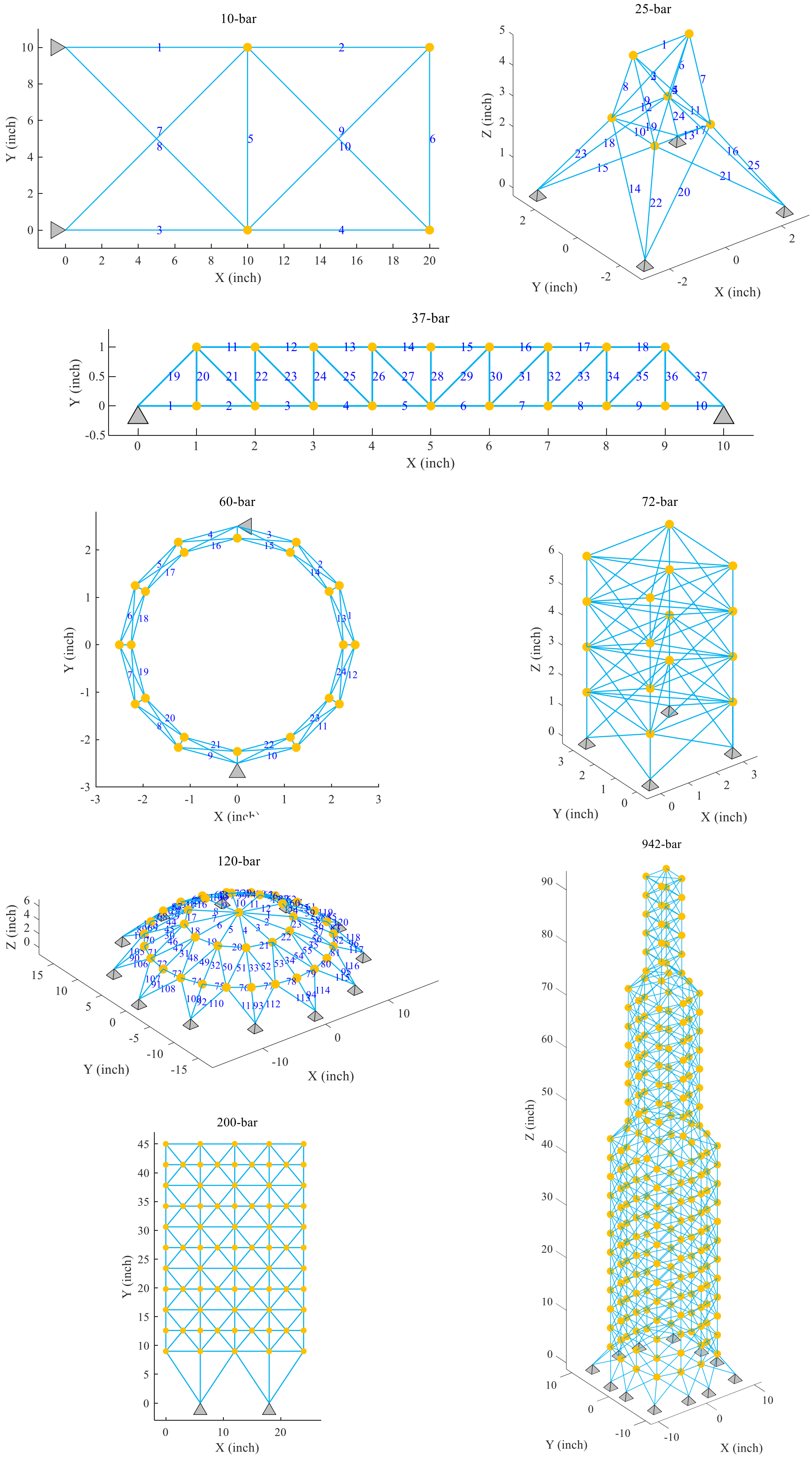


Fig. 4. Topology of truss optimization problems.

Table 4
Parameter settings of the competing algorithms.

Algorithm	Parameter
IMOMRFO	Scaling factor $s = 2$.
SHAMODE-WO	Memory size $h = 5$; memory index $k = 1$.
MSCEA	Decrease trend of the dynamic constraint boundary $c_p = 5$.
MOCS	Discovery probability $p_a = 0.25$; shape parameter of Lévy flight $\beta = 3/2$.
MO-Ring-PSO-SCD	Maximum size of the individual best archive $n_{\text{PBA}} = 5$; maximum size of the neighborhood best archive $n_{\text{NBA}} = 3 \cdot n_{\text{PBA}}$.
CAEAD	Initial ε value $\varepsilon_k = 10^8$; minimum ε value $\varepsilon_{\min} = 10^{-4}$; change threshold $\delta_{\text{threshold}} = 10^{-2}$; max change $\delta_{\max} = 1$; proportion factor $\tau = 0.05$.
IBMSMA	Random reset probability $z = 0.03$; maximum regrouping interval iteration $N_{\max} = 40$; minimum regrouping interval iteration $N_{\min} = 10$; scaling factor $s = \text{rand}(0.2, 0.8)$.
RPBILDE	Learning rate $lr_0 = 0.25$; mutation probability $m_p = 0.05$; mutation shift $m_s = 0.2$; crossover probability $p_c = 0.7$; scaling factor $s = 0.8$; selecting probability $p_s = 0.5$.
MOEA/D-DQN	Discount factor $\gamma = 0.5$; learning rate $lr = 0.01$; size of reward window is 100; neighborhood size is 10; frequency of weights update is 0.05; numbers of neurons in hidden layers are $\{128, 256, 128, 64, 32\}$; size of experience replay pool is 512; minibatch size is 16; update interval of target network is 10.
HHEA-DQN	Discount factor $\gamma = 0.5$; learning rate $lr = 0.01$; size of reward window is 100; numbers of neurons in hidden layers are $\{32, 64, 32, 16, 16\}$; size of experience replay pool is 100; mini-batch size is 32; update interval of target network is 5; size of external archive is 100.
HHEA-PPO	Discount factor $\gamma = 0.98$; parameter of bias-variance tradeoff $\lambda = 0.95$; learning rate of actor network $lr_a = 0.001$; learning rate of critic network $lr_c = 0.01$; iteration of policy updates $epochs = 10$; clip range $\varepsilon = 0.02$; numbers of neurons in hidden layers are $\{32, 64, 32, 16, 16\}$; size of replay buffer is 100; mini-batch size is 32; size of external archive is 100.

*The population size is 100, the maximum number of evaluations is 50000, and the neural network is trained using the Adam optimizer (Kingma & Ba, 2017).

4.2. Performance metrics

Hypervolume (HV) (While et al., 2006): Since the true PF of the truss problem is not available, the HV is used as a performance metric. The HV value is calculated as

$$\text{HV}(PF, r^*) = \bigcup_{i=1}^{|PF|} v(PF_i, r^*), \tag{19}$$

where $|PF|$ denotes the number of solutions, and $v(PF_i, r^*)$ represents the hypervolume created between the i-th solution and the reference point r^*. During the calculation of HV, the reference point for each truss problem was 1.1 times the largest objective value obtained from 30 random runs of all algorithms. The reference points for the eight truss optimization problems are listed in Table 5.

To evaluate the time and stability of finding feasible solutions when solving truss problems, the methods from the references (Duman et al., 2021; Gürgen et al., 2022) are employed. Specifically, the success rate (SR), average calculation durations (ACDs), and average number of fitness evaluations (AFEs) are used to further evaluate the performance of the algorithms. To obtain these parameters, it is necessary to define feasible solutions.

Table 5
Reference points for truss problems.

Problem	Mass	Compliance
10-bar	2.113796e + 04	2.286985e + 05
25-bar	1.518560e + 04	6.846669e + 04
37-bar	7.460509e + 03	4.033068e + 04
60-bar	1.134696e + 04	1.051983e + 05
72-bar	3.867375e + 04	1.500580e + 05
120-bar	1.140247e + 05	1.788624e + 06
200-bar	1.596036e + 05	2.831675e + 05
942-bar	3.934378e + 06	2.573770e + 05

Feasible solution (FS): The FS is different from the feasible solution in constrained optimization problems (solutions that satisfy all the constraints of the problem). For specific descriptions, please refer to (Öztürk & Kahraman, 2023). In this study, FS is defined by the average HV of the competing algorithms. Specifically, for the i-th truss problem, its FS can be expressed as

$$\text{FS}_i = \frac{1}{N_{\text{CA}}} \sum_{j=1}^{N_{\text{CA}}} \frac{1}{K} \sum_{k=1}^{K} \text{HV}_{i,j,k}, \tag{20}$$

where $\text{HV}_{i,j,k}$ represents the HV value of the i-th problem for the j-th algorithm in the k-th run, N_{CA} is the number of competing algorithms, and K is the number of independent runs. The FSs calculated for the eight truss problems are listed in Table 6.

Success rate (SR): SR is used to determine the frequency with which an algorithm successfully finds the FS. It can be defined as

$$\text{SR} = \frac{k}{K} \times 100, \tag{21}$$

where k is the number of successful runs where the FS is found, and K is the total number of runs.

Average calculation durations (ACDs): ACDs represent the average time taken by the algorithm to find the FS, calculated only for successful runs. It can be expressed as

$$\text{ACDs} = \frac{1}{k} \sum_{j=1}^{k} t_{\text{CD}_j}, \tag{22}$$

where k is the number of successful runs, and t_{CD_j} is the time taken to find a feasible solution in the j-th successful run.

Average number of fitness evaluations (AFEs): AFEs quantify the average computational effort required by an algorithm to find the FS, calculated for successful runs. It is defined as

$$\text{AFEs} = \frac{1}{k} \sum_{j=1}^{k} \text{FEs}_j, \tag{23}$$

where FEs_j is the number of fitness evaluations needed to find the FS in the j-th successful run. These metrics provide a comprehensive evaluation of the algorithm's performance, including solution quality, computational complexity, and stability.

4.3. Experimental results and analysis

4.3.1. Overall performance analysis

The mean and standard deviation of HV for all competing algorithms are listed in Table 7, and statistical analyses were performed using Friedman's average ranking and Wilcoxon rank sum test. The results show that HHEA-PPO obtained the optimal HV values in the six truss problems and outperformed the competing algorithms not only in terms of average convergence accuracy, but also in terms of stability. According to the Friedman's average ranking, HHEA-PPO ranks first overall, followed by IBMSMA and SHAMODE-WO. In terms of average HV results, HHEA-PPO improved by 0.30 % and 0.12 % over IBMSMA and SHAMODE-WO, respectively. These three algorithms adopted multi-swarm collaboration strategies, indicating that multi-swarm collaboration is an efficient search strategy in truss optimization. In HHEA-PPO, agents select LLHs by the state of the individual, allowing them to use multiple LLHs in a single iteration. This is similar to multi-swarm cooperation, but with more precise selection. Although HHEA-PPO performs well on most problems, it is inferior to IBMSMA and HHEA-DQN on the 25-bar truss problem. CAEAD significantly outperforms HHEA-PPO on the 60-bar truss problem. This will be analyzed

Table 6
Feasible solutions for each truss optimization problem.

Problem	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar
Feasible solution	0.680769	0.728839	0.752982	0.643146	0.741413	0.601100	0.829099	0.953455

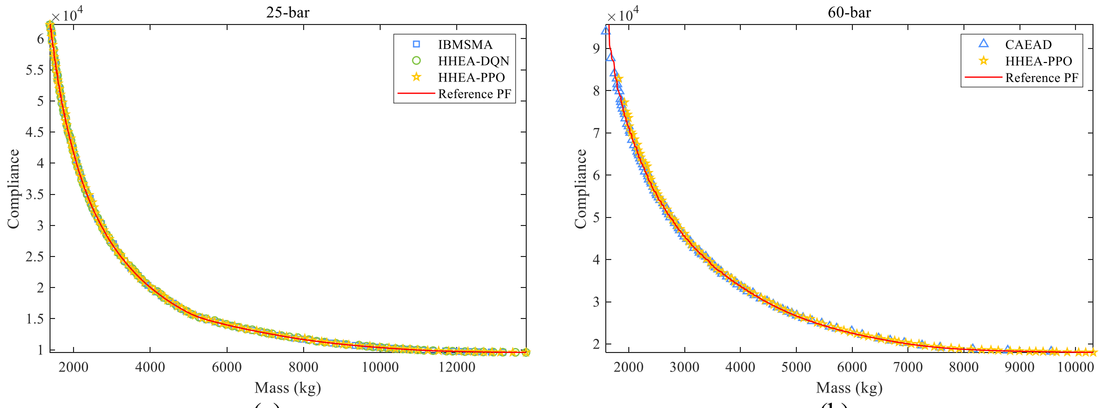
Table 7
HV values obtained by all competing algorithms.

Algorithm	Index	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar	FAR	+/-/=
MOCS	Mean	6.7541e-1 -	7.2603e-1 -	7.4434e-1 -	6.1823e-1 -	7.2880e-1 -	5.9557e-1 -	8.2465e-1 -	9.4786e-1 -	11.00 (12)	0/8/0
	STD	2.06e-3	9.83e-4	2.02e-3	8.77e-3	3.72e-3	1.74e-3	1.87e-3	2.93e-3	11.25 (13)	
DSC-MOSOS	Mean	6.7240e-1 -	7.2591e-1 -	7.3962e-1 -	5.9962e-1 -	7.1984e-1 -	5.9222e-1 -	8.1444e-1 -	9.4192e-1 -	12.13 (13)	0/8/0
	STD	1.93e-3	5.42e-4	1.60e-3	8.62e-3	3.73e-3	1.89e-3	2.14e-3	1.70e-3	10.75 (12)	
MO-Ring-PSO-SCD	Mean	6.3811e-1 -	7.0997e-1 -	7.1231e-1 -	5.3917e-1 -	6.7964e-1 -	5.5418e-1 -	7.6676e-1 -	8.7235e-1 -	13.75 (14)	0/8/0
	STD	4.40e-3	2.35e-3	4.11e-3	1.64e-2	8.13e-3	5.64e-3	3.69e-3	4.18e-3	13.13 (14)	
MOEA/D-DQN	Mean	6.7702e-1 -	7.2681e-1 -	7.5124e-1 -	3.1734e-1 -	7.2142e-1 -	5.8997e-1 -	8.2953e-1 -	9.4960e-1 -	10.38 (11)	0/8/0
	STD	1.89e-3	**1.57e-5**	**1.10e-4**	6.85e-3	2.67e-3	**5.35e-5**	6.22e-4	3.46e-4	5.00 (4)	
IMOMRFO	Mean	6.7821e-1 -	7.2643e-1 -	7.4933e-1 -	6.4361e-1 -	7.3743e-1 -	5.9780e-1 -	8.2658e-1 -	9.5500e-1 -	8.88 (9)	0/8/0
	STD	9.38e-4	7.53e-4	1.63e-3	2.82e-3	1.63e-3	1.68e-3	1.12e-3	8.97e-4	8.50 (9)	
MSCEA	Mean	6.7933e-1 -	7.2911e-1 -	7.5025e-1 -	6.3031e-1 -	7.4099e-1 -	6.0096e-1 -	8.0755e-1 -	8.5230e-1 -	9.38 (10)	0/8/0
	STD	9.40e-4	1.92e-4	1.21e-3	8.38e-3	1.58e-3	4.14e-4	9.27e-3	1.46e-2	9.25 (11)	
NSGA-II	Mean	6.8000e-1 -	7.2866e-1 -	7.5240e-1 -	6.3239e-1 -	7.4249e-1 -	6.0118e-1 -	8.1243e-1 -	8.7237e-1 -	7.88 (8)	0/8/0
	STD	8.50e-4	1.80e-4	8.28e-4	8.81e-3	6.45e-4	1.77e-4	1.46e-2	1.90e-2	8.38 (8)	
HHEA-PPO-DSC	Mean	6.7973e-1 -	7.2804e-1 -	7.5195e-1 -	6.4253e-1 -	7.4181e-1 -	6.0020e-1 -	8.2993e-1 -	9.5793e-1 -	6.88 (7)	0/8/0
	STD	1.06e-3	5.22e-4	4.14e-3	5.96e-3	1.00e-3	4.85e-4	5.16e-4	5.11e-4	8.00 (7)	
RPBILDE	Mean	6.8070e-1 -	7.2862e-1 -	7.5249e-1 -	6.4840e-1 -	7.4231e-1 -	6.0140e-1 -	8.2780e-1 -	9.5187e-1 -	6.00 (6)	0/8/0
	STD	3.10e-4	1.52e-4	4.86e-4	2.51e-3	4.74e-4	2.81e-4	4.60e-4	1.11e-3	4.44 (3)	
HHEA-DQN	Mean	6.8189e-1 -	7.2981e-1 +	7.5399e-1 -	6.4068e-1 -	7.3881e-1 -	6.0190e-1 -	8.3081e-1 -	9.5480e-1 -	4.75 (5)	1/7/0
	STD	4.78e-4	9.06e-5	1.40e-3	1.03e-2	5.06e-3	5.62e-4	7.67e-4	4.25e-3	9.00 (10)	
CAEAD	Mean	6.8196e-1 -	7.2860e-1 -	7.5523e-1 =	**6.5314e-1** +	7.4342e-1 -	6.0194e-1 -	8.2853e-1 -	9.2365e-1 -	4.50 (4)	1/6/1
	STD	1.43e-4	4.36e-4	2.53e-4	2.71e-3	9.23e-4	1.57e-4	2.25e-3	9.87e-3	6.00 (6)	
IBMSMA	Mean	6.8212e-1 =	**7.3001e-1** +	7.5304e-1 -	6.4073e-1 -	7.4054e-1 -	6.0180e-1 -	8.3107e-1 -	9.5815e-1 -	4.00 (2)	1/6/1
	STD	3.53e-4	1.35e-4	4.86e-4	7.59e-3	1.98e-3	5.47e-4	3.54e-4	4.99e-4	5.94 (5)	
SHAMODE-WO	Mean	6.8153e-1 -	7.2898e-1 -	7.5398e-1 -	6.4943e-1 =	7.4228e-1 -	6.0156e-1 -	8.3076e-1 -	9.5858e-1 -	4.13 (3)	0/7/1
	STD	1.69e-4	1.89e-4	3.68e-4	3.84e-3	7.22e-4	2.51e-4	2.18e-4	**2.48e-4**	3.88 (2)	
HHEA-PPO	Mean	**6.8221e-1**	7.2974e-1	**7.5524e-1**	6.5025e-1	**7.4404e-1**	**6.0226e-1**	**8.3136e-1**	**9.5883e-1**	**1.38 (1)**	-
	STD	**1.13e-4**	7.53e-5	1.71e-4	**2.19e-3**	**4.16e-4**	9.17e-5	**1.86e-4**	2.64e-4	**1.50 (1)**	

*The best result is shown in bold, '+', '-' and '=' are the results of the Wilcoxon rank sum test, and FAR indicates the Friedman's average ranking. In each problem-specific ranking, blue, red, and yellow represent the first, second, and third algorithms, respectively. Gray represents the other competing algorithms in the top. Gray represents the other competing algorithms in the top ten for that problem.

in detail in the following discussions. In conclusion, HHEA-PPO performs excellently on most problems, with significant advantages in convergence accuracy and stability. However, for the 25-bar and 60-bar truss problems, HHEA-PPO still needs to improve its balance between exploration and exploitation to further enhance its performance. In addition, the multi-swarm collaboration strategy shows significant advantages in truss optimization.

Fig. 5**(a)** displays the best PFs obtained by IBMSMA, HHEA-DQN, and HHEA-PPO for the 25-bar truss optimization problem. It can be seen that all three algorithms can find an ideal PF, but the distribution of a small number of solutions in HHEA-PPO is poor. This may be due to the fact that after the DCD method removes the most crowded solutions, the solutions generated by HHEA-PPO are not as good as those removed. This is because the reward mechanism of HHEA-PPO is designed to be



(a) (b)

Fig. 5. Optimal PFs obtained by algorithms. (a) 25-bar truss problem. (b) 60-bar truss problem.

dependent on whether there is improvement or not, not on the degree of improvement. For this problem, IBMSMA and HHEA-DQN converge faster in the later iterations.

Fig. 5**(b)** illustrates the best PFs obtained by CAEAD and HHEA-PPO for the 60-bar truss problem. It is clear that HHEA-PPO focuses more on optimizing the compliance of the truss, while CAEAD focuses more on optimizing the mass. In terms of truss mass, the solutions obtained by CAEAD are better than the current reference PF, showing its unique advantage in this problem. In fact, HHEA-PPO converges more slowly on this problem and has not yet reached a state of convergence stagnation (see Fig. 6). Another reason is that the crowding distance in CAEAD is not normalized, while the scales of the two objective functions are different. In contrast, HHEA-PPO normalizes the objective functions to obtain more uniformly distributed solutions and thus focuses more on optimizing the compliance of the truss.

To analyze the convergence process of HHEA-PPO and other competing algorithms, the average convergence curves of these algorithms are plotted in Fig. 6. It can be seen that HHEA-PPO, HHEA-PPO-DSC, and HHEA-DQN show slow convergence in the early iterations, but faster convergence in the later iterations. This is due to the fact that they all use online training and online application for optimization, which facilitates their use. In the early iterations, the agents have not yet learned effective HLS, resulting in slower convergence. HHEA-PPO outperforms HHEA-DQN in terms of convergence speed and accuracy, except for the 25-bar truss problem. This indicates that stochastic policy-based PPO is more suitable for HLS design than value-based DQN. However, the results of the 25-bar truss problem show that the PPO still faces challenges in balancing exploration and exploitation. In this problem, HHEA-PPO converges rapidly in the early iterations, and the HLS quickly reaches a stable state. Even with the stochastic policy, there is still insufficient exploration of other LLHs in the later iterations. As a result, HHEA-DQN with an $\varepsilon-$greedy strategy outperforms HHEA-PPO in later iterations. Moreover, HHEA-PPO-DSC, which integrates DSC, performs worse than HHEA-PPO on all truss problems, indicating that DCD is more suitable for multi-objective truss problems. Although DSC achieves a good balance between the decision space and the objective space, considering the distribution in the decision space tends to weaken the Pareto advantages in the objective space due to the much higher dimension of the decision variables compared to the objective space. Therefore, the DCD that focuses on the convergence of the objective space achieves better HV values. In summary, HHEA-PPO performs well in most cases, but still needs to find a better balance between exploration and exploitation for specific problems. In addition, DCD is more suitable than DSC for dealing with problems with high-dimensional decision variables.

The HV results of all the competing algorithms on eight truss problems are shown in Fig. 7, obtained by independently running 30 experiments. This figure shows that HHEA-PPO performs well on the 10-bar and 37-bar truss problems, with the median and upper quartile of its HV values higher than those of the other algorithms, indicating its superior and stable overall performance. IBMSMA and HHEA-DQN also perform well, but HHEA-PPO has a slight advantage in terms of convergence accuracy and stability. For the 25-bar problem, IBMSMA performs the best with the highest median HV value and a more concentrated distribution. HHEA-DQN comes next, while HHEA-PPO, although performing well in the early stages, shows slightly insufficient stability and convergence speed in the later stages, resulting in its overall HV value being slightly lower than IBMSMA and HHEA-DQN. On the 60-bar problem, CAEAD performs well, especially on the upper quartile and median HV values, demonstrating its unique advantage on this problem. HHEA-PPO comes next, although its distribution is more centralized, it is slightly slower in convergence speed. For the 72-bar problem, the performance of HHEA-PPO and IBMSMA is relatively close, but HHEA-PPO has a slight advantage in terms of stability. For the 120-bar and 200-bar truss problems, the performance of HHEA-PPO is still excellent, especially in terms of the median and distribution range of the HV values, showing its efficient convergence performance. Especially on the 200-bar problem, HHEA-PPO significantly outperforms other algorithms. On the 942-bar problem, HHEA-PPO has the most concentrated distribution of HV values and the highest median, indicating its high performance and stability on large-scale complex problems.

In addition, MOEA/D-DQN performs less effectively than other algorithms on most problems. This is because the algorithm lacks a constraint handling mechanism and is not suited to solve the problem. Specifically, Fig. 8 shows the PF obtained by MOEA/D-DQN performing 5000 and 10,000 evaluations for the 60-bar truss problem. It can be observed that MOEA/D-DQN optimizes on the basis of the pre-assigned weights, and the number of feasible solutions becomes smaller and smaller, resulting in worse HV values.

4.3.2. *Stability and efficiency analysis*

In the study of optimization algorithms, it is crucial to evaluate the stability of the algorithms. To further analyze the stability of HHEA-PPO and competing algorithms. This study evaluates the performance and strengths and weaknesses of each algorithm on different problems by analyzing three indicators: SR, ACDs, and AFEs.

SR measures the percentage of algorithms that reach the expected goal in multiple independent runs. A higher SR indicates that the algorithm has good stability and reliability on different problems. The SRs of all competing algorithms are counted in Table 8. It can be seen that the SR of different algorithms varies significantly across the truss problems. HHEA-PPO achieves nearly 100 % success rate on all eight truss problems, showing the most stable performance. In the eight problems, HHEA-PPO improves the SR by 9.07 %, 60.27 %, and 45.67 % compared to SHAMODE-WO, IBMSMA, and HHEA-DQN, respectively. SHAMODE-WO, CAEAD, and IBMSMA each achieve close to 100 % success rate on four problems. HHEA-DQN and HHEA-PPO-DSC perform well on the 25-bar and 942-bar problems, respectively, but their overall SR is slightly lower than the top four algorithms.

ACDs evaluate the average computation duration required by the algorithm to find the FS. Shorter ACDs mean that the algorithm can find solutions faster with the same computational resources, demonstrating its computational efficiency. Table 9 records the ACDs of each algorithm on different problems. It can be seen that IBMSMA has the shortest ACDs on the 10-bar, 37-bar, and 942-bar problems. HHEA-PPO has the shortest ACDs on the 25-bar and 200-bar problems, and its overall performance is second only to IBMSMA. CAEAD has the shortest ACDs for the 60-bar and 120-bar problems. NSGA-II has the fastest solution speed on the 72-bar problem. It is noteworthy that despite the inclusion of an online learning process, HHEA-PPO's ACDs to obtain FS are still better than most competing algorithms, indicating that the PPO-based HLS is highly efficient.

The SR and ACDs of all the competing algorithms on the eight truss problems are shown in Fig. 9. As can be seen, HHEA-PPO has a higher SR than other algorithms for all eight problems, and also has shorter ACDs for the 25-bar, 37-bar, 200-bar, and 942-bar problems, demonstrating its superior overall performance. Although IBMSMA has the shortest overall ACDs, its SR is lower than HHEA-PPO on the 37-bar, 60-bar, 72-bar, and 120-bar problems, indicating a gap in stability. CAEAD shows a balanced performance in both SR and ACDs, but performs slightly worse on the 25-bar, 200-bar, and 942-bar problems.

ACDs include both the processing time of competing algorithms and the evaluation time of the truss problems. For small-scale problems, ACDs are mainly influenced by the algorithm's operators, while for large-scale problems, ACDs are mainly influenced by the number of fitness evaluations. To eliminate the influence of the algorithm's processing time, it is necessary to count the effective evaluations of the algorithms. For most real-world optimization problems, the fitness evaluation time is usually much higher than the algorithm processing time. Therefore, the number of fitness evaluations usually better reflects the efficiency of the algorithm. The AFEs of all competing algorithms on

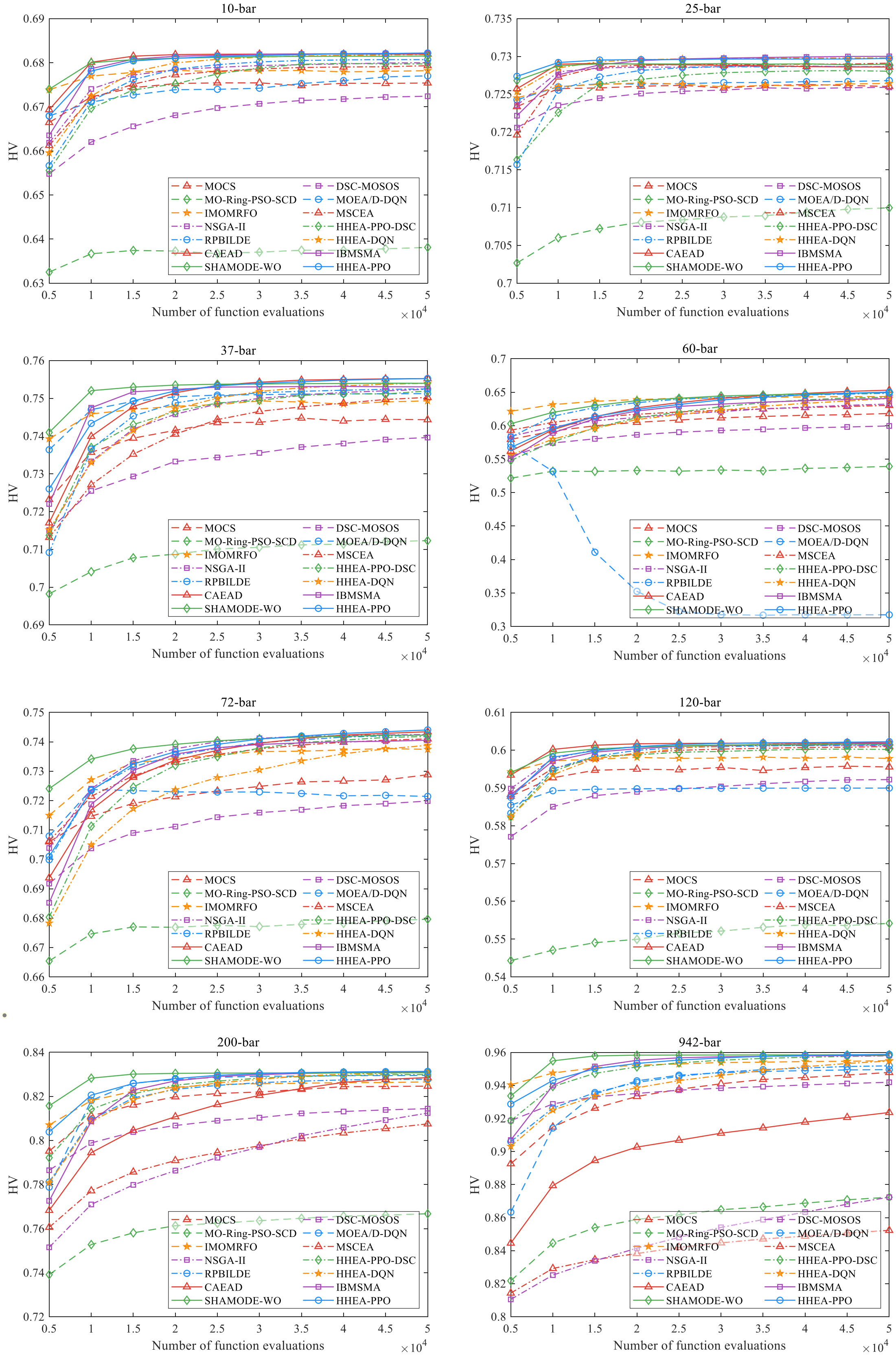


Fig. 6. Average convergence curves for all competing algorithms.

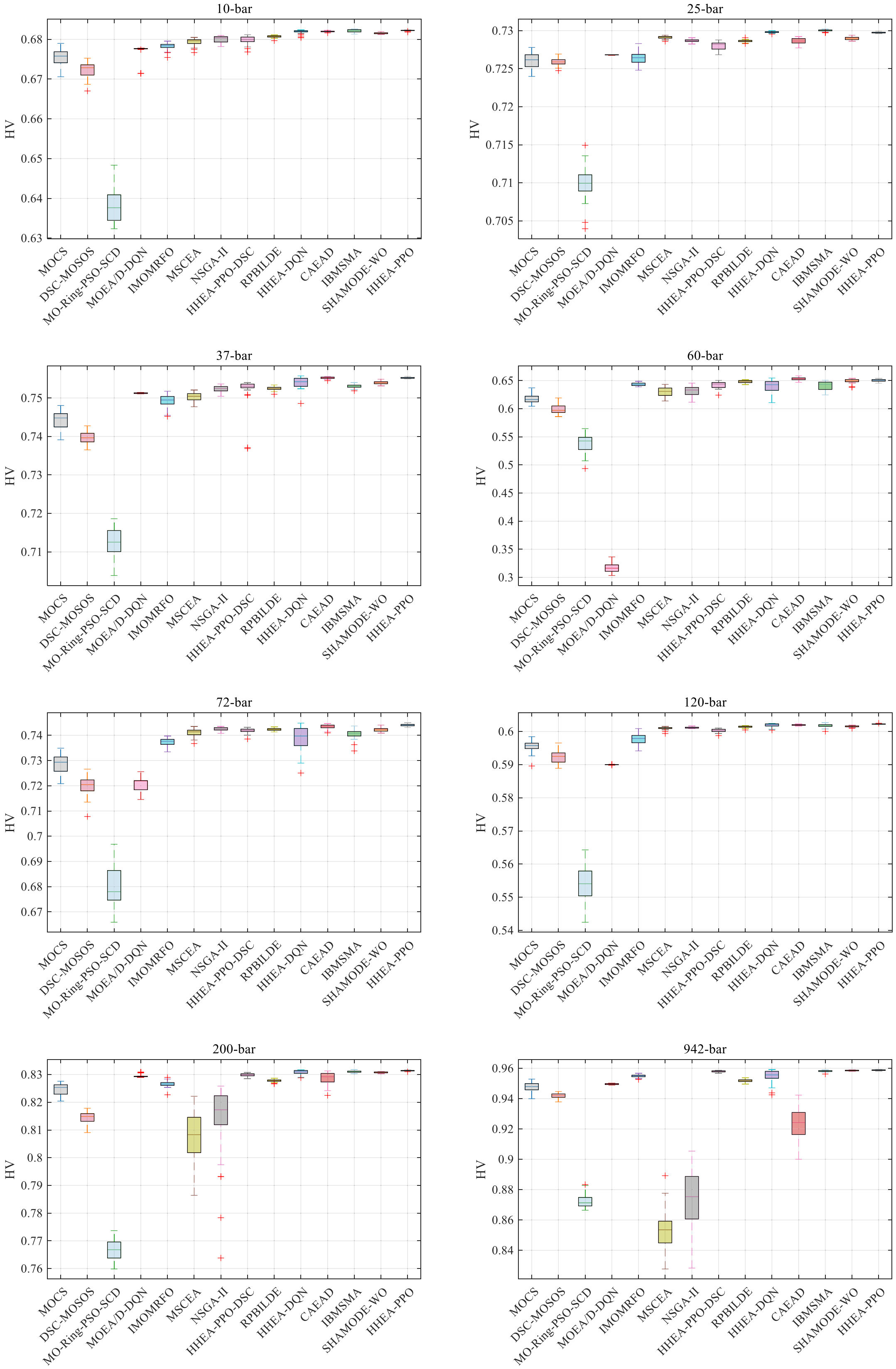


Fig. 7. HV results for 30 independent runs of all competing algorithms.

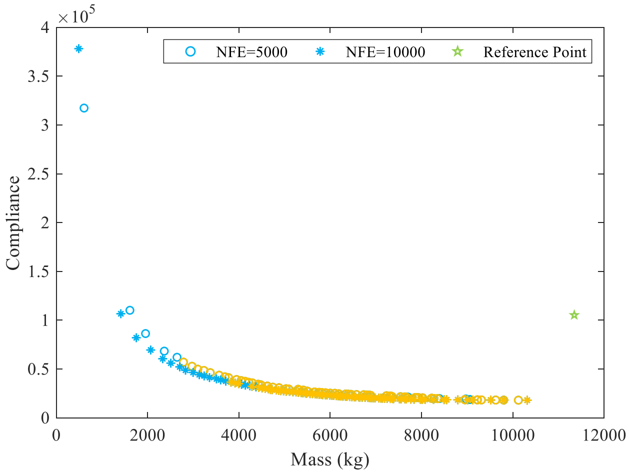


Fig. 8. Convergence analysis of MOEA/D-DQN on the 60-bar truss problem. Blue denotes individuals in the current population that do not satisfy constraints, and yellow denotes the PF composed of feasible solutions. NFE denotes the number of function evaluations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

different problems are counted in Table 10. It can be seen that SHAMODE-WO has the fewest AFEs on the 37-bar, 60-bar, 200-bar, and 942-bar problems, demonstrating its efficient optimization capability. HHEA-PPO and CAEAD have the fewest AFEs on the 25-bar, 10-bar, and 120-bar problems, respectively, and also have relatively low AFEs on other problems, further verifying their efficiency.

Based on the above analysis, HHEA-PPO performs well in SR, ACDs, and AFEs metrics, demonstrating its strong performance in truss optimization problems. IBMSMA, SHAMODE-WO, and CAEAD perform better in ACDs and AFEs, respectively. Among them, SHAMODE-WO is considered to be the most advanced algorithm for solving multi-objective truss optimization problems (Panagant et al., 2021). The value-based HHEA-DQN is comparable to SHAMODE-WO in terms of average convergence accuracy, but has lower stability. In contrast, the policy-based HHEA-PPO significantly outperforms SHAMODE-WO in terms of stability, validating the effectiveness and efficiency of the policy-based PPO-assisted HHEA.

5. Conclusion

In this study, the HHEA-PPO algorithm is proposed to improve the efficiency of HHEA in solving multi-objective truss optimization problems by integrating PPO. The HHEA-PPO algorithm improves the performance of multi-objective truss optimization through the following innovations: (1) The PPO-based HLS effectively handles continuous state spaces. (2) The integration of ten different LLHs enhances the search capability. (3) The DCD mechanism ensures the diversity and convergence of the Pareto set.

The HHEA-PPO algorithm was tested on eight truss problems and compared with thirteen state-of-the-art algorithms. The experimental results show that HHEA-PPO achieves the best HV values in six out of eight truss problems, and significantly outperforms the competing algorithms in terms of convergence accuracy and stability. According to the Friedman's average ranking, HHEA-PPO ranked first overall, with average HV values improved by 0.30 % and 0.12 % over IBMSMA and

Table 8
Success rates (%) of all competing algorithms.

Algorithm	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar	FAR
HHEA-PPO	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**1.88 (1)**
SHAMODE-WO	**100**	70	**100**	93.33	86.67	93.33	**100**	**100**	3.38 (2)
CAEAD	**100**	33.33	**100**	**100**	93.33	**100**	53.33	0	4.38 (3.5)
IBMSMA	**100**	**100**	50	50	26.67	93.33	**100**	**100**	4.38 (3.5)
HHEA-DQN	93.33	**100**	76.67	46. 67	40	83.33	93.33	73.33	5.56 (5)
RPBILDE	43.33	6.67	10	96.67	96.667	93.33	0	6.67	6.13 (6)
HHEA-PPO-DSC	10	0	53.33	56.67	76.67	0	96.67	**100**	6.69 (7)
NSGA-II	20	13.33	26.67	13.33	93.33	66.67	0	0	7.81 (8)
MSCEA	0	90	0	3.33	46.67	36.67	0	0	9.25 (9)
IMOMRFO	0	0	0	53.33	0	0	0	93.33	10.00 (10)
MOEA/D-DQN	0	0	0	0	0	0	93.33	0	10.88 (11)
DSC-MOSOS	0	0	0	0	0	0	0	0	11.56 (13)
MO-Ring-PSO-SCD	0	0	0	0	0	0	0	0	11.56 (13)
MOCS	0	0	0	0	0	0	0	0	11.56 (13)

*The best result is shown in bold, and FAR indicates the Friedman's average ranking.

Table 9
Average calculation durations (s) of all competing algorithms.

Algorithm	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar	FAR
HHEA-PPO	14.47	**8.62**	29.12	63.63	63.05	64.38	**96.16**	439.28	3.13 (2)
SHAMODE-WO	50.56	38.73	62.37	116.78	134.02	177.30	107.08	434.10	5.75 (6)
CAEAD	6.58	11.61	23.55	**50.09**	58.57	**40.60**	162.88	–	3.63 (3)
IBMSMA	**5.54**	10.36	**21.16**	54.34	54.43	72.79	102.01	**365.82**	**2.13 (1)**
HHEA-DQN	13.12	10.64	34.75	60.04	69.15	67.10	131.68	717.92	4.25 (4)
RPBILDE	317.77	348.23	347.27	235.45	339.65	446.15	–	2131.19	8.63 (8)
HHEA-PPO-DSC	59.16	–	80.88	123.86	134.73	–	219.30	743.10	8.25 (7)
NSGA-II	10.15	13.55	33.73	60.13	**47.35**	65.43	–	–	5.38 (5)
MSCEA	–	21.47	–	91.74	76.36	135.07	–	–	8.75 (9)
IMOMRFO	–	–	–	50.45	–	–	–	658.39	9.38 (10)
MOEA/D-DQN	–	–	–	–	–	–	615.69	–	11.06 (11)
DSC-MOSOS	–	–	–	–	–	–	–	–	11.56 (13)
MO-Ring-PSO-SCD	–	–	–	–	–	–	–	–	11.56 (13)
MOCS	–	–	–	–	–	–	–	–	11.56 (13)

*The best result is shown in bold, and FAR indicates the Friedman's average ranking.

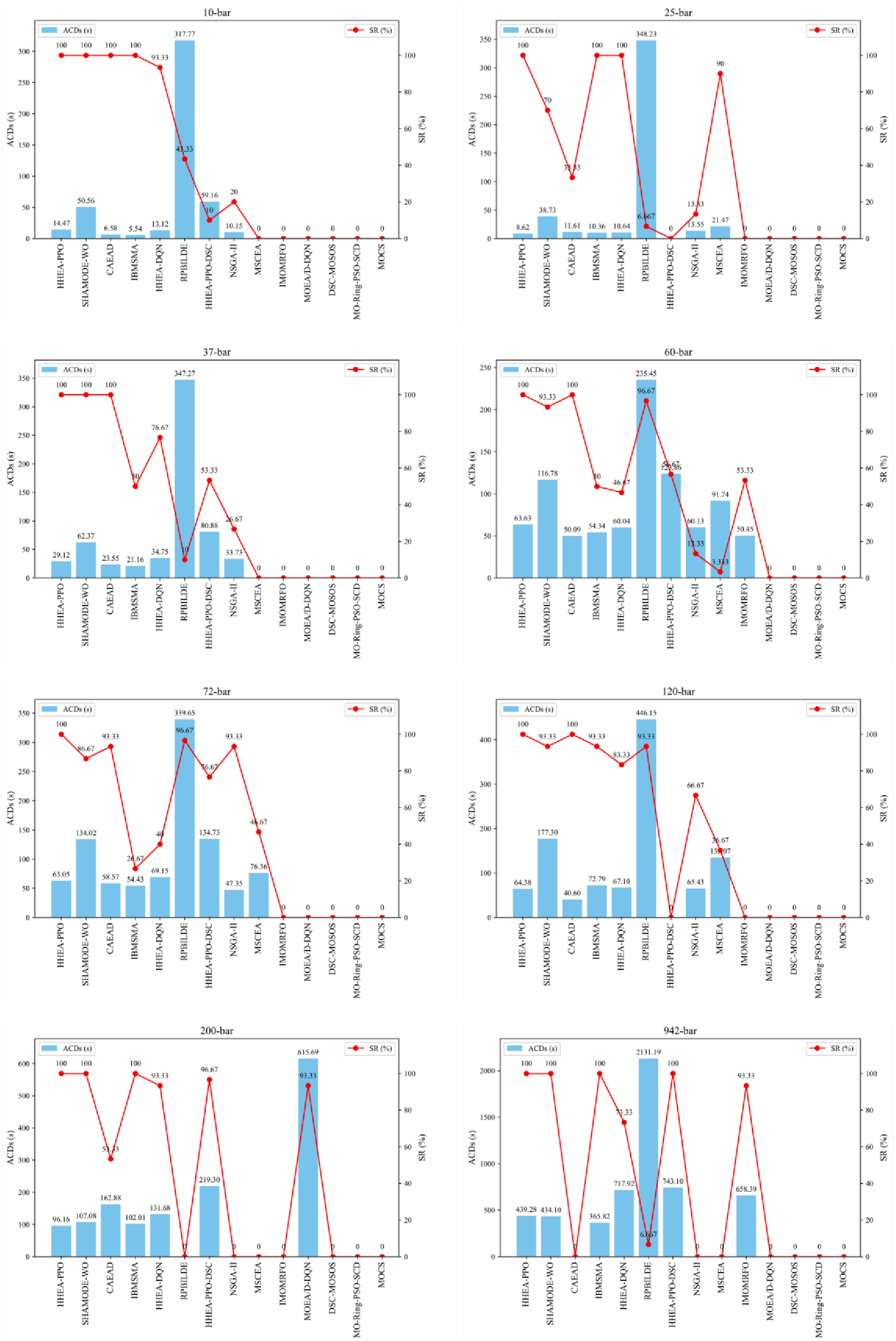


Fig. 9. Success rate and average computation duration of all competing algorithms.

Table 10
Average number of fitness evaluations of all competing algorithms.

Algorithm	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar	FAR
HHEA-PPO	18756.26	**8902.64**	23651.33	34942.99	32010.15	21042.14	22129.67	19720.55	3.00 (2)
SHAMODE-WO	15594.10	11400.44	**16006.39**	**26574.58**	31727.46	23039.99	**11931.77**	**9808.49**	**1.88 (1)**
CAEAD	**11749.39**	12784.46	23151.44	32209.03	35009.80	**13964.65**	40118.53	–	4.13 (3)
IBMSMA	15757.26	14692.84	24529.45	36922.69	33271.42	26689.49	25184.20	16569.39	4.25 (4)
HHEA-DQN	20984.51	10784.05	30801.61	34067.03	36377.71	22481.85	30244.05	32352.71	4.50 (5)
RPBILDE	37697.74	30349.08	43607.38	32474.65	37001.76	30319.02	–	48451.12	7.13 (6)
HHEA-PPO-DSC	37493.04	–	35930.07	40264.68	41892.52	–	33247.59	23765.45	7.63 (8)
NSGA-II	38819.36	22627.02	43747.37	43723.28	**30897.80**	25296.07	–	–	7.50 (7)
MSCEA	–	20229.34	–	48852.49	41487.13	42732.69	–	–	9.63 (9)
IMOMRFO	–	–	–	33535.60	–	–	–	26683.06	9.75 (10)
MOEA/D-DQN	–	–	–	–	–	–	35678.96	–	10.94 (11)
DSC-MOSOS	–	–	–	–	–	–	–	–	11.56 (13)
MO-Ring-PSO-SCD	–	–	–	–	–	–	–	–	11.56 (13)
MOCS	–	–	–	–	–	–	–	–	11.56 (13)

*The best result is shown in bold, and FAR indicates the Friedman's average ranking.

SHAMODE-WO, respectively. For the SR metric, HHEA-PPO achieved a success rate close to 100 % for all eight truss problems. Compared to SHAMODE-WO, IBMSMA, and HHEA-DQN, HHEA-PPO's SR improved by 9.07 %, 60.27 %, and 45.67 %, respectively. For the ACDs metric, despite the inclusion of the online learning process, HHEA-PPO achieved the shortest ACDs on the 25-bar and 200-bar problems, with an overall performance second only to IBMSMA. For the AFEs metric, HHEA-PPO showed the lowest AFEs on the 25-bar, 10-bar, and 120-bar problems. In addition, due to the influence of deep neural network training, HHEA-PPO is more suitable for large-scale multi-objective truss problems. On the one hand, in small-scale truss problems, PPO may learn strategies too early and easily fall into local optima. On the other hand, the long training time of HHEA-PPO leads to lower efficiency in small-scale cases.

Although HHEA-PPO performs well on most truss optimization problems, there are some weaknesses. First, HHEA-PPO incorporates an online learning process, which increases the computational duration of the algorithm to some extent, resulting in potentially longer computation times for small-scale problems. Second, HHEA-PPO may not be suitable for real-time systems because the algorithm requires training in the early stages of iteration, resulting in longer execution times (Yin & Xiang, 2024). Finally, because HHEA-PPO incorporates a deep reinforcement learning algorithm, it increases the setup parameters of the algorithm, requiring repeated adjustments to achieve optimal performance. This increases the complexity of applying the algorithm.

To further improve the performance and applicability of HHEA-PPO, future research will focus on the following areas: (1) Explore and integrate more advanced deep reinforcement learning algorithms, such as deep deterministic policy gradient (Lillicrap et al., 2019) and twin delayed deep deterministic policy gradient (Fujimoto et al., 2018), to improve the learning efficiency of the algorithm. (2) Integrate more LLHs to provide the algorithm with more exploratory and exploitative tendencies, and adapt to more diverse search space characteristics. (3) Design the hyperparameters of the evolutionary algorithm as part of the action space, thereby improving the adaptive ability of the algorithm. (4) Develop methods for offline training and online application by pre-collecting samples to train HLS, thus reducing execution time in real-time systems. (5) Apply HHEA-PPO to more real-world problems, such as topology and shape variable optimization (Anosri et al., 2022), truck-multi drone delivery system (Yılmaz et al., 2024), and economic dispatch of combined heat and power (Ozkaya, Duman, et al., 2024), to verify its generality and effectiveness.

CRediT authorship contribution statement

Shihong Yin: Investigation, Methodology, Software, Writing – original draft. **Zhengrong Xiang:** Conceptualization, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62373191.

References

Ahmed, M., & Babu, G. R. M. (2024). Hyper-heuristic multi-objective online optimization for cyber security in big data. *International Journal of System Assurance Engineering and Management, 15*(1), 314–323. https://doi.org/10.1007/s13198-022-01727-w

Almeida, C. P., Gonçalves, R. A., Venske, S., Lüders, R., & Delgado, M. (2020). Hyper-heuristics using multi-armed bandit models for multi-objective optimization. *Applied Soft Computing, 95*, Article 106520. https://doi.org/10.1016/j.asoc.2020.106520

Anosri, S., Panagant, N., Bureerat, S., & Pholdee, N. (2022). Success history based adaptive multi-objective differential evolution variants with an interval scheme for solving simultaneous topology, shape and sizing truss reliability optimisation. *Knowledge-Based Systems, 253*, Article 109533. https://doi.org/10.1016/j.knosys.2022.109533

Azizi, M., Aickelin, U., Khorshidi, H. A., & Shishehgarkhaneh, M. B. (2022). Shape and size optimization of truss structures by Chaos game optimization considering frequency constraints. *Journal of Advanced Research, 41*, 89–100. https://doi.org/10.1016/j.jare.2022.01.002

Cao, Z., Lin, C., & Zhou, M. (2021). A knowledge-based cuckoo search algorithm to schedule a flexible job shop with sequencing flexibility. *IEEE Transactions on Automation Science and Engineering, 18*(1), 56–69. https://doi.org/10.1109/TASE.2019.2945717

Carvalho, É. C. R., Carvalho, J. P. G., Bernardino, H. S., Lemonge, A. C. C., Hallak, P. H., & Vargas, D. E. C. (2024). Solving multi-objective truss structural optimization problems considering natural frequencies of vibration and automatic member grouping. *Evolutionary Intelligence, 17*(2), 653–678. https://doi.org/10.1007/s12065-022-00804-0

Carvalho, J. P. G., Carvalho, É. C. R., Vargas, D. E. C., Hallak, P. H., Lima, B. S. L. P., & Lemonge, A. C. C. (2021). Multi-objective optimum design of truss structures using differential evolution algorithms. *Computers & Structures, 252*, Article 106544. https://doi.org/10.1016/j.compstruc.2021.106544

Carvalho, J. P. G., Vargas, D. E. C., Jacob, B. P., Lima, B. S. L. P., Hallak, P. H., & Lemonge, A. C. C. (2024). Multi-objective structural optimization for the automatic member grouping of truss structures using evolutionary algorithms. *Computers & Structures, 292*, Article 107230. https://doi.org/10.1016/j.compstruc.2023.107230

David, P., Mareš, T., & Chakraborti, N. (2023). Evolutionary multi-objective optimization of truss topology for additively manufactured components. *Materials and Manufacturing Processes, 38*(15), 1922–1931. https://doi.org/10.1080/10426914.2023.2196325

Deb, K. (2011). Multi-objective optimisation using evolutionary algorithms: An introduction. In *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing* (pp. 3–34). London: Springer. https://doi.org/10.1007/978-0-85729-652-8_1.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197. https://doi.org/10.1109/4235.996017

Deb, K., Sindhya, K., & Okabe, T. (2007). Self-adaptive simulated binary crossover for real-parameter optimization. *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, 1187–1194. https://doi.org/10.1145/1276958.1277190.

Degertekin, S. O. (2012). Improved harmony search algorithms for sizing optimization of truss structures. *Computers & Structures, 92–93*, 229–241. https://doi.org/10.1016/j.compstruc.2011.10.022

Dokeroglu, T., Kucukyilmaz, T., & Talbi, E.-G. (2024). Hyper-heuristics: A survey and taxonomy. *Computers & Industrial Engineering, 187*, Article 109815. https://doi.org/10.1016/j.cie.2023.109815

Duman, S., Akbel, M., & Kahraman, H. T. (2021). Development of the multi-objective adaptive guided differential evolution and optimization of the MO-ACOPF for wind/PV/tidal energy sources. *Applied Soft Computing, 112*, Article 107814. https://doi.org/10.1016/j.asoc.2021.107814

Eid, H. F., Garcia-Hernandez, L., & Abraham, A. (2022). Spiral water cycle algorithm for solving multi-objective optimization and truss optimization problems. *Engineering with Computers, 38*(S2), 963–973. https://doi.org/10.1007/s00366-020-01237-y

Escalante, H. J., Yao, Q., Tu, W.-W., Pillay, N., Qu, R., Yu, Y., & Houlsby, N. (2021). Guest editorial: Automated machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(9), 2887–2890. https://doi.org/10.1109/TPAMI.2021.3077106

Fairclough, H., & Gilbert, M. (2020). Layout optimization of simplified trusses using mixed integer linear programming with runtime generation of constraints. *Structural and Multidisciplinary Optimization, 61*(5), 1977–1999. https://doi.org/10.1007/s00158-019-02449-7

Fujimoto, S., Hoof , H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods . Proceedings of the 35th International Conference on Machine Learning, 1587–1596. https://proceedings.mlr.press/v80/fujimoto18a.html.

Gandomi, A. H., Talatahari, S., Yang, X.-S., & Deb, S. (2013). Design optimization of truss structures using cuckoo search algorithm. *The Structural Design of Tall and Special Buildings, 22*(17), 1330–1349. https://doi.org/10.1002/tal.1033

Gholizadeh, S., & Poorhoseini, H. (2016). Seismic layout optimization of steel braced frames by an improved dolphin echolocation algorithm. *Structural and Multidisciplinary Optimization, 54*(4), 1011–1029. https://doi.org/10.1007/s00158-016-1461-y

Gomes, H. M. (2011). Truss optimization with dynamic constraints using a particle swarm algorithm. *Expert Systems with Applications, 38*(1), 957–968. https://doi.org/10.1016/j.eswa.2010.07.086

Gürgen, S., Kahraman, H. T., Aras, S., & Altın, İ. (2022). A comprehensive performance analysis of meta-heuristic optimization techniques for effective organic rankine cycle design. *Applied Thermal Engineering, 213*, Article 118687. https://doi.org/10.1016/j.applthermaleng.2022.118687

Ho-Huu, V., Duong-Gia, D., Vo-Duy, T., Le-Duc, T., & Nguyen-Thoi, T. (2018). An efficient combination of multi-objective evolutionary optimization and reliability analysis for reliability-based design optimization of truss structures. *Expert Systems with Applications, 102*, 262–272. https://doi.org/10.1016/j.eswa.2018.02.040

Ho-Huu, V., Hartjes, S., Visser, H. G., & Curran, R. (2018). An improved MOEA/D algorithm for bi-objective optimization problems with complex Pareto fronts and its application to structural optimization. *Expert Systems with Applications, 92*, 430–446. https://doi.org/10.1016/j.eswa.2017.09.051

Ho-Huu, V., Vo-Duy, T., Luu-Van, T., Le-Anh, L., & Nguyen-Thoi, T. (2016). Optimal design of truss structures with frequency constraints using improved differential evolution algorithm based on an adaptive mutation scheme. *Automation in Construction, 68*, 81–94. https://doi.org/10.1016/j.autcon.2016.05.004

Jawad, F. K. J., Ozturk, C., Dansheng, W., Mahmood, M., Al-Azzawi, O., & Al-Jemely, A. (2021). Sizing and layout optimization of truss structures with artificial bee colony algorithm. *Structures, 30*, 546–559. https://doi.org/10.1016/j.istruc.2021.01.016

Jiang, F., Wang, L., & Bai, L. (2021). An improved whale algorithm and its application in truss optimization. *Journal of Bionic Engineering, 18*(3), 721–732. https://doi.org/10.1007/s42235-021-0041-z

Kahraman, H. T., Akbel, M., & Duman, S. (2022). Optimization of optimal power flow problem using multi-objective manta ray foraging optimizer. *Applied Soft Computing, 116*, Article 108334. https://doi.org/10.1016/j.asoc.2021.108334

Kahraman, H. T., Akbel, M., Duman, S., Kati, M., & Sayan, H. H. (2022). Unified space approach-based dynamic switched crowding (DSC): A new method for designing Pareto-based multi/many-objective algorithms. *Swarm and Evolutionary Computation, 75*, Article 101196. https://doi.org/10.1016/j.swevo.2022.101196

Kaveh, A., Hamedani, K. B., & Kamalinejad, M. (2022). Improved slime mould algorithm with elitist strategy and its application to structural optimization with natural frequency constraints. *Computers & Structures, 264*, Article 106760. https://doi.org/10.1016/j.compstruc.2022.106760

Kaveh, A., & Khayatazad, M. (2013). Ray optimization for size and shape optimization of truss structures. *Computers & Structures, 117*, 82–94. https://doi.org/10.1016/j.compstruc.2012.12.010

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, *4*, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968.

Khodadadi, N., & Mirjalili, S. (2022). Truss optimization with natural frequency constraints using generalized normal distribution optimization. *Applied Intelligence, 52*(9), 10384–10397. https://doi.org/10.1007/s10489-021-03051-5

Kingma, D. P., & Ba, J. (2017). *Adam: A method for stochastic optimization* (arXiv:1412.6980). arXiv. https://doi.org/10.48550/arXiv.1412.6980.

Kumar, S., Jangir, P., Tejani, G. G., & Premkumar, M. (2022). MOTEO: A novel physics-based multiobjective thermal exchange optimization algorithm to design truss structures. *Knowledge-Based Systems, 242*, Article 108422. https://doi.org/10.1016/j.knosys.2022.108422

Kumar, S., Panagant, N., Tejani, G. G., Pholdee, N., Bureerat, S., Mashru, N., & Patel, P. (2023). A two-archive multi-objective multi-verse optimizer for truss design. *Knowledge-Based Systems, 270*, Article 110529. https://doi.org/10.1016/j.knosys.2023.110529

Kumar, S., Tejani, G. G., & Mirjalili, S. (2019). Modified symbiotic organisms search for structural optimization. *Engineering with Computers, 35*(4), 1269–1296. https://doi.org/10.1007/s00366-018-0662-y

Kumar, S., Tejani, G. G., Pholdee, N., & Bureerat, S. (2021). Multi-objective modified heat transfer search for truss optimization. *Engineering with Computers, 37*(4), 3439–3454. https://doi.org/10.1007/s00366-020-01010-1

Kumar, S., Tejani, G. G., Pholdee, N., Bureerat, S., & Mehta, P. (2021). Hybrid heat transfer search and passing vehicle search optimizer for multi-objective structural optimization. *Knowledge-Based Systems, 212*, Article 106556. https://doi.org/10.1016/j.knosys.2020.106556

Kupwiwat, C., Hayashi, K., & Ohsaki, M. (2024). Multi-objective optimization of truss structure using multi-agent reinforcement learning and graph representation. *Engineering Applications of Artificial Intelligence, 129*, Article 107594. https://doi.org/10.1016/j.engappai.2023.107594

Lamberti, L., & Pappalettere, C. (2004). Improved sequential linear programming formulation for structural weight minimization. *Computer Methods in Applied Mechanics and Engineering, 193*(33), 3493–3521. https://doi.org/10.1016/j.cma.2003.12.040

Lemonge, A. C. C., Carvalho, J. P. G., Hallak, P. H., Vargas, D., & E. c.. (2021). Multi-objective truss structural optimization considering natural frequencies of vibration and global stability. *Expert Systems with Applications, 165*, Article 113777. https://doi.org/10.1016/j.eswa.2020.113777

Li, C., Li, S., Shi, L., Zhao, Y., Zhang, S., & Wang, S. (2024). A compass-based hyper-heuristic for multi-objective optimization problems. *Swarm and Evolutionary Computation, 87*, Article 101530. https://doi.org/10.1016/j.swevo.2024.101530

Li, Z., Shi, L., Yue, C., Shang, Z., & Qu, B. (2019). Differential evolution based on reinforcement learning with fitness ranking for solving multimodal multiobjective problems. *Swarm and Evolutionary Computation, 49*, 234–244. https://doi.org/10.1016/j.swevo.2019.06.010

Liang, J., Qiao, K., Yue, C., Yu, K., Qu, B., Xu, R., Li, Z., & Hu, Y. (2021). A clustering-based differential evolution algorithm for solving multimodal multi-objective optimization problems. *Swarm and Evolutionary Computation, 60*, Article 100788. https://doi.org/10.1016/j.swevo.2020.100788

Lieu, Q. X., Do, D. T. T., & Lee, J. (2018). An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. *Computers & Structures, 195*, 99–112. https://doi.org/10.1016/j.compstruc.2017.06.016

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2019). *Continuous control with deep reinforcement learning* (arXiv:1509.02971). arXiv. https://doi.org/10.48550/arXiv.1509.02971.

Lin, Q., Lin, W., Zhu, Z., Gong, M., Li, J., & Coello, C. A. C. (2021). Multimodal multiobjective evolutionary optimization with dual clustering in decision and objective spaces. *IEEE Transactions on Evolutionary Computation, 25*(1), 130–144. https://doi.org/10.1109/TEVC.2020.3008822

Liu, J., & Xia, Y. (2022). A hybrid intelligent genetic algorithm for truss optimization based on deep neutral network. *Swarm and Evolutionary Computation, 73*, Article 101120. https://doi.org/10.1016/j.swevo.2022.101120

Liu, Y., Ishibuchi, H., Yen, G. G., Nojima, Y., & Masuyama, N. (2019). Handling imbalance between convergence and diversity in the decision space in evolutionary multi-modal multi-objective optimization. *IEEE Transactions on Evolutionary Computation, 1–1*. https://doi.org/10.1109/TEVC.2019.2938557

Luo, Q., Yin, S., Zhou, G., Meng, W., Zhao, Y., & Zhou, Y. (2023). Multi-objective equilibrium optimizer slime mould algorithm and its application in solving engineering problems. *Structural and Multidisciplinary Optimization, 66*(5), 114. https://doi.org/10.1007/s00158-023-03568-y

Nguyen-Van, S., Nguyen, K. T., Luong, V. H., Lee, S., & Lieu, Q. X. (2021). A novel hybrid differential evolution and symbiotic organisms search algorithm for size and shape optimization of truss structures under multiple frequency constraints. *Expert Systems with Applications, 184*, Article 115534. https://doi.org/10.1016/j.eswa.2021.115534

Ozkaya, B., Duman, S., Kahraman, H. T., & Guvenc, U. (2024). Optimal solution of the combined heat and power economic dispatch problem by adaptive fitness-distance balance based artificial rabbits optimization algorithm. *Expert Systems with Applications, 238*, Article 122272. https://doi.org/10.1016/j.eswa.2023.122272

Ozkaya, B., Kahraman, H. T., Duman, S., Guvenc, U., & Akbel, M. (2024). Combined heat and power economic emission dispatch using dynamic switched crowding based multi-objective symbiotic organism search algorithm. *Applied Soft Computing, 151*, Article 111106. https://doi.org/10.1016/j.asoc.2023.111106

Öztürk, H. T., & Kahraman, H. T. (2023). Meta-heuristic search algorithms in truss optimization: Research on stability and complexity analyses. *Applied Soft Computing, 145*, Article 110573. https://doi.org/10.1016/j.asoc.2023.110573

Panagant, N., & Bureerat, S. (2018). Truss topology, shape and sizing optimization by fully stressed design based on hybrid grey wolf optimization and adaptive differential evolution. *Engineering Optimization, 50*(10), 1645–1661. https://doi.org/10.1080/0305215X.2017.1417400

Panagant, N., Bureerat, S., & Tai, K. (2019). A novel self-adaptive hybrid multi-objective meta-heuristic for reliability design of trusses with simultaneous topology, shape and sizing optimisation design variables. *Structural and Multidisciplinary Optimization, 60*(5), 1937–1955. https://doi.org/10.1007/s00158-019-02302-x

Panagant, N., Pholdee, N., Bureerat, S., Yildiz, A. R., & Mirjalili, S. (2021). A comparative study of recent multi-objective metaheuristics for solving constrained truss optimisation problems. *Archives of Computational Methods in Engineering, 28*(5), 4031–4047. https://doi.org/10.1007/s11831-021-09531-8

Pham, H.-A., & Tran, T.-D. (2022). Optimal truss sizing by modified Rao algorithm combined with feasible boundary search method. *Expert Systems with Applications, 191*, Article 116337. https://doi.org/10.1016/j.eswa.2021.116337

Pierezan, J., dos Santos Coelho, L., Cocco Mariani, V., de Vasconcelos, H., Segundo, E., & Prayogo, D. (2021). Chaotic coyote algorithm applied to truss optimization problems. *Computers & Structures, 242*, Article 106353. https://doi.org/10.1016/j.compstruc.2020.106353

Poulsen, P. N., Olesen, J. F., & Baandrup, M. (2020). Truss optimization applying finite element limit analysis including global and local stability. *Structural and Multidisciplinary Optimization, 62*(1), 41–54. https://doi.org/10.1007/s00158-019-02468-4

Renkavieski, C., & Parpinelli, R. S. (2021). Meta-heuristic algorithms to truss optimization: Literature mapping and application. *Expert Systems with Applications, 182*, Article 115197. https://doi.org/10.1016/j.eswa.2021.115197

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. *Proceedings of the 32nd International Conference on Machine Learning, 37*, 1889–1897. https://proceedings.mlr.press/v37/schulman15.html.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal policy optimization algorithms* (arXiv:1707.06347). arXiv. https://doi.org/10.48550/arXiv.1707.06347.

Sonmez, M. (2011). Artificial bee colony algorithm for optimization of truss structures. *Applied Soft Computing, 11*(2), 2406–2418. https://doi.org/10.1016/j.asoc.2010.09.003

Storn, R., & Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*, 341–359.

Sun, L., & Li, K. (2020). Adaptive operator selection based on dynamic Thompson sampling for MOEA/D. In T. Bäck, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, & H. Trautmann (Eds.), *Parallel Problem Solving from Nature – PPSN XVI* (Vol. 12270, pp. 271–284). Springer International Publishing. https://doi.org/10.1007/978-3-030-58115-2_19.

Techasen, T., Wansasueb, K., Panagant, N., Pholdee, N., & Bureerat, S. (2019). Simultaneous topology, shape, and size optimization of trusses, taking account of uncertainties using multi-objective evolutionary algorithms. *Engineering with Computers, 35*(2), 721–740. https://doi.org/10.1007/s00366-018-0629-z

Tejani, G. G., Kumar, S., & Gandomi, A. H. (2021). Multi-objective heat transfer search algorithm for truss optimization. *Engineering with Computers, 37*(1), 641–662. https://doi.org/10.1007/s00366-019-00846-6

Tejani, G. G., Pholdee, N., Bureerat, S., Prayogo, D., & Gandomi, A. H. (2019). Structural optimization using multi-objective modified adaptive symbiotic organisms search. *Expert Systems with Applications, 125*, 425–441. https://doi.org/10.1016/j.eswa.2019.01.068

Tejani, G. G., Savsani, V. J., Bureerat, S., Patel, V. K., & Savsani, P. (2019). Topology optimization of truss subjected to static and dynamic constraints by integrating simulated annealing into passing vehicle search algorithms. *Engineering with Computers, 35*(2), 499–517. https://doi.org/10.1007/s00366-018-0612-8

Tejani, G. G., Savsani, V. J., & Patel, V. K. (2016). Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization. *Journal of Computational Design and Engineering, 3*(3), 226–249. https://doi.org/10.1016/j.jcde.2016.02.003

Tian, Y., Cheng, R., Zhang, X., & Jin, Y. (2017). PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [Educational Forum]. *IEEE Computational Intelligence Magazine, 12*(4), 73–87. https://doi.org/10.1109/MCI.2017.2742868

Tian, Y., Li, X., Ma, H., Zhang, X., Tan, K. C., & Jin, Y. (2023). Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization. *IEEE Transactions on Emerging Topics in Computational Intelligence, 7*(4), 1051–1064. https://doi.org/10.1109/TETCI.2022.3146882

Venske, S. M., Almeida, C. P., Lüders, R., & Delgado, M. R. (2022). Selection hyper-heuristics for the multi and many-objective quadratic assignment problem. *Computers & Operations Research, 148*, Article 105961. https://doi.org/10.1016/j.cor.2022.105961

Vo, N., Tang, H., & Lee, J. (2024). A multi-objective grey wolf-cuckoo search algorithm applied to spatial truss design optimization. *Applied Soft Computing, 155*, Article 111435. https://doi.org/10.1016/j.asoc.2024.111435

While, L., Hingston, P., Barone, L., & Huband, S. (2006). A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation, 10*(1), 29–38. https://doi.org/10.1109/TEVC.2005.851275

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82. https://doi.org/10.1109/4235.585893

Wu, S., Heidari, A. A., Zhang, S., Kuang, F., & Chen, H. (2023). Gaussian bare-bone slime mould algorithm: Performance optimization and case studies on truss structures. *Artificial Intelligence Review, 56*(9), 9051–9087. https://doi.org/10.1007/s10462-022-10370-7

Xu, N., Shi, Z., Yin, S., & Xiang, Z. (2024). A hyper-heuristic with deep Q-network for the multi-objective unmanned surface vehicles scheduling problem. *Neurocomputing, 596*, Article 127943. https://doi.org/10.1016/j.neucom.2024.127943

Yang, X.-S., & Deb, S. (2013). Multiobjective cuckoo search for design optimization. *Computers & Operations Research, 40*(6), 1616–1624. https://doi.org/10.1016/j.cor.2011.09.026

Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation, 3*(2), 82–102. https://doi.org/10.1109/4235.771163

Yi, W., Qu, R., Jiao, L., & Niu, B. (2023). Automated design of metaheuristics using reinforcement learning within a novel general search framework. *IEEE Transactions on Evolutionary Computation, 27*(4), 1072–1084. https://doi.org/10.1109/TEVC.2022.3197298

Yin, S., Luo, Q., & Zhou, Y. (2023). IBMSMA: An indicator-based multi-swarm slime mould algorithm for multi-objective truss optimization problems. *Journal of Bionic Engineering, 20*(3), 1333–1360. https://doi.org/10.1007/s42235-022-00307-9

Yin, S., & Xiang, Z. (2024). Adaptive operator selection with dueling deep Q-network for evolutionary multi-objective optimization. *Neurocomputing, 581*, Article 127491. https://doi.org/10.1016/j.neucom.2024.127491

Yılmaz, C., Cengiz, E., & Kahraman, H. T. (2024). A new evolutionary optimization algorithm with hybrid guidance mechanism for truck-multi drone delivery system. *Expert Systems with Applications, 245*, Article 123115. https://doi.org/10.1016/j.eswa.2023.123115

Yue, C., Qu, B., & Liang, J. (2018). A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems. *IEEE Transactions on Evolutionary Computation, 22*(5), 805–817. https://doi.org/10.1109/TEVC.2017.2754271

Yue, C., Suganthan, P. N., Liang, J., Qu, B., Yu, K., Zhu, Y., & Yan, L. (2021). Differential evolution using improved crowding distance for multimodal multiobjective optimization. *Swarm and Evolutionary Computation, 62*, Article 100849. https://doi.org/10.1016/j.swevo.2021.100849

Zhang, Y., Tian, Y., Jiang, H., Zhang, X., & Jin, Y. (2023). Design and analysis of helper-problem-assisted evolutionary algorithm for constrained multiobjective optimization. *Information Sciences, 648*, Article 119547. https://doi.org/10.1016/j.ins.2023.119547

Zhang, Z., Tang, Q., Chica, M., & Li, Z. (2023). Reinforcement learning-based multiobjective evolutionary algorithm for mixed-model multimanned assembly line balancing under uncertain demand. *IEEE Transactions on Cybernetics, 1–14*. https://doi.org/10.1109/TCYB.2022.3229666

Zhang, Z.-Q., Wu, F.-C., Qian, B., Hu, R., Wang, L., & Jin, H.-P. (2023). A Q-learning-based hyper-heuristic evolutionary algorithm for the distributed flexible job-shop scheduling problem with crane transportation. *Expert Systems with Applications, 234*, Article 121050. https://doi.org/10.1016/j.eswa.2023.121050

Zhao, F., Liu, Y., Zhu, N., Xu, T., & Jonrinaldi. (2023). A selection hyper-heuristic algorithm with Q-learning mechanism. *Applied Soft Computing, 147*, Article 110815. https://doi.org/10.1016/j.asoc.2023.110815

Zhao, W., Zhang, Z., Mirjalili, S., Wang, L., Khodadadi, N., & Mirjalili, S. M. (2022). An effective multi-objective artificial hummingbird algorithm with dynamic elimination-based crowding distance for solving engineering design problems. *Computer Methods in Applied Mechanics and Engineering, 398*, Article 115223. https://doi.org/10.1016/j.cma.2022.115223

Zhong, C., Li, G., Meng, Z., Li, H., & He, W. (2023). Multi-objective SHADE with manta ray foraging optimizer for structural design problems. *Applied Soft Computing, 134*, Article 110016. https://doi.org/10.1016/j.asoc.2023.110016

Zou, J., Sun, R., Yang, S., & Zheng, J. (2021). A dual-population algorithm based on alternative evolution and degeneration for solving constrained multi-objective optimization problems. *Information Sciences, 579*, 89–102. https://doi.org/10.1016/j.ins.2021.07.078