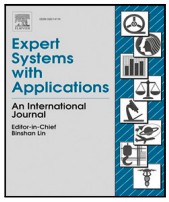
Expert Systems With Applications 233 (2023) 120905

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Great Wall Construction Algorithm: A novel meta-heuristic algorithm for engineer problems

Ziyu Guan [a], Changjiang Ren [a,*], Jingtai Niu [a], Peixi Wang [a], Yizi Shang [b]

[a] Department of Hydraulic and Ecological Engineering, Nanchang Institute of Technology, Nanchang, 330099, Jiangxi, China
[b] State Key Laboratory of Simulation and Regulation of Water Cycles in River Basins, China Institute of WaterResources and Hydropower Research, Beijing, 100038, China

ARTICLE INFO

Keywords:
Meta-heuristic
CEC-2017
Great Wall Construction
Benchmark test functions
Engineering design problems

ABSTRACT

In recent years, the optimization community has witnessed a surge in the popularity of population-based optimization methods. However, many of these methods suffer from various shortcomings, including unclear performance characteristics, incomplete validation, excessive reliance on metaphors, inadequate exploration and exploitation components, and compromised trade-offs between exploration and exploitation in real-world scenarios. As a result, users often find themselves needing to extensively modify and fine-tune these methods to achieve faster convergence, stable balance, and high-quality results. To shift the optimization community's focus towards performance rather than metaphorical changes, we propose a general population-based optimization technique called the Great Wall Construction Algorithm (GWCA). This study presents GWCA as a simple yet robust method with competitive performance for efficiently solving constrained and unconstrained problems. GWCA draws inspiration from the competition and elimination mechanisms observed among workers during the construction of the ancient Great Wall. It introduces a mathematical model of the labor movement to simulate the algorithm's dynamics. Unlike other methods that employ multiple models to generate new solutions, GWCA randomly assigns a single predefined motion model to each worker in every iteration. This unique approach showcases GWCA's dynamic nature, simple structure, high convergence performance, and ability to deliver satisfactory solution quality, thus outperforming existing optimization methods in terms of efficiency. To validate GWCA, we conduct extensive comparisons with popular and advanced algorithms on the IEEE CEC 2017 benchmark suite across different dimensions (D = 10, 30, 50, 100). Additionally, GWCA is applied to solve 16 constrained engineering problems and 6 NP-Hard problems, demonstrating its applicability in handling constrained and complex nonlinear problems. Finally, we compare GWCA's optimized solutions with those obtained from 33 advanced meta-heuristic algorithms, including the winner of CEC 2017. The results confirm the effectiveness of the proposed optimizer in solving a wide range of single-objective problems, surpassing popular base optimizers, advanced variants of existing methods, and several CEC winners. We present GWCA as an open-source population-based method that can serve as a standard optimization tool across various domains of artificial intelligence and machine learning. It exhibits a range of exploratory and exploitative features, offering high performance and optimization capabilities. The method is highly flexible, scalable, and can be further extended in terms of structure and application to accommodate diverse forms of optimization scenarios. https://github.com/guangian/Great-Wall-Construction-Algorithm-a-novel-meta-heuristic-algorithm-for-global-optimization.

1. Introduction

Over the past few decades, optimization techniques have been employed to solve challenging problems in science, engineering, economics, and business (Garip et al., 2022; Wansasueb et al., 2022; Yuan et al., 2021). Companies and institutions can improve profits, increase efficiency, and reduce time and costs through optimization. Before the emergence of approximate algorithms, scholars generally adopted traditional optimization techniques to solve optimization problems. Although these techniques have received extensive attention in various fields, they cannot obtain effective solutions because they are only

* Corresponding author.
E-mail addresses: 1909275821@qq.com (Z. Guan), 2014994517@nit.edu.cn, 971932670@qq.com (C. Ren), niujingtai@163.com (J. Niu), 1966313009@qq.com (P. Wang), Shangyzshang@foxmail.com (Y. Shang).

https://doi.org/10.1016/j.eswa.2023.120905
Received 18 September 2022; Received in revised form 22 June 2023; Accepted 22 June 2023
Available online 4 July 2023

suitable for local optimization. Besides, a deterministic algorithm, especially the gradient-based approach, cannot solve the difficulty and high computational cost required to determine the cost function derivative, restricting its applicability in solving complex real-world problems. Thus, scholars are convinced to develop approximation algorithms.

Approximation algorithms can be divided into two categories: heuristics and meta-heuristics (Ezugwu et al., 2021). Heuristic algorithms are usually designed for particular problems, such as Ant Colony Optimization (ACO) (Dorigo et al., 2006), inspired by the foraging behavior of ants in nature, which has been widely utilized in TSP and other problems. However, the efficiency of heuristic algorithms depends on initial parameters (Akbaripour & Masehian, 2013; Fathollahi-Fard et al., 2020). Moreover, they may easily fall into local optimum. Therefore, meta-heuristic algorithms are more popular than heuristic ones because they can solve almost any problem. These algorithms randomly search all solutions in the solution space and reduce the solution space size to find the optimal solution to an optimization problem (Abdel-Basset et al., 2018). Therefore, they can provide optimal solutions to complex problems with limited computing power. Meta-heuristic algorithms can be divided into five categories: Evolution-inspired, Nature-inspired, Human-inspired, and Natural-like inspired.

1. **Evolution-inspired:** Evolution-inspired algorithms establish mathematical models for the biological evolution process. They randomly generate a population, and each individual in the population corresponds to a solution. In the evolution process, individuals will experience natural selection, genetic variation, and other processes to generate new offspring (Holland, 1992). Finally, an alternative plan is applied to determine the remaining offspring and parents. The famous algorithms in this category are Genetic Algorithm (GA) (Holland, 1992), Clonal Selection Algorithm (CSA) (De Castro & Von Zuben, 2000), Evolution Strategies Algorithm (ESA) (Bäck, 1995), Evolutionary Programming Algorithm (EPA) (Zhang & Xu, 1999), Biogeography Based Optimization (BBO) (Simon, 2008) and Differential Evolution (DE) (Das & Suganthan, 2010).
2. **Nature-inspired:** Nature-inspired algorithms are inspired by the group behavior of organisms in nature. They can find goals and solutions in the problem space inspired by the collective behavior of most organisms (Dhal et al., 2019). One of the most interesting kinds of these algorithms is nature-inspired swarm intelligence (SI) inspired by biological behavior, such as animals, plants, and bacteria (Parpinelli & Lopes, 2011). One of the most famous algorithms is Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), which simulates the predation behavior of birds. Furthermore, various nature-inspired algorithms have been proposed in recent years, such as Hunger Games Search (HGS) (Yang et al., 2021a), Artificial Plant Optimization Algorithm(APOA) (Cui & Cai, 2013), Invasive Weed Optimization (IWO) (Mehrabian & Lucas, 2006), Sparrow Search Algorithm (SSA) (Xue & Shen, 2020), Red Deer Algorithm (RDA) (Fathollahi-Fard & Hajiaghaei-Keshteli, 2016), and Coronavirus Herd Immunity Optimizer (CHIO) (Al-Betar et al., 2021).
3. **Human-inspired:** Human-inspired algorithms are inspired by human behaviors. These behaviors include human and social behaviors. Human-inspired algorithms mathematically describe the behavioral strategies of humans in response to different environments. Human groups find optimal solutions through information exchange between individuals. The commonly used algorithms in this category are Teaching–Learning Based Optimization (TLBO) (Rao et al., 2011), Social-Based Algorithm (SBA) (Ramezani & Lotfi, 2013), and Social Emotional Optimization Algorithm (SEOA) (Xu et al., 2010). Recently published ones include Social Network Search (SNS) (Talatahari et al., 2021), Social Engineering Optimizer (SEO) (Fathollahi-Fard et al., 2018), Political Optimizer (PO) (Askari et al., 2020b), Ludo Game-based Swarm Intelligence (LGSI) (Singh et al., 2019), Heap-Based Optimizer (HBO) (Askari et al., 2020a),Giza Pyramids Construction (GPC) (Harifi et al., 2021) and Team Game Algorithm (TGA) (Mahmoodabadi et al., 2018).
4. **Natural-like inspired:** Scholars have employed the law of natural phenomena or nonlinear sciences, such as physics, chemistry, and mathematical laws, to solve optimization problems. One of the basic algorithms in this category is Simulated Annealing (SA) (Kirkpatrick et al., 1983), which simulates the annealing process of a solid matter. More well-known algorithms are Gravitational Search Algorithm (GSA) (Rashedi et al., 2009), Chemical Reaction Optimization (CRO) (Lam & Li, 2012) and Lightning Search Algorithm (LSA) (Shareef et al., 2015). A few recently published algorithms are Chaos Game Optimization (CGO) (Talatahari & Azizi, 2021), Student Psychology Based Optimization algorithm (SPBO) (Das et al., 2020), and Arithmetic Optimization Algorithm (AOA) (Abualigah et al., 2021a).

In summary, the following conclusions can be drawn by comparing different features and advantages of the metaheuristic algorithms:

- Metaheuristic algorithms are often inspired by some nature laws or mathematical theories (Abualigah et al., 2022b).
- Meta-heuristic algorithms have a derivation-free mechanism that treats the optimization problem as a black-box model with less dependence on mathematical conditions (Zandavi et al., 2019).
- The computational complexity of a metaheuristic algorithm determines its search rate for finding an approximate and appropriate solution (Dong & Zhou, 2016).

Meta-heuristic algorithms reach the sweet spot through two main approaches: (1) single solution and (2) population-based solutions. A single solution-based algorithm such as Tabu Search (TS) starts its process with a candidate solution and improves it during iterations. The population-based metaheuristic algorithms initialize random solutions such that the population tends toward the best promising region. Compared to single solution-based algorithms, population-based metaheuristics have two advantages: exploration and exploitation (Črepinšek et al., 2013; Ezugwu et al., 2021). Exploration provides finding promising regions within the search space. The development finds more precise solutions in promising regions to avoid getting stuck in a local optimum. Combining the two search strategies enables the algorithm to avoid premature convergence and quickly converge to the globally optimal solution (Črepinšek et al., 2013).

According to the No Free Lunch Theory (NFL) (Wolpert & Macready, 1997), no algorithm can solve all optimization problems. Hence, proposing a new algorithm is a research hotspot for researchers. This paper proposes a new meta-heuristic algorithm called GWCA. Fig. 1 shows the classification of metaheuristic algorithms. The following sections will analyze and evaluate the feasibility and reliability of the algorithm. The main contributions of this work are as follows:

1. It introduces a novel antique GWCA (Great Wall Construction Algorithm) that effectively simulates the competition and elimination mechanism among workers during the construction of the Great Wall.
2. In contrast to existing literature, the proposed GWCA is evaluated on a comprehensive set of benchmark problems, including 29 CEC-2017 benchmark functions with varying dimensions (10, 30, 50, and 100), 16 benchmark-constrained engineering design optimization problems, and 5 NP-hard problems. Additionally, a comparative analysis is conducted with 33 well-known metaheuristics, including 5 state-of-the-art (SOTA) metaheuristics, namely L-SHADE, LSHADEcnEpSin, FDB-TLABC, FDB-AEO, and FDBAGDE.

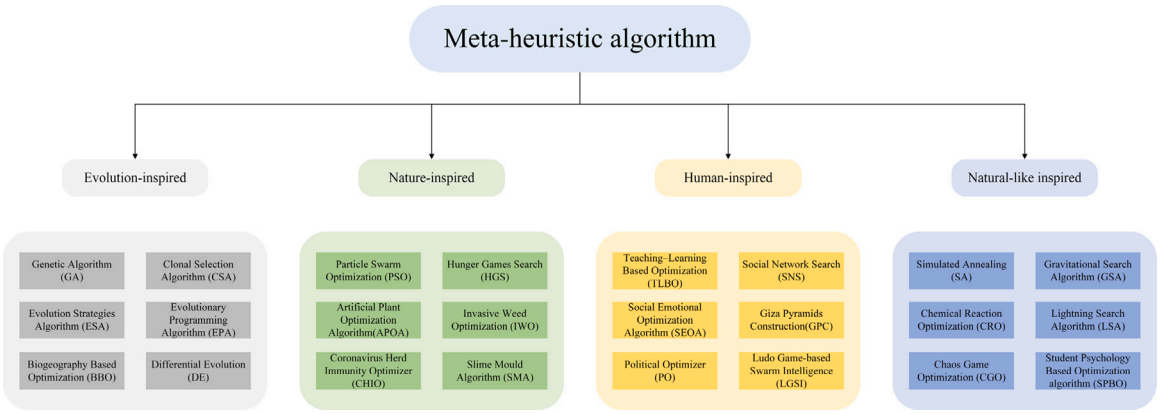


Fig. 1. The classification of meta-heuristic algorithms.

3. The GWCA is applied to dam safety detection, and a new KICA-GWCA-SVM model is proposed, which enhances the accuracy and comprehensiveness of capturing the spatio-temporal deformation pattern of the dam.

The remainder of this paper is organized as follows. Section 2 introduces the relate work. Section 3 describes the GWCA. Section 4 presented the experimental setting in this paper. The results of experiments and discussion about GWCA are presented in Section 5. Section 6 shows the results of the parametric sensitivity analysis of the GWCA. Section 7 provides the application in dam safety monitoring. Section 8 presents conclusions and prospects. Appendices A and B show mathematical models for constraint problems and engineering problems, respectively.

2. Relate work

In recent decades, there has been an exponential growth in the size and complexity of optimization problems (Zhan et al., 2022). Traditional methods, including gradient descent and Newton's method, have proven inadequate in meeting the evolving engineering demands. Consequently, there is an urgent requirement for policymakers, developers, and computer scientists to employ deterministic or approximate algorithm families within a reasonable timeframe (Kashani et al., 2022). This necessity arises from the need to identify solutions that are both feasible and explainable while offering a comprehensive level of detail for addressing diverse problem domains. Examples of such domains include closed-loop supply chain design (Mosallanezhad et al., 2021b), agri-food supply chain Gholian-Jouybari et al. (2023), rescue supply chain networks (Mosallanezhad et al., 2021a), fixed-cost transportation problems (Sadeghi-Moghaddam et al., 2019), agricultural closed-loop supply chain network (Rajabi-Kafshgar et al., 2023) and sustainable planning and decision-making (Chouhan et al., 2022).

Finding optimal solutions to multimodal rotation or combination problems without any gradient information about the objective function is a challenging task. As users increasingly emphasize the accurate estimation of optimal solutions, they selectively utilize solutions based on their level of accuracy (Jiang et al., 2017). Consequently, meta-heuristic algorithms (MAs) have garnered significant attention and found successful applications across various subfields and application scenarios in machine learning, engineering, and science. One of the primary drivers behind this trend is the rapid emergence of new real-world problems characterized by higher complexity and challenge. As these problems become more demanding, so does the demand for these solvers. MAs possess distinctive attributes, including measures to evade local optima, streamline the solution process, and operate independently of gradient information. These features empower MAs to offer satisfactory solutions for complex problems that frequently entail numerous local optima and arduous search spaces. At the core of all MAs lies their iterative exploration and exploitation strategy, which enables them to progressively approach the optimal solution by continuously exploring the search space and developing potential solutions.

Nonetheless, previous population-based optimization methods have exhibited certain gaps, issues, and limitations (Bai et al., 2021). For instance, the ACO algorithm is characterized by slow search speed (Nayar et al., 2021), while the PSO algorithm is prone to premature convergence (Gad, 2022). Similarly, the MBO algorithm often struggles to escape local optima (Feng et al., 2021), restricting its ability to explore broader search spaces. The ABC algorithm, when confronted with constraint problems and composite functions, exhibits inadequate convergence speed, thereby potentially getting trapped in local minima (Tang et al., 2021). More recently, novel approaches inspired by human activities and animal behavior have gained popularity. However, a variety of studies indicate that these methods have not received comprehensive research in their original works. Furthermore, their mathematical models suffer from structural flaws, mediocre performance, problematic verification methods, apparent similarities in structure, and slight modifications of components (Chen et al., 2020). These issues may impact the reliability analysis of solvers. Additionally, the optimization community has not adequately addressed aspects such as algorithm performance, complexity, parameter tuning, comparisons with state-of-the-art optimizers, verification using the CEC competition set, verification using constrained test sets, and the effectiveness of complex nonlinear. These considerations play a vital role for decision-makers and practitioners when tackling real-world problems (Xue et al., 2020). These controversies motivate us to conduct further research into algorithmic behavior and develop more robust frameworks, particularly considering the significant effort and modifications required to overcome the drawbacks associated with local optima and stagnation in these popular methods. Although these issues may not be readily apparent to general or inexperienced users, these methods still present inherent complexities. Hence, we aim to emphasize additional aspects of this research and draw comparisons with alternative approaches to redirect the focus of the field toward performance.

Despite the numerous metaheuristic algorithms proposed, the principle of No Free Lunch (Wolpert & Macready, 1997) in the World demonstrates that no single algorithm can universally provide optimal solutions for all optimization problems. Given that each algorithm exhibits its own strengths in specific optimization scenarios, researchers persistently endeavor to explore and develop improved algorithms. Moreover, the inspiration behind metaheuristic algorithms often stems from natural or mathematical laws (Tzanetos & Dounias, 2021; Yang, 2020), thus highlighting the need for further investigation into ancient ruins as potential sources of inspiration. Concurrently, engineers seek a straightforward and intelligent optimization algorithm capable

Table 1
Differences between GWCA and other metaheuristics.

Differences	GWCA	Other meta-heuristic algorithms
Inspiration	Ancient relics	Focus on natural and mathematical laws
Coding difficulty	Four main steps and simple parameters	Require many steps, and it is challenging to understand and encode their parameters
Convergence performance	Strong ability to jump out of local optimum, suitable for solving engineering problems	Converge too early and tend to fall into local optimum

of addressing engineering problems and providing novel solutions to intricate global optimization challenges. Therefore, this work proposes a new meta-heuristic algorithm, the Great Wall Construction Algorithm: a novel meta-heuristic algorithm for engineer problems (GWCA), which is inspired by the competition and elimination behavior of workers during the construction of the Great Wall. Table 1 shows the differences between GWCA and other metaheuristic algorithms.

3. Great wall construction algorithm

The Great Wall is a wall as the main body, with many cities, barriers, pavilions, and a combination of defense systems (Waldron, 1990). It was constructed in the Western Zhou Dynasty. Furthermore, it is the first military project in ancient China and one of the world's seven wonders (Jing, 2015). Archaeologists found that the construction method of the Great Wall will develop over time. Workers' management was a critical issue in the Great Wall construction. Due to the lack of hardware facilities and a large amount of rammed earth employed in the Great Wall (Waldron, 1990), its construction has been optimized. This section describes the construction method, the inspiration, and the proposed algorithm in detail.

3.1. The construction

Different people have different views on the Great Wall construction methods. In a most recognized method, people continuously transport rammed earth to the mountains through tools and construct defense facilities (Jing, 2015; Yang, 2018). According to historical records, Qin Shihuang employed about one million laborers while constructing the Great Wall, accounting for one-twentieth of the national population at that time (Jing, 2015). Without using tools, the construction time of the Great Wall was extended from ten years to twenty years or even longer with workers, including enslaved people, coolies, mud tilers, carpenters, and supervisors.

In addition to the project construction challenges, advanced staff management mechanisms and construction management methods are also necessary. After completing the Great Wall, the project should also consider issues such as diet, housing, wage payments for workers, and workers' work organization. In summary, the Great Wall construction process of logistical requirements is stringent.

3.2. Inspiration

As mentioned earlier, workers included soldiers, coolies, and carpenters. They were led by the emperor's agent, who was a foreman. The workers were responsible for moving their respective stones under the supervision of the emperor's agent and regularly reported the task to the emperor's agent. Each worker performs this task. Fig. 2 illustrates the effect of these factors, including simple workers and emperor agents.

Every worker on the construction site has a corresponding position. The emperor agent will arrange the workers' work according to their task report. Excellent workers can become engineers to enjoy better welfare. Thus, workers compete to obtain a higher ranking to promote welfare. The competition among workers is shown in Fig. 3. Besides, physical exhaustion during rammed earth transportation can lead to a period of rest for workers. If a worker feels tired, he (or she) will

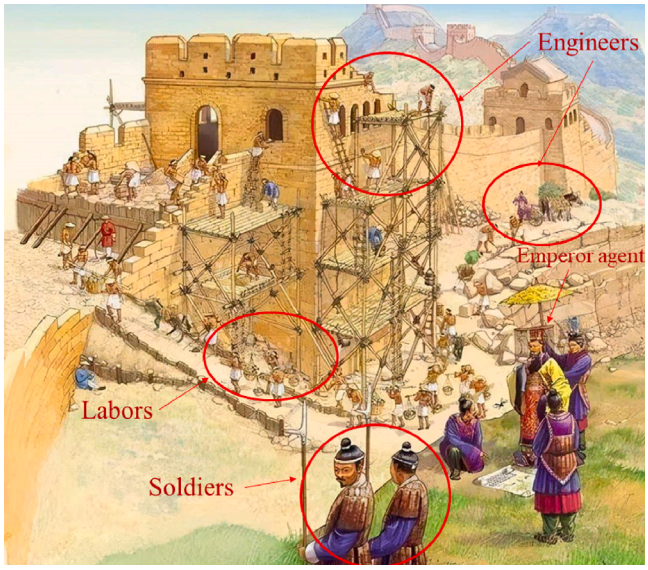


Fig. 2. Subjective perception of emperor agents and workers.

be replaced by energetic workers. New alternatives are required under the need for labor force improvement. In addition to the ranking competition, there is motivational competition among workers because everyone can obtain experience and professional knowledge. Workers created new tools during the Great Wall construction according to their acquired knowledge. These tools can help them get closer to the top of the mountain, the installation site of the Great Wall. It is worth noting that slope and terrain complexity affect workers' movement.

3.3. The proposed algorithm

Assuming rammed earth is scattered in the construction site, workers must transport rammed earth to the designated installation site. The initial position and cost of each rammed soil are known. The task reports of all workers were collected and analyzed. The workers were divided into engineers, laborers, and soldiers. In the transportation process, the ruggedness of the mountain road and the slope of the mountain road will affect the transportation speed of workers. Besides, due to the different work efficiency of each worker, the effect of accelerating the construction period can be achieved by replacing workers. Therefore, emperor agents would recruit new workers to replace inefficient workers. Algorithm 1 describes the pseudo-code of GWCA. Fig. 4 shows the flow chart of the algorithm. This algorithm has the following rules:

1. The construction site of the Great Wall is on the top of the mountain.
2. Suppose there is only one road from the bottom to the top of the mountain.
3. In the GWCA, the angle between the mountain road and the horizon is variable.
4. The solution is derived from the worker's position because the workers carry the stones.

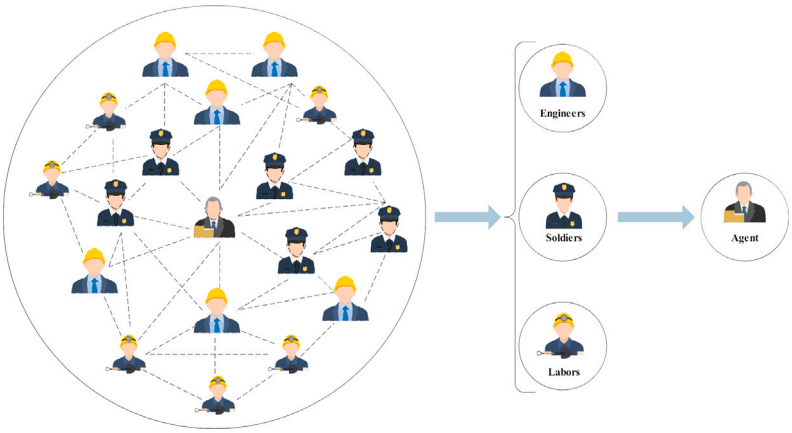


Fig. 3. Competition among workers.

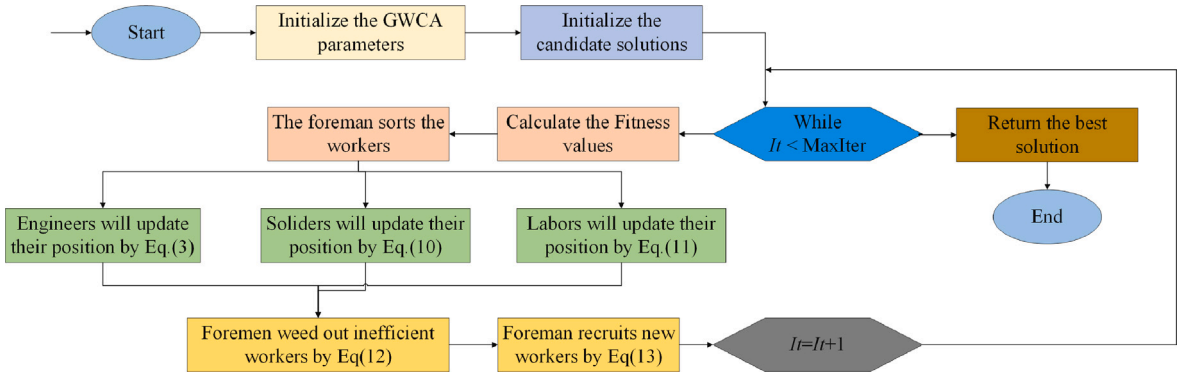


Fig. 4. Flowchart of the GWCA.

5. The mountain road complexity will affect the workers' movement speed.
6. Some workers may be changed during the construction process.

The following subsection will introduce the mathematical model of each algorithm.

Algorithm 1: The training algorithm of MACR.

Input: Number of worker N, Max Iteration $MaxIter$, Parameter P, Q.
Output: Po_{best}, $f(Po_{best})$.

```
Initialize the following parameters:
(1) Initialize the position of the worker (Po);
(2) Generate worker's task report (f(Po));
(3) The foreman sorts the workers by job based on the task report;
(4) Determine the best workers (Po_best);
for 1 → MaxIter do
    for 1 → N do
        Calculate and update the position of engineers (Eq. 3);
        Calculate and update the position of Soldiers (Eq. (10));
        Calculate and update the position of Labors (Eq. (11));
        Determine the new position of workers;
    end
    Sort solutions for the next iteration;
end
return Po_best.
```

3.4. Mathematical model and algorithm

Experienced workers tend to get the best benefit. In the construction of the Great Wall, the emperor would select the best worker from all the workers on the site as the project manager. Therefore, workers will learn from good workers and improve their own work efficiency. In addition, inefficient workers will be eliminated. This optimization process is the basis of the current algorithm. Fig. 3 shows the general model of this competition mode. Before the start of the construction period, each worker will be assigned different tasks, including engineers, soldiers, and laborers. Engineers use tools such as carts to improve their efficiency. The military supervises the worker and will go to the worker closest to him and supervise him or her. Laborers are responsible for moving heavy objects such as rocks. Finally, the project manager will weed out the inefficient workers after finishing the schedule innovation. Almost all meta-heuristics apply a set of operations to generate new solutions. The following sections will introduce the mathematical models of an engineer movement, labor movement, slave movement, and the worker elimination process.

In this subsection, the movement of workers is simulated on the slope. Assuming the worker moves along the hillside, adjust the coordinate axis to make the x-axis level on the hillside. The stress diagram of workers is shown in Fig. 5. Besides, we decompose and draw the quality force of workers.

3.4.1. Initialization phase

In this algorithm, the location of workers is considered as the solution candidates (Lac_n), comprising some elements represented as

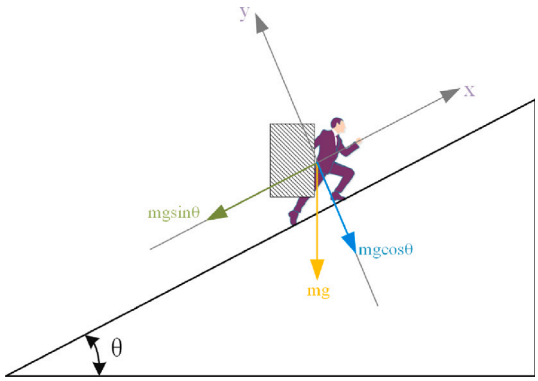


Fig. 5. The stress diagram of workers.

decision variables(Po_i^j). The mathematical representation of these two aspects is as follows:

$$Lac_n = \begin{bmatrix} Lac_1 \\ Lac_2 \\ \vdots \\ Lac_i \\ \vdots \\ Lac_n \end{bmatrix} = \begin{bmatrix} Po_1^1 & Po_1^2 & \cdots & Po_1^j & \cdots & Po_1^d \\ Po_2^1 & Po_2^2 & \cdots & Po_2^j & \cdots & Po_2^d \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ Po_i^1 & Po_i^2 & \cdots & Po_i^j & \cdots & Po_i^d \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ Po_n^1 & Po_n^2 & \cdots & Po_n^j & \cdots & Po_n^d \end{bmatrix}, \begin{cases} i = 1, 2, \ldots, n. \\ j = 1, 2, \ldots, d. \end{cases} \tag{1}$$

where d is the number of decision variables in each solution candidate, and n is the number of workers considered to be the solution candidates.

In the initial stage of optimization, Po_i^j is a chaotic sequence generated by the logistic chaotic mapping to randomly determine the values of its decision variables as defined by the optimization problem, and the initial position of Po_i^j is randomly determined in the search space as follows:

$$Po_{i+1}^j(0) = (UB_j - LB_j)cxl_j + LB_j, \begin{cases} cxl_{j+1} = 4 * cxl_j * (1 - cxl_j) \\ i = 1, 2, \ldots, n. \\ j = 1, 2, \ldots, d. \end{cases} \tag{2}$$

where $Po_i^j(0)$ determines the initial value of the location of workers; cxl is Logistic chaotic sequence; LB_j and UB_j denote the minimum and maximum values for jth the decision variable, respectively; $Rand(0, 1)$ is a random number in the interval [0, 1].

The following subsections establish a mathematical model for assigning tasks to workers. The models of engineers, laborers, and slaves are denoted by E_i, L_i, and S_i, respectively. Besides, there is competition among workers to get better welfare.

3.4.2. Mathematical model of the engineer movement (exploitation)

As mentioned earlier, the emperor's agent will select some workers as engineers. Engineers will skillfully employ tools to carry stones to improve work efficiency. Engineers compete with the most efficient workers (the best candidate solution). The following formulas are presented in this regard.

$$E_i(t+1) = E_{best}(t) + R + E_i(t)v_i(t) * R_2 \tag{3}$$

$$R = (-1)^k(X_{best}(t) - E_i(t))R_1 \tag{4}$$

where k represents an integer between 0 and 1; $v_i(t)$ denotes the speed update formula of the ith engineers in the tth iteration, and its mathematical model is as follows Eq. (5); $E_{best}(t)$ is the most efficient worker that represents the best candidate solution in the tth iteration; $E_i(t+1)$ is the position of the ith engineer in the t+1th iteration; $E_i(t)$ is the position of the ith engineer in the tth iteration; R_1 is a random number uniformly distributed in the interval [0, 1]; R_1 and R_2 provides random weights for the engineer. R is the engineer's promotion radius, which is calculated based on the difference between the efficiency of the ith worker and the engineer's leader.

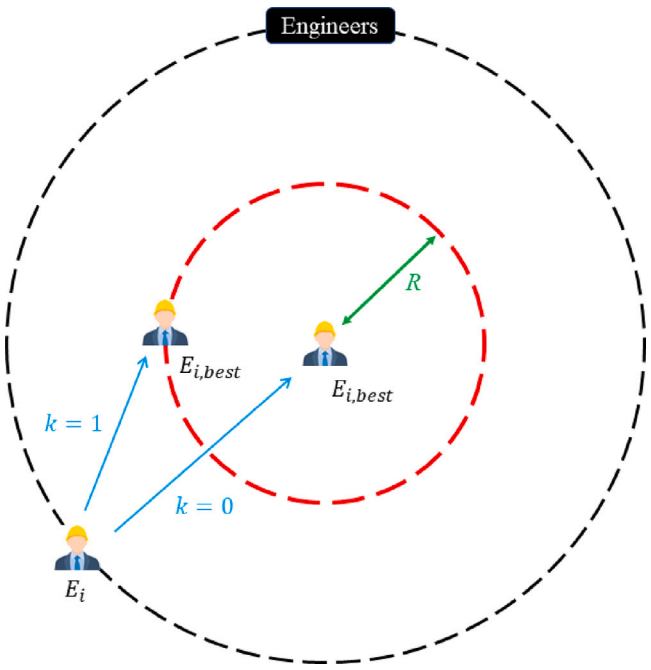


Fig. 6. Detail of engineer movement.

Moreover, the final effect of R is reflected by multiplying its value by a random integer in [−1, 1], where if the integer is positive, it advances toward the agent's position and vice versa. The engineer's position update model is shown in Fig. 6. With the operators proposed so far, the GWCA algorithm allows its search agent to update its position based on the positions of E_{best}, S_{best}, and L_{best}; and is called the leader. However, when using these operators, the GWCA algorithm is prone to local solution stagnation.

$$v_i(t) = (\frac{T \times TL}{m} - g\frac{H(t)}{sin\theta_i})C(t) \times gampdf(t \mid P, Q) \tag{5}$$

where m is the mass of the rock, g is gravitational acceleration, θ_i is a random number between [0,80] that represents the angle between the slope of the ith worker's location and the horizon, T denotes the thrust generated by the tool, *gampdf* represents the probability density function of the gamma distribution (Eq. (8)) (Yu et al., 2014), P and Q delegate two parameters in the gamma probability distribution function, and TL represents the engineer's tool wear level, which will decrease linearly from 1 to 0. $C(t)$ is the fractal dimension (Shen, 2002; Theiler, 1990) that denotes the terrain complexity of the mountain road in the tth iteration, $H(t)$ indicates the height of the ith worker from the top of the mountain in the tth iteration. The mathematical models of $H(t)$ and $C(t)$ are shown in following:

$$H(t) = 1 - \frac{t}{MaxIter} \tag{6}$$

$$C(t) = log((Cmax - Cmin)\frac{MaxIter - t}{MaxIter} + Cmin) \tag{7}$$

where t in Eqs. (6) and (7) denotes the current iteration; *MaxIter* represents the maximum number of iterations; C_{max} and C_{min} are constants.

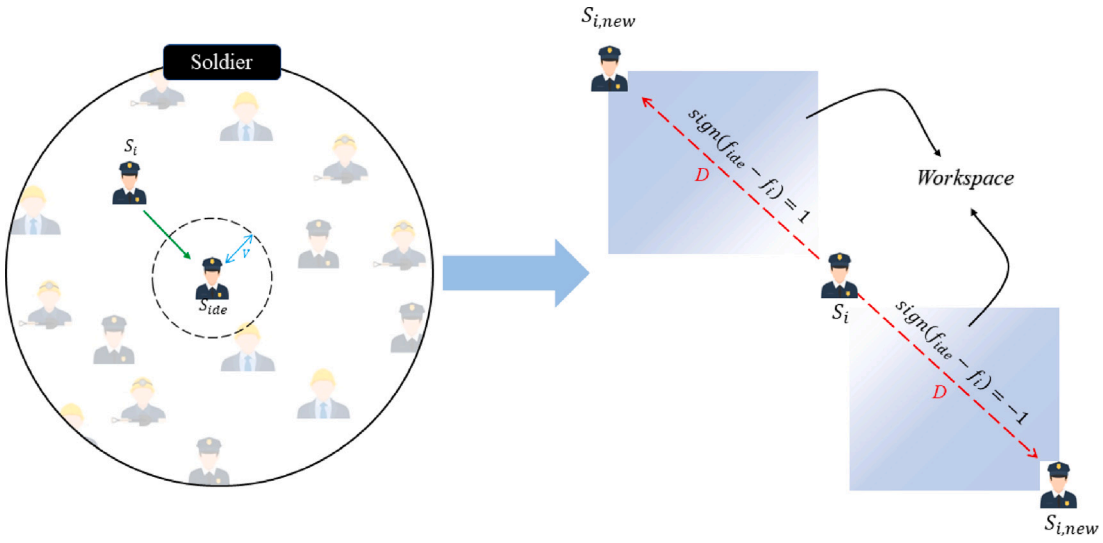


Fig. 7. Details of the soldier movement.

$$gampdf(t \mid P, Q) = \frac{Q^{-P} \times t^{P-1} \times exp(-t/Q)}{\Gamma(P)} \tag{8}$$

3.4.3. Mathematical model of soldier movement (exploration)

Eq. (9) determines the speed expression of the soldiers. Eq. (10) shows the displacement of the soldier relative to the previous position. Soldiers will go to the nearest workers and supervise their work, and their movement status is shown in Fig. 7. Workers search mainly based on the positions of E_{best}, S_{best} and L_{best}. They diverge from each other in finding the leader and converge from each other in replacing the leader. To mathematically model the process, we use the symbolic function sign to determine the solider's next position. This represents the exploration and runs the GWCA for global search. Fig. 7 also shows that the worker approaches the leader when $sign(f(ide) - f(i)) = 1$; when $sign(f(ide) - f(i)) = -1$, the solider will deviate from the leader.

It is worth mentioning that the descent process of the efficiency function v is nonlinear. Moreover, we deliberately require that v provides random values at all times in order to emphasize exploration not only during the initial iterations but also during the final iterations. This component is very useful in the case of local optimization stagnation, especially in the final iteration. Another component of GWCA that facilitates exploration is R_1, R_2, R_3, and R_4. In GWCA, R_1, R_2, R_3, and R_4 provide random weights for workers. These parameters help GWCA to exhibit a more stochastic behavior throughout the optimization process, facilitating exploration and avoiding local optimization.

$$v_i(t) = mg\frac{H(t)}{sin\theta_i}C(t) \times gampdf(t/MaxIter \mid P, Q) \tag{9}$$

$$\begin{aligned} S_i(t+1) = {} & S_i(t) + (S_{best}(t) - S_i(t)) * R_3 + sign(f(ide) - f(i)) \\ & \times (Po_{ide} - S_i(t))v_i(t) * R_4 \end{aligned} \tag{10}$$

where ide represents the soldier closest to the ith worker; S_{Best} stands for top performing soldier; $S_i(t+1)$ is the position of the ith labor in the $t+1$th iteration; $S_i(t)$ is the position of the ith labor in the tth iteration; R_3 and R_4 is a random number uniformly generated in the interval [0, 1].

3.4.4. Mathematical model of labor movement

The agent attributes all workers except engineers and soldiers to laborers. Due to their weak body, laborers could not undertake the task of transporting the rammed earth alone. labors transport rammed earth by relay. labors lined up from the bottom to the top of the mountain, passing a basket of rammed earth up so that everyone only should walk a short distance. Eq. (11) shows the mathematical model of slaves. The labor transportation process is shown in Fig. 2. The motion model of the laborer is shown in Fig. 8.

$$L_i(t+1) = L_i(t) + 2(L_{best}(t) - S_i)R_5 + (PB_i(t) - S_i) \times gampdf(t/MaxIter \mid P, Q). \tag{11}$$

where $L_i(t+1)$ is the position of the ith labor in the $t+1$th iteration; $L_i(t)$ is the position of the ith labor in the tth iteration, PB_i stands for personal best; L_{Best} stands for top performing labor.

3.4.5. Selection of worker models

Currently, many algorithms define several models to create new solutions, and each agent of the algorithm must go through all these models repeatedly. In contrast, in the GWCA algorithm, only one of the three pre-defined models, the so-called movement model, will be randomly selected and executed for each worker in each iteration of the algorithm. In other words, all the motions described here are the real behaviors of the workers during the construction process, and it seems correct to assume that these behaviors only occur at one specific time (iteration). Thus, as shown in Fig. 9, these behaviors occur by using a uniformly distributed random procedure.

In summary, the search process starts with the creation of a random population of workers (candidate solutions) in the GWCA. During the iterative process, workers estimate the possible positions of the leader based on E_{best}, S_{best} and L_{best}. Each candidate solution updates the distance between it and the leader. The random switching between the three work modes of engineer, soldier and laborer controls the exploration and development of GWCA. Finally, the GWCA algorithm is terminated by satisfying an end criterion.

3.4.6. Mathematical model for replacing and recombination

During the Great Wall construction, the works will be eliminated in proportion p, as the workers lost their ability to work or were inefficient, the p of works is 10% in GWCA, and the emperor's agents would recruit new workers to replace them. Substitute the position of the worker obtained from Eq. (3), Eq. (10), and Eq. (11) into the evaluation function, and then eliminate El workers with higher fitness. This will ensure that the construction period cannot be postponed. The mathematical model for replacing workers with agents is shown in Eq. (12).

$$EL = ceil(N * p) \tag{12}$$

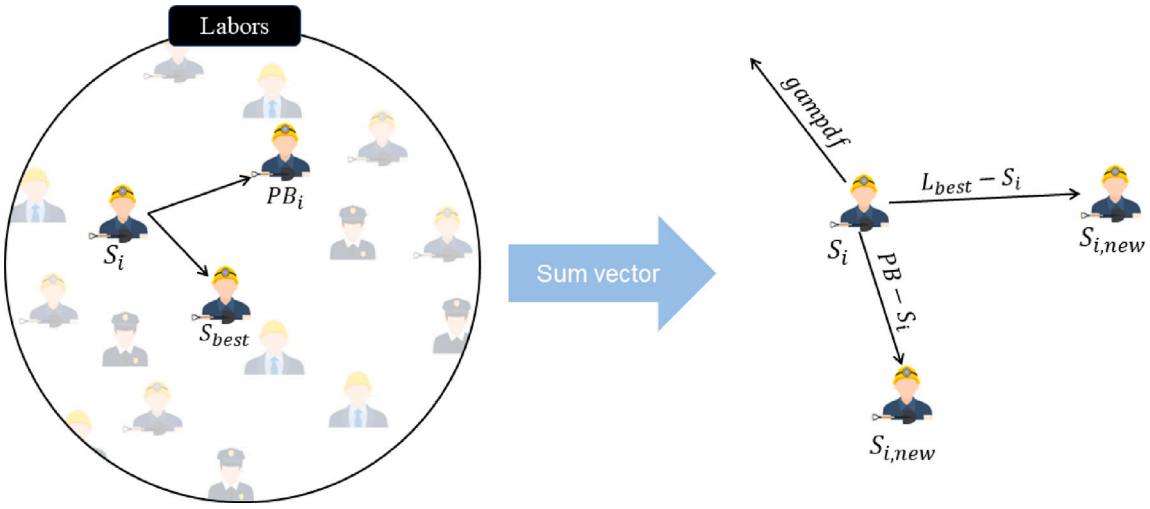


Fig. 8. Detail of Labors movement.

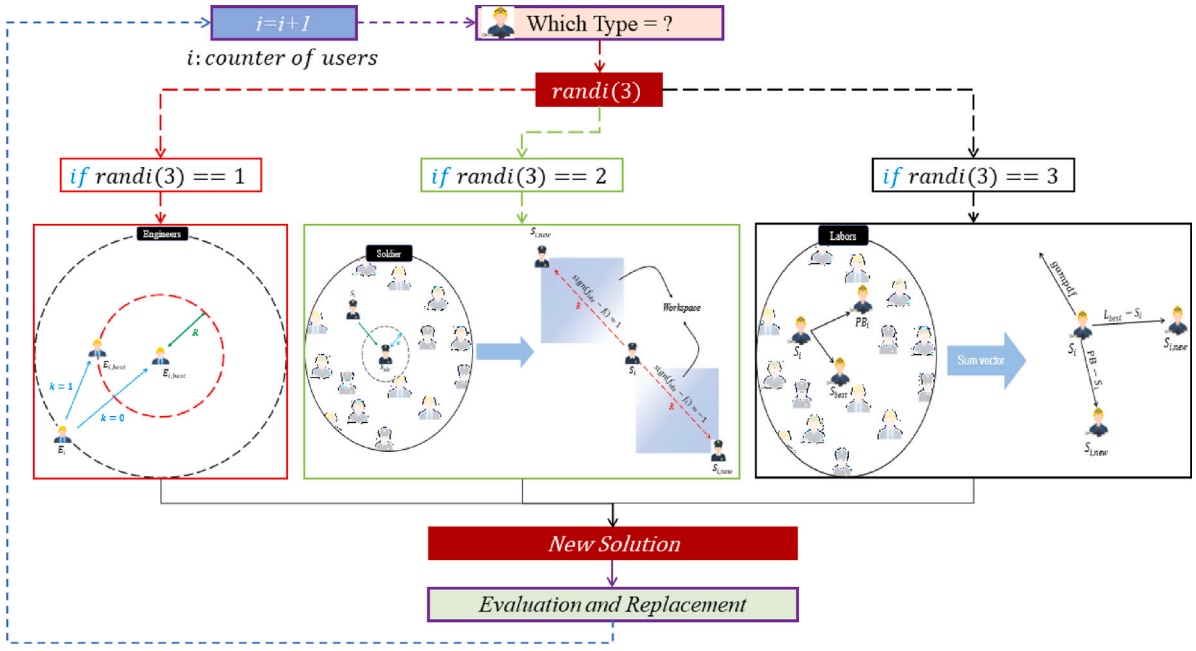


Fig. 9. Process of choosing decision movement model.

$$Po^{j}_{i,new} = (UB_j - LB_j)R_6 + LB_j, \begin{cases} i = 1, 2, \ldots, El. \\ j = 1, 2, \ldots, d. \end{cases} \tag{13}$$

$$PN = [PR, PE] \tag{14}$$

Where *ceil* is an integer function that returns the smallest integer greater than or equal to the specified expression, R_6 is a random number uniformly distributed in the interval [0, 1], PR is the position of works that is retained after elimination, PN is the position of a newly recruited worker, El represents the number of eliminated workers.

This updated rule governs the movement of the newly recruited workers towards the better positions found by the retained workers. We highlight that the effectiveness of this update is based on its ability to balance exploration and exploitation in the search space.

3.5. Computational complexity

This subsection discusses the computational complexity of the proposed GWCA. The algorithm's computational complexity depends on three factors: (1) the population initialization process, (2) the fitness function evaluation, and (3) the solution update.

1. **Population initialization process:** The complexity of the initialization process is $O(N \times D)$, where N denotes the population size, and D denotes the number of parameters (dimensions) in the test problem.
2. **Adaptation function evaluation:** The agent fitness needs $O(MaxIter \times N \times D)$ times, where MaxIter denotes the number of iterations.

3. **Solution update:** The complexity of the solution update is $O(El)$, where El represents the elimination and recruitment behavior of workers.

Therefore, the computational complexity of the proposed GWCA is $O(MaxIter \times N \times D \times El)$. The following section will employ various benchmarking functions and real optimization problems to verify and validate the performance of the proposed GWCA in solving optimization problems.

4. Experimental setup

When testing and comparing algorithm performance, ensuring fairness and compliance with standards is critical. This section describes the conditions and parameter settings for conducting the experimental study to ensure that the data obtained from the experimental study are reliable. The experimental study is based on the competition conditions and criteria defined in CEC17 and CEC2020-Realworld as references. The experimental working conditions prepared considering these criteria are given below.

- The experiment of this paper is divided into three parts: (1) 116 IEEE-2017 benchmark functions are used to test the performance of competing algorithms in search spaces with different characteristics. These functions are designed in 30, 50, and 100 dimensions to represent small, medium, and large search spaces. (2) 16 constrained engineering problems were used to test the algorithm's search performance in unknown spaces. (3) 5 NP-hard problems are used to investigate the scalability or applicability of GWCA in complex mixed integer nonlinear problems.
- To ensure fairness among competing algorithms, this paper uses the maximum number of objective function evaluations (MAXFEs) as the termination criterion for the search process. The MAXFEs were set to 10,000*D (D is the problem dimension) and 100,000 in IEEE-2017 benchmark functions and constraint problems for all algorithms, respectively. The independent times of all algorithms are 51 and 25 in IEEE-2017 benchmark functions and constraint problems, respectively.
- The nonparametric Friedman and Wilcoxon tests were applied to test the statistical validity of the experimental study. Wilcoxon's two-by-two test was performed at the 5% level of significance.
- The parameter settings and code sources in the algorithms are shown in Table 2. The experiment was performed on MATLAB R2021b version using a 64-bit Core i7 processor with 2.11 GHz and 16 GB main memory.

As shown in Table 3, there are four types of functions in the benchmark test suite. Each problem type was used to investigate and test the different capabilities of the algorithms. Each problem type was used to investigate and test the different capabilities of the algorithms. Unimodal functions were used to examine the development capability of the algorithm, and multimodal functions were used to examine the exploration capability of the algorithm. Hybrid problems were used to test the algorithm's balanced searchability. The performance of algorithms in high-complexity search spaces and their local search and global search capabilities was tested with composition functions.

In order to verify the performance of GWCA proposed in this paper, 34 competing MHS algorithms were used for experimental research (Recently proposed algorithms: Aquila Optimizer (AO) Abualigah et al., 2021b, Arithmetic Optimization Algorithm (AOA) Abualigah et al., 2021a, Artificial Electric Field Algorithm (AEFA) Anita & Yadav, 2019, Black Widow Optimization Algorithm (BWOA) Hayyolalam & Kazem, 2020, Coronavirus herd immunity optimizer (CHIO) Al-Betar et al., 2020, Dingo Optimization Algorithm (DOA) Peraza-Vazquez et al., 2021, Dynamic Differential Annealed Optimization (DDAO) Ghafil & Jármai, 2020, Giza Pyramids Construction (GPC) Harifi et al., 2021, Grey Wolf Optimizer (GWO) Mirjalili et al., 2014, Hunger Games Search (HGS) Yang et al., 2021b, Harris hawks optimization (HHO) Heidari et al., 2019, Hunter–prey optimizer (HPO) Yang et al., 2021b, Jumping Spider Optimization Algorithm (JSOA) Peraza-Vázquez et al., 2021, Moth–Flame Optimization (MFO) Shehab et al., 2020, Pelican Optimization Algorithm (POA) Trojovskỳ & Dehghani, 2022, Pathfinder Algorithm (PFA) Yapici & Cetinkaya, 2019, Runge Kutta optimization (RUN) Ahmadianfar et al., 2021, Reptile Search Algorithm (RSA) Abualigah et al., 2022a, Rat Swarm Optimizer (RSO) Dhiman et al., 2021, Sine Cosine Algorithm (SCA) Mirjalili, 2016, Sailfish Optimization Algorithm (SFO) Shadravan et al., 2019, Skill optimization algorithm (SOA) Ramshanker & Chakraborty, 2022, Supply-Demand-Based Optimization (SDO) Zhao et al., 2019, Spotted Hyena Optimizer (SHO) Dhiman & Kumar, 2017, Sparrow Search algorithm (SSA) Xue & Shen, 2020, Tunicate Swarm Algorithm (TSA) Kaur et al., 2020, Transient search algorithm (TSO) Qais et al., 2020, War Strategy Optimization Algorithm (WSOA) Ayyarao et al., 2022; State-of-the-art algorithm: LSHADE Piotrowski, 2018, LSHADEcnEpSin Awad et al., 2017, FDB-AGDE Guvenc et al., 2021, FDB-TLABC Duman et al., 2022, LRFDB-COA Duman et al., 2021). The parameter Settings of all algorithms used in the experimental study are given in Table 2. The algorithms used in this paper are all from their original authors.

5. Results and analysis

This section presents a statistical analysis of the data obtained from three experimental studies.

- The first experimental study tests and analyzes the search performance of GWCA developed in this study using the benchmark function. The results of the first experimental study are given in Section 5.1. (Comparison of GWCA and MHS algorithms to IEEE-2017 benchmark test functions)
- The second experiment verifies the performance of GWCA in constraint problems and design problems. The purpose is to study GWCA's performance and efficiency in exploring unknown spaces. The results of the second experimental study are given in Section 5.2. (Comparison of GWCA and MHS algorithms to engineer problems)
- Finally, GWCA is used to solve a complex mixed integer nonlinear problem (Grid map - robot routing). The results are compared with those of other state-the-of-art optimization algorithms in the literature. The results of the third experimental study are presented in Section 5.3. (Comparison of GWCA and MHS algorithms to NP-Hard problems)

5.1. Comparison of GWCA and MHS algorithms to benchmark test functions

In this section, we test the performance of GWCA and compare it with 33 of the latest and most powerful competing algorithms in the literature. 29 benchmark test problems from the CEC17 benchmark test suites were used for the comparison. In addition, we analyze the data from the experimental study using Friedman and Wilcoxon tests. The results of the analysis are presented in the following sections.

5.1.1. Evaluation of the CEC-2017 benchmark test functions

In experimental studies with more than two competitors, the Friedman test is one of the most commonly used statistical methods based on performance comparison algorithms. In this study, 34 competing algorithms for search performance, including L-SHADE, and LSHADEcnEpSinh (these are the winning algorithms for CEC 2017), were analyzed using the Friedman test. Thus, four experiments were performed for 29 problems and D = 10, 30, 50, and 100 dimensions in the CEC 2017 benchmark suite. The Friedman rankings obtained by the

Table 2
The parameter settings of all algorithms.

Algorithms	Parameters setting	Reference
GWCA	P = 9; Q = 6; T = 8.3; m = 3; C_{max} = exp(3); C_{min} = exp(2)	
FDB-AGDE	None	Guvenc et al. (2021)
FDB-TLABC	limit = 200; CR = 0.5;	Duman et al. (2022)
LRFDB-COA	n_{coy} = 5; n_{packs} = 20;	Duman et al. (2021)
L-SHADE	p-best rate = 0.11; arc_{rate} = 1.4; memory size = 5	Piotrowski (2018)
LSHADEcnEpSin	p_b = 0.4; p_s = 0.5	Awad et al. (2017)
Arithmetic Optimization Algorithm (AOA)	MOP_{Max} = 1; MOP_{Min} = 0.2; Alpha = 5; Mu = 0.499;	Abualigah et al. (2021a)
Aquila Optimizer (AO)	u = .0265; r0 = 10; omega = .005; alpha = 0.1; delta = 0.1; phi0 = 3*pi/2;	Abualigah et al. (2021b)
Artificial Electric Field Algorithm (AEFA)	Rnorm = 2; tag = 1; FCheck = 1; Rpower = 1;	Anita and Yadav (2019)
Black Widow Optimization Algorithm (BWOA)	$-1 < \beta < 1$; $0.4 <$ m < 0.9	Hayyolalam and Kazem (2020)
Coronavirus herd immunity optimizer (CHIO)	SpreadingRate = 0.05; MaxAge = 100; C0 = 1;	Al-Betar et al. (2020)
Dingo Optimization Algorithm (DOA)	$-2 <$ beta < 2	Peraza-Vazquez et al. (2021)
Dynamic Differential Annealed Optimization (DDAO)	MaxSubIt = 1000; T0 = 2000; ALPHA = 0.995	Ghafil and Jármai (2020)
Giza Pyramids Construction (GPC)	G = 9.8; Tetha = 14; Mu_{Min} = 1; Mu_{Max} = 10; pSS = 0.5	Harifi et al. (2021)
Grey Wolf Optimizer (GWO)	None	Mirjalili et al. (2014)
Hunger Games Search (HGS)	VC2 = 0.03;	Yang et al. (2021b)
HarrisHawk Optimization (HHO)	$-1 <$ E0 < 1	Heidari et al. (2019)
Hunter–prey optimizer (HPO)	B = 0.1;	Yang et al. (2021b)
Jumping Spider Optimization Algorithm (JSOA)	gravity = 9.80665; v_o = 100	Peraza-Vázquez et al. (2021)
Moth–Flame Optimization (MFO)	None	Shehab et al. (2020)
Pelican Optimization Algorithm (POA)	None	Trojovskỳ and Dehghani (2022)
Pathfinder Algorithm (PFA)	None	Yapici and Cetinkaya (2019)
Runge Kutta optimization (RUN)	None	Ahmadianfar et al. (2021)
Reptile Search Algorithm (RSA)	Alpha = 0.1; Beta = 0.005	Abualigah et al. (2022a)
Rat Swarm Optimizer (RSO)	x = 1; y = 5;	Dhiman et al. (2021)
Sine Cosine Algorithm (SCA)	a = 2;	Mirjalili (2016)
Sailfish Optimization Algorithm (SFO)	A = 4; e = 0.001; SFpercent = 0.3;	Shadravan et al. (2019)
Skill optimization algorithm (SOA)	None	Ramshanker and Chakraborty (2022)
Supply-Demand-Based Optimization (SDO)	None	Zhao et al. (2019)
Spotted Hyena Optimizer (SHO)	None	Dhiman and Kumar (2017)
Sparrow Search algorithm (SSA)	None	Xue and Shen (2020)
Tunicate Swarm Algorithm (TSA)	P_{min} = 1; P_{max} = 4	Kaur et al. (2020)
Transient search algorithm (TSO)	K = 1	Qais et al. (2020)
War Strategy Optimization Algorithm (WSOA)	R = 0.8;	Ayyarao et al. (2022)

Table 3
Descriptions of CEC-2017 benchmark test functions.

No.	Functions	Properties	Search range	Dim	f_{min}
Unimodal functions					
F1	Shifted and Rotated Bent Cigar Function	N, S	[−100, 100]	10	100
F2	Shifted and Rotated Zakharov Function	U, N	[−100, 100]	10	300
Simple multimodal functions					
F3	Shifted and Rotated Rosenbrock's Function	M, N, L	[−100, 100]	10	400
F4	Shifted and Rotated Rastrigin's Function	M, N, L	[−100, 100]	10	500
F5	Shifted and Rotated Expanded Scaffer's F6 Function	M, N, A, L	[−100, 100]	10	600
F6	Shifted and Rotated Lunacek Bi_Rastrigin Function	M, N, A, C	[−100, 100]	10	700
F7	Shifted and Rotated Non-Continuous Rastrigin's Function	M, N, A, L	[−100, 100]	10	800
F8	Shifted and Rotated Levy Function	M, N, L	[−100, 100]	10	900
F9	Shifted and Rotated Schwefel's Function	M, N, L	[−100, 100]	10	1000
Hybrid function					
F10	Hybrid Function 1 (N = 3)	M, N, D	[−100, 100]	10	1100
F11	Hybrid Function 2 (N = 3)	M, N, D	[−100, 100]	10	1200
F12	Hybrid Function 3 (N = 3)	M, N, D	[−100, 100]	10	1300
F13	Hybrid Function 4 (N = 4)	M, N, D	[−100, 100]	10	1400
F14	Hybrid Function 5 (N = 4)	M, N, D	[−100, 100]	10	1500
F15	Hybrid Function 6 (N = 4)	M, N, D	[−100, 100]	10	1600
F16	Hybrid Function 6 (N = 5)	M, N, D	[−100, 100]	10	1700
F17	Hybrid Function 6 (N = 5)	M, N, D	[−100, 100]	10	1800
F18	Hybrid Function 6 (N = 5)	M, N, D	[−100, 100]	10	1900
F19	Hybrid Function 6 (N = 6)	M, N, D	[−100, 100]	10	2000
Composition function					
F20	Composition Function 1 (N = 3)	M, N, A, D	[−100, 100]	10	2100
F21	Composition Function 2 (N = 3)	M, N, A, D	[−100, 100]	10	2200
F22	Composition Function 3 (N = 4)	M, N, A, D	[−100, 100]	10	2300
F23	Composition Function 4 (N = 4)	M, N, A, D	[−100, 100]	10	2400
F24	Composition Function 5 (N = 5)	M, N, A, D	[−100, 100]	10	2500
F25	Composition Function 6 (N = 5)	M, N, A, D	[−100, 100]	10	2600
F26	Composition Function 7 (N = 6)	M, N, A, D	[−100, 100]	10	2700
F27	Composition Function 8 (N = 6)	M, N, A, D	[−100, 100]	10	2800
F28	Composition Function 9 (N = 3)	M, N, A, D	[−100, 100]	10	2900
F29	Composition Function 10 (N = 3)	M, N, A, D	[−100, 100]	10	3000

Table 4
Friedman Mean Rank Test Of IEEE-2017 benchmark functions.

Algorithms	Mean				Best				Std				Mean rank	Final rank
	1	2	3	4	5	6	7	8	9	10	11	12		
	10D	30D	50D	100D	10D	30D	50D	100D	10D	30D	50D	100D		
GWCA	10.10	8.59	8.31	8.07	9.97	7.93	7.67	7.81	12.21	13.59	12.69	13.21	10.01	6.00
FDB-AGDE	3.09	3.47	3.93	4.00	4.12	3.62	4.00	4.00	4.95	4.31	5.66	5.59	4.23	3.00
FDB-TLABC	4.90	4.66	4.69	4.79	8.47	5.21	5.17	4.88	6.66	5.28	6.24	6.83	5.65	4.00
FDB-AEO	4.48	5.79	6.48	7.03	7.41	6.55	7.52	7.66	6.97	6.55	8.00	9.38	6.99	5.00
L-SHADE	2.69	1.98	1.83	1.90	4.60	3.50	2.67	2.86	3.52	2.55	3.00	1.79	2.74	2.00
LSHADEcnEpSin	3.12	1.95	1.55	1.62	4.59	3.53	2.45	2.14	3.81	1.86	2.41	1.90	2.58	1.00
AOA	25.86	24.00	24.86	26.76	23.69	24.76	25.72	27.45	24.93	23.21	21.86	22.76	24.66	25.00
AO	15.83	13.52	13.28	12.83	17.03	15.07	14.28	13.34	17.21	15.93	17.00	17.14	15.20	14.00
AEFA	17.59	10.64	8.45	7.28	20.97	13.09	9.90	8.90	12.69	9.45	7.72	7.72	11.20	8.00
BWOA	21.48	23.34	24.72	25.93	18.45	22.52	23.48	25.14	25.59	27.10	27.62	28.03	24.45	24.00
CHIO	33.86	32.59	31.59	28.97	22.31	17.28	15.55	14.24	34.00	34.00	34.00	34.00	27.70	31.00
DDAO	21.38	26.00	26.45	27.21	26.83	29.10	29.52	30.03	14.38	15.86	15.24	14.59	23.05	23.00
GPC	16.38	17.59	18.83	19.52	22.00	22.00	21.76	21.62	13.24	12.03	13.83	13.69	17.71	19.00
GWO	13.31	11.59	11.00	11.14	14.00	12.48	11.59	11.69	13.62	14.14	14.76	16.45	12.98	12.00
HGS	12.79	10.10	9.34	9.41	9.67	8.90	8.59	9.38	15.34	12.55	13.03	14.45	11.13	7.00
HHO	19.86	16.76	15.86	14.97	17.52	17.07	16.55	15.21	21.34	17.62	16.34	15.62	17.06	17.00
HPO	11.83	10.10	9.97	10.07	10.17	9.66	9.53	10.07	14.45	14.93	15.66	13.97	11.70	10.00
JSOA	26.31	28.69	29.21	29.45	24.38	26.10	28.07	28.38	28.31	29.00	28.03	26.55	27.71	32.00
MFO	14.72	16.79	16.93	17.38	10.90	13.79	15.90	15.69	18.38	21.79	23.24	23.41	17.41	18.00
POA	10.62	15.14	15.76	16.03	12.83	16.10	16.76	16.93	14.03	18.90	18.72	19.76	15.97	15.00
PFA	11.90	10.48	9.90	10.41	11.31	10.14	9.52	10.14	14.76	14.38	16.28	16.21	12.12	11.00
RUN	11.03	9.31	10.21	9.97	14.28	11.07	11.41	10.62	12.10	13.28	12.86	11.86	11.50	9.00
RSA	29.97	29.24	28.86	27.72	31.14	30.48	29.31	28.28	25.90	24.69	23.38	22.62	27.63	30.00
RSO	25.03	25.55	24.86	24.17	28.79	27.93	27.48	25.48	20.79	21.79	21.93	22.62	24.70	26.00
SCA	18.69	21.07	20.93	20.90	23.83	23.66	23.55	22.97	13.69	15.21	14.79	16.00	19.61	20.00
SFO	29.90	30.76	31.00	31.52	31.72	32.14	32.24	32.38	22.14	19.66	18.14	19.28	27.57	29.00
SOA	18.03	19.97	20.79	20.97	19.72	21.34	22.14	22.21	16.66	18.14	19.90	21.41	20.11	21.00
SDO	26.62	30.79	31.24	31.79	29.34	32.17	32.21	32.72	19.69	22.14	22.41	23.93	27.92	33.00
SHO	32.86	32.97	32.83	32.72	33.52	33.52	33.41	33.03	30.52	28.93	26.07	24.00	31.20	34.00
SSA	14.83	13.41	12.52	11.14	11.33	12.36	11.81	11.17	17.83	17.52	16.03	12.38	13.53	13.00
TSA	28.14	25.38	24.45	22.83	25.93	25.03	24.31	23.31	28.10	27.00	24.72	24.28	25.29	27.00
TSO	27.28	28.48	28.17	27.76	25.62	27.17	27.38	27.52	26.07	26.28	25.86	24.38	26.83	28.00
WSOA	15.48	14.93	15.66	16.62	10.66	12.72	14.69	16.24	18.48	19.55	20.76	21.66	16.45	16.00
DOA	15.03	19.38	20.55	22.14	7.91	17.00	18.86	21.52	22.66	25.79	26.79	27.55	20.43	22.00

algorithms for these experiments are given in Table 4. The average algorithm ranking is given in the last column of Table 4. Although GWCA cannot dominate all of its competitors, GWCA can outperform 82.35% of the algorithms inside the comparison algorithm library (6/34).

The Friedman test was used for multiple comparisons between more than two MHS algorithms, while the Wilcoxon test was the most commonly used method for pairwise comparisons between two algorithms. Therefore, we have analyzed the results of pairwise comparisons between GWCA and other methods in Table 5. The scores given in each cell of Table 5 show the number of test problems in which FDB-TLABC fails relative to its competitors, the number of test problems in which the two competing algorithms perform similarly, and the number of test problems in which the GWCA implementation outperforms its competitors. Based on the results in Table 5, it can be seen that the Wilcoxon pairwise comparisons and Friedman test results overlap. The performance of the L-SHADE algorithm in the CEC-2017 suite is successful. The number of GWCA's outperforming FDB-AEO increases as the dimensionality of the problem increases. This suggests that FDB-AEO is effective in small search spaces. The results of the two-by-two comparison demonstrate the competitiveness of the GWCA, FDB-AGDE, FDB-TLABC, FDB-AEO, L-SHADE, and LSHADEcnEpSin methods.

In conclusion, according to the results of the statistical analysis, GWCA showed very strong competitiveness. Moreover, the LSHADESPACMA algorithm outperforms its competitors and ranks first among the 34 competing MHS methods. This indicates that no MHS method has the ability to solve all optimization problems or provide superior performance to its competitors for all problems, but some powerful and competitive MHS methods are able to succeed in every optimization problem. Therefore, it is necessary to conduct a study specific to each optimization problem. In the following sections, we will investigate the utility and scalability of the algorithms on constrained engineering problems and complex nonlinear problems. This is one of the research focuses in the field of MHS algorithms.

5.1.2. Convergence analysis

This subsection focuses on the convergence analysis of GWCA and five competing algorithms, namely FDBAGDE, AEFA, DOA, GWO, and HGS. The objective is to assess the search performance of these algorithms across different types of problems, including unimodal, multimodal, hybrid, and composition problems. To achieve this, one representative problem from each problem type in the CEC 2017 benchmark suite is selected for evaluation.

The box plots presented in Fig. 10 illustrates the convergence analysis of the five algorithms, namely FDBAGDE, AEFA, DOA, GWO, and HGS, on the CEC 2017 benchmark suite. The box plots are based on the error values obtained from all independent runs (51 times). For the F2 Unimodal problem, all five algorithms consistently converge to the global optimal solution across all dimensions (30, 50, and 100). This indicates that the algorithms effectively perform the optimization task, achieving stable convergence. In the case of the F3 Multimodal problem, the convergence performances of the algorithms vary across all dimensions. As the problem size increases, the error values of the algorithms also increase. Notably, GWCA and FDBAGDE demonstrate better convergence performance compared to the other algorithms. This highlights the challenging nature of the F3 problem, which requires strong exploration capabilities to navigate its complex search space. Considering the F11 hybrid problem, FDBAGDE consistently exhibits the lowest error values and the most stable convergence across all dimensions (10, 30, 50, and 100). However, in lower-dimensional search spaces, FDBAGDE and GWCA show comparable performance. The F11 problem serves as a test for evaluating the algorithms' ability to balance exploitation and exploration. From Fig. 10, it can be observed that GWCA maintains a more balanced search compared to its competitors. For the F21 combinatorial problem, GWCA's convergence is affected by the presence of multiple local optima. However, it is worth noting that GWCA still manages to obtain reliable solutions, particularly as the problem dimensionality increases. This demonstrates the balanced search performance of GWCA across all three dimensions. Overall,

Table 5
Wilcoxon's two-by-two Test Of IEEE-2017 benchmark functions.

Algorithms	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	10D				30D				50D				100D			
GWCA	R+	R−	P-value	+/−/=	R+	R−	P-Value	+/−/=	R+	R−	P-Value	+/−/=	R+	R−	P-Value	+/−/=
FDB-AGDE	7.00	428.00	5.32E−06	2/27/0	9.00	426.00	6.53E−06	2/27/0	47.00	388.00	2.27E−04	3/26/0	116.00	319.00	0.03	7/22/0
FDB-TLABC	29.00	406.00	4.58E−05	2/27/0	84.00	351.00	3.89E−03	4/25/0	108.00	327.00	1.79E−02	4/25/0	115.00	320.00	0.03	6/23/0
FDB-AEO	4.00	431.00	3.90E−06	2/27/0	79.00	356.00	2.75E−03	5/24/0	131.00	304.00	6.14E−02	7/22/0	167.00	268.00	0.27	9/20/0
L-SHADE	0.00	435.00	2.56E−06	0/29/0	22.00	413.00	2.36E−05	2/27/0	2.00	433.00	3.17E−06	1/28/0	0.00	435.00	0.00	0/29/0
LSHADEcnEpSin	5.00	430.00	4.33E−06	1/28/0	4.00	431.00	3.90E−06	1/28/0	0.00	435.00	2.56E−06	0/29/0	1.00	434.00	0.00	1/28/0
AOA	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0
AO	369.00	66.00	1.05E−03	25/4/0	369.00	66.00	1.05E−03	24/5/0	382.00	53.00	3.75E−04	25/4/0	393.00	42.00	0.00	24/5/0
AEFA	358.00	77.00	2.38E−03	19/10/0	293.00	142.00	1.03E−01	16/13/0	247.00	188.00	5.24E−01	14/15/0	232.00	203.00	0.75	13/16/0
BWOA	411.00	24.00	2.86E−05	28/1/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0
CHIO	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	434.00	1.00	0.00	28/1/0
DDAO	410.00	25.00	3.15E−05	27/2/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0
GPC	320.00	115.00	2.67E−02	21/8/0	434.00	1.00	2.85E−06	28/1/0	433.00	2.00	3.17E−06	28/1/0	435.00	0.00	0.00	29/0/0
GWO	336.00	99.00	1.04E−02	20/9/0	282.00	153.00	1.63E−01	14/15/0	278.00	157.00	1.91E−01	13/16/0	311.00	124.00	0.04	14/15/0
HGS	307.00	128.00	5.30E−02	20/9/0	354.00	81.00	3.16E−03	20/9/0	295.00	140.00	9.38E−02	17/12/0	295.00	140.00	0.09	17/12/0
HHO	435.00	0.00	2.56E−06	29/0/0	433.00	2.00	3.17E−06	28/1/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0
HPO	245.00	190.00	5.52E−01	17/12/0	365.00	70.00	1.43E−03	22/7/0	374.00	61.00	7.14E−04	21/8/0	386.00	49.00	0.00	23/6/0
JSOA	412.00	23.00	2.60E−05	28/1/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0
MFO	374.00	61.00	7.14E−04	23/6/0	430.00	5.00	4.33E−06	27/2/0	432.00	3.00	3.51E−06	28/1/0	418.00	17.00	0.00	25/4/0
POA	169.00	266.00	2.94E−01	14/15/0	383.00	52.00	3.45E−04	24/5/0	425.00	10.00	7.23E−06	28/1/0	433.00	2.00	0.00	28/1/0
PFA	293.00	142.00	1.03E−01	18/11/0	339.00	96.00	8.61E−03	19/10/0	309.00	126.00	4.79E−02	18/11/0	298.00	137.00	0.08	18/11/0
RUN	252.00	183.00	4.56E−01	19/10/0	262.00	173.00	3.36E−01	16/13/0	325.00	110.00	2.01E−02	21/8/0	373.00	62.00	0.00	23/6/0
RSA	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0
RSO	423.00	12.00	8.85E−06	27/2/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0
SCA	405.00	30.00	5.03E−05	27/2/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0
SFO	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0
SOA	366.00	69.00	1.32E−03	24/5/0	429.00	6.00	4.80E−06	28/1/0	430.00	5.00	4.33E−06	28/1/0	433.00	2.00	0.00	28/1/0
SDO	432.00	3.00	3.51E−06	28/1/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0
SHO	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0
SSA	324.00	111.00	2.13E−02	24/5/0	385.00	50.00	2.92E−04	25/4/0	423.00	12.00	8.85E−06	27/2/0	404.00	31.00	0.00	25/4/0
TSA	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0
TSO	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0
WSOA	303.00	132.00	6.45E−02	23/6/0	369.00	66.00	1.05E−03	26/3/0	394.00	41.00	1.35E−04	27/2/0	419.00	16.00	0.00	28/1/0
DOA	341.00	94.00	7.57E−03	22/7/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	2.56E−06	29/0/0	435.00	0.00	0.00	29/0/0

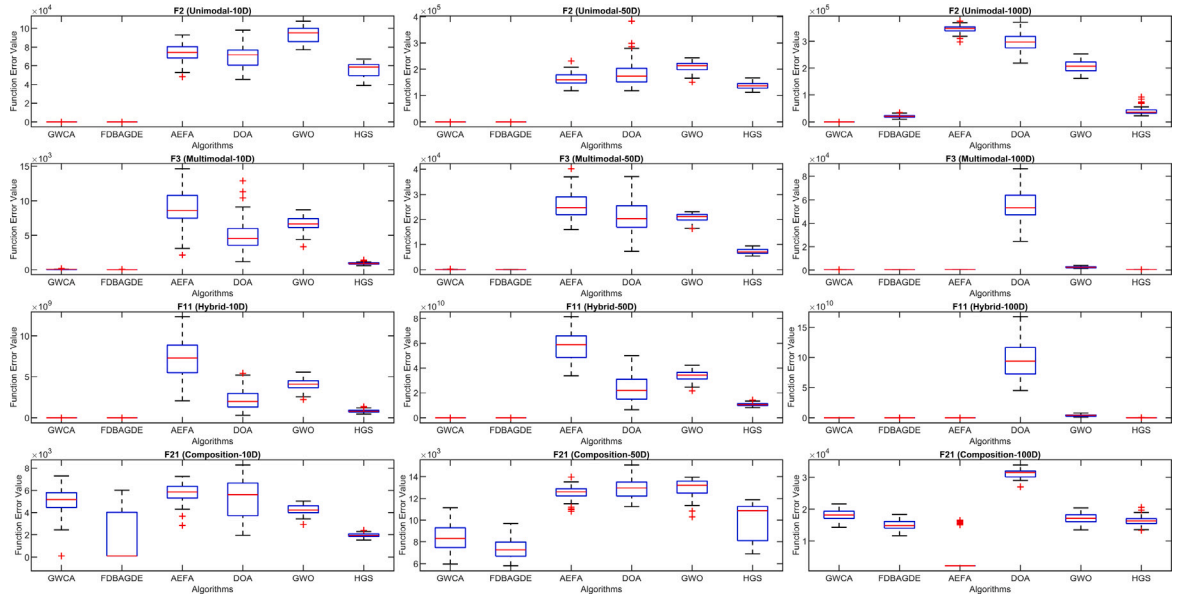


Fig. 10. Boxplot of CEC-2017 functions for different algorithms.

the convergence analysis reveals the strengths and weaknesses of the algorithms on different problem types within the CEC 2017 benchmark suite, providing insights into their exploration and exploitation capabilities.

Furthermore, the convergence performance of the five algorithms, namely FDBAGDE, AEFA, DOA, GWO, and HGS, was examined on the standard test set of the CEC 2017 benchmark suite. To conduct this analysis, four different problem types were selected from the test functions in CEC 2017. The convergence curves of these algorithms were generated for dimensions D = 10, 30, 50, and 100, and are presented in Fig. 11. In Fig. 11, the convergence curves of GWCA are highlighted in red to facilitate the assessment of its performance. From Fig. 11, it can be observed that GWCA exhibits a rapid convergence at the initial stage of the iteration and steadily progresses towards the optimal value. This demonstrates the algorithm's ability to effectively navigate the search space and converge towards the global optimum. The convergence curves of the other algorithms can also be observed in Fig. 11, allowing for a comprehensive comparison of their convergence behaviors. By analyzing these curves, it is possible to gain insights into the relative performance of the algorithms on the CEC 2017 benchmark suite. In summary, the convergence analysis conducted on the CEC 2017 standard test set showcases GWCA's ability to converge quickly and approach the optimal solution. The presented convergence curves enable a detailed examination of the algorithm's performance in comparison to the other analyzed algorithms.

Lastly, in order to comprehensively evaluate the performance of GWCA, this study examined two crucial aspects: exploration and exploitation. Unlike previous approaches that primarily rely on convergence plots and final results for performance assessment, we employed the methodology proposed by Hussain et al. (2019) to analyze the

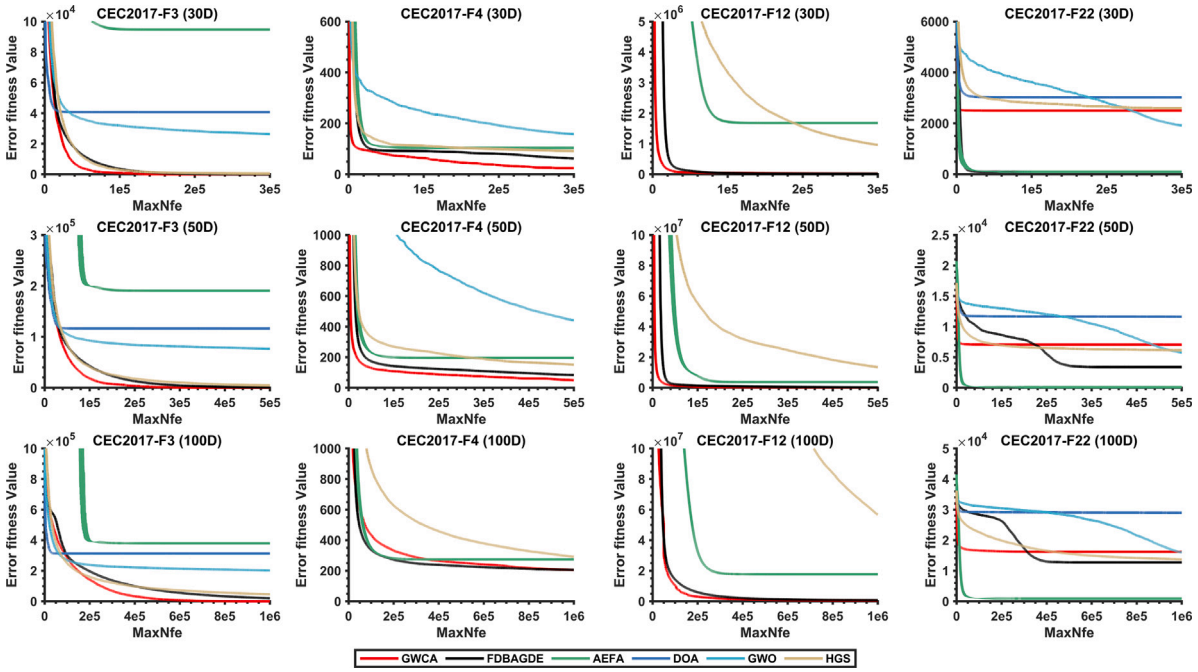


Fig. 11. Convergence analysis of CEC-2017.

trade-off between exploration and exploitation in GWCA. To ensure fairness in the experiment, we adopted the experimental settings of CEC2017 to calculate the change curve of the exploration-exploitation ratio. Fig. 12 presents the change curve illustrating the percentage of exploration and exploitation in GWCA. The results reveal that, throughout the search process, GWCA consistently maintained a balance between exploration and exploitation, with approximately 10% allocated to exploration and 90% to exploitation. This outcome demonstrates the remarkable capability of GWCA to effectively balance the exploration of the search space and the exploitation of promising regions. By incorporating this comprehensive analysis of the exploration-exploitation trade-off, beyond conventional convergence plots and final results, we gain deeper insights into the performance of GWCA. The observed balance achieved by GWCA highlights its ability to effectively explore and exploit the problem landscape, further validating its effectiveness as an optimization method. In conclusion, the utilization of the (Hussain et al., 2019) proposed methodology allows for a rigorous assessment of GWCA's performance, showcasing its adeptness in balancing exploration and exploitation during the search process.

5.2. Comparison of GWCA and MHS algorithms to engineer problems

In this subsection, we select 16 well-known engineering problems to demonstrate the applicability of the proposed algorithm to real engineering optimization problems. In addition, this paper compares the GWCA approach with 34 of the latest and most powerful competing algorithms in the literature. The data from the experimental studies are analyzed using Friedman and Wilcoxon tests. The Feasibility Rate (FR) and Success Rate (SR) (Kumar et al., 2020) are also given in this section. The analysis and algorithm complexity results are presented in the following sections.

5.2.1. Constraint handling

The constrained optimization problems used in this paper are given as Eq. (15)(Abualigah et al., 2022b). Furthermore, we use the method of penalty function to deal with the inequality constraint by taking the amount of violation of the constraint as part of the function, and the penalty function corresponding to this objective function is shown in Eq. (16).

$$\begin{aligned} &\min f(X) \\ &X = \left\{x_{(11)}, \ldots, x_{(1j)}, \ldots, x_{(1n)}\right\} \\ &s.t. \quad g_i(X) \le 0, j = 1, 2, 3, \ldots, m \\ &h_k(X) = 0, k = 1, 2, 3, \ldots, l \\ &LB_{(j)} \le x_{(ij)} \le UB_{(j)}, \quad j = 1, 2, 3, \ldots, n \end{aligned} \tag{15}$$

where m and l denote the number of the equation and inequality constraints, respectively, and n denotes the number of variables.

$$J_{aug}(x) = f(x) + \sum_{i=1}^{n} k_i \cdot G_i + \sum_{i=1}^{n} k_i \cdot H_i \tag{16}$$

where $k_i (= 10^{10})$ is the penalty function coefficient and G_i (see Eq. (17)) and H_i (see Eq. (18)) control the equation constraint and inequality constraint, respectively.

$$G_i = \begin{cases} 0 & \text{if } x_i \le 0 \\ x_i^2 & \text{else} \end{cases} \qquad i = 1, 2, \ldots, n \tag{17}$$

$$H_i = \begin{cases} 0 & \text{if } |x_i - 0.0001| \le 0 \\ x_i^2 & \text{else} \end{cases} \qquad i = 1, 2, \ldots, n \tag{18}$$

5.2.2. Statistical analysis

The Friedman test, commonly employed in experimental studies involving multiple competitors, was utilized to analyze the search performance of 34 algorithms, including GWCA. This analysis aimed to assess the performance of GWCA in comparison to other algorithms. To evaluate GWCA's performance, we focused on 18 well-established constrained engineering problems representing three domains: constrained problems, mechanical engineering, and civil engineering. The specifics of these 16 engineering design problems can be found in Table 6. By subjecting the algorithms to the Friedman test, we were able to statistically evaluate their relative performance across the set of engineering design problems. This approach provides a robust and objective assessment of GWCA's effectiveness in comparison to its counterparts. The inclusion of diverse constrained engineering problems from various domains ensures a comprehensive evaluation of GWCA's capabilities in tackling real-world optimization challenges.

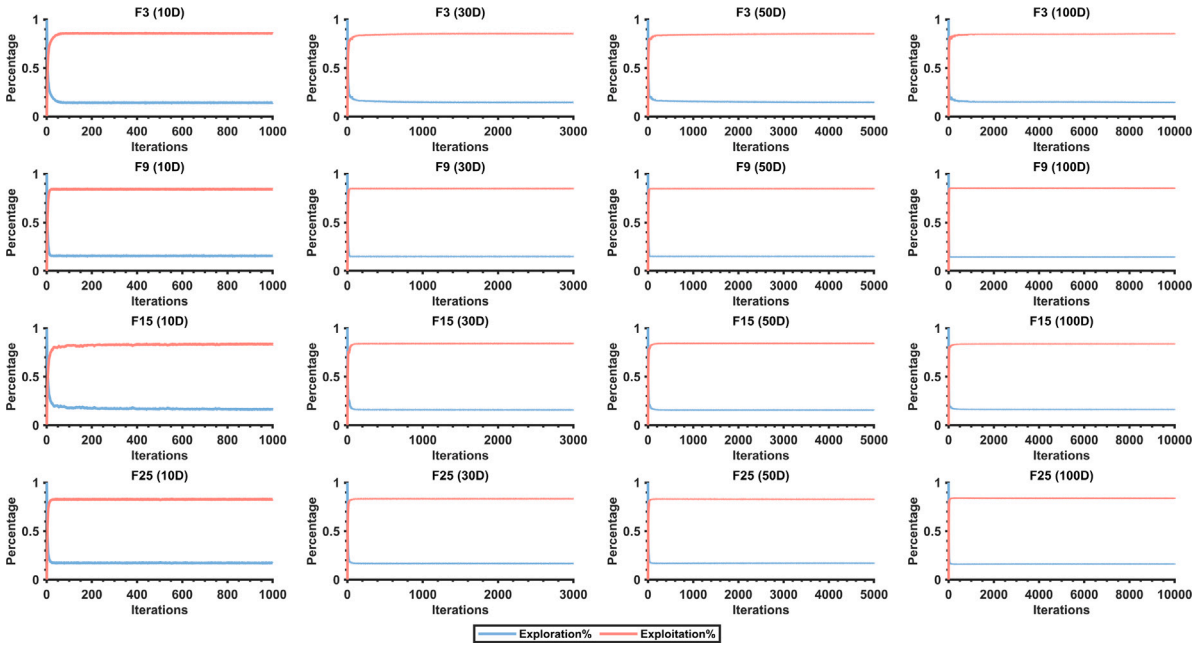


Fig. 12. The curve of the percentage of exploration and exploitation of GWCA.

Table 6
Descriptions of engineer problems (g for inequality constraint and geq for equation constraint).

Types	No	Name	Dim	g	geq	f_{min}	Formulation	Reference
Constraint	RW-1	Constraint Problem 1	7	4	0	6.80630E+02	Constrain problem 1	Eskandar et al. (2012)
	RW-2	Constraint Problem 2	5	6	0	−3.06655E+04	A.2	Eskandar et al. (2012)
	RW-3	Constraint Problem 3	10	1	0	−1.00000E+00	A.3	Eskandar et al. (2012)
Mechanical engineering	RW-4	Speed Reducer	7	11	0	2.99442E+03	B.1	Talatahari et al. (2021)
	RW-5	Rolling Element Bearing	10	9	0	−8.55385E+04	B.3	Talatahari et al. (2021)
	RW-6	Multiple disk clutch brake design problems	5	8	0	3.13660E−01	B.4	Talatahari et al. (2021)
	RW-7	Design of gear train	4	0	0	2.70000E−12	B.6	Talatahari et al. (2021)
	RW-8	Piston lever	4	4	0	8.41270E+00	B.7	Talatahari et al. (2021)
	RW-9	Corrugated bulkhead design	4	6	0	6.84296E+00	B.8	Talatahari et al. (2021)
	RW-10	Car side impact design	11	10	0	2.28430E+01	B.9	Talatahari et al. (2021)
Civil engineering	RW-11	Three-bar truss design problem	2	3	0	2.63896E+02	B.10	Talatahari et al. (2021)
	RW-12	Cantilever beam	5	1	0	1.33996E+00	B.11	Talatahari et al. (2021)
	RW-13	Minimize I-beam vertical deflection	4	2	0	1.30700E−02	B.12	Talatahari et al. (2021)
	RW-14	Tubular column design	2	6	0	2.64864E+01	B.13	Talatahari et al. (2021)
	RW-15	Design of welded beam design	4	7	0	1.72485E+00	B.14	Talatahari et al. (2021)
	RW-16	Reinforced concrete beam design	3	2	0	3.59208E+02	B.15	Talatahari et al. (2021)

To assess GWCA's capability in locating global minima, we conducted the Friedman ranking and Wilcoxon pairwise comparison tests, the results of which are presented in Table 7. The Friedman test and Wilcoxon test outcomes confirm that GWCA achieved the top-ranking position among the 34 algorithms analyzed. This finding underscores GWCA's superior convergence performance and its ability to escape from local optima.

To further evaluate the stability of GWCA, we examined the Friedman rankings obtained for the experimental algorithms based on three metrics: mean, best, and standard. The results are summarized in

Table 7
Friedman Mean Rank Test and Wilcoxon pairwise comparison results Of constraint engineer problems.

Name	RW-1	RW-2	RW-3	RW-4	RW-5	RW-6	RW-7	RW-8	RW-9	RW-10	RW-11	RW-12	RW-13	RW-14	RW-15	RW-16	Mean Rank	Final Rank	P-Value	+/−/=
GWCA	8	8.5	4	7.5	7	16	12	8	8.5	3	8	9.5	13	9	7.5	10.5	8.75	1		
FDB-AGDE	3	8.5	12	7.5	7	16	12	8	8.5	3	8	9.5	13	9	7.5	10.5	8.94	3	1.00E+00	1/1/14
FDB-TLABC	7	8.5	15	7.5	7	16	12	8	8.5	5	8	9.5	13	9	7.5	10.5	9.50	6	7.50E−01	2/1/13
FDB-AEO	6	16	13	15	16	16	12	25	8.5	6	8	9.5	13	9	7.5	10.5	11.94	12	4.69E−02	6/1/9
L-SHADE	3	8.5	11	7.5	7	16	12	8	8.5	3	8	9.5	13	9	7.5	10.5	8.88	2	1.00E+00	1/1/14
LSHADEcnEpSin	34	34	33	34	34	31.5	30.5	34	34	34	34	34	34	34	32	34	33.44	34	4.38E−04	16/0/0
AOA	27	28	30	29	27	16	29	30	25	11	29	27	24.5	31	28	28	26.22	30	6.10E−05	15/0/1
AO	17	21	2	23	25	16	24	18	19	23	23	20	13	23.5	22	21	19.41	21	8.54E−04	13/1/2
AEFA	20	27	10	32	17	31.5	12	33	32	19	33	9.5	28	32	31	10.5	23.59	25	2.44E−04	13/0/3
BWOA	3	8.5	18	7.5	7	16	12	8	8.5	21	8	9.5	13	9	7.5	10.5	10.44	10	5.00E−01	2/1/13
CHIO	25	33	27	20	26	16	30.5	29	27	29	27	29	31	27	24	33	27.09	31	6.10E−05	15/0/1
DDAO	30	26	17	24	29	16	12	28	23	30	19	30	29	25	25	25	24.25	26.5	1.22E−04	14/0/2
GPC	23	24	28	27	23	16	28	20	1	24	32	22	27	33	30	30	24.25	26.5	3.36E−03	14/1/1
GWO	15	19	20	21	18	16	12	17	16	18	17.5	9.5	13	18.5	17	10.5	16.13	17	9.77E−04	11/0/5
HGS	14	8.5	3	7.5	7	16	24	8	29	7	31	9.5	13	9	7.5	10.5	12.78	15	9.38E−02	5/1/10
HHO	18	20	22	19	20	16	12	8	18	9	17.5	19	13	20	20	10.5	16.38	19	9.77E−04	11/0/5
HPO	12	8.5	6	7.5	7	16	12	8	8.5	14.5	8	9.5	13	9	7.5	10.5	9.84	8	2.50E−01	3/0/13
JSOA	21	8.5	19	18	22	16	12	21	17	22	8	25	13	9	17	10.5	16.19	18	3.91E−03	9/0/7
MFO	13	8.5	21	7.5	7	16	26	8	8.5	14.5	16	9.5	13	9	7.5	10.5	12.22	13	6.25E−02	5/0/11
POA	10	8.5	7	7.5	7	16	12	8	8.5	14.5	8	9.5	13	9	7.5	10.5	9.78	7	2.50E−01	3/0/13
PFA	9	8.5	25	14	7	16	12	8	15	10	8	9.5	13	9	7.5	10.5	11.38	11	6.25E−02	5/0/11
RUN	16	17	16	16	14	16	12	8	2	20	8	9.5	13	9	15	10.5	12.63	14	1.95E−01	7/1/8
RSA	26	32	29	30	28	16	27	24	28	26	25	23	13	29	26	22	25.25	28	1.22E−04	14/0/2
RSO	28	31	34	33	33	33	32	26	31	32	26	31	33	30	29	31	30.81	32	4.38E−04	16/0/0
SCA	22	25	31	26	19	16	12	19	24	25	24	24	26	22	23	27	22.81	24	1.22E−04	14/0/2
SFO	31	30	23	1	30	16	34	32	33	33	30	33	1	26	33	29	25.94	29	7.30E−02	13/2/1
SOA	19	18	5	17	15	16	12	16	21	8	8	9.5	13	9	17	10.5	13.38	16	3.91E−03	9/0/7
SDO	32	1	24	28	31	1	1	27	26	1	21	32	2	23.5	27	23	18.78	20	1.09E−01	11/5/0
SHO	33	29	32	31	32	34	33	31	30	31	28	28	32	28	34	32	31.13	33	4.38E−04	16/0/0
SSA	11	8.5	1	7.5	7	16	24	8	8.5	14.5	8	9.5	13	9	7.5	10.5	10.22	9	3.75E−01	3/1/12
TSA	29	23	26	25	21	16	12	22	22	27	22	21	24.5	21	21	26	22.41	23	1.22E−04	14/0/2
TSO	24	22	14	22	24	16	12	23	20	28	20	26	30	18.5	19	24	21.41	22	1.22E−04	14/0/2
WSOA	3	8.5	8	7.5	7	16	12	8	8.5	14.5	8	9.5	13	9	7.5	10.5	9.41	4	5.00E−01	2/1/13
DOA	3	8.5	9	7.5	7	16	12	8	8.5	14.5	8	9.5	13	9	7.5	10.5	9.47	5	5.00E−01	2/1/13

Table 8
Friedman rankings of competing algorithms in the different index (Constraint engineer problems).

Algorithms	Best	Mean	Std	Mean rank	Final rank	Algorithms	Best	Mean	Std	Mean rank	Final rank
GWCA	8.75	7.16	9.31	8.41	4.00	JSOA	16.19	21.69	21.56	19.81	22.00
FDB_AGDE	8.94	6.84	4.97	6.92	2.00	MFO	12.22	12.50	12.31	12.34	12.00
fdb_tlabc	9.50	6.06	5.28	6.95	3.00	POA	9.78	7.28	8.47	8.51	5.00
FDB_AEO	11.94	6.47	7.41	8.60	6.00	PFA	11.38	11.53	9.94	10.95	8.00
L_SHADE	8.88	6.09	4.44	6.47	1.00	RUN	12.63	10.81	12.75	12.06	11.00
LSHADEcnEpSin	33.44	32.81	32.13	32.79	34.00	RSA	25.25	24.25	22.06	23.85	28.00
AOA	26.22	25.38	21.75	24.45	29.00	RSO	30.81	30.19	30.06	30.35	31.00
AO	19.41	19.22	19.56	19.40	19.00	SCA	22.81	19.50	15.88	19.40	19.00
AEFA	23.59	23.25	21.81	22.89	27.00	SFO	25.94	24.94	25.50	25.46	30.00
BWOA	10.44	20.69	23.06	18.06	18.00	SOA	13.38	13.56	15.66	14.20	14.00
CHIO	27.09	33.88	34.00	31.66	33.00	SDO	18.78	19.25	21.31	19.78	21.00
DDAO	24.25	21.00	17.19	20.81	24.00	SHO	31.13	30.63	30.31	30.69	32.00
GPC	24.25	22.06	21.44	22.58	26.00	SSA	10.22	10.84	11.88	10.98	9.00
GWO	16.13	12.78	11.94	13.61	13.00	TSA	22.41	19.88	19.69	20.66	23.00
HGS	12.78	14.00	15.81	14.20	14.00	TSO	21.41	23.00	23.00	22.47	25.00
HHO	16.38	18.00	18.66	17.68	17.00	WSOA	9.41	10.47	10.25	10.04	7.00
HPO	9.84	11.41	14.06	11.77	10.00	DOA	9.47	17.59	21.56	16.21	16.00

Table 9
Statistics result Of GWCA and advanced algorithms on engineer problems.

RW1	Best	Mean	STD	FR	SR	RW2	Mean	Best	STD	FR	SR
GWCA	6.8063807591E+02	6.8073516703E+02	7.5031212264E−02	100	0	GWCA	−3.0665538692E+04	−3.0665538692E+04	1.1858360095E−11	100	0
FDB-AGDE	6.8063005742E+02	6.8063005820E+02	6.5405452499E−07	100	0	FDB-AGDE	−3.0665538692E+04	−3.0665538692E+04	1.4851986287E−11	100	0
FDB-TLABC	6.8063163188E+02	6.8063679271E+02	3.1213047687E−03	100	0	FDB-TLABC	−3.0665538692E+04	−3.0665538692E+04	1.1647215137E−11	100	0
L-SHADE	6.8063005737E+02	6.8063005737E+02	3.1134422756E−13	100	0	L-SHADE	−3.0665538692E+04	−3.0665538692E+04	1.4851986287E−11	100	0
LSHADEcnEpSin	6.8063005737E+02	6.8063007361E+02	7.9982402990E−05	100	0	LSHADEcnEpSin	−3.2471948154E+04	−3.2178346388E+04	1.0864940630E+02	0	0
POA	6.8064793431E+02	6.8100152912E+02	5.3578780950E−01	100	0	POA	−3.0665538692E+04	−3.0665538692E+04	1.3200734161E−11	100	0
WSOA	6.8063005737E+02	6.8063005737E+02	1.5306878326E−11	100	0	WSOA	−3.0665538692E+04	−3.0665538692E+04	1.1974054152E−11	100	0
RW3						RW4					
GWCA	−9.9938857970E−01	−2.7899142924E−01	3.4024557344E−01	100	0	GWCA	2.9944243820E+03	2.9944243820E+03	3.0786541192E−13	100	0
FDB-AGDE	−6.1370840519E−01	−2.7610798783E−01	1.3290980400E−01	100	0	FDB-AGDE	2.9944243820E+03	2.9944243820E+03	3.8272692633E−13	100	0
FDB-TLABC	−4.6841270512E−01	−1.4309771251E−01	1.0786888482E−01	100	0	FDB-TLABC	2.9944243820E+03	2.9944243820E+03	0.0000000000E+00	100	0
L-SHADE	−5.0257198927E−01	−2.7754618777E−01	1.0844192278E−01	100	0	L-SHADE	2.9944243820E+03	2.9944243820E+03	1.6077746777E−13	100	0
LSHADEcnEpSin	−6.3925560933E−01	−4.2325791674E−01	1.2366720883E−01	100	0	LSHADEcnEpSin	2.7437629532E+03	2.8344300499E+03	2.4316916870E+01	0	0
POA	−8.9043318598E−01	−4.1534434274E−01	1.5897032938E−01	100	0	POA	2.9944243820E+03	2.9979928798E+03	6.0182962950E+00	100	0
WSOA	−8.0288076489E−01	−1.5806709719E−01	2.8287244568E−01	100	0	WSOA	2.9944243820E+03	3.0009941968E+03	3.0960459036E+01	100	0
RW5						RW6					
GWCA	−8.5539194532E+04	−8.5539194532E+04	2.5895322935E−11	100	0	GWCA	3.1365661053E−01	3.1365661053E−01	1.6996749444E−16	100	0
FDB-AGDE	−8.5539194532E+04	−8.5539194532E+04	5.9407945150E−12	100	0	FDB-AGDE	3.1365661053E−01	3.1365661053E−01	1.6996749444E−16	100	0
FDB-TLABC	−8.5539194532E+04	−8.5539194532E+04	1.5146113579E−11	100	0	FDB-TLABC	3.1365661053E−01	3.1365661053E−01	1.6996749444E−16	100	0
L-SHADE	−8.5539194532E+04	−8.5539194532E+04	9.3932208891E−12	100	0	L-SHADE	3.1365661053E−01	3.1365661053E−01	1.6996749444E−16	100	0
LSHADEcnEpSin	−9.5816910906E+04	−9.2089912366E+04	3.9044523294E+03	0	0	LSHADEcnEpSin	−3.0789600891E+06	−1.2959912470E+06	7.2708524152E+05	0	0
POA	−8.5539194532E+04	−8.5512971387E+04	8.5945667851E+01	100	0	POA	3.1365661053E−01	3.1365661053E−01	1.6996749444E−16	100	0
WSOA	−8.5539194532E+04	−8.5500973112E+04	1.0563825454E+02	100	0	WSOA	3.1365661053E−01	3.2103244177E−01	1.2071537477E−02	100	0
RW7						RW8					
GWCA	2.7008571489E−12	1.1606300089E−10	2.5364131796E−10	100	100	GWCA	8.4126983227E+00	9.7486316145E+01	8.0583439395E+01	100	40
FDB-AGDE	2.7008571489E−12	7.5914091931E−12	8.8822592245E−12	100	100	FDB-AGDE	8.4126983227E+00	1.6103112918E+02	3.1796845549E+01	100	4
FDB-TLABC	2.7008571489E−12	5.9612251784E−12	7.6244875757E−12	100	100	FDB-TLABC	8.4126983227E+00	2.7499902130E+01	5.2754244439E+01	100	88
L-SHADE	2.7008571489E−12	5.1461331710E−12	6.7583858952E−12	100	100	L-SHADE	8.4126983227E+00	1.4838552624E+02	5.2754244439E+01	100	12
LSHADEcnEpSin	5.1524773041E−13	4.0944748788E−12	7.3103116249E−12	8	100	LSHADEcnEpSin	−7.4822284231E+07	−5.5758488309E+07	1.1435707569E+07	0	0
POA	2.7008571489E−12	2.7071529087E−11	3.9512112686E−11	100	100	POA	8.4126983227E+00	4.0224704669E+01	6.4935986035E+01	100	80
WSOA	2.7008571489E−12	2.7722356262E−09	6.0099292524E−09	100	92	WSOA	8.4126983227E+00	1.2293592117E+02	7.2890463544E+01	100	28
RW9						RW10					
GWCA	6.8429580095E+00	6.8429580095E+00	3.9988459992E−13	100	100	GWCA	2.2843027327E+01	2.3003507968E+01	2.1179380049E−01	100	0
FDB-AGDE	6.8429580095E+00	6.8429580095E+00	1.8129866073E−15	100	100	FDB-AGDE	2.2842969193E+01	2.2842969193E+01	5.5228759124E−12	100	0
FDB-TLABC	6.8429580095E+00	6.8429580095E+00	3.0119595631E−15	100	100	FDB-TLABC	2.2843127477E+01	2.2860012967E+01	7.5483423565E−02	100	0
L-SHADE	6.8429580095E+00	6.8429580095E+00	1.7855847292E−15	100	100	L-SHADE	2.2842969193E+01	2.2842969193E+01	1.4192338936E−14	100	0
LSHADEcnEpSin	−1.2818966931E+07	−8.6661634015E+05	3.0537724574E+06	92	92	LSHADEcnEpSin	−3.6268590356E+02	−6.5117630206E+01	8.8063661866E+01	0	0
POA	6.8429580095E+00	6.8429580646E+00	1.9371289674E−07	100	80	POA	2.3355261879E+01	2.3415636727E+01	1.4517164630E−01	100	0
WSOA	6.8429580095E+00	6.8429580095E+00	1.8129866073E−15	100	100	WSOA	2.3355261864E+01	2.3512589866E+01	2.1512930564E−01	100	0
RW11						RW12					
GWCA	2.6389584387E+02	2.6389599908E+02	1.8586307745E−04	100	0	GWCA	1.3399601110E+00	1.3399699258E+00	9.4196416512E−06	100	0
FDB-AGDE	2.6389584340E+02	2.6389585631E+02	2.0925914018E−05	100	0	FDB-AGDE	1.3399563738E+00	1.3399564271E+00	8.6341388760E−08	100	0
FDB-TLABC	2.6389584338E+02	2.6389584338E+02	5.6843418861E−14	100	100	FDB-TLABC	1.3399565690E+00	1.3399572840E+00	5.2751979208E−07	100	0
L-SHADE	2.6389584338E+02	2.6389584338E+02	0.0000000000E+00	100	100	L-SHADE	1.3399563606E+00	1.3399563606E+00	0.0000000000E+00	100	0
LSHADEcnEpSin	−8.9136952856E+01	2.4036138403E+02	8.3207816374E+01	92	92	LSHADEcnEpSin	−2.0966641930E+01	−1.8351638066E+01	1.6113099449E+00	0	0
POA	2.6389584338E+02	2.6389584338E+02	2.0097183471E−14	100	100	POA	1.3399563618E+00	1.3399630036E+00	9.8180292565E−06	100	0
WSOA	2.6389584338E+02	2.6389584338E+02	2.8421709430E−14	100	100	WSOA	1.3399563606E+00	1.3399563606E+00	0.0000000000E+00	100	0
RW13						RW14					
GWCA	1.3074118905E−02	1.3074118905E−02	4.0577627785E−13	100	0	GWCA	2.6486361471E+01	2.6486361471E+01	3.8517326090E−14	100	100
FDB-AGDE	1.3074118905E−02	1.3074118905E−02	5.3114842012E−18	100	0	FDB-AGDE	2.6486361471E+01	2.6486361471E+01	3.6259732147E−15	100	100
FDB-TLABC	1.3074118905E−02	1.3074118905E−02	1.4347301055E−16	100	0	FDB-TLABC	2.6486361471E+01	2.6486361471E+01	3.6259732147E−15	100	100
L-SHADE	1.3074118905E−02	1.3074118905E−02	5.3114842012E−18	100	0	L-SHADE	2.6486361471E+01	2.6486361471E+01	3.6259732147E−15	100	100
LSHADEcnEpSin	−1.1313722878E+14	−4.8886696599E+12	2.2585898928E+13	0	0	LSHADEcnEpSin	2.6486361471E+01	2.6486361471E+01	3.6259732147E−15	100	100
POA	1.3074118905E−02	1.3074118905E−02	5.3114842012E−18	100	0	POA	2.6486361471E+01	2.6486361471E+01	3.6259732147E−15	100	100
WSOA	1.3074118905E−02	1.3074118905E−02	5.3114842012E−18	100	0	WSOA	2.6486361471E+01	2.6486361471E+01	3.6259732147E−15	100	100
RW15						RW16					
GWCA	1.7248523086E+00	1.7283615631E+00	9.3693260839E−03	100	40	GWCA	3.5920800000E+02	3.5920800000E+02	5.8015571435E−14	100	100
FDB-AGDE	1.7248523086E+00	1.7248523086E+00	6.7986997776E−16	100	100	FDB-AGDE	3.5920800000E+02	3.5920800000E+02	5.8015571435E−14	100	100
FDB-TLABC	1.7248523086E+00	1.7248523086E+00	7.4061113154E−16	100	100	FDB-TLABC	3.5920800000E+02	3.5920800000E+02	5.8015571435E−14	100	100
L-SHADE	1.7248523086E+00	1.7248523086E+00	6.7986997776E−16	100	100	L-SHADE	3.5920800000E+02	3.5920800000E+02	5.8015571435E−14	100	100
LSHADEcnEpSin	−3.8700617995E+03	−7.2711355949E+02	1.1118011209E+03	40	40	LSHADEcnEpSin	3.5920800000E+02	3.5981640000E+02	1.2418912996E+00	88	80
POA	1.7248523086E+00	1.7248560406E+00	1.8629246254E−05	100	92	POA	3.5920800000E+02	3.5920800000E+02	5.8015571435E−14	100	100
WSOA	1.7248523086E+00	1.7248523086E+00	8.4476591986E−14	100	100	WSOA	3.5920800000E+02	3.5946672000E+02	8.9716774351E−01	100	92

Table 8, which also includes the average algorithm ranking and the final ranking. Additionally, Table 9 provides the statistical analysis results of eight competing algorithms across the 16 constrained engineering problems. From Table 9, it is evident that GWCA consistently obtains optimal solutions or solutions that closely approximate the optimal solutions. Notably, LSHADEcnEpSin fails to generate a sufficient number of feasible solutions, further highlighting GWCA's superiority over LSHADEcnEpSin in this experiment. Although GWCA did not secure the first place among all competing algorithms, it outperformed 88.23% of them (4/34), with its solutions closely matching or even surpassing those obtained by FDB-AGDE, FDB-TLABC, and L-SHADE. These results demonstrate GWCA's robust search ability in unknown search spaces.

In conclusion, the comprehensive analysis reveals GWCA's overall advantage over its competitors, with the L-SHADE algorithms demonstrating superior performance as well. This indicates that no single metaheuristic method can universally solve all optimization problems or outperform competitors in every scenario. However, a select few powerful and competitive metaheuristic methods, including GWCA, can achieve success across a wide range of optimization problems.

5.2.3. Convergence analysis

This subsection presents information about the convergence analysis of the GWCA and its 7 strong competing algorithms. In order to show the search performance of the algorithms on constraint engineer problems, 9 problems were selected to analyze the convergence performances.

Fig. 13 illustrates the convergence curve of GWCA and its competitors in the constraint engineering problem. It is evident from the plot that GWCA achieves faster and more efficient convergence compared to the other algorithms. This observation highlights GWCA's ability to accurately perform neighbor search tasks in both unknown search spaces and highly constrained conditions. Consequently, the results demonstrate the superior convergence capability of GWCA relative to its competitors in the context of constraint engineering problems.

The efficiency and effectiveness of GWCA in solving real-world problems are evaluated using three qualitative metrics: particle trajectory, search history, and optimization history. These metrics provide insights into the behavior and performance of GWCA in locating optimal solutions. Fig. 14 showcases the analysis of these three metrics. To enhance the reader's comprehension of the particle's trajectory in GWCA and its ability to find the optimal point, the evaluation is conducted by solving mathematical functions with intentionally reduced numbers of particles and iterations (30/100). It is important to note that the dimensions of the problem remain unchanged throughout the analysis.

One of the qualitative metrics used to assess GWCA's performance is the search history, which captures the particle positions throughout the iterations. Analyzing the search history provides insights into how GWCA achieves its best solution and reveals any patterns in the particles' exploration of the search space. The results demonstrate GWCA's ability to efficiently aggregate particles, indicating a balanced approach to exploitation and exploration.

The second metric, the average fitness history (convergence curve), tracks the fitness of the best particle (Work1) from the initial to the final iteration. This metric illustrates how the algorithm progressively approximates the global optimal solution over iterations. From Fig. 14, it is evident that GWCA exhibits rapid convergence in the early iterations, which contributes to its superiority over other optimization methods.

The third qualitative metric examines the trajectory of the particle. The third column in Fig. 14 represents the positional changes of the first particle in the first dimension. Across all tested functions, there is an initial abrupt oscillation followed by a fade-out in the later iterations. These sudden changes indicate exploratory behavior in the global search space during the initial iterations, followed by stabilization through local search as the number of iterations increases. This trend, as stated in Bergh and Engelbrecht (2006), ensures that the algorithm eventually converges to a global or local optimum point. It can be observed from the trajectory curves that the magnitude and frequency of fluctuations are influenced by the complexity of the search domain. More complex domains exhibit greater fluctuations. In RW16, there is a lack of significant rise or fall even after numerous iterations, whereas in RW1, the fluctuations are more pronounced in both magnitude and frequency, persisting for a larger number of iterations. This behavior suggests that GWCA benefits from well-balanced exploration and exploitation strategies.

5.2.4. Algorithm complexity

This section focuses on comparing the computational complexity of the top five algorithms among the 34 considered in this study, namely FDB-AGDE, FDB-TLABC, FDB-SDO, LRFDB-COA, WSOA, and POA. Detailed information regarding the complexity of these algorithms has been presented in the preceding subsections. Table 10 provides an overview of the computational complexity of these five competing algorithms. From the results presented in Table 10, it is evident that GWCA exhibits the lowest computational complexity among the considered algorithms. To calculate the algorithmic complexity, the following procedures have been employed.

1. $T_1 = \frac{\sum_{i=1}^{18} t_{1i}}{18}$: where t_{1i} is the computation time required to evaluate the function for 100000 times for problem i.
2. $T_2 = \frac{\sum_{i=1}^{18} t_{2i}}{18}$: where t_{2i} is the computation time required by the algorithm for 100000 function evaluations for problem i.
3. The algorithmic complexity is evaluated using $T_1, T_2, T_3 = \frac{T_2 - T_1}{T_1}$.

5.3. Comparison of GWCA and MHS algorithms to NP-hard problems

In the previous experiments, we evaluated the performance of GWCA and other algorithms on benchmark test functions, constraint problems, and engineering design problems. However, for simple standard test problems, exact optimization techniques can often find optimal solutions. Therefore, in this subsection, we aim to investigate the scalability and applicability of GWCA to more challenging problem domains, specifically mixed integer nonlinear problems. Addressing such problems is a key focus of research and development in the field of heuristic optimization. To assess GWCA's performance on these complex problems, we apply it to five representative mixed integer nonlinear problems and compare the results with those obtained by some of the best-performing algorithms as indicated in Table 8. Table 11 provides a description of these problems. All results presented in this subsection are obtained by running each algorithm 25 times for each NP-Hard problem, with populations randomly initialized for each run.

Table 12 presents the statistical analysis results of the data obtained from GWCA and the state of the art algorithms applied to the NP-Hard problems. The table also includes the outcomes of two non-parametric tests conducted. The findings indicate that GWCA consistently achieves solutions that either outperform or are on par with other algorithms across all five NP-Hard problems. Both the Friedman ranking and Wilcoxon pairwise comparison results support the leading position of GWCA in the NP-Hard problem domain. Additionally, Fig. 15 showcases the shortest path planning (NP1) for the robot, illustrating the solution obtained by GWCA in comparison with other algorithms.

In conclusion, the comprehensive analysis of the three experiments conducted demonstrates the overall superiority of GWCA over its competitors. The performance of GWCA was evaluated on both constrained real-world optimization problems and complex nonlinear problems, including 16 constrained engineering problems and 5 NP-hard problems. The results obtained from two non-parametric statistical tests provide strong evidence that the GWCA method consistently outperforms 33

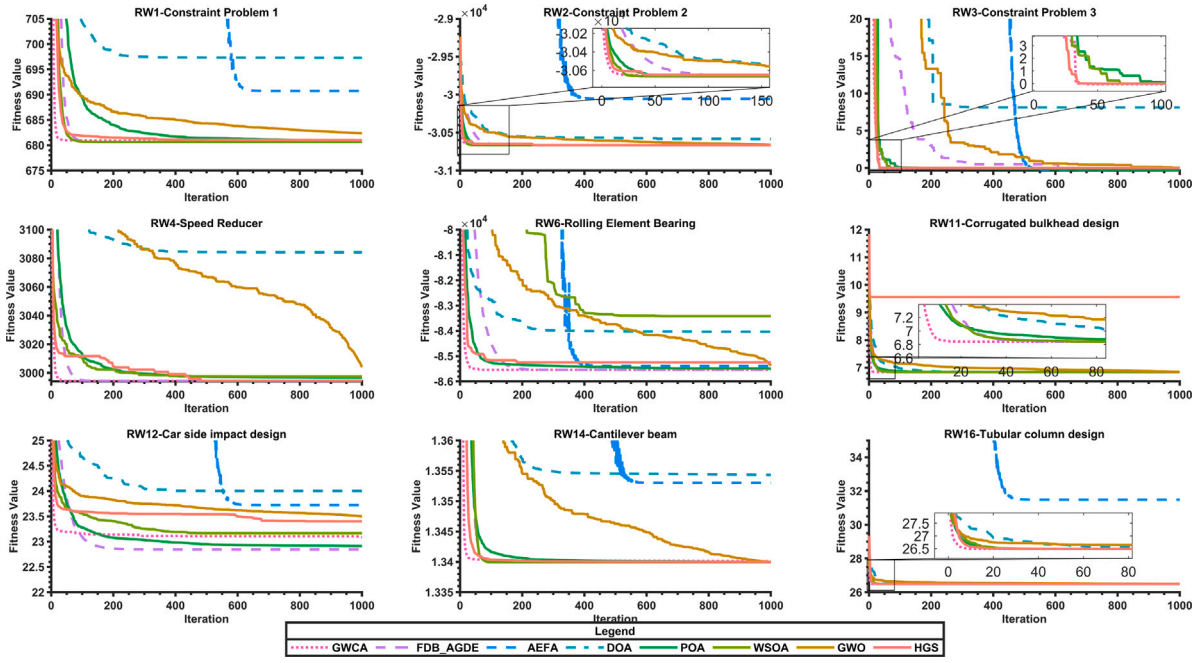


Fig. 13. Convergence analysis of GWCA.

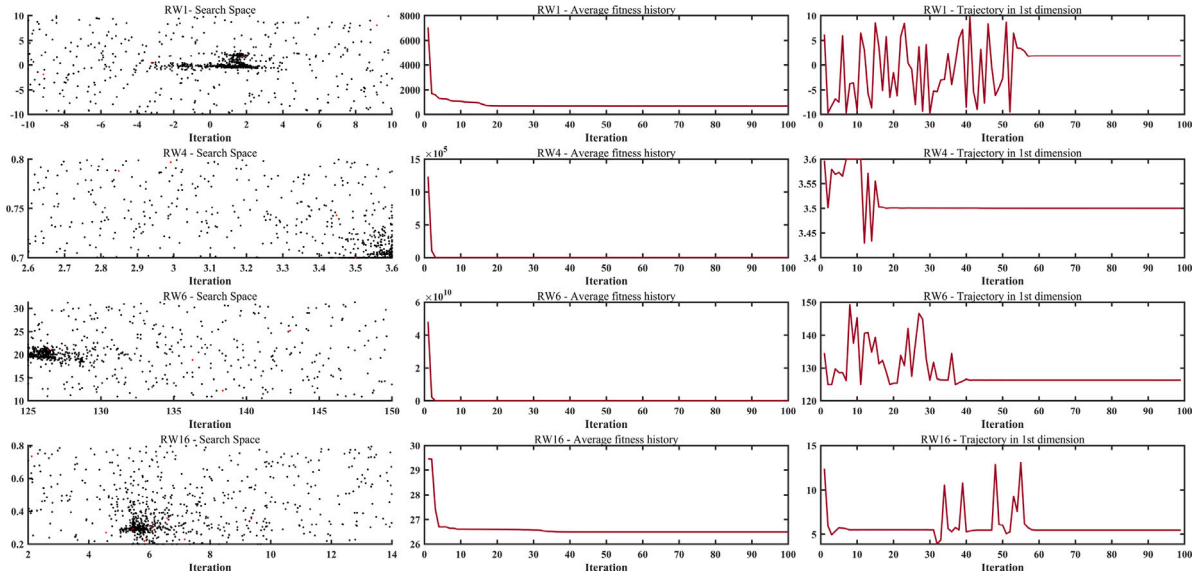


Fig. 14. Qualitative metrics: search history, average fitness history and trajectory in 1st dimension.

Table 10
The computational complexity of some competing algorithms.

T_1	3.1563						
Name	T_2	T_3	Rank	Name	T_2	T_3	Rank
GWCA	3.5060	0.1108	1	L-SHADE	3.8129	0.2080	2
FDB-AGDE	4.4006	0.3942	5	POA	5.7439	0.8198	6
FDB-TLABC	4.0346	0.2783	4	WSOA	3.8794	0.2291	3

other methods in finding better solutions for real-world engineering design problems. Moreover, GWCA excels in its ability to search for global minima, surpassing all competing algorithms in this regard. These findings highlight the successful, stable, and robust search performance of GWCA across different types of problem spaces.

6. Sensitivity analysis

The proposed GWCA algorithm employs four parameters, i.e., the maximum number of iterations, number of workers, and Gamma density function parameters P(Q). This section verifies the sensitivity of

Table 11
The description of NP-Hard problems.

No.	Name	Baseline
NP1	Raster maps - robot pathfinding	Yang (2023)
NP2	Logistics center location problem: factory-center-demand point	Yang (2023)
NP3	Multi-row shop floor scheduling - considering AVG partitioning	Yang (2023)
NP4	Power system bus optimization based on tide calculation	Yang (2023)
NP5	Optimization study of cold chain distribution logistics vehicle scheduling for a dairy company	Yang (2023)

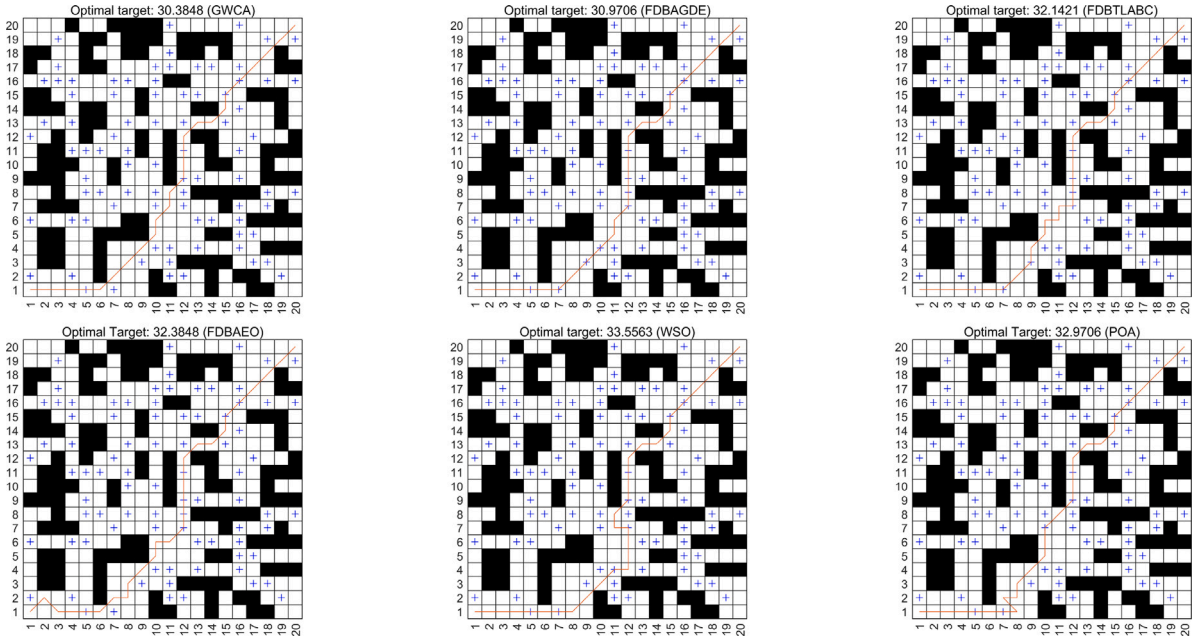


Fig. 15. Raster maps - robot pathf.

Table 12
Statistics Result Of GWCA and advanced algorithms on NP-Hard Peoblems.

NP1	Best	Mean	Std	Rank	NP2	Best	Mean	Std	Rank
GWCA	3.04E+01	3.29E+01	2.42E+00	2		3.52E+06	3.58E+06	2.97E+04	2
FDB-AGDE	3.10E+01	3.15E+01	4.40E−01	1		3.58E+06	3.61E+06	1.75E+04	3
FDB-TLABC	3.21E+01	3.39E+01	1.11E+00	3		3.94E+06	4.07E+06	6.48E+04	7
FDB-AEO	3.24E+01	3.59E+01	2.12E+00	6		3.61E+06	3.70E+06	5.60E+04	5
WSO	3.21E+01	3.52E+01	1.67E+00	4		3.65E+06	4.02E+06	1.17E+05	6
POA	3.30E+01	3.68E+01	8.00E−01	7		3.57E+06	3.69E+06	6.38E+04	4
L-SHADE	3.36E+01	3.57E+01	1.16E+00	5		3.50E+06	3.52E+06	1.08E+04	1
NP3					NP4				
GWCA	4.19E−01	4.30E−01	9.37E−03	6		3.32E+05	3.32E+05	1.78E−10	3.5
FDB-AGDE	4.19E−01	4.20E−01	1.53E−03	1		3.32E+05	3.32E+05	1.78E−10	3.5
FDB-TLABC	4.19E−01	4.25E−01	3.85E−03	4		3.32E+05	3.32E+05	1.81E−10	3.5
FDB_AEO	4.19E−01	4.24E−01	4.17E−03	3		3.32E+05	3.32E+05	1.78E−10	3.5
WSO	4.19E−01	4.33E−01	7.99E−03	7		3.32E+05	3.32E+05	1.78E−10	3.5
POA	4.19E−01	4.27E−01	5.90E−03	5		3.32E+05	3.32E+05	9.28E+01	7
L-SHADE	4.19E−01	4.23E−01	5.79E−03	2		3.32E+05	3.32E+05	1.78E−10	3.5
NP5						Mean rank	Final rank	P-value	+/−/=
GWCA	1.01E+09	1.01E+09	6.08E−07	4		3	1		
FDB-AGDE	1.01E+09	1.01E+09	6.08E−07	4		3.6	2	0.5	2/0/3
FDB-TLABC	1.01E+09	1.01E+09	6.08E−07	4		4.5	7	0.5	2/0/3
FDB-AEO	1.01E+09	1.01E+09	6.08E−07	4		4.4	6	0.5	2/0/3
WSO	1.01E+09	1.01E+09	6.08E−07	4		4.3	5	0.5	2/0/3
POA	1.01E+09	1.01E+09	6.08E−07	4		4.2	4	0.5	2/0/3
L-SHADE	1.01E+09	1.01E+09	6.08E−07	4		4	3	1	1/1/3

the GWCA to these four parameters while considering fixed values for thrust T, the weight of stone m, C_{max}, and C_{min}.

1. **Maximum number of iterations MaxIter:** GWCA was simulated for different numbers of iterations (200, 400, 600, 800,

Table 13
Sensitivity analysis of the proposed GWCA.

Setting	F1	RW4	NP3	Setting	F1	RW4	NP3
N				P			
20	128011803.3	2994.4244	0.4406597	2	3661.891154	2994.4244	0.430632
40	5422.103783	2994.4244	0.4359308	4	4334.51961	2994.4244	0.4316697
60	4855.292794	2994.4244	0.4349167	6	4638.127137	2994.4244	0.4323996
80	3122.831994	2994.4244	0.4355071	8	2922.350542	2994.4244	0.4311824
100	3499.591544	2994.4244	0.4326243	10	4200.633413	2994.4244	0.4322644
MaxIter				Q			
200	3954.175825	2994.4244	0.442069	2	5444.03375	2994.4244	0.4282
400	5389.948029	2994.4244	0.4406834	4	5942.749306	2994.4244	0.43051
600	4654.593806	2994.4244	0.4350525	6	5048.089043	2994.4244	0.43468
800	4945.196931	2994.4244	0.4337512	8	4259.622282	2994.4244	0.41918
1000	3435.391084	2994.4244	0.4324005	10	3399.66685	2994.4244	0.44503

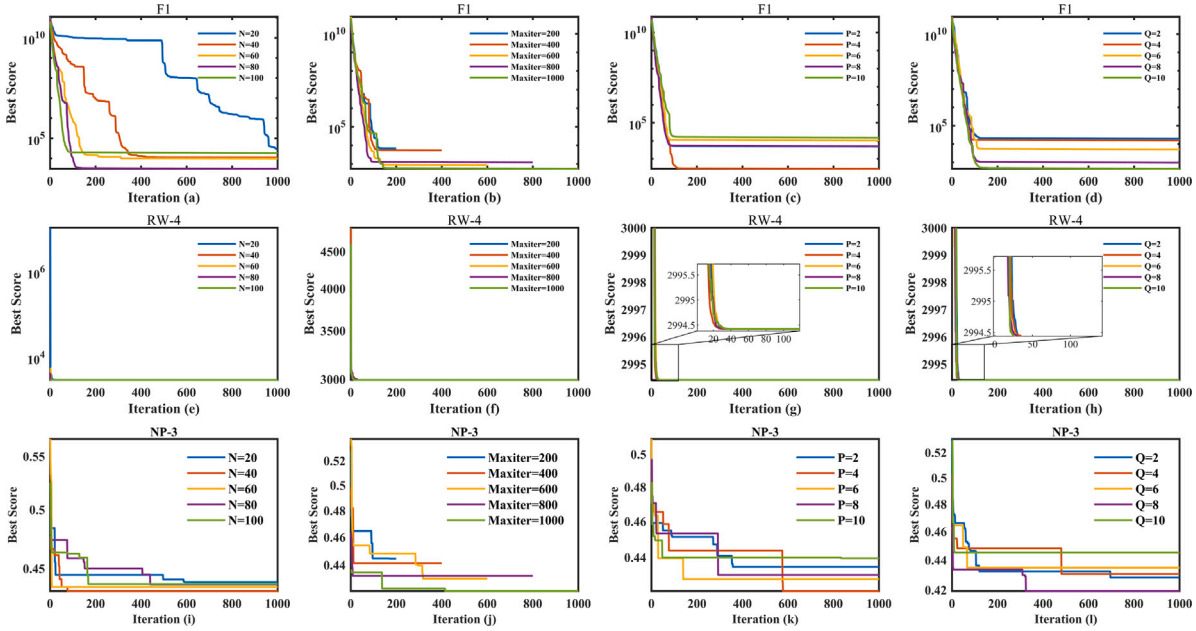


Fig. 16. Sensitivity analysis of GWCA.

and 1000). Section 6 and Fig. 16(a, e, and (i) show the iteration variations on various benchmark test functions. The results indicate that GWCA converges toward the optimal solution when iterations increase.

2. **Number of workers N:**GWCA is simulated for different numbers of workers (i.e., 20, 40, 60, 80, and 100). Section 6 and Fig. 16(b, f, and j) show the variation in the benchmark test functions for the different numbers of workers. As shown in Fig. 16, the fitness function value decreases while increasing the number of search agents.
3. **Gamma density function parameters P:** GWCA was run for different values of P (e.g., 2, 4, 6, 8, and 10) while the other parameters were kept unchanged. Section 6 and Fig. 16(c, g, and k) show the variation of P over different benchmark test functions. The results indicate that the GWCA gives better optimal results for P=8.
4. **Gamma density function parameters Q:** GWCA was simulated for different Q values (i.e., 0.2, 0.4, 0.6, 0.8, and 1) by keeping the other parameters fixed. Table 13 and Fig. 16(d, b, and l) show the effect of Q on various benchmark test functions. It can be concluded that GWCA provides better optimal results for Q=10.

7. Application in dam safety monitoring

In recent years, there has been significant development in China's extra-high arch dams (EHAD), with the construction of several 300 m high arch dams including Xiaowan, Jinping, Xiluodu, and Baihetan. Ensuring the safety and stability of EHADs has become a prominent concern in dam inspection. To address this issue, researchers have developed a spatial and temporal monitoring model to assess the impact of dam displacement on dam operation. This model is based on prototype monitoring data. Unlike traditional models that rely on single measurement points, the proposed model takes into account the spatial deformation field and aims to capture the intrinsic connections within it. By utilizing the model gram, it is possible to reasonably characterize the spatial deformation field's intrinsic connections. However, the dam spatial and temporal monitoring model involves numerous factors, leading to the issue of severe multicollinearity. As a result, the traditional regression model is inadequate in reducing the influence of multicollinearity on the prediction results. To overcome this limitation, we have constructed a highly expressive KICA-GWCA-SVM spatio-temporal monitoring model specifically designed for extra-high arch dams. This model effectively mitigates the influence of multicollinearity on dam deformation monitoring, providing a more accurate

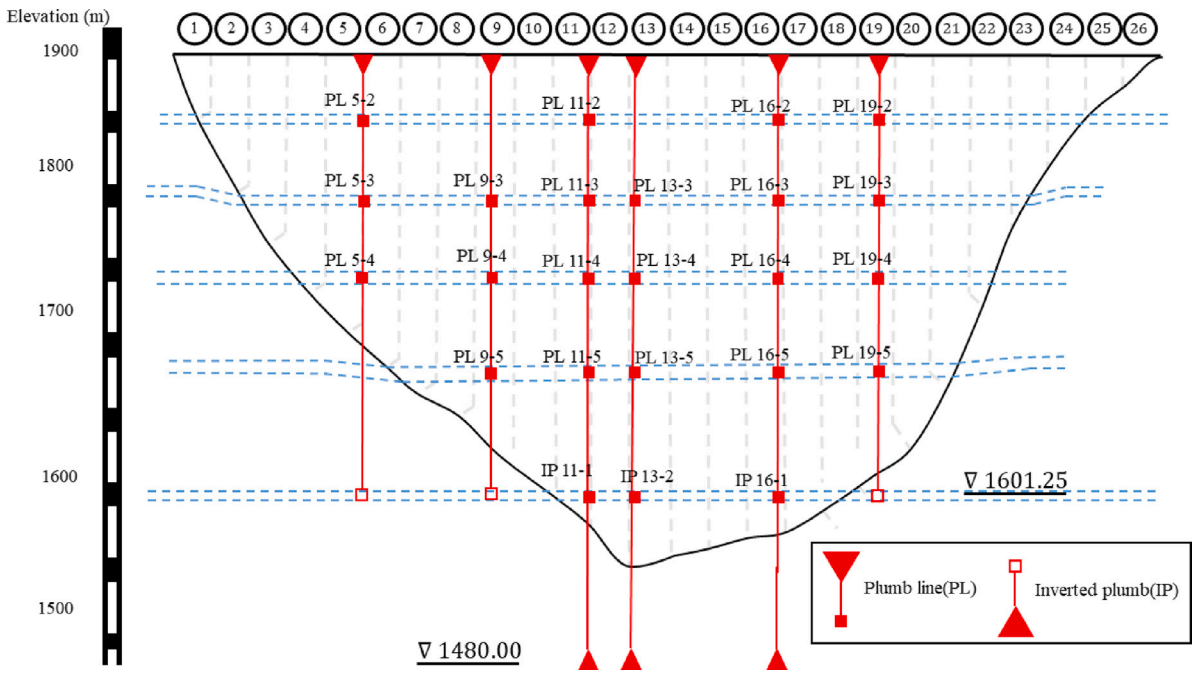


Fig. 17. The location of each monitoring point.

and comprehensive understanding of the overall spatio-temporal deformation behavior of the dam. The construction of the model involves the following steps:

1. Introduce the Kernel Independent Component Analysis (KICA) method to achieve the non-linear extraction of displacement data and influence factors from multiple measurement points. This method removes the interference of redundant information and enables the simultaneous monitoring of multiple measurement points with a small amount of integrated data analysis.
2. The optimal solution of the parameters in the support vector machine model is sought through the advantages of GWCA in parameter finding. The optimal solutions obtained by GWCA are substituted into the parameter of the SVM model.
3. Finally, a KICA-GWCA-SVM model is developed to realize the non-linear prediction of EHAD monitoring data to obtain EHAD's displacement data.

A hydropower plant in Sichuan Province, China, is used as an example to verify the validity of the KICA-GWCA-SVM model. The model parameters for this hydropower plant are a total installed capacity of 8.4 million kW, maximum dam height of 305 m, crest elevation of 1885 m, crest width of 16 m, and normal reservoir storage level of 1880 m. The dam consists of 26 sections and 24 monitoring points (See Fig. 17). Six sections (5#, 9#, 11#, 13#, 16#, and 19#) were selected for statistical analysis from June 16, 2013, to September 28, 2015. A total of 274 radial displacement monitoring data were collected during this period and the measured water level process line and displacement values for each measurement point are shown in Fig. 18.

The influencing factors for dam displacement are water pressure, temperature, and aging. Considering the lagging effect of these factors on the displacement of the dam, the water pressure factors are taken as H, H2, H3, and H4, where H represents the water depth in front of the dam. The temperature factors are taken as $sin\frac{2\pi it}{365}, cos\frac{2\pi it}{365}, sin\frac{4\pi it}{365}, and\ cos\frac{4\pi it}{365}$. The time limitation factors are θ and $ln\theta$, where $\theta = (monitoringdayorder - basedayorder)/100$. The experimental procedure and details of this experiment are as follows:

1. KICA was used to extract three independent displacement components (IC1, IC2, and IC3) from the displacement data of 24 measurement points and the 10 environmental influences mentioned above. The contribution rates of IC1, IC2, IC3, and other independent components were 95.12%, 3.36%, 0.99%, and 0% respectively. The method effectively eliminated the interference of redundant information to the model and integrated 99.3% of the main information from the multi-survey point displacement monitoring data.

2. The contribution weights of the three independent components were combined and analyzed to obtain an integrated data IC, i.e. IC = IC1 × 95.12% + IC2 × 3.36% + IC3 × 0.99%. The independent components of the environmental impact factor and the effector IC were used as inputs and outputs of the model. The training sample for the SVM model was the first 220 data of the monitoring day series and the prediction sample was the last 54 data.

3. The root mean square error (RMSE) was used as the fitness function and the proposed GWCA algorithm was used to determine the optimal parameters of the SVM.

4. Finally, a KICA-GWCA-SVM Spatio-temporal monitoring model was established to reflect the overall Spatio-temporal deformation property of the dam.

In this subsection, the parameters of the GWCA are the number of populations (50) and the maximum number of iterations (200). The best parameters for the penalty factor and kernel function obtained by training the support vector machine model were c = 0.28199 and σ = 4.0315. Fig. 19 shows the complete set of fit and prediction results calculated from this model.

We used the complex correlation coefficient (R^2) as the accuracy evaluation index, mean squared error (MSE), mean absolute error (MAE), and mean absolute percentage (MAPE) to further evaluate the fitting and prediction accuracy of the model. The statistical results of the model are shown in Table 14. As shown in Table 14, the proposed model can weaken the effects of multicollinearity to a certain extent and is, therefore, more suitable for the spatio-temporal monitoring of dams.

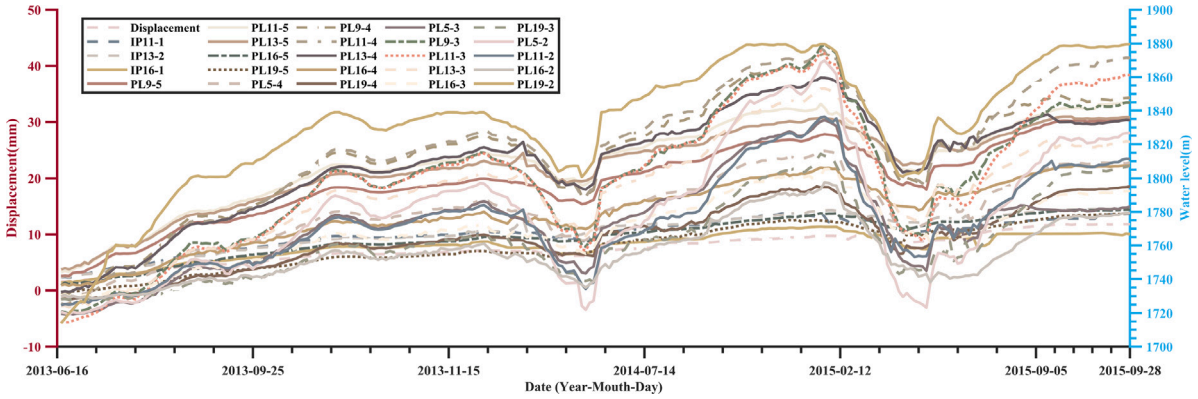


Fig. 18. The process line of reservoir water level and the measured displacement value of the measuring point.

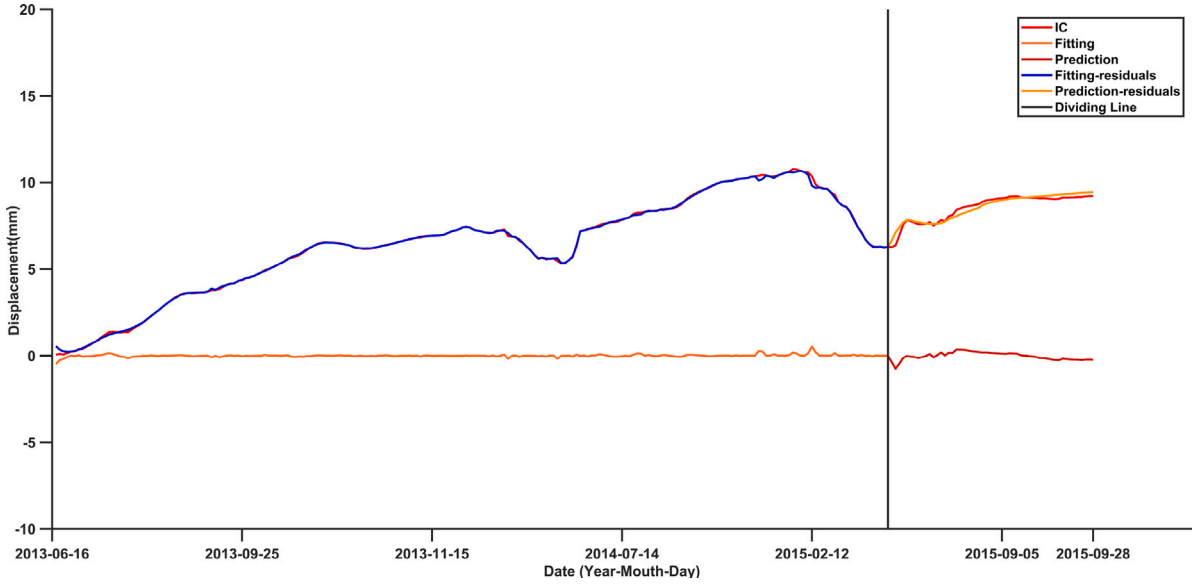


Fig. 19. The fitting and prediction of KICA-GWCA-SVM.

Table 14
Calculation results of model error and complex correlation coefficient.

MAE (mm)		MSE (mm)		MAPE (mm)		RMSE (mm)		R^2	
Fitting	Prediction	Fitting	Prediction	Fitting	Prediction	Fitting	Prediction	Fitting	Prediction
0.03623	0.17709	0.0054912	0.047728	0.063788	0.021421	0.074103	0.21847	0.99968	0.95988

8. Conclusions and future prospects

This paper presents a novel meta-heuristic algorithm called the Great Wall Construction Algorithm (GWCA), which is inspired by the ancient Great Wall construction process. Unlike conventional meta-heuristics that draw inspiration from natural or mathematical laws, GWCA simulates the movement patterns of workers during the Great Wall construction. The algorithm is population-based and controlled by worker movements. The primary objective of this paper is to evaluate the basic performance of GWCA using 29 IEEE CEC2017 benchmark functions with varying dimensions (D = 10, 30, 50, and 100). The results obtained from GWCA are compared with those of 33 advanced meta-heuristic algorithms. The experimental findings demonstrate that GWCA is effective and efficient in obtaining optimal solutions. Furthermore, this paper evaluates the optimization performance of GWCA on 16 engineering design problems and 6 NP-Hard problems. The results indicate that GWCA can successfully address production engineering problems while reducing the associated costs. Consequently, it can be considered as a primary optimization method in various domains such as Health Hygiene, Social Networking, and Energy Climate, among others.

While this paper focuses on analyzing the effectiveness of GWCA and its basic performance, the algorithm holds potential for further research directions. Firstly, GWCA can be applied to optimize the architecture of artificial neural networks, enhancing classification and prediction efficiency based on its convergence performance and ability to escape local optima. Additionally, GWCA can be employed in parameter optimization for proton-exchange membrane fuel cell (PEMFC) models, neural network training, support vector machine marginal adjustment, parameter adjustment for multiple machine learning algorithms, multilevel image thresholding problems, optimal heat exchanger design problems, PID controller step response tuning, solar PV parameter estimation, and other engineering optimization problems. However, it should be noted that GWCA may have limitations

in practical time-constrained optimization problems due to its slower convergence rate.

Secondly, although this paper presents the fundamental structure of GWCA, the algorithm can be explored in various areas, and researchers can investigate its variants to enhance its performance. Potential research directions include developing binary versions of GWCA, incorporating additional emperor agents to lead different worker groups, combining GWCA with other well-known optimization algorithms, and exploring the integration of greedy search, co-evolutionary methods, quantum computing, parallel computing, and other mechanisms to improve GWCA's convergence rate.

Lastly, future work can involve developing multi-objective and discrete versions of GWCA to tackle multi-objective optimization problems such as group network optimization and robot path planning.

CRediT authorship contribution statement

Ziyu Guan: Conceptualization, Supervision, Methodology, Software, Investigation, Validation, Writing – original draft. **Changjiang Ren:** Funding acquisition, Supervision, Methodology, Writing – original draft, Writing – review & editing, Software. **Jingtai Niu:** Writing – original draft, Visualization, Investigation. **Peixi Wang:** Writing – original draft, Investigation. **Yizi Shang:** Supervision, Writing – review.

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

The research was supported by the National Natural Science Foundation of China (Grant No. 51909116) and Research on the Spatiotemporal Analysis Method of Deformation Behavior and Time-Varying Risk Rate for Super-High Arch Dam (Grant No. 51969018). We thank Editideas (www.editideas.cn) for its linguistic assistance during the preparation of this manuscript.

Appendix A. Mathematical model of constraint problems

A.1. Constrain problem 1

Consider:

$$X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]$$

Minimize:

$$f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

Subject to:

$$g_1(x) = 127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \geqslant 0$$
$$g_2(x) = 282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 \geqslant 0$$
$$g_3(x) = 196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 \geqslant 0$$
$$g_4(x) = -4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7 \geqslant 0$$

Variable range:

$$-10 \leqslant x_i \leqslant 10 \quad i = 1, 2, 3, 4, 5, 6, 7$$

A.2. Constrain problem 2

Consider:

$$X = [x_1, x_2, x_3, x_4, x_5]$$

Minimize:

$$f(x) = 5.3578547x_3^3 + 0.8356891x_1x_5 + 37.293239x_1 + 40729.141$$

Subject to:

$$g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leqslant 0$$
$$g_2(x) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 - 0.0022053x_3x_5 \leqslant 0$$
$$g_3(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leqslant 0$$
$$g_4(x) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leqslant 0$$
$$g_5(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leqslant 0$$
$$g_6(x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leqslant 0$$

Variable range:

$$78 \leqslant x_1 \leqslant 10233 \leqslant x_2 \leqslant 4527 \leqslant x_i \leqslant 45 \quad i = 3, 4, 5$$

A.3. Constrain problem 3

Consider:

$$X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}]$$

Minimize:

$$f(x) = -(\sqrt{n})^n \cdot \prod_{i=1}^{n} x_i$$

Subject to:

$$h(x) = \sum_{i=1}^{n} x_i^2 = 1$$

Variable range:

$$0 \leqslant x_i \leqslant 1 \quad i = 1, \ldots, n$$

Appendix B. Mathematical model of engineering model

B.1. Speed reducer

Consider:

$$X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, z, l_1, l_2, d_1, d_2];$$

minimize:

$$f(X) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2),$$

subject to:

$$g_1(X) = \frac{27}{x_1 x_2^2 x_3} - 1 \le 0$$
$$g_2(X) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0$$
$$g_3(X) = \frac{1.93 x_4^3}{x_2 x_6^4 x_3} - 1 \le 0$$
$$g_4(X) = \frac{1.93 x_5^3}{x_2 x_7^4 x_3} - 1 \le 0$$
$$g_5(X) = \frac{\sqrt{\left(745 x_4 / x_2 x_3\right)^2 + 16.9\times 10^6}}{110 x_6^3} - 1 \le 0$$
$$g_6(X) = \frac{\sqrt{\left(745 x_5 / x_2 x_3\right)^2 + 157.5\times 10^6}}{85 x_7^3} - 1 \le 0$$
$$g_7(X) = \frac{x_2 x_3}{40} - 1 \le 0$$
$$g_8(X) = \frac{5 x_2}{x_1} - 1 \le 0$$
$$g_9(X) = \frac{x_1}{12 x_2} - 1 \le 0$$
$$g_{10}(X) = \frac{1.5 x_6 + 1.9}{x_4} - 1 \le 0$$
$$g_{11}(X) = \frac{1.1 x_7 + 1.9}{x_5} - 1 \le 0$$

variable range:

$$2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8$$
$$x_3 \in \{17, 18, 19, \ldots, 28\}, 7.3 \le x_4, x_5 \le 8.3$$
$$2.9 \le x_6 \le 3.9, 5 \le x_7 \le 5.5.$$

B.2. Tension/compression spring design

Consider:

$$X = [x_1, x_2, x_3] = [D, d, N];$$

minimize:

$$f(X) = \left(x_3 + 2\right) x_2 x_1^2,$$

subject to:

$$g_1(X) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0$$
$$g_2(X) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} - 1 \le 0$$
$$g_3(X) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0$$
$$g_4(X) = \frac{x_1 + x_2}{1.5} - \le 0$$

variable range:

$$0.05 \le x_1 \le 2$$
$$0.25 \le x_2 \le 1.3$$
$$2 \le x_3 \le 15.$$

B.3. Rolling element bearing

Consider:

$$X = [D_m, D_b, Z, f_i, f_o, K_{Dmin}, K_{Dmax}, \varepsilon, e, \xi]$$

max:

$$C_d = f_c Z^{2/3} D_b^{1.8} if D \leqslant 25.4 \text{ mm}$$
$$C_d = 3.647 f_c Z^{2/3} D_b^{1.4} \quad if D \succ 25.4 \text{ mm}$$

subject to:

$$g_1(x) = \frac{\phi_0}{2\sin^{-1}(D_b/D_m)} - Z + 1 \geqslant 0$$
$$g_2(x) = 2D_b - K_{D\min}(D - d) \geqslant 0$$
$$g_3(x) = K_{D\max}(D - d) - 2D_b \geqslant 0$$
$$g_4(x) = \zeta B_w - D_b \leqslant 0$$
$$g_5(x) = D_m - 0.5(D + d) \geqslant 0$$
$$g_6(x) = (0.5 + e)(D + d) - D_m \geqslant 0$$
$$g_7(x) = 0.5\left(D - D_m - D_b\right) - \varepsilon D_b \geqslant 0$$
$$g_8(x) = f_i \geqslant 0.515$$
$$g_9(x) = f_o \geqslant 0.515$$

variable range:

$$0.4 \leqslant K_{D\min} \leqslant 0.5, 0.6 \leqslant K_{D_{\max}} \leqslant 0.7, 0.3 \leqslant \varepsilon \leqslant 0.4$$
$$0.02 \leqslant e \leqslant 0.1, 0.6 \leqslant \zeta \leqslant 0.85$$

where,

$$f_c = 37.91\left[1 + \left\{1.04\left(\frac{1-\gamma}{1+\gamma}\right)^{1.72}\left(\frac{f_i(2f_o - 1)}{f_0(2f_i - 1)}\right)^{0.41}\right\}^{10/3}\right]^{-0.3}$$
$$\times\left[\frac{\gamma^{0.3}(1-\gamma)^{1.39}}{(1+\gamma)^{1/3}}\right]\left[\frac{2f_i}{2f_i - 1}\right]^{0.41}$$
$$\phi_0 = 2\pi - 2\times\cos^{-1}\left(\frac{\left[\{(D-d)/2 - 3(T/4)\}^2 + \{D/2 - T/4 - D_b\}^2 - \{d/2 + T/4\}^2\right]}{2\{(D-d)/2 - 3(T/4)\}\{D/2 - T/4 - D_b\}}\right)$$
$$\gamma = \frac{D_b}{D_m}, \quad f_i = \frac{r_i}{D_b}, \quad f_0 = \frac{r_0}{D_b} \quad T = D - d - 2D_b$$
$$D = 160, \quad d = 90, \quad B_w = 30, \quad r_i = r_o = 11.033$$
$$0.5(D + d) \leqslant Dm \leqslant 0.6(D + d), 0.15(D - d) \leqslant Db \leqslant 0.45(D - d)$$
$$4 \leqslant Z \leqslant 50, 0.515 \leqslant f_i \text{ and } f_o \leqslant 0.6$$

B.4. Multiple disk clutch brake design problems

Consider:

$$X = [r_i, r_o, t, F, Z]$$

minimize:

$$f(X) = \pi\left(r_o^2 - r_i^2\right) t(Z + 1)\rho$$

subject to:

$$g_1(x) = r_o - r_i - \Delta r \geqslant 0$$
$$g_2(x) = l_{\max} - (Z + 1)(t + \delta) \geqslant 0$$
$$g_3(x) = p_{\max} - p_{rz} \geqslant 0$$
$$g_4(x) = p_{\max} v_{\text{sr max}} - p_{rz} v_{sr} \geqslant 0$$
$$g_5(x) = v_{\text{sr max}} - v_{\text{sr}} \geqslant 0$$
$$g_6(x) = T_{\max} - T \geqslant 0$$
$$g_7(x) = M_h - s M_s \geqslant 0$$
$$g_8(x) = T \geqslant 0$$

variable range:

$$x_1 \in \{60, 61, 62, \ldots, 80\}$$
$$x_2 \in \{90, 91, 92, \ldots, 110\}$$
$$x_3 \in \{1, 1.5, 2, \ldots, 3\}$$
$$x_4 \in \{600, 601, 602, \ldots, 1000\}$$
$$x_5 \in \{2, 3, 4, \ldots, 9\}$$

where,

$$M_h = \frac{2}{3}\mu F Z \frac{r_0^3 - r_i^3}{r_0^2 - r_i^2}, \quad p_{rz} = \frac{F}{\pi\left(r_0^2 - r_i^2\right)} v_{rz} = \frac{2\pi n\left(r_0^3 - r_i^3\right)}{90\left(r_0^2 - r_i^2\right)}, \quad T = \frac{I_z \pi n}{30(M_h + M_f)}$$
$$\Delta r = 20 \text{ mm}, \quad I_z = 55 \text{ kgmmkm}^2, \quad p_{\max} = 1 \text{ MPa}, \quad F_{\max} = 1000 \text{ N},$$
$$T_{\max} = 15 \text{ s}, \quad \mu = 0.5, \quad s = 1.5, \quad M_s = 40\text{Nm},$$
$$M_f = 3 \text{ Nm}, \quad n = 250 \text{ rpm}, v_{\text{srmax}} = 10 \text{ m/s}, l_{\max} = 30 \text{ mm},$$
$$r_{i\min} = 60, r_{\text{imax}} = 80, \quad r_{o\min} = 90, r_{o\max} = 110,$$
$$t_{\min} = 1.5, t_{\max} = 3, \quad F_{\min} = 600, \quad F_{\max} = 1000, \quad Z_{\min} = 2, Z_{\max} = 9.$$

B.5. Pressure vessel design

Consider:

$$X = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L];$$

Minimize:

$$f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3,$$

Subject to:

$$g_1(X) = -x_1 + 0.0193x_3 \le 0$$
$$g_2(X) = -x_2 + 0.00954x_3 \le 0$$
$$g_3(X) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \le 0$$
$$g_4(X) = x_4 - 240 \le 0$$

Variable range:

$$x_1, x_2 \in \{1 \times 0.0625, 2 \times 0.0625, 3 \times 0.0625, \ldots, 1600 \times 0.0625\},$$
$$10 \le x_3,$$
$$x_4 \le 200.$$

B.6. Three-bar truss design problem

Consider:

$X = [x_1, x_2] = [A_1, A_2]$;

Minimize:

$f(X) = (2\sqrt{2}x_1 + x_2) * l$;

Subject to:

$$g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0;$$
$$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0;$$
$$g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \le 0;$$

Variable range:

$0 \le x_1, x_2 \le 1$;

where

$l = 100$ cm, $P = 2$ KN/cm^2, $\sigma = 2$ KN/cm^2.

B.7. Design of gear train

Consider:

$X = [x_1, x_2, x_3, x_4] = [n_A, n_B, n_C, n_D]$;

Minimize:

$f(X) = (\frac{1}{6.391} - \frac{x_3x_2}{x_1x_4})^2$;

Variable range:

$x_1, x_2, x_3, x_4 \in \{12, 13, \ldots, 60\}$;

B.8. Cantilever beam design problem

Consider:

$X = [x_1, x_2, x_3, x_4, x_5]$;

Minimize:

$f(X) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5)$;

Subject to:

$$g(x) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} \le 1;$$

Variable range:

$0.01 \le x_1, x_2, x_3, x_4, x_5 \le 100$.

B.9. Design of I-shaped beam

Consider:

$X = [x_1, x_2, x_3, x_4] = [b, h, t_w, t_f]$;

Minimize:

$$f(X) = \frac{5000}{x_3(x_2 - 2x_4)^3/12 + (x_1x_4^3/6) + 2bx_4(x_2 - x_4/2)^2};$$

Subject to:

$$g_1(x) = 2x_1x_3 + x_3(x_2 - 2x_4) \le 300;$$
$$g_2(x) = \frac{18x_2 \times 10^4}{x_3(x_2 - 2x_4)^3 + 2x_1x_3(4x_4^2 + 3x_2(x_2 - 2x_4))} + \frac{15x_1 \times 10^3}{(x_2 - 2x_4)x_3^2 + 2x_3x_1^3} \le 56;$$

Variable range:

$10 \le x_1 \le 50$;
$10 \le x_2 \le 80$;
$0.9 \le x_3 \le 5$;
$0.9 \le x_4 \le 5$.

B.10. Tubular column design problem

Consider:

$X = [x_1, x_2] = [d, t]$;

Minimize:

$f(X) = 9.8x_1x_2 + 2x_1$;

Subject to:

$$g_1(X) = \frac{P}{\pi x_1x_2\sigma_y} - 1 \le 0;$$
$$g_2(X) = \frac{8PL^2}{\pi^3 Ex_1x_2(x_1^2 + x_2^2)} - 1 \le 0;$$
$$g_3 1(X) = \frac{2.0}{x_1} - 1 \le 0;$$
$$g_4(X) = \frac{x_1}{14} \le 0;$$
$$g_5(X) = \frac{0.2}{x_2} - 1 \le 0;$$
$$g_6(X) = \frac{x_2}{8} - 1 \le 0;$$

Variable range:

$2 \le x_1 \le 14$;
$0.2 \le x_2 \le 0.8$.

B.11. Piston lever

Consider:

$X = [x_1, x_2, x_3, x_4] = [H, B, D, X]$;

minimize:

$$f(X) = \frac{1}{4}\pi x_3^2\left(L_2 - L_1\right)$$

subject to:

$$g_1(X) = QL\cos\theta - R \times F \le 0$$
$$g_2(X) = Q\left(L - x_4\right) - M_{\max} \le 0$$
$$g_3(X) = 1.2\left(L_2 - L_1\right) - L_1 \le 0,$$
$$g_4(X) = \frac{x_3}{2} - x_2 \le 0,$$

where,

$$R = \frac{|-x_4(x_4\sin\theta+x_1)+x_1(x_2-x_4\cos\theta)|}{\sqrt{(x_4-x_2)^2+x_1^2}},$$
$$F = \frac{\pi P x_3^2}{4},$$
$$L_1 = \sqrt{(x_4-x_2)^2+x_1^2},$$
$$L_2 = \sqrt{(x_4\sin\theta+x_1)^2+(x_2-x_4\cos\theta)^2},$$
$$\theta = {}^{45^\circ}, Q = 10,000\text{lbs}, L = 240\text{in}, M_{\max} = 1.8\times10^6\text{lbs in}, P = 1500\text{psi},$$

variable range:

$$0.05 \le x_i \le 500, \quad i = 1,\dots,3,$$
$$0.05 \le x_5 \le 120.$$

B.12. Corrugated bulkhead design

Consider:

$$X = [x_1, x_2, x_3, x_4];$$

minimize:

$$f(X) = \frac{5.885x_4\left(x_1+x_3\right)}{x_1+\sqrt{\left|x_3^2-x_2^2\right|}}$$

subject to:

$$g_1(X) = -x_4x_2\left(0.4x_1+\tfrac{x_3}{6}\right)+8.94\left(x_1+\sqrt{|\,x_3^2-x_2^2\,|}\right)\le 0,$$
$$g_2(X) = -x_4x_2^2\left(0.2x_1+\tfrac{x_3}{12}\right)+2.2\left(8.94\left(x_1+\sqrt{\left|x_3^2-x_2^2\right|}\right)\right)^{4/3}\le 0,$$
$$g_3(X) = -x_4+0.0156x_1+0.15\le 0,$$
$$g_4(X) = -x_4+0.0156x_3+0.15\le 0,$$
$$g_5(X) = -x_4+1.05\le 0,$$
$$g_6(X) = -x_3+x_2\le 0,$$

variable range:

$$0\le x_1,x_2,x_3\le 100$$
$$0\le x_4\le 5.$$

B.13. Car side impact design

Consider:

$$X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}]$$

Minimize:

$$f(X) = 1.98+4.90x_1+6.67x_2+6.98x_3+4.01x_4+1.78x_5+2.73x_7$$

Subject to:

$$\begin{aligned} g_1(X) =& 1.16-0.3717x_2x_4-0.00931x_2x_{10}\\ &-0.484x_3x_9+0.01343x_6x_{10}-1\le 0\end{aligned}$$
$$g_2(X) = 46.36-9.9x_2-12.9x_1x_2+0.1107x_3x_{10}-32\le 0$$
$$\begin{aligned} g_3(X) =& 33.86+2.95x_3+0.1792x_3-5.057x_1x_2\\ &-11.0x_2x_8-0.0215x_5x_{10}-9.98x_7x_8+22.0x_8x_9-32\le 0\end{aligned}$$
$$\begin{aligned} g_4(X) =& 28.98+3.818x_3-4.2x_1x_2+0.0207x_5x_{10}\\ &+6.63x_6x_9-7.7x_7x_8+0.32x_9x_{10}-32\le 0\end{aligned}$$
$$\begin{aligned} g_5(X) =& 0.261-0.0159x_1x_2-0.188x_1x_8-0.019x_2x_7\\ &+0.0144x_3x_5+0.0008757x_5x_{10}+0.08045x_6x_9\\ &+0.00139x_8x_{11}+0.00001575x_{10}x_{11}-0.32\le 0\end{aligned}$$
$$\begin{aligned} g_6(X) =& 0.214+0.00817x_5-0.131x_1x_8-0.0704x_1x_9+0.03099x_2x_6\\ &-0.018x_2x_7+0.0208x_3x_8+0.121x_3x_9-0.00364x_5x_6\\ &+0.0007715x_5x_{10}-0.0005354x_6x_{10}+0.00121x_8x_{11}\\ &+0.00184x_9x_{10}-0.02x_2^2-0.32\le 0\end{aligned}$$
$$\begin{aligned} g_7(X) =& 0.74-0.61x_2-0.163x_3x_8+0.001232x_3x_{10}\\ &-0.166x_7x_9+0.227x_2^2-0.32\le 0\end{aligned}$$
$$\begin{aligned} g_8(X) =& 4.76-0.5x_4-0.19x_2x_3-0.0122x_4x_{10}\\ &+0.009325x_6x_{10}+0.000191x_{11}^2-4\le 0\end{aligned}$$
$$\begin{aligned} g_9(X) =& 10.58-0.674x_1x_2-1.95x_2x_8\\ &+0.02054x_3x_{10}-0.0198x_4x10+0.028x_6x10-9.9\le 0\end{aligned}$$
$$\begin{aligned} g_{10}(X) =& 16.45-0.489x_3x_7-0.843x_5x_6\\ &+0.0432x_9x_{10}-0.0556x_9x11-0.000786x_{11}^2-15.7\le 0\end{aligned}$$

Variable range:

$$0.5\le x_1,x_2,x_3,x_4,x_5,x_6,x_7\le 1.5$$
$$x_8,x_9 = \{0.192, 0.345\}$$
$$-30\le x_{10},x_{11}\le +30$$

B.14. Design of welded beam design

Consider:

$$X = [x_1, x_2, x_3, x_4] = [h, l, t, b];$$

minimize:

$$f(X) = 1.10471x_1^2x_2+0.04811x_3x_4\left(14.0+x_2\right),$$

subject to:

$$\begin{aligned} g_1(X) &= \tau(X)-\tau_{\max}\le 0\\ g_2(X) &= \sigma(X)-\sigma_{\max}\le 0\\ g_3(X) &= \delta(X)-\delta_{\max}\le 0\\ g_4(X) &= x_1-x_4\le 0\\ g_5(X) &= P-P_c(X)\le 0\\ g_6(X) &= 0.125-x_1\le 0\\ g_7(X) &= 1.10471x_1^2+0.04811x_3x_4\left(14.0+x_2\right)-5.0\le 0,\\ \tau(X) &= \sqrt{(\tau')^2+2\tau'\tau''\frac{x_2}{2R}+(\tau'')^2},\\ \tau' &= \frac{P}{\sqrt{2}x_1x_2},\\ \tau'' &= \frac{MR}{J},\\ M &= P\left(L+\frac{x_2}{2}\right),\\ R &= \sqrt{\frac{x_2^2}{4}+\left(\frac{x_1+x_3}{2}\right)^2},\\ J &= 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4}+\left(\frac{x_1+x_3}{2}\right)^2\right]\right\},\\ \sigma(\vec{X}) &= \frac{6PL}{x_4x_3^2},\\ \delta(\vec{X}) &= \frac{6PL^3}{Ex_3^2x_4}\\ P_c(\vec{X}) &= \frac{4.013E\sqrt{x_3^2x_4^6/36}}{L^2}\left(1-\frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),\\ P &= 6000\text{lb}, \end{aligned}$$

$$
\begin{aligned}
L &= 14\text{in}, \\
\delta_{\max} &= 0.25\text{in}, \\
E &= 30 \times 10^6 \text{psi} \\
G &= 12 \times 10^6 \text{psi} \\
\tau_{\max} &= 13,600\text{psi} \\
\sigma_{\max} &= 30,000\text{psi},
\end{aligned}
$$

variable range:

$$
\begin{aligned}
&0.1 \le x_1, x_4 \le 2 \\
&0.1 \le x_2, x_3 \le 10.
\end{aligned}
$$

B.15. Reinforced concrete beam design problem

Consider:

$$X = [x_1, x_2, x_3] = [A_s, b, h];$$

Minimize:

$$f(X) = 2.9x_1 + 0.6x_2x_3;$$

Subject to:

$$g_1(X) = \frac{x_2}{x_3} - 4 \le 0;$$

$$g_2(X) = 180 + 7.375\frac{x_1^2}{x_3} - x_1x_2 \le 0;$$

Variable range:

$$x_1 \in 6, 6.16, 6.32, 6.6, 7, 7.11, 7.2, 7.8, 7.9, 8, 8.4;$$

$$x_2 \in 28, 29, 30, \ldots, 40;$$

$$5 \le x_3 \le 10. \tag{19}$$

References

Abdel-Basset, M., Abdel-Fatah, L., & Sangaiah, A. K. (2018). Metaheuristic algorithms: A comprehensive review. *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, 185–231.

Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z. W., & Gandomi, A. H. (2022). Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications*, *191*, Article 116158.

Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*, *376*, Article 113609.

Abualigah, L., Elaziz, M. A., Khasawneh, A. M., Alshinwan, M., Ibrahim, R. A., Al-qaness, M. A., Mirjalili, S., Sumari, P., & Gandomi, A. H. (2022). Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: A comprehensive survey, applications, comparative analysis, and results. *Neural Computing and Applications*, 1–30.

Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A., & Gandomi, A. H. (2021). Aquila optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, *157*, Article 107250.

Ahmadianfar, I., Heidari, A. A., Gandomi, A. H., Chu, X., & Chen, H. (2021). RUN beyond the metaphor: An efficient optimization algorithm based on runge kutta method. *Expert Systems with Applications*, *181*, Article 115079.

Akbaripour, H., & Masehian, E. (2013). Efficient and robust parameter tuning for heuristic algorithms.

Al-Betar, M. A., Alyasseri, Z. A. A., Awadallah, M. A., & Abu Doush, I. (2021). Coronavirus herd immunity optimizer (CHIO). *Neural Computing and Applications*, *33*(10), 5011–5042.

Al-Betar, M. A., Alyasseri, Z. A. A., Awadallah, M. A., & Doush, I. A. (2020). Coronavirus herd immunity optimizer (CHIO). *Neural Computing and Applications*, (5).

Anita, & Yadav, A. (2019). AEFA: Artificial electric field algorithm for global optimization. *Swarm and Evolutionary Computation*, *48*, 93–108. http://dx.doi.org/10.1016/j.swevo.2019.03.013, URL https://www.sciencedirect.com/science/article/pii/S2210650218305030.

Askari, Q., Saeed, M., & Younas, I. (2020). Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Systems with Applications*, *161*, Article 113702.

Askari, Q., Younas, I., & Saeed, M. (2020). Political optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowledge-Based Systems*, *195*, Article 105709.

Awad, N. H., Ali, M. Z., & Suganthan, P. N. (2017). Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving CEC2017 benchmark problems. In *2017 IEEE congress on evolutionary computation* (pp. 372–379). http://dx.doi.org/10.1109/CEC.2017.7969336.

Ayyarao, T. S., Ramakrishna, N., Elavarasan, R. M., Polumahanthi, N., Rambabu, M., Saini, G., Khan, B., & Alatas, B. (2022). War strategy optimization algorithm: A new effective metaheuristic algorithm for global optimization. *IEEE Access*, *10*, 25073–25105.

Bäck, T. (1995). Evolution strategies: An alternative evolutionary algorithm. In *European conference on artificial evolution* (pp. 1–20). Springer.

Bai, B., Guo, Z., Zhou, C., Zhang, W., & Zhang, J. (2021). Application of adaptive reliability importance sampling-based extended domain PSO on single mode failure in reliability engineering. *Information Sciences*, *546*, 42–59.

Bergh, F., & Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information Sciences*, *176*(8), 937–971.

Chen, H., Heidari, A. A., Zhao, X., Zhang, L., & Chen, H. (2020). Advanced orthogonal learning-driven multi-swarm sine cosine optimization: Framework and case studies. *Expert Systems with Applications*, *144*, Article 113113.

Chouhan, V. K., Khan, S. H., & Hajiaghaei-Keshteli, M. (2022). Sustainable planning and decision-making model for sugarcane mills considering environmental issues. *Journal of Environmental Management*, *303*, Article 114252.

Črepinšek, M., Liu, S. H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys*, *45*(3), 1–33.

Cui, Z., & Cai, X. (2013). Artificial plant optimization algorithm. In *Swarm intelligence and bio-inspired computation* (pp. 351–365). Elsevier.

Das, B., Mukherjee, V., & Das, D. (2020). Student psychology based optimization algorithm: A new population based optimization algorithm for solving optimization problems. *Advances in Engineering Software*, *146*, Article 102804.

Das, S., & Suganthan, P. N. (2010). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, *15*(1), 4–31.

De Castro, L. N., & Von Zuben, F. J. (2000). The clonal selection algorithm with engineering applications. In *Proceedings of GECCO, Vol. 2000* (pp. 36–39).

Dhal, K. G., Ray, S., Das, A., & Das, S. (2019). A survey on nature-inspired optimization algorithms and their application in image enhancement domain. *Archives of Computational Methods in Engineering*, *26*(5), 1607–1638.

Dhiman, G., Garg, M., Nagar, A., Kumar, V., & Dehghani, M. (2021). A novel algorithm for global optimization: rat swarm optimizer. *Journal of Ambient Intelligence and Humanized Computing*, *12*(8), 8457–8482.

Dhiman, G., & Kumar, V. (2017). Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Advances in Engineering Software*, *114*, 48–70.

Dong, W., & Zhou, M. (2016). A supervised learning and control method to improve particle swarm optimization algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *47*(7), 1135–1148.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, *1*(4), 28–39.

Duman, S., Kahraman, H. T., Guvenc, U., & Aras, S. (2021). Development of a levy flight and FDB-based coyote optimization algorithm for global optimization and real-world ACOPF problems. http://dx.doi.org/10.1007/s00500-021-05654-z.

Duman, S., Kahraman, H. T., Sonmez, Y., Guvenc, U., Kati, M., & Aras, S. (2022). A powerful meta-heuristic search algorithm for solving global optimization and real-world solar photovoltaic parameter estimation problems. *Engineering Applications of Artificial Intelligence*, *111*, http://dx.doi.org/10.1016/j.engappai.2022.104763.

Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers and Structures*, *110*, 151–166.

Ezugwu, A. E., Shukla, A. K., Nath, R., Akinyelu, A. A., Agushaka, J. O., Chiroma, H., & Muhuri, P. K. (2021). Metaheuristics: A comprehensive overview and classification along with bibliometric analysis. *Artificial Intelligence Review*, *54*(6), 4237–4316.

Fathollahi-Fard, A., & Hajiaghaei-Keshteli, M. (2016). Red deer algorithm (RDA); A new optimization algorithm inspired by red deer's mating.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2018). The social engineering optimizer (SEO). *Engineering Applications of Artificial Intelligence*, *72*, 267–293. http://dx.doi.org/10.1016/j.engappai.2018.04.009, URL https://www.sciencedirect.com/science/article/pii/S0952197618300873.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., Tian, G., & Li, Z. (2020). An adaptive Lagrangian relaxation-based algorithm for a coordinated water supply and wastewater collection network design problem. *Information Sciences*, *512*, 1335–1359.

Feng, Y., Deb, S., Wang, G.-G., & Alavi, A. H. (2021). Monarch butterfly optimization: A comprehensive review. *Expert Systems with Applications*, *168*, Article 114418.

Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: A systematic review. *Archives of Computational Methods in Engineering*, 1–31.

Garip, Z., Karayel, D., & Erhan Çimen, M. (2022). A study on path planning optimization of mobile robots based on hybrid algorithm. *Concurrency Computations: Practice and Experience*, *34*(5), Article e6721.

Ghafil, H. N., & Jármai, K. (2020). Dynamic differential annealed optimization: New metaheuristic optimization algorithm for engineering applications. *Applied Soft Computing*, *93*, Article 106392.

Gholian-Jouybari, F., Hashemi-Amiri, O., Mosallanezhad, B., & Hajiaghaei-Keshteli, M. (2023). Metaheuristic algorithms for a sustainable agri-food supply chain considering marketing practices under uncertainty. *Expert Systems with Applications*, *213*, Article 118880.

Guvenc, U., Duman, S., Kahraman, H. T., Aras, S., & Katı, M. (2021). Fitness–distance balance based adaptive guided differential evolution algorithm for security-constrained optimal power flow problem incorporating renewable energy sources. *Applied Soft Computing*, *108*, Article 107421. http://dx.doi.org/10.1016/j.asoc.2021.107421, URL https://www.sciencedirect.com/science/article/pii/S1568494621003446.

Harifi, S., Mohammadzadeh, J., Khalilian, M., & Ebrahimnejad, S. (2021). Giza pyramids construction: An ancient-inspired metaheuristic algorithm for optimization. *Evolutionary Intelligence*, *14*(4), 1743–1761.

Hayyolalam, V., & Kazem, A. A. P. (2020). Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, *87*, Article 103249.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, *97*, 849–872.

Holland, J. H. (1992). Genetic algorithms. *Scientific American*, *267*(1), 66–73.

Hussain, K., Salleh, M. N. M., Cheng, S., & Shi, Y. (2019). On the exploration and exploitation in popular swarm-based metaheuristic algorithms. *Neural Computing and Applications*, *31*, 7665–7683.

Jiang, Q., Shao, F., Lin, W., Gu, K., Jiang, G., & Sun, H. (2017). Optimizing multistage discriminative dictionaries for blind image quality assessment. *IEEE Transactions on Multimedia*, *20*(8), 2035–2048.

Jing, A. (2015). *A History of the Great Wall of China*. Singapore: World Scientific.

Kashani, A. R., Camp, C. V., Rostamian, M., Azizi, K., & Gandomi, A. H. (2022). Population-based optimization in structural engineering: A review. *Artificial Intelligence Review*, 1–108.

Kaur, S., Awasthi, L. K., Sangal, A., & Dhiman, G. (2020). Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence*, *90*, Article 103541.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks, Vol. 4* (pp. 1942–1948). IEEE.

Kirkpatrick, S., Gelatt. Jr., C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.

Kumar, A., Wu, G., Ali, M. Z., Mallipeddi, R., Suganthan, P. N., & Das, S. (2020). A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation*, *56*, Article 100693.

Lam, A., & Li, V. O. (2012). Chemical reaction optimization: A tutorial. *Memetic Computing*, *4*(1), 3–17.

Mahmoodabadi, M., Rasekh, M., & Zohari, T. (2018). TGA: Team game algorithm. *Future Computing and Informatics Journal*, *3*(2), 191–199.

Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics*, *1*(4), 355–366.

Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, *96*, 120–133.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61. http://dx.doi.org/10.1016/j.advengsoft.2013.12.007, URL https://www.sciencedirect.com/science/article/pii/S0965997813001853.

Mosallanezhad, B., Chouhan, V. K., Paydar, M. M., & Hajiaghaei-Keshteli, M. (2021). Disaster relief supply chain design for personal protection equipment during the COVID-19 pandemic. *Applied Soft Computing*, *112*, Article 107809.

Mosallanezhad, B., Hajiaghaei-Keshteli, M., & Triki, C. (2021). Shrimp closed-loop supply chain network design. *Soft Computing*, *25*(11), 7399–7422.

Nayar, N., Gautam, S., Singh, P., & Mehta, G. (2021). Ant colony optimization: A review of literature and application in feature selection. *Inventive Computation and Information Technologies*, 285–297.

Parpinelli, R. S., & Lopes, H. S. (2011). New inspirations in swarm intelligence: A survey. *International Journal of Bio-Inspired Computation*, *3*(1), 1–16.

Peraza-Vazquez, H., Peña-Delgado, A., Echavarría-Castillo, G., Morales-Cepeda, A., Velasco, J., & Ruiz Perez, F. (2021). A bio-inspired method for engineering design optimization inspired by dingoes hunting strategies. *Mathematical Problems in Engineering*, *2021*, 1–19. http://dx.doi.org/10.1155/2021/9107547.

Peraza-Vázquez, H., Peña-Delgado, A. F., Echavarría-Castillo, G., Morales-Cepeda, A. B., Velasco-Álvarez, J., & Ruiz-Perez, F. (2021). A bio-inspired method for engineering design optimization inspired by dingoes hunting strategies. *Mathematical Problems in Engineering*, *2021*.

Piotrowski, A. P. (2018). L-SHADE optimization algorithms with population-wide inertia. *Information Sciences*, *468*, 117–141. http://dx.doi.org/10.1016/j.ins.2018.08.030, URL https://www.sciencedirect.com/science/article/pii/S0020025518306352.

Qais, M. H., Hasanien, H. M., & Alghuwainem, S. (2020). Transient search optimization: A new meta-heuristic optimization algorithm. *Applied Intelligence*, *50*(11), 3926–3941.

Rajabi-Kafshgar, A., Gholian-Jouybari, F., Seyedi, I., & Hajiaghaei-Keshteli, M. (2023). Utilizing hybrid metaheuristic approach to design an agricultural closed-loop supply chain network. *Expert Systems with Applications*, Article 119504.

Ramezani, F., & Lotfi, S. (2013). Social-based algorithm (SBA). *Applied Soft Computing*, *13*(5), 2837–2856.

Ramshanker, A., & Chakraborty, S. (2022). Maiden application of skill optimization algorithm on cascaded multi-level neuro-fuzzy based power system stabilizers for damping oscillations. *International Journal of Renewable Energy Research (IJRER)*, *12*(4), 2152–2167.

Rao, R. V., Savsani, V. J., & Vakharia, D. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, *43*(3), 303–315.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, *179*(13), 2232–2248.

Sadeghi-Moghaddam, S., Hajiaghaei-Keshteli, M., & Mahmoodjanloo, M. (2019). New approaches in metaheuristics to solve the fixed charge transportation problem in a fuzzy environment. *Neural Computing and Applications*, *31*(1), 477–497.

Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial Intelligence*, *80*, 20–34.

Shareef, H., Ibrahim, A. A., & Mutlag, A. H. (2015). Lightning search algorithm. *Applied Soft Computing*, *36*, 315–333.

Shehab, M., Abualigah, L., Al Hamad, H., Alabool, H., Alshinwan, M., & Khasawneh, A. M. (2020). Moth–flame optimization algorithm: variants and applications. *Neural Computing and Applications*, *32*, 9859–9884.

Shen, G. (2002). Fractal dimension and fractal growth of urbanized areas. *International Journal of Geographical Information Science*, *16*(5), 419–437.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, *12*(6), 702–713.

Singh, P. R., Abd Elaziz, M., & Xiong, S. (2019). Ludo game-based metaheuristics for global and engineering optimization. *Applied Soft Computing*, *84*, Article 105723.

Talatahari, S., & Azizi, M. (2021). Chaos game optimization: A novel metaheuristic algorithm. *Artificial Intelligence Review*, *54*(2), 917–1004.

Talatahari, S., Bayzidi, H., & Saraee, M. (2021). Social network search for global optimization. *IEEE Access*, *9*, 92815–92863.

Tang, J., Liu, G., & Pan, Q. (2021). A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. *IEEE/CAA Journal of Automatica Sinica*, *8*(10), 1627–1643.

Theiler, J. (1990). Estimating fractal dimension. *Journal of the Optical Society of America A*, *7*(6), 1055–1073.

Trojovskỳ, P., & Dehghani, M. (2022). Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors*, *22*(3), 855.

Tzanetos, A., & Dounias, G. (2021). Nature inspired optimization algorithms or simply variations of metaheuristics? *Artificial Intelligence Review*, *54*(3), 1841–1862.

Waldron, A. (1990). *Cambridge studies in chinese history, literature and institutions*, *The great wall of china: From history to myth*. London: Cambridge University Press, URL https://books.google.com.hk/books?id=e4ybBQAAQBAJ.

Wansasueb, K., Panmanee, S., Panagant, N., Pholdee, N., Bureerat, S., & Yildiz, A. R. (2022). Hybridised differential evolution and equilibrium optimiser with learning parameters for mechanical and aircraft wing design. *Knowledge-Based Systems*, *239*, Article 107955.

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

Xu, Y., Cui, Z., & Zeng, J. (2010). Social emotional optimization algorithm for nonlinear constrained optimization problems. In *International conference on swarm, evolutionary, and memetic computing* (pp. 583–590). Springer.

Xue, J., & Shen, B. (2020). A novel swarm intelligence optimization approach: sparrow search algorithm. *Systems Science & Control Engineering*, *8*(1), 22–34.

Xue, X., Zhang, K., Tan, K. C., Feng, L., Wang, J., Chen, G., Zhao, X., Zhang, L., & Yao, J. (2020). Affine transformation-enhanced multifactorial optimization for heterogeneous problems. *IEEE Transactions on Cybernetics*, *52*(7), 6217–6231.

Yang, J. R. (2018). *The application of fuzzy logic and virtual reality in the study of ancient methods and materials used for the construction of the Great Wall of China in Jinshanling* (Ph.D. thesis), The Ohio State University.

Yang, X. S. (2020). Nature-inspired optimization algorithms: Challenges and open problems. *Journal of Computer Science*, *46*, Article 101104.

Yang, Z. (2023). AFO solving real-world problems. https://github.com/TwilightArchonYz/A-new-Nature-inspired-optimization-algorithm-AFO/releases/tag/1.3.

Yang, Y., Chen, H., Heidari, A. A., & Gandomi, A. H. (2021). Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Systems with Applications*, *177*, Article 114864.

Yang, Y., Chen, H., Heidari, A. A., & Gandomi, A. H. (2021). Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Systems with Applications*, *177*, Article 114864. http://dx.doi.org/10.1016/j.eswa.2021.114864, URL https://www.sciencedirect.com/science/article/pii/S0957417421003055.

Yapici, H., & Cetinkaya, N. (2019). A new meta-heuristic optimizer: Pathfinder algorithm. *Applied Soft Computing*, *78*, 545–568.

Yu, N., Delrieu, G., Boudevillain, B., Hazenberg, P., & Uijlenhoet, R. (2014). Unified formulation of single-and multimoment normalizations of the raindrop size distribution based on the gamma probability density function. *Journal of Applied Meteorology and Climatology*, *53*(1), 166–179.

Yuan, M., Li, Y., Zhang, L., & Pei, F. (2021). Research on intelligent workshop resource scheduling method based on improved NSGA-II algorithm. *Robotics and Computer-Integrated Manufacturing*, *71*, Article 102141.

Zandavi, S. M., Chung, V. Y. Y., & Anaissi, A. (2019). Stochastic dual simplex algorithm: A novel heuristic optimization algorithm. *IEEE Transactions on Cybernetics*, *51*(5), 2725–2734.

Zhan, Z. H., Shi, L., Tan, K. C., & Zhang, J. (2022). A survey on evolutionary computation for complex continuous optimization. *Artificial Intelligence Review*, 1–52.

Zhang, J. H., & Xu, X. H. (1999). An efficient evolutionary programming algorithm. *Computers & Operations Research*, *26*(7), 645–663.

Zhao, W., Wang, L., & Zhang, Z. (2019). Supply-demand-based optimization: A novel economics-inspired algorithm for global optimization. *IEEE Access*, *7*, 73182–73206.