



An adaptive improved gray wolf optimization algorithm with dynamic constraint handling for mechanism-constrained optimization problems

Yanhua Lei and Mengzhen Huang

Department of Mechanical and Electrical Engineering, Zhoukou Vocational and Technical College,
Zhoukou, Henan 466000, China

Correspondence: Yanhua Lei (2008080004@zkvtc.edu.cn)

Received: 5 August 2025 – Revised: 9 September 2025 – Accepted: 17 September 2025 – Published: 3 November 2025

Abstract. To address constrained optimization problems in mechanical design, this study proposes an enhanced gray wolf optimization (GWO) algorithm. First, a novel individual memory optimization strategy is developed to expand the population's exploration scope and mitigate the risk of individuals pursuing misguided search trajectories. Second, a position update strategy incorporating differential variation is proposed to balance the local and global search capabilities of individual populations. Lastly, a discrete crossover strategy is proposed to promote information diversity across individual dimensions within the population. By integrating these three improvement strategies with the GWO, a novel improved gray wolf optimization (IGWO) algorithm is developed, which not only preserves robust global and local search capabilities but also demonstrates accelerated convergence performance. To validate the effectiveness, feasibility, and generalizability of the proposed algorithms, three representative mechanical design optimization cases and the Z3 parallel mechanism scale parameter optimization case were employed. Empirical findings reveal that the IGWO algorithm effectively resolves the targeted optimization problem, demonstrating superior performance relative to other benchmark algorithms in comparative analyses.

1 Introduction

Advancements in intelligent algorithms have facilitated the extensive application of diverse intelligent algorithms in addressing mechanical design optimization challenges. Mechanical design optimization problems are frequently formulated as constrained optimization tasks, which inherently involve both objective functions and constraints. These constraints, typically categorized into inequality constraints and equality constraints, further restrict the exploration scope of the feasible region. Consequently, constrained optimization problems typically manifest hybrid characteristics, discontinuous traits, discrete attributes, and nonlinear properties, rendering them arduous to resolve (Chen et al., 2022, Chakraborty et al., 2021). Two primary approaches for addressing constrained optimization problems are mathematical induction (Atta, 2024) and intelligent algorithms (Cheng et al., 2025). The mathematical induction-based solving ap-

proach can be characterized as a sequential exploration of the solution space, initiated from a predefined starting point and guided by gradient-derived information, progressing iteratively until either a local or global optimal state is reached. This method is plagued by two primary limitations: challenging gradient computation and a pronounced tendency to converge to local optima; furthermore, the efficacy of the solution process exhibits significant dependence on the selection of initial points. In contrast, intelligent algorithms exhibit resilience to initial point selection, with each agent evolving autonomously – a feature that endows them with advantages including structural simplicity, minimal control requirements, and robust generalization capacity. Consequently, population-based intelligent algorithms have demonstrated extensive applicability across diverse domains, such as optimal mechanism design (Ye et al., 2025), communication systems (Shuo et al., 2021), transportation planning

optimization (Tang et al., 2024), and computer science research (Wen et al., 2025b).

Intelligent algorithms, a distinct category of computational methodologies, derive inspiration from biological intelligence, natural phenomena, or human cognitive processes. Their primary objective lies in addressing intricate challenges by emulating, extrapolating, and enhancing human cognitive capacities – including learning, reasoning, optimization, and adaptation. Intelligent algorithms typically manifest adaptability, self-learning capability, parallel processing, and global search proficiency, empowering them to pinpoint near-optimal solutions amid uncertain and dynamic milieus. Therefore, intelligent algorithms are widely employed to address real-world constrained optimization problems in engineering. The primary ones include the genetic algorithm (Wang et al., 2022), differential evolutionary algorithm (Li et al., 2023), gray wolf optimization algorithm (Zeng et al., 2025), gravitational search algorithm (Rashedi et al., 2009), artificial bee colony (ABC) algorithm (Gürçan et al., 2022), simulated annealing algorithm (Subrata et al., 2017), and teaching-learning-based optimization algorithm (Zhang et al., 2017), among others. Parsopoulos and Vrahatis (2012) addressed four classical benchmark problems in constrained mechanical design optimization – specifically, the spring optimization problem, pressure vessel optimization problem, welded beam optimization problem, and wheel train optimization problem – by employing standard particle swarm optimization (PSO) algorithms. Chen et al. (2024) proposed the accelerated teaching-learning-based optimization (ATLBO) algorithm to enhance the population update speed of the original teaching-learning-based optimization (TLBO) algorithm by integrating a differential evolution (DE)-based variation strategy. They further validated the effectiveness and feasibility of this improved strategy through typical benchmark test functions, mechanical optimization case studies, and parallel mechanism optimization examples. Sobia et al. (2024) proposed a novel meta-heuristic algorithm, termed the imitation-based cognitive learning optimizer (CLO), to address mechanical design optimization problems. Additionally, three representative mechanical optimization cases and 100 benchmark test functions were selected for experimental validation. The experimental results demonstrate that the CLO algorithm outperforms 12 state-of-the-art counterparts. Mridula and Tapan (2017) employed neutrosophic optimization (NSO) to optimize welded beam problem instances and experimentally demonstrated that the NSO algorithm outperforms other iterative methods.

The gray wolf optimization (GWO) algorithm, belonging to the category of intelligent algorithms, was proposed by Mirjalili et al. (2014) and draws inspiration from the population structure and hunting behavior of gray wolves. Twenty-nine classical test functions and three classical mechanical design optimization examples were employed in the experiments. The experimental results demonstrate that the GWO algorithm outperforms classical algorithms such

as the PSO algorithm and the differential evolution algorithm. Meidani et al. (2022) proposed an adaptive GWO algorithm and verified its feasibility and validity through classical test functions. Wang et al. (2025) restructured the hierarchical architecture of the GWO, which enables direct information transmission from the alpha wolf to all subordinate wolves, thereby accelerating the population's convergence rate. Furthermore, they developed two novel learning strategies that synergistically reduce feature dimensionality while mitigating the risk of entrapment in local optima. Chen et al. (2025) incorporated an exponentially decreasing convergence factor, a per-generation elite reselection strategy, and a Cauchy mutation operator into the GWO algorithm, proposing the strengthened gray wolf optimization (SGWO). Singh et al. (2025) integrated a teaching-learning-based optimization (TLBO) algorithm into the GWO, enhancing its search capability, and proposed the hybrid GWO-TLBO algorithm.

The position update strategy of the traditional GWO algorithm is prone to falling into local optima and exhibits weak global search capability, whereas the search space accessible to individuals within the population remains relatively limited. This paper proposes an improved gray wolf optimization (IGWO) algorithm based on these analyses, which aims to address the aforementioned issues and is further applied to mechanical design optimization cases. The main contributions of this study are as follows:

1. The optimal individual memory strategy, the position update strategy incorporating differential variance, and the discrete crossover operation are integrated into the GWO algorithm to develop an improved variant, which simultaneously enhances local search capability, global search capability, and convergence speed.
2. Three classical mechanical optimization cases (welded beam optimization, spring optimization, and pressure vessel optimization), along with the optimization of scale parameters for the Z3 parallel mechanism, are employed as test functions. Experimental results demonstrate that the IGWO algorithm outperforms the other three comparator algorithms.

The remaining chapters of this paper are structured as follows: Sect. 2 presents the methodology for transforming constrained optimization problems into unconstrained optimization problems, along with the fundamental framework of the gray wolf optimization (GWO) algorithm; Sect. 3 details the improved GWO (IGWO) algorithm and its operational procedure; and Sect. 4 evaluates the performance of the IGWO algorithm through three classical mechanical design optimization cases and the dimensional parameter optimization problem of the Z3 parallel mechanism. Finally, Sect. 5 concludes the study and outlines future research directions.

2 Problem description and gray wolf optimizer

In this section, constrained optimization problems are introduced, along with methods for converting them into unconstrained formulations. The GWO algorithm for solving such problems is also presented.

2.1 Structural description

2.1.1 Constrained optimization problem

Without loss of generality, the constrained optimization problem can be formulated as

$$\begin{cases} \min f(x) \\ \text{s.t. } g_p(x) \leq 0, p = 1, 2, \dots, m \\ h_q(x) = 0, q = 1, 2, \dots, n \\ x \in S_0 \end{cases}, \quad (1)$$

where $f(x)$ denotes the objective function; x denotes the decision variable; $g_p(x)$ represents an inequality constraint; p denotes the number of inequality constraints, where $p = 1, 2, \dots, m$; $h_q(x)$ represents an equational constraint; q denotes the number of equational constraints, where $q = 1, 2, \dots, n$; S_0 denotes the search space of decision variables, where $S_0 = \{x \in R_n | x_{\min,i} \leq x \leq x_{\max,i}\}$; i denotes the decision variable in the i th dimension, where $i = 1, 2, \dots, D$; $x_{\min,i}$ denotes the minimum value of the decision variable in the i th dimension; and $x_{\max,i}$ denotes the maximum value of the decision variable in the i th dimension. The feasible domain, defined as the set of all points in the search space S for the decision variable that satisfy the constraints, is denoted as S (i.e., $S = \{x \in S_0, g_p(x) \leq 0, h_q(x) = 0\}$). Points within this feasible domain are termed “feasible solutions”. For the inequality constraint $g_p(x) \leq 0$, if there exists a point x satisfying $g_p(x) = 0$, then the point x is termed an “active constraint” of $g_p(x)$. Similarly, a point x is said to be positively bounded by $h_q(x)$ if there exists a point y satisfying $g_p(x) = 0$. Suppose that for some $x^* \in S_0$, there exists a constant $\varepsilon > 0$ such that $\forall x \in S_0 \cap \{x | \|x - x^*\| < \varepsilon\}$. Then, x^* is defined as a locally optimal solution of $f(x)$ if it satisfies $f(x) \geq f(x^*)$ for all x in a neighborhood of x^* . If the equality $f(x) \geq f(x^*)$ holds for all $x \in S_0$, then x^* is said to be a globally optimal solution of $f(x)$.

2.2 Constrained optimization problem

It is inevitable that practical engineering optimization problems are inherently subject to constraints due to limitations such as those related to the environment, space, and other factors. Several types of constraint handling methods are commonly used, including the ε -constraint technique, the Deb criterion, and the penalty function method (also referred to as the feasibility rule). Among them, the ε -constrained processing technique is a process that introduces the parameter ε

to control the weights of each objective function, aiming to find the technique’s myopic optimal solution while ensuring constraint satisfaction. The penalty function method incorporates a corresponding penalty factor into individuals that violate the constraints, thereby reducing the probability of selecting infeasible individuals. Nevertheless, the relevance between the penalty factor and the optimization problem must be carefully considered during the design process. The feasibility rule, proposed by Deb (Wen et al., 2025a), which states that feasible solutions always outperform infeasible ones and thus results in the evolutionary population never accessing infeasible solutions, is one of the most classical constraint handling methods.

In summary, the Deb criterion is selected in this paper to address the inequality constraints. Therefore, the degree to which the candidate solution x violates the inequality constraint can be defined as follows:

$$f_{ic}(x) = \sum_{i=1}^m \max\{0, g_p(x)\} \quad p = 1, 2, \dots, m, \quad (2)$$

where $g_p(x) \leq 0$ indicates that the p th inequality constraint has not been violated; the opposite is indeed violated. $f_{ic}(x)$ denotes the extent to which all inequality constraints are violated, with larger values indicating more severe violations.

In summary, Eq. (1) can be transformed into an unconstrained optimization problem by integrating the method proposed in Wen et al. (2025a) with the Deb criterion:

$$\min F(x) = -[1 - \text{sign}(f_{ic}(x))]f(x) + \text{sign}(f_{ic}(x)) \times [\Phi + f_{ic}(x)], \quad (3)$$

where $\text{sign}(\cdot)$ is the sign function and Φ is a sufficiently large constant; in this paper, we take $\Phi = 10^5$.

2.3 Gray wolf optimizer algorithm

Mirjalili et al. (2014) proposed a gray wolf optimization (GWO) algorithm inspired by the population structure and hunting behavior of gray wolves, characterized by its simple architecture and minimal control parameters. The GWO algorithm primarily comprises three core components: social structure, prey herding, and prey capturing. Its main operational steps are detailed as follows:

1. Social structure: the population was hierarchically structured into α -wolves, β -wolves, and γ -wolves based on individual fitness values. During the evolutionary process, candidate wolves updated their positions by following the guidance of α -wolves, β -wolves, and γ -wolves.
2. Rounding up prey: during the hunting process, the Euclidean distance between the n th gray wolf and the α -wolf, β -wolf, and γ -wolf needs to be calculated, which

is formulated as follows:

$$\begin{cases} D_\alpha = C_\alpha \cdot x_\alpha - x_n \\ D_\beta = C_\beta \cdot x_\beta - x_n \\ D_\gamma = C_\gamma \cdot x_\gamma - x_n \end{cases}, \quad (4)$$

where D_α , D_β , and D_γ are the Euclidean distances between candidate gray wolves and α -wolves, β -wolves, and γ -wolves, respectively, and C_α , C_β , and C_γ are the swing factors corresponding to the α -wolf, β -wolf, and γ -wolf, respectively, where each parameter represents a random value uniformly sampled from the interval $[0, 2]$. x_n denotes the position of the n th iteration in the GWO.

3. Prey hunting: gray wolves converge toward their prey, with a mathematical model describing this process formulated as follows:

$$\begin{cases} y_\alpha = x_\alpha - A_\alpha \cdot D_\alpha \\ y_\beta = x_\beta - A_\beta \cdot D_\beta \\ y_\gamma = x_\gamma - A_\gamma \cdot D_\gamma \end{cases}, \quad (5)$$

where y_α , y_β , and y_γ denote the temporary positions generated by wolves x_α , x_β , and x_γ , respectively, and A_α , A_β , and A_γ are the convergence factors for the α -wolf, β -wolf, and γ -wolf, respectively.

The new position of the current gray wolf is determined by averaging the three temporary gray wolf positions generated by Eq. (5), which is calculated as follows:

$$x_n^{t+1} = \frac{y_\alpha + y_\beta + y_\gamma}{3}. \quad (6)$$

3 Improving the gray wolf optimization algorithm

In this section, the novel individual memory optimization strategy, position update strategy incorporating differential variation, and discrete crossover strategy are integrated into the GWO framework, with the IGWO algorithm flowchart illustrated subsequently.

3.1 Novel individual memory optimization strategy

As described in the previous section, GWO algorithms utilize the positions of the α -wolf, β -wolf, and γ -wolf to guide the swarm in pursuing the optimal solution, and such algorithms can be classified as elite optimization algorithms. According to Zhang et al. (2025), the search strategy of the elite optimization algorithm enhances the exploitation of its population while diminishing the exploration of the population. The GWO algorithm, on the other hand, relies solely on the current position information of the α -wolf, β -wolf, and γ -wolf during its position update process, without leveraging the historical optimal positions of individuals within the population. This limitation may consequently reduce the search

range available to individuals in the swarm. To summarize, the historical optimal position during evolution is introduced not only to expand the population's search range but also to prevent individuals within the population from exploring in the wrong direction. Therefore, when the GWO algorithm evolves to generation t , the historical optimal position experienced by the n th gray wolf is denoted as $p_{\text{best},n}^{(t)}$, and the first three fitness function values within this historical optimal position are denoted as w_α , w_β , and w_γ . Meanwhile, a guide wolf w_g is introduced during a location update, which is closely associated with the historical optimal value of each updated individual. The guide wolf of the n th individual can be denoted as

$$x_{w_g,n}^t = \frac{2p_{\text{best},n}^t + p_{\text{best},r}^t}{3}, \quad (7)$$

where $p_{\text{best},r}^{(t)}$ is the historical optimum of a random individual.

3.2 Position update strategy incorporating differential variation

During the evolutionary process of the GWO, individuals in the population require stronger global search capability in the early stages, whereas in the later stages, they demand enhanced local search ability and accelerated convergence speed. According to Zhang et al. (2025), the position update strategy presented in Eq. (6) exhibits favorable local search capability and rapid convergence speed; however, it tends to fall into local optima and demonstrates poor global search capability. According to Wen et al. (2022), the DEr1 mutation strategy is reported to exhibit strong global search capabilities while demonstrating relatively weak local exploitation capabilities. Based on this analysis, a three-stage velocity update formulation is proposed to fully exploit the complementary advantages of the GWO position update strategy and the DEr1 mutation strategy. In the early stage of the algorithm, individuals within the population exhibit strong global search capabilities; during the mid-stage, the algorithm demonstrates effective balancing between global search and local search capabilities. By the late stage, it further develops robust local search capability accompanied by accelerated convergence speed. In summary, the new location update strategy is presented below:

$$z_n^t = \begin{cases} x_{w_g,n}^t & 0 \leq \rho < \kappa_1 \\ x_{w_g,n}^t + F(p_{\text{best},r_1}^t - p_{\text{best},r_2}^t) & \kappa_1 \leq \rho < \kappa_2 \\ p_{\text{best},r_1}^t + F(p_{\text{best},r_2}^t - p_{\text{best},r_3}^t) & \kappa_2 \leq \rho < \kappa_3 \end{cases}, \quad (8)$$

where z_n^t denotes the position of the n th gray wolf subsequent to the mutation operation; r_1 , r_2 , and r_3 denote three distinct random integers sampled from the interval $[1, \text{NP}]$;

Algorithm 1: Improved Gray Wolf Optimizer

Input:
Output:
Begin

1. Population initialization, and set $p_{best,n}^0 = x_n^0$ ($n=1, 2, \dots, NP$);
2. Ranked in ascending order, the calculated function fitness values determine the positions of w_α , w_β , w_γ , and w_δ .
3. **for** $t=1: T$ **do**
4. **for** $n=1: NP$ **do**
5. Roundups and trapping are conducted using Eqs. (4) and (5).
6. The position update is performed according to Eqs. (8) and (9), yielding the new gray wolf position x_n^{t+1} ;
7. Compute the fitness value $f(x_n^{t+1})$ of function x_n^{t+1} and update parameters $p_{best,n}^{t+1}$ and $f(p_{best,n}^{t+1})$ accordingly.
8. **End**
9. Sort $f(p_{best,n}^{t+1})$ in ascending order and update the positions of w_α , w_β , w_γ , and w_δ .
10. **End**

F denotes the scale factor; and A is a random number uniformly distributed in the interval $[0, 1]$. κ_1 and κ_2 denote random perturbation probabilities sampled uniformly from the interval $[0, 1]$. In this study, we set $\kappa_1 = 0.25$ and $\kappa_2 = 0.5$.

3.3 Discrete crossover strategy

According to Mirjalili et al. (2014), the GWO addresses high-dimensional optimization problems by representing each wolf's position as a D -dimensional coordinate. As the dimensionality increases, the wolves' search efforts in individual dimensions gradually diminish, resulting in unbalanced exploration across different dimensions. Meanwhile, the linear decreasing strategy of the traditional GWO algorithm also struggles to adapt to high-dimensional search spaces, further diminishing the wolf pack's search efficiency. According to Wen et al. (2025b), the discrete crossover strategy can integrate the advantages of parent individuals, facilitate solution evolution, maintain population diversity, avoid premature convergence, balance exploration and exploitation, and enhance optimization efficiency. To summarize, a discrete crossover operation is performed between the mutated wolf position and the historical optimal position of the current wolf to generate the next-generation wolf position. The specific expression of this operation is provided below:

$$x_n^{t+1} = \begin{cases} z_{n,d}^t & \phi < CR \\ p_{best,n,d}^{t+1} & \text{else if} \end{cases}, \quad (9)$$

where CR denotes the crossover probability; in this study, we set $CR = 0.9$. ϕ denotes a uniformly distributed random number within the closed interval $[0, 1]$. $z_{n,d}^t$ denotes the d -dimensional variable corresponding to the position of the n th gray wolf. $p_{best,n,d}^{t+1}$ is the d -dimensional variable of the historically optimal position of the n th gray wolf.

3.4 Pseudocode of the IGWO

The IGWO algorithm is developed by integrating three key strategies – the optimal individual memory strategy, the position update strategy with differential variances, and the discrete crossover operation – into the baseline GWO algorithm presented in Sect. 2.3. The pseudocode of the IGWO algorithm is presented in Algorithm 1.

4 Mechanical optimization example

In this section, the IGWO, GWO, ABC, and PSO algorithms were employed to solve three classic mechanical optimization problems and the Z3 parallel mechanism scale parameter optimization problem.

4.1 Typical examples of mechanical optimization

Case 1: welded beam optimization. The welded beam optimization problem aims to minimize the beam's fabrication cost under constraints such as shear stress, bending stress, buckling load on the rod, and end perturbations of the beam. The welded beam structure is schematically shown in Fig. 1, where the optimization variables are defined as $x = (x_1, x_2, x_3, x_4)^T = (h, l_2, a, b)^T$. The mathematical model of the problem can be expressed as follows:

$$\min f(x) = 1.1047x_1^2x_2 + 0.04811x_3x_4(14 + x_2). \quad (10)$$

The constraints are

$$\begin{cases} g_1(x) = \tau(x) - \tau_{\max} \leq 0 \\ g_2(x) = \sigma(x) - \sigma_{\max} \leq 0 \\ g_3(x) = x_1 - x_4 \leq 0 \\ g_4(x) = 0.10471x_1^2 + 0.4811x_3x_4(14.0 + x_2) - 5 \leq 0 \\ g_5(x) = 0.125 - x_1 \leq 0 \\ g_6(x) = \delta(x) - \delta_{\max} \leq 0 \\ g_7(x) = P - P_c(x) \leq 0 \end{cases}, \quad (11)$$

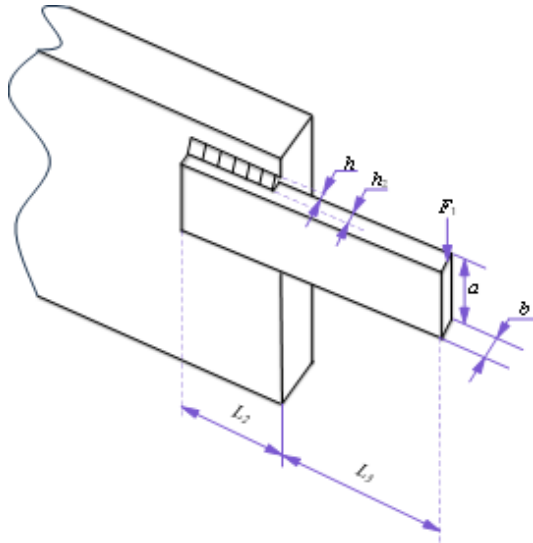


Figure 1. Optimization of welded beams.

$$\text{where } \tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \tau' = \frac{P}{\sqrt{2x_1x_2}},$$

$$\tau'' = \frac{MR}{J}, M = P(L + \frac{x_2}{2}), R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2},$$

$$J = 2 \left\{ \frac{x_1x_2}{\sqrt{2}} \left[\frac{x_2^2}{12} + (\frac{x_1+x_3}{2})^2 \right] \right\}, \delta(x) = \frac{4PL^3}{Ex_3^3x_4}, \sigma(x) = \frac{6PL}{x^4x_3^2},$$

$$\text{and } P_c(x) = \frac{4.013E\sqrt{\frac{a^2b^6}{36}}}{L_3^2} \left(1 - \frac{a}{2L_3}\sqrt{\frac{E}{4G}} \right).$$

In the formula, each parameter is expressed in imperial units, with its value or the range of values specified. $F_1 = 2761.6$ kg, $L_3 = 35.56$ cm, $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi, $\tau_{\max} = 13\,600$ psi, $\sigma_{\max} = 30\,000$ psi, and $\delta_{\max} = 0.635$ cm.

Here, the design variables have to be in the following ranges:

$$0.1 \leq x_1 \leq 2.0; 0.1 \leq x_2 \leq 10; 0.1 \leq x_3 \leq 10; 0.1 \leq x_4 \leq 2.0.$$

To verify the effectiveness and feasibility of the IGWO algorithm, the GWO, PSO, and ABC (Mridula and Tapan, 2017) algorithms are employed as comparative benchmarks.

The control parameters of each algorithm are aligned with those reported in the original literature. To ensure the fairness of experimental outcomes, the population size NP and number of iterations T for all algorithms are set as follows: NP = 100 and $T = 400$. Equations (10) and (11) are first transformed into unconstrained optimization problems based on Eq. (3) and then solved separately using five algorithms. The four algorithms were run independently and randomly 50 times, and the average evolution curve of the objective function values was obtained, as shown in Fig. 4. The statistical results from 50 independent random runs of the five algorithms for solving the welded beam optimization problem are presented in Table 1. Here, f_w , f_{av} , f_b , and f_{std} denote the worst-case, mean, optimal, and standard deviation of

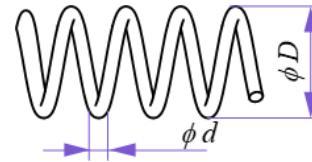


Figure 2. Schematic diagram of spring structure.

the optimal objective function values across 50 independent algorithm runs, respectively.

As shown in Table 1, the IGWO algorithm yields the optimal values for f_w , f_{av} , f_b , and f_{std} , followed by the PSO, GWO, and ABC algorithms. Based on Fig. 4, it can be observed that during the pre-evolutionary algorithm period, the GWO and PSO algorithms converge faster than the IGWO algorithm. As the algorithm progresses to its later stages, the IGWO algorithm exhibits a superior convergence rate compared to other comparative algorithms. Moreover, the convergence accuracy of the IGWO algorithm is significantly better than that of several other comparative algorithms.

Case 2: spring optimization.

The spring optimization problem can be formulated as minimizing the spring weight subject to the constraints of minimum deflection, shear stress, surge frequency, and outside diameter limits. The optimization variables include the wire diameter d , average coil diameter D , and effective number of active coils N . The design variables can be mathematically represented as $x = (x_1, x_2, x_3)^T = (d, D, N)^T$. A schematic diagram of the spring structure is illustrated in Fig. 1. The mathematical model of the problem can be expressed as follows:

$$\min f(x) = (x_3 + 2)x_2x_1^2. \quad (12)$$

The constraints are

$$\begin{cases} g_1(x) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \\ g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \\ g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \\ g_4(x) = \frac{x_2 + x_1}{1.5} - 1 \leq 0 \end{cases}, \quad (13)$$

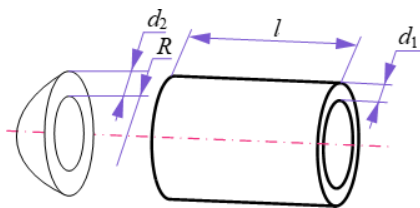
where the design variables have to be in the following ranges:

$$0.05 \leq x_1 \leq 2; 0.25 \leq x_2 \leq 1.3; 2 \leq x_3 \leq 15.$$

Based on Eq. (3), Eqs. (12) and (13) were transformed into unconstrained optimization models and solved separately using four algorithms. The control parameters of the four algorithms were consistent with those in Case 1. Each algorithm was run 50 independent times, with the average evolutionary curve of the objective function presented in Fig. 4 and the performance metrics of the four algorithms shown in Table 1. Table 1 shows that the IGWO algorithm achieves optimal values in terms of f_w , f_{av} , f_b , and f_{std} , followed by

Table 1. Statistical performance metrics of the four algorithms.

Case	Algorithm	f_w	f_b	f_{av}	f_{std}
Case 1	IGWO	2.359361	2.359361	2.359361	7.9669×10^{-9}
	GWO	2.370631	2.360022	2.362294	2.2771×10^{-3}
	ABC	3.385900	2.522356	2.917150	0.2537
	PSO	2.362543	2.359365	2.360262	9.0966×10^{-4}
Case 2	IGWO	0.0127	0.0127	0.0127	1.5756×10^{-11}
	GWO	0.0132	0.0127	0.0128	1.2581×10^{-4}
	ABC	0.0133	0.0127	0.0130	1.6406×10^{-4}
	PSO	0.0154	0.0127	0.0134	7.6588×10^{-4}
Case 3	IGWO	6.0597×10^3	6.0905×10^3	6.0622×10^3	8.4439
	GWO	6.4109×10^3	6.0599×10^3	6.0957×10^3	47.0244
	ABC	6.3900×10^3	5.9758×10^3	6.1105×10^3	95.5921
	PSO	6.8204×10^3	6.0597×10^3	6.1466×10^3	157.8196

**Figure 3.** Pressure vessel structure sketch.

the GWO, ABC, and PSO algorithms. Meanwhile, Fig. 4 illustrates that during the early evolutionary stage, the GWO algorithm exhibits faster convergence speed than the IGWO algorithm. However, in the late evolutionary stage, the IGWO algorithm demonstrates significantly superior convergence speed and convergence accuracy compared to several other benchmark algorithms.

Case 3: pressure vessel optimization. The pressure vessel constrained optimization problem is described as a problem of minimizing the cost of a pressure vessel, subject to constraints on material cost, forming cost, welding cost, and shell thickness. The design variables are the shell thickness d_1 , head thickness d_2 , inner shell radius R , and cylindrical length l . These design variables can be mathematically represented as $x = (x_1, x_2, x_3, x_4)^T = (d_1, d_2, R, l)^T$. A schematic diagram of the pressure vessel's structural configuration is shown in Fig. 1. The mathematical model of the problem can be expressed as follows:

$$\min f(x) = 0.6224x_1x_3x_4 + 1.7811x_2x_3^2 + 3.166x_1^2x_4 + 19.84x_1^2x_3. \quad (14)$$

The constraints are

$$\begin{cases} g_1(x) = -x_1 + 0.0193 \leq 0 \\ g_2(x) = -x_2 + 0.00954 \leq 0 \\ g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^2 + 129600 \leq 0 \\ g_4(x) = x_4 - 240 \leq 0 \end{cases}, \quad (15)$$

where the design variables have to be in the following ranges:

$$x_1 \geq 0; x_2 \leq 99; x_3 \geq 10; x_4 \leq 200.$$

Equations (14) and (15) were first transformed into unconstrained optimization models based on Eq. (3) and then solved individually using four distinct algorithms. The control parameters of the four algorithms were consistent with those in Case 1. Each algorithm was independently run for 50 trials, with the average evolution curve of the objective function shown in Fig. 4 and the performance indexes of the four algorithms presented in Table 1. According to Table 1, the IGWO algorithm achieves the optimal values for f_w , f_{av} , f_b , and f_{std} , followed by the GWO, ABC, and PSO algorithms. Additionally, Fig. 4 illustrates that during the early stages of the algorithms, ABC and GWO exhibit faster convergence than IGWO. Conversely, in the later stages, IGWO demonstrates significantly superior convergence speed and accuracy compared to the other evaluated algorithms.

4.2 Dimensional optimization for parallel mechanisms

An example of scale parameter optimization for a Z3 parallel mechanism, reported in Wen et al. (2025a), is selected to further validate the generality of the IGWO algorithm. The structural diagram of the Z3 parallel mechanism is shown in Fig. 5, which primarily consists of three PRS (P: prismatic, R: revolute, S: spherical) pivot chains, along with movable and fixed platforms featuring identical structures and sizes. The P sub-axis intersects the fixed platform at point A_i ,

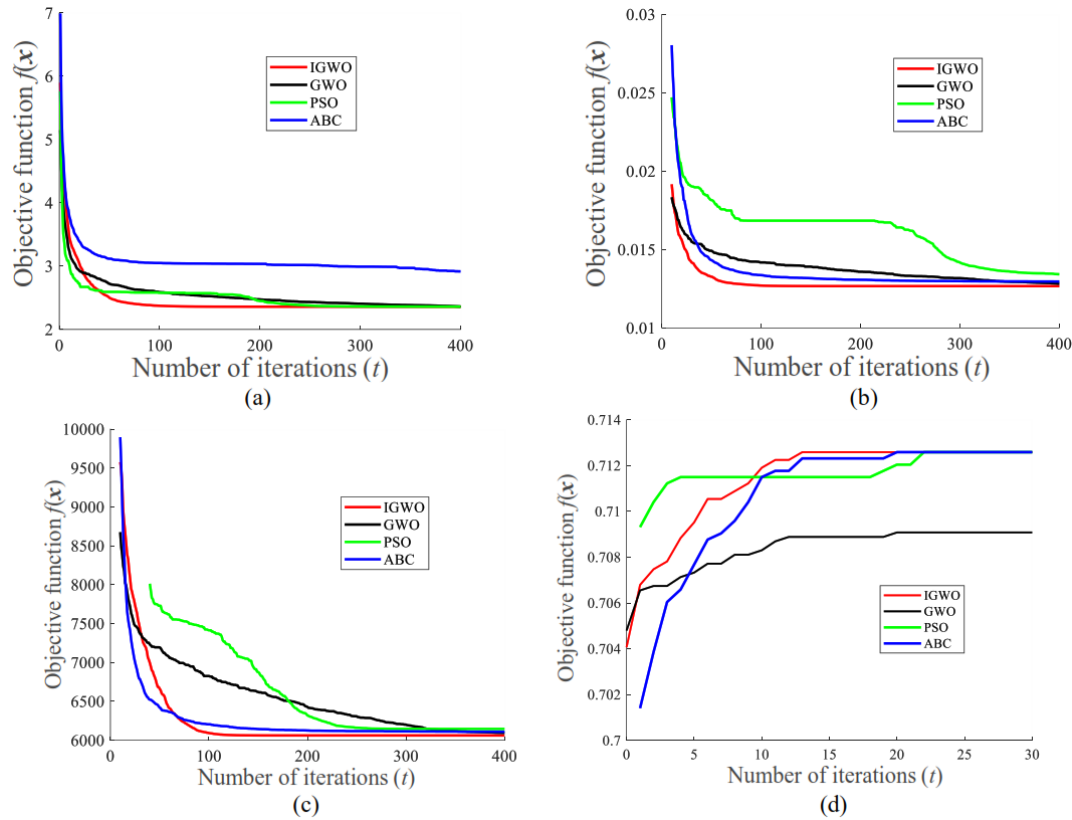


Figure 4. Average evolution curves of the algorithms. (a) Case 1, (b) case 2, (c) case 3, and (d) Z3 parallel mechanism.

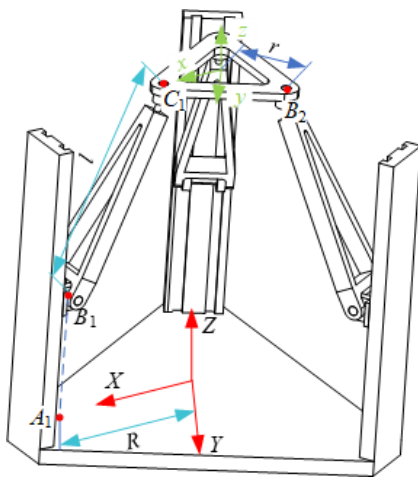


Figure 5. Z3 parallel mechanism (Wen et al., 2025a).

which lies on a circle with radius R centered at O , where the angles between A_1O , A_2O , and A_3O are each 120° . The length of the connecting rod B_iC_i is denoted by l . The center of the S-vic intersects the moving platform at point C_i , which lies on a circle with an outer radius r centered at o . Additionally, the angle among C_1P , C_2P , and C_3P is 120° .

In summary, the scale parameter optimization problem of the Z3 parallel mechanism can be succinctly formulated as follows: for a given scale parameter R , the optimal combination of the mechanism's scale parameters l and r enables the Z3 parallel mechanism to achieve an optimal effective transmission workspace. Therefore, the optimization variables in the scale parameter optimization problem for the Z3 parallel mechanism can be defined as

$$x = [x_1, x_2]^T = [r, l]^T. \quad (16)$$

According to Wen et al. (2025a), the objective function is defined as

$$\max f(x) = \max\{\kappa_{ETW}(r, l)\}, \quad (17)$$

where κ_{ETW} denotes the mathematical expression used to calculate the maximum radius of the internal tangent circle in the Z3 parallel mechanism via motion/force transfer metrics; specifically, $\kappa_{ETW} = \min\{\theta | \eta_{LTI}(\varphi, \theta) = 0.7, \forall \varphi \in [0, 2\pi]\}$.

The constraints can be expressed as follows:

$$\begin{cases} g_1(x) = 0.75 - \eta_{GTI} \leq 0 \\ g_2(x) = 0.05 - \frac{r}{l} \leq 0 \end{cases}, \quad (18)$$

where η_{GTI} denotes the average of the transfer metrics for all attitudes (φ, θ) at a fixed altitude Z_3 , which is calculated as

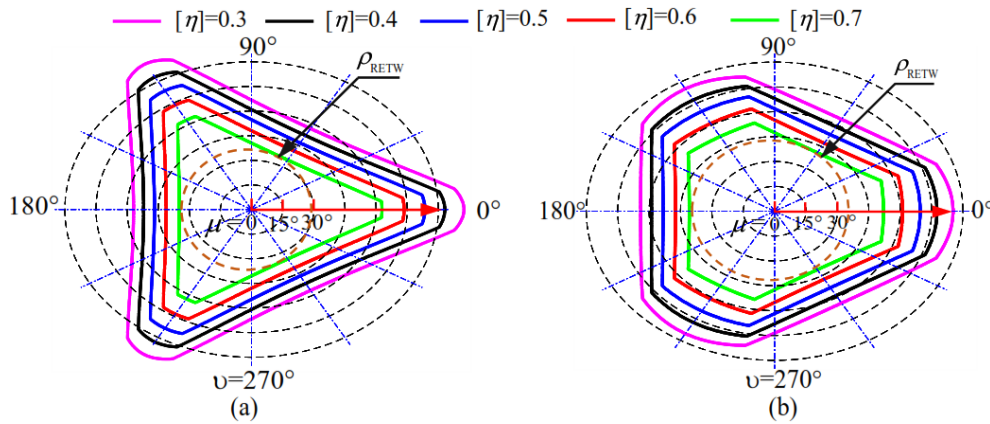


Figure 6. Performance comparison of the Z3 parallel mechanism before and after optimization. (a) Pre-optimization. (b) Optimization design.

shown below:

$$\eta_{GTI} = \frac{\int_0^{2\pi} \int_0^{\kappa_{ETW}} \eta_{LTI}(\varphi, \theta) \sin \theta d\varphi d\theta}{2\pi(1 - \cos \kappa_{ETW})} - 0.75 - \eta_{GTI} \leq 0. \quad (19)$$

Equal scaling of the mechanism's scale parameters does not affect its performance; therefore, the parameters of the Z3 parallel mechanism are set as $R = 100$ cm and $Z_P = 100$ mm. Equations (16) and (17) are transformed into the unconstrained optimization model presented in Eq. (3) and subsequently solved using four comparative algorithms – each employing identical parameter settings to those specified in Sect. 4.1. Each algorithm is run independently 50 times. The average evolution curve of the objective function is shown in Fig. 6. Meanwhile, based on the 50 optimization results obtained from the IGWO algorithm, a set of optimized mechanism scale parameters are selected as the mechanism parameters. The performance of these optimized scale parameters is then compared with that of the empirical scale parameters, and the ETW cross-section of the mechanism with fixed height is plotted, as shown in Fig. 6.

Figure 6 demonstrates that the optimized mechanism achieves significantly better performance than the pre-optimization state, aligning with the results reported by Wen et al. (2025a). As shown in Table 1, the optimization results of the IGWO, PSO, and ABC algorithms are consistent. However, Fig. 4 reveals that the IGWO algorithm exhibits a significantly faster convergence speed compared to the PSO and ABC algorithms. Meanwhile, the IGWO algorithm achieves significantly better optimization results than the GWO algorithm. However, as evidenced by Fig. 4, the GWO algorithm demonstrates a markedly faster convergence speed during the pre-evolutionary stage compared to IGWO. In the later stages of optimization, the GWO algorithm's position update mechanism tends to cause premature convergence to local optima. Conversely, the incorporation of the DE position update strategy effectively mitigates this issue, enabling the algorithm to escape local optima and converge

more efficiently toward the global optimum. In summary, the proposed novel individual memory optimization strategy, differential-variation-based position update mechanism, and discrete crossover operator integrated with the GWO algorithm collectively enhance its global exploration capability, local exploitation efficiency, and convergence performance.

5 Conclusion

To enhance the convergence speed, convergence accuracy, and ability to solve high-dimensional constrained optimization problems of the traditional GWO algorithm, an IGWO algorithm is proposed. By integrating an optimal individual memory strategy, a position update strategy with differential variance, and a discrete crossover factor into the traditional GWO algorithm, the IGWO effectively balances local search capability, global search capability, and convergence speed. To validate the performance of the IGWO algorithm, three typical mechanical design optimization problems and the scale parameter optimization problem of the Z3 parallel mechanism were employed as test cases. The experimental results demonstrate that the IGWO algorithm outperforms the PSO, GWO, and ABC algorithms.

In future research, investigations on the IGWO algorithm can be extended to multi-objective optimization. To address its core challenges – including conflicting objectives and the need to balance convergence with solution set diversity – we propose designing a composite adaptive evaluation mechanism that integrates dominance relations, distribution indicators, and convergence metrics. Additionally, optimizing the rules for dynamic social hierarchy updating and group collaboration strategies is expected to mitigate the loss of solution set diversity caused by premature convergence. In addition, effective constraint processing techniques can be further integrated with the IGWO algorithm to expand its applicability to a broader range of practical engineering scenarios.

Code and data availability. No data sets were used in this paper. The MATLAB source code developed by the authors can be obtained by contacting the corresponding author by email.

Author contributions. YL: writing the original draft, idea, methodology, validation, and investigation. MH: modeling, simulation, and data processing.

Competing interests. The contact author has declared that neither of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors. Views expressed in the text are those of the authors and do not necessarily reflect the views of the publisher.

Review statement. This paper was edited by Pengyuan Zhao and reviewed by two anonymous referees.

References

- Atta, S.: An Improved Harmony Search Algorithm Using Opposition-Based Learning and Local Search for Solving the Maximal Covering Location Problem, *Eng. Optimiz.*, 56, 1298–1317, <https://doi.org/10.1080/0305215x.2023.2244907>, 2024.
- Chakraborty, S., Saha, A., Chakraborty, R., and Saha, M.: An Enhanced Whale Optimization Algorithm for Large Scale Optimization Problems, *Knowl.-Based. Syst.*, 233, 107543, <https://doi.org/10.1016/j.knsys.2021.107543>, 2021.
- Chen, G., Tang, Q., Li, X., Wu, W., Wang, K., and Wu, K.: A generic framework for ETW-based dimensional synthesis of parallel mechanism, *Proc. Inst. Mech. Eng. C J. Mec. Eng. Sci.*, 239, 2914–2929, <https://doi.org/10.1177/09544062241302205>, 2024.
- Chen, Q., Sun, J., Palade, V., Wu, X., and Shi, X.: An improved Gaussian distribution based quantum-behaved particle swarm optimization algorithm for engineering shape design problems, *Eng. Optimiz.*, 5, 743–769, <https://doi.org/10.1080/0305215x.2021.1900154>, 2022.
- Chen, X., Ye, C., and Zhang, Y.: Strengthened grey wolf optimization algorithms for numerical optimization tasks and AutoML, *Swarm and Evolutionary Computation*, 94, 101891, <https://doi.org/10.1016/j.swevo.2025.101891>, 2025.
- Cheng, X., Guo, Z., and Ke, J.: Research on fuzzy proportional–integral–derivative (PID) control of bolt tightening torque based on particle swarm optimization (PSO), *Mech. Sci.*, 16, 209–225, <https://doi.org/10.5194/ms-16-209-2025>, 2025.
- Gürçan, Y., Burhanettin, D., and Doğan, A.: Artificial Bee Colony Algorithm with Distant Savants for constrained optimization, *Appl. Soft. Comput.*, 116, 108343, <https://doi.org/10.1016/j.asoc.2021.108343>, 2022.
- Li, J., Li, G., Wang, Z., and Cui, L.: Differential evolution with an adaptive penalty coefficient mechanism and a search history exploitation mechanism, *Expert. Syst. App.*, 23, 120530, <https://doi.org/10.1016/j.eswa.2023.120530>, 2023.
- Meidani, K., Hemmasian, A., Mirjalili, S., and Farimani, A.: Adaptive grey wolf optimizer, *Neural. Comput. Applic.*, 34, 7711–7731, <https://doi.org/10.1007/s00521-021-06885-9>, 2022.
- Mirjalili, S., Mirjalili, S. M., and Lewis, A.: Grey Wolf Optimizer, *Adv. Eng. Softw.*, 69, 46–61, <https://doi.org/10.1016/j.advengsoft.2013.12.007>, 2014.
- Mridula, S. and Tapan, K.: Optimization of Welded Beam Structure Using Neutrosophic Optimization Technique: A Comparative Study, *Int. J. Fuzzy Syst.*, 20, 847–860, <https://doi.org/10.1007/s40815-017-0362-6>, 2017.
- Parsopoulos, K. and Vrahatis, M.: Unified Particle Swarm Optimization for Solving Constrained Engineering Optimization Problems, *Lecture Notes in Computer Science*, 3612, 582–591, 2012.
- Rashedi, E., Hossein, N., and Saeid, S.: GSA: A Gravitational Search Algorithm, *Inform. Sci.*, 179, 2232–2248, <https://doi.org/10.1016/j.ins.2009.03.004>, 2009.
- Shuo, O., Dong, D., Xu, Y., and Xiao, L.: Communication optimization strategies for distributed deep neural network training: A survey, *J. Parallel Distrib. Comput.*, 149, 52–65, <https://doi.org/10.1016/j.jpdc.2020.11.005>, 2021.
- Singh, H., Saxena, S., Sharma, H., Kamboj, V., Arora, K., Joshi, G., and Cho, W.: An integrative TLBO-driven hybrid grey wolf optimizer for the efficient resolution of multi-dimensional, nonlinear engineering problems, *Scientific Reports*, 15, 11205, <https://doi.org/10.1038/s41598-025-89458-3>, 2025.
- Sobia, T., Kashif, Z., and Irfan, Y.: Imitation-based Cognitive Learning Optimizer for solving numerical and engineering optimization problems, *Cogn. Syst. Res.*, 86, 101237, <https://doi.org/10.1016/j.cogsys.2024.101237>, 2024.
- Subrata, S., Izabela, N., and Ilkyeong, M.: Optimal retailer investments in green operations and preservation technology for deteriorating items, *J. Clean. Prod.*, 140, 1514–1527, <https://doi.org/10.1016/j.jclepro.2016.09.229>, 2017.
- Tang, B., Yang, Z., Jiang, H., and Hu, Z.: Development of pedestrian collision avoidance strategy based on the fusion of Markov and social force models, *Mech. Sci.*, 15, 17–30, <https://doi.org/10.5194/ms-15-17-2024>, 2024.
- Wang, L., Yang, Z., Chen, X., Zhang, R., and Zhou, Y.: Research on adaptive speed control method of an autonomous vehicle passing a speed bump on the highway based on a genetic algorithm, *Mech. Sci.*, 13, 647–657, <https://doi.org/10.5194/ms-13-647-2022>, 2022.
- Wang, Y., Yin, Y., Zhao, H., Liu, J., Xu, C., and Dong, W.: Grey wolf optimizer with self-repulsion strategy for feature selection, *Scientific Reports*, 15, 12807, <https://doi.org/10.1038/s41598-025-97224-8>, 2025.
- Wen, S., Ji, A., Che, L., and Yang, Z.: Time-varying external archive differential evolution algorithm with applications to parallel mechanisms, *Appl. Mathmat. Model.*, 114, 745–769, <https://doi.org/10.1016/j.apm.2022.10.026>, 2022.
- Wen, S., Ji, A., Che, L., Wu, H., Wu, S., Sun, Y., Xu, Y., and Zhen, H.: Dimensional optimization of parallel mechanisms based on differential evolution, *J. Mech. Design*, 147, 124–501, <https://doi.org/10.1115/1.4068550>, 2025a.

- Wen, S., Gharbi, Y., Xu, Y., Liu, X., Sun, Y., Wu, X., Lee, H., Che, L., and Ji, A.: Dynamic neighbourhood particle swarm optimisation algorithm for solving multi-root direct kinematics in coupled parallel mechanisms, *Expert Syst. Appl.*, 268, 126–315, <https://doi.org/10.1016/j.eswa.2024.126315>, 2025b.
- Ye, H., Niu, Z., Han, D., Su, X., Chen, W., Tu, G., and Zhu, T.: Size optimization method of the Watt-II six-bar mechanism based on particle swarm optimization, *Mech. Sci.*, 16, 237–244, <https://doi.org/10.5194/ms-16-237-2025>, 2025.
- Zeng, Q., Li, X., Chen, Y., Yang, M., Liu, X., Liu, Y., and Xiu, S.: Device-Driven Service Allocation in Mobile Edge Computing with Location Prediction, *Sensors-Basel*, 25, 3025, <https://doi.org/10.3390/s25103025>, 2025.
- Zhang, K., Lin, L., Huang, X., Liu, Y., and Zhang, Y.: A Sustainable City Planning Algorithm Based on TLBO and Local Search, *IOP Conf. Ser. Mater. Sci. Eng.*, 242, 012120, <https://doi.org/10.1088/1757-899x/242/1/012120>, 2017.
- Zhang, Y., Gao, Y., Huang, L., and Xie, X.: An improved grey wolf optimizer based on attention mechanism for solving engineering design problems, *Symmetry*, 17, 50, <https://doi.org/10.3390/sym17010050>, 2025.