https://doi.org/10.3311/PPci.40442

Periodica Polytechnica Civil Engineering, 69(3), pp. 925–943, 2025

Multi-objective Variants of Water Strider Algorithm for Construction Engineering Optimization Problems

Ali Kaveh[1*], Ali Akbar Shirzadi Javid[1], Yasin Vazirinia[1]

[1] School of Civil Engineering, Iran University of Science and Technology, P.O. Box 16846-13114, Narmak, Tehran, Iran
[*] Corresponding author, e-mail: alikaveh@iust.ac.ir

Received: 17 March 2025, Accepted: 16 June 2025, Published online: 02 July 2025

Abstract

Many engineering problems require optimizing multiple conflicting objectives simultaneously, necessitating efficient exploration of the design space for balanced solutions. The Water Strider Algorithm (WSA) is a robust metaheuristic technique that demonstrates superior performance compared to traditional evolutionary algorithms. This study evaluates two multi-objective variants of WSA: the Grid-based Multi-objective Water Strider Algorithm (GMOWSA) and the Non-dominated Sorting Water Strider Algorithm (NSWSA). Both variants incorporate a selection mechanism that archives and prioritizes high-quality solutions, emulating the natural behavior of water striders. The proposed methods are tested on nine multi-objective benchmark functions and three construction engineering optimization problems to assess their effectiveness. Comparative analysis against three state-of-the-art algorithms demonstrates that GMOWSA and NSWSA achieve competitive results, showcasing their potential for solving complex multi-objective optimization challenges in engineering applications.

Keywords

construction site layout planning, water strider algorithm, non-dominated sorting, grid-based multi objective, truss structures, project scheduling

1 Introduction

Many engineering and scientific applications involve optimizing multiple, often conflicting, objectives simultaneously [1]. Unlike single-objective optimization, which seeks a single optimal solution, multi-objective optimization aims to find a set of Pareto-optimal solutions, none of which can be improved without compromising another [2].

Meta-heuristic algorithms are widely used for optimization problems [3–5]. These methods efficiently find good, though not necessarily optimal, solutions. While solution quality is not guaranteed, meta-heuristics can be highly effective when well-suited to the problem.

Multi-objective optimization (MOO) addresses problems with conflicting objectives, seeking a Pareto-optimal set of solutions rather than a single optimum. Multi-objective algorithms, such as multi-objective vibrating particles system (MOVPS) [2], Non-dominated Sorting Genetic Algorithm (NSGA-II) [6] and Multi-objective Particle Swarm Optimization (MOPSO) [7], Multi-Objective Ant Lion Optimizer (MOALO) [8], and Multi-objective Colliding Bodies Optimization (MOCBO) [9], are some of the well-known metahuristics in solving multi-objective problems [10, 11].

The Water Strider Algorithm (WSA) is a nature-inspired metaheuristic optimization algorithm that mimics the social behavior and foraging strategies of water striders in their natural habitat [12]. The Water Strider Algorithm (WSA) has shown promise in single-objective optimization. The original WSA simulates the movement, mating, and predation avoidance mechanisms of water striders to solve complex optimization problems. The algorithm employs a population-based approach, where individuals represent water striders, and their positions are updated based on interactions such as territorial behavior, ripple communication, and feeding patterns. Since its inception, several variants of WSA have been proposed to enhance its performance, including the Improved Water Strider Algorithm (IWSA) [13], which incorporates adaptive parameters and chaotic maps to balance exploration and exploitation. Another version is the Binary Water Strider Algorithm (BWSA) [14], designed

Cite this article as: Kaveh, A., Shirzadi Javid, A. A., Vazirinia, Y. "Multi-objective Variants of Water Strider Algorithm for Construction Engineering Optimization Problems", Periodica Polytechnica Civil Engineering, 69(3), pp. 925–943, 2025. https://doi.org/10.3311/PPci.40442

for discrete optimization problems by modifying the position update rules. Overall, WSA and its variants have demonstrated competitive performance in diverse applications, including engineering design [13], image processing and machine learning [15]. A multi-objective version of the Water Strider Algorithm (MOWSA) [16] has been presented, which integrates elitist non-dominated sorting and crowding distance mechanisms into the original WSA to create MOWSA.

This study introduces two multi-objective variants for Water strider algorithm called NSWSA and GMOWSA, which enables WSA to deal with multi-objective optimization problems. Similar to MOWSA, NSWSA employs elitist non-dominated sorting and crowding distance mechanisms while incorporating an external archive to preserve diversity and expedite convergence. A roulette wheel selection mechanism guides the search towards promising regions. In contrast, GMOWSA employs the non-dominated sorting principles and grid-based selection mechanism to have a front of various solutions. The proposed algorithms are evaluated on nine benchmark functions and three real-world construction engineering problems (two spatial truss dome problems, a time-cost-risk trade-off project scheduling problem, and a construction site layout planning problem). Both variants demonstrate competitive performance compared to state-of-the-art algorithms.

The rest of this article is organized as follows:

Section 2 provides a background on multi-objective optimization (MOO) fundamentals, including concepts like problem formulation, Pareto dominance, optimality, optimal sets, and fronts. After an explanation of the WSA, Section 3 introduces the proposed algorithms, NSWSA and GMOWSA. Section 4 evaluates the proposed methods using numerical examples. The application of the proposed algorithms in engineering optimization problems provided in this section, and Section 6 concludes the paper and outlines potential future research directions.

2 Preliminaries

In this section, the concepts of MOO problems, non-dominated sorting and crowding distance concepts, grid-based selection mechanisms and brief description of the original WSA algorithm is presented.

2.1 Basic definitions for MOO

Multi-objective optimization compares solutions based on multiple criteria (objectives) [17]. Such problems can be formulated as a minimization problem:

$$\min_{x\in\Omega} F(\vec{x}) = \left\{f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x}), \ldots, f_o(\vec{x})\right\}, \tag{1}$$

such that:

$$g_e(\vec{x}) = 0, e = 1, 2, \ldots, E, \tag{2}$$

$$h_i(\vec{x}) \geq 0, \ i = 1, 2, \ldots, I, \tag{3}$$

$$L_v \leq x_v \leq U_v, \ v = 1, 2, \ldots V, \tag{4}$$

where $F(x)$ is the optimization variables vector, o is the number of objective functions, I is the number of inequality constraints, E is the number of equality constraints, g_e is the e-th equality constraints, h_i indicates the i-th inequality constraints, and $[L_v, U_v]$ are the boundaries of v-th variable.

Relational operators are not efficient in comparing the solutions of multi-objective problems. In this case, there must be other operators. For simplicity, we define the fundamentals of multi-objective optimization (minimization) as follows:

Definition 1 (Pareto Dominance) Considering two vectors like: $\underset{y}{\rightarrow} = \left(x_1, x_2, \ldots, x_k\right)$ and $\underset{Y}{\rightarrow} = \left(y_1, y, \ldots, y_k\right)$. Vector $\vec{X}$ is called to dominate $\vec{Y}$ (denotes as $\underset{X}{\rightarrow} \prec \underset{Y}{\rightarrow}$) if and only if [10]:

$$\begin{aligned} &\forall i \in \{1, 2, \ldots, k\} : f_i\left(\underset{x}{\rightarrow}\right) \leq f_i\left(\underset{y}{\rightarrow}\right) \wedge \exists i \in \\ &\{1, 2, \ldots, k\} \, f_i\left(\underset{x}{\rightarrow}\right) < f_i\left(\underset{y}{\rightarrow}\right). \end{aligned} \tag{5}$$

The Pareto optimality's definition is presented as follows [18, 19]:

Definition 2 (Pareto Optimality [17]) A solution $\underset{x}{\rightarrow} \in X$ is assumed Pareto-optimal if and only if:

$$\left\{\nexists \underset{y}{\rightarrow} \in X \mid \underset{x}{\rightarrow} \prec \underset{y}{\rightarrow}\right\}. \tag{6}$$

Definition 3 (Pareto optimal set) The set all Pareto-optimal solutions is defined as follows:

$$P_s = \left\{\underset{x}{\rightarrow}, \underset{y}{\rightarrow} \in X \mid \nexists \underset{x}{\rightarrow} \prec \underset{y}{\rightarrow}\right\}. \tag{7}$$

Definition 4 (Pareto optimal front) A set including the values of objective functions for Pareto solutions set:

$$P_f = \left\{f\left(\underset{x}{\rightarrow}\right) \mid \underset{x}{\rightarrow} \in P_s\right\}. \tag{8}$$

A solution is Pareto optimal if no other solution in the search space dominates it. Such solutions are also known as non-dominated.

2.2 The grid-based diversity maintenance mechanism

To maintain diversity within a limited archive size, an adaptive grid-based mechanism is employed. This mechanism divides the two-dimensional coordinate space into small boxes to assess the distribution of the Pareto optimal front [20].

2.2.1 Grid setting

The extreme values of the objective functions (F1 (minimum value ($f_1^{\min}$) and maximum value ($f_1^{\max}$)) and F2 (minimum value ($f_2^{\min}$) and maximum value ($f_2^{\max}$))) can be determined from the current archive set. All optimal individuals can be plotted in a two-dimensional coordinate system [20]. The grid division, which depends on the number of non-dominated solutions, ensures that all squares are achievable.

$$\begin{cases} \delta_{F_1} = \dfrac{\left|F_1^{\max} - F_1^{\min}\right|}{N_Q} \\ \delta_{F_2} = \dfrac{\left|F_2^{\max} - F_1^{\min}\right|}{N_Q} \end{cases}. \tag{9}$$

The feasible region divided into $N_Q \times N_Q$ pieces, and the δ_{F_1} and δ_{F_1} are the square lengths in F_1 and F_2 objective directions. Those optimal individuals scattered on those pieces need to keep specific distribution properties to ensure the Pareto front's diversity distribution [20]. For each optimal individual, it satisfies:

$$\left(F_1^{*(i)}, F_2^{*(i)}\right) \in \bigcup_{j=1,\ldots,N_Q} \left\{ \begin{bmatrix} F_1^{\min} + \left(j - \dfrac{3}{2}\right)\delta_{F_1}, F_1^{\min} + \left(j - \dfrac{1}{2}\right)\delta_{F_1} \end{bmatrix} \right. \\ \left. \cap \left[F_2^{\max} + \left(\dfrac{1}{2} - j\right)\delta_{F_2}, F_1^{\max} + \left(\dfrac{3}{2} - j\right)\delta_{F_2} \right] \right\}. \tag{10}$$

After sorting all optimal individuals by objective F_1 from the smallest to largest {1, 2, 3, ...}, the grid setting for the ith individual has to fulfill the following equation:

$$\left(F_1^{*(j)}, F_2^{*(j)}\right) \in \left[F_1^{\min} + \left(i_{k_j} - \frac{3}{2}\right)\delta_{F_1}, F_1^{\min} + \left(i_{k_j} - \frac{1}{2}\right)\delta_{F_1} \right] \\ \cap \left[F_2^{\max} + \left(\frac{1}{2} - i_{k_j}\right)\delta_{F_2}, F_1^{\max} + \left(\frac{3}{2} - i_{k_j}\right)\delta_{F_2} \right]. \tag{11}$$

2.2.2 Grid based diversity maintenance mechanism of Pareto front

In case that the number of Pareto optimal individuals exceeds the archive size, a mechanism is required to keep elite individuals considering the limited size of the archive set. The grid-based diversity mechanism determines whether a newly generated solution can be added to the full archive. The newly generated individuals (j_1 and j_2) cannot be added to the archive set if they are dominated by the Pareto optimal individuals (i_1 and i_2), as seen in Fig. 1 [20].

If both the newly generated individual (j_3) is in the same grid with Pareto optimal individual (i_3), the diversity distribution of the Pareto front determines which one is replaced [20]. The main procedures of Grid-based mechanism:

Step 1: Add the newly generated individual to the archive set, determine the number of individuals in each grid $\left\{k_1,\ k_2,\ \ldots, k_i,\ \ldots, k_{N_Q}\right\}$, which satisfies $\sum_{i=1}^{N_Q} k_i = N_Q + 1$ and $k_i \in N$.

Step 2: Find the $k_{\max} = \max_{i=1:N_Q}(k_i)$, and if there are N_{grid} grids that contain $k_{\max}$ individuals, and select all individuals $\left\{L_1,\ L_2,\ \ldots, L_{N_{\text{grid}}} \times k_{\max}\right\}$ in these grids.

Step 3: Calculate the density degree of these individuals with the following metrics:

$$\begin{cases} \text{Den}(L_i) = \displaystyle\sum_{j \neq L_i, j \in Q} \frac{1}{\text{dis}(L_i, j)} \\ \text{dis}(L_i, j) = \sqrt{\left(F_1^{L_i} + F_1^{j}\right)^2 + \left(F_2^{L_i} + F_2^{j}\right)^2} \end{cases}. \tag{12}$$

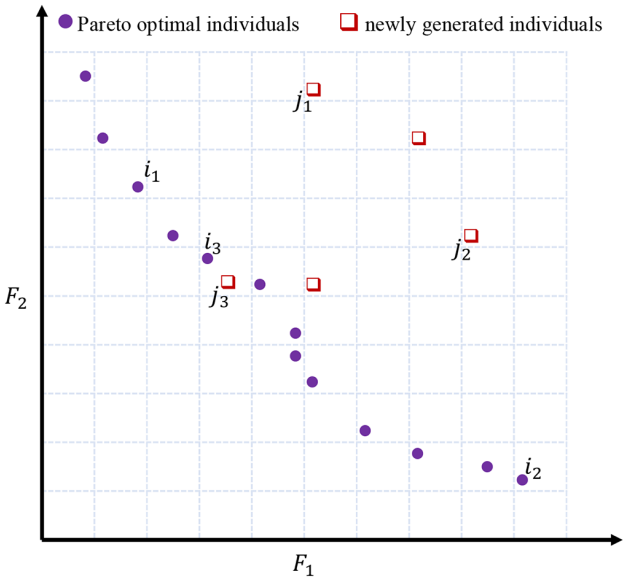


Fig. 1 Grid-based diversity maintenance mechanism of Pareto front

Step 4: Select the individual with the largest density, and remove this individual from the archive set; then, the archive set can be maintained appropriately to a particular extent. Especially when the newly generated individual has a better minimum value in F_1 or F_2, this new individual can be added directly to the archive, and the person with the highest density should be removed.

3 The proposed multi-objective algorithms

This section outlines the GMOWSA and NSWSA algorithms. We begin with a brief overview of the Water Strider Algorithm (WSA), followed by a detailed description of the proposed methods.

3.1 WSA

The overall process of WSA consists of five steps, i.e., birth, territory establishment, mating, feeding, death, and succession [12]. Each step is described below, along with the corresponding mathematical expressions.

3.1.1 Birth

Water striders emerge from eggs laid on the water's surface. To simulate this, initial positions for the water striders (*WS*s) are generated randomly using the Eq. (13):

$$WS_i^0 = LB + \text{rand} \times (Ub - Lb), i = 1, 2, \ \ldots, \ nws, \tag{13}$$

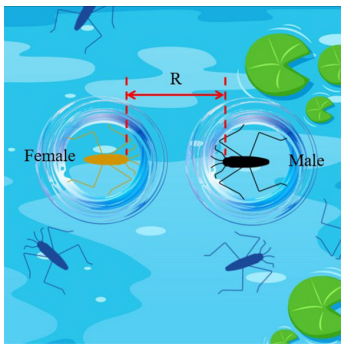
where WS_i^0 denotes the *i*th water strider's initial position. $Ub - Lb$ are the upper and lower boundaries of the bottom and top allowed values, respectively; rand is a uniformly distributed number in range [0, 1]; *nws* is the number of *WS*s. The fitness of the *WS*s' position on the lake is calculated using an objective function.

3.1.2 Territory establishment

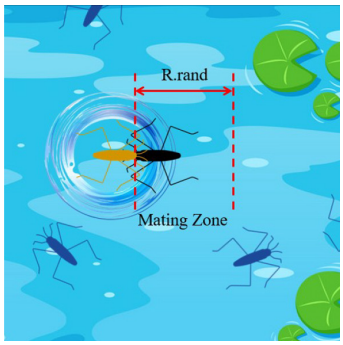
*WS*s maintain territories for living, mating, and feeding. The following procedure is used to establish the *nt* number of territories and assign *WS*s to the territories. Initially, *WS*s are sorted according to their fitness and are separated into $\frac{nws}{nt}$ bunches orderly. The *j*th member of each bunch is located in the *j*th territory, where *j* = 1, 2, ..., *nt*. Consequently $\frac{nws}{nt}$ number of *WS*s reside within each territory. The fittest water strider in each territory is designated as the female (optimal foraging habitat), while the least fit is the male (keystone).

3.1.3 Mating

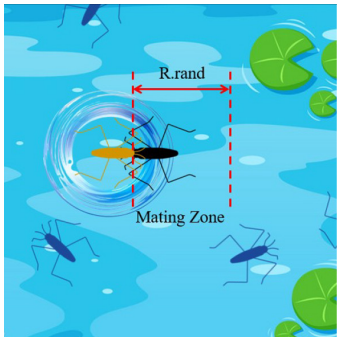
Mating is a crucial step in the water striders' life cycle. The keystone emits courtship signals, and the female responds with either attraction or repulsion signals. The chance of sending an attraction response is *p*, and consequently (1 – *p*) is the chance of sending a repulsion signal. To simplify the analysis, *p* is set to 50%, since the female's response is unpredictable. If the female sends an attraction ripple, the pair moves towards each other and mates. As shown in Fig. 2 (a), a circle wave is considered. The new position will be set to a location among them after mating, resulting in a new offspring position (Fig. 2 (b)). The male will be mounted by the female, then dismounted and pushed away, if the female declines the request, as depicted in Fig. 2 (c). The keystone's new position will be calculated by Eq. (14), regardless of whether it mates or is repelled.



(a)



(b)



(c)

Fig. 2 The position updating and mating behavior of the water striders (a) description, (b) description, (c) description

$$\begin{cases} WS_i^{t+1} = WS_i^t + R \times \text{rand} & \text{if mating happens with probability of } p, \\ WS_i^{t+1} = WS_i^t + R \times (1 + \text{rand}) & \text{otherwise} \end{cases} \tag{14}$$

where WS_i^t represents the position of ith WS in the tth cycle; *rand* is a random number in range [0, 1]; R is a vector pointing from the male's position $\left(WS_i^{t-1}\right)$ to the female's position $\left(WS_F^{t-1}\right)$ within the same territory. The female is selected using a mechanism like the roulette wheel. The length of R is equal to the Euclidean distance between the male $\left(WS_i^{t-1}\right)$ and female WSs $\left(WS_F^{t-1}\right)$ (the radius of ripple wave) as shown Eq. (15) and Fig. 2 (a).

$$R = WS_F^{t-1} - WS_i^{t-1}. \tag{15}$$

3.1.4 Feeding

After mating or repulsion, the WSs must feed to recover. If the objective function value at the new position is higher than the previous value, the water strider has successfully found food. However, if the value is lower, the WSs must move towards the optimal habitat with the highest fitness. This is achieved using Eq. (16), which updates the position based on the best WS of the lake $\left(WS_{BL}^t\right)$, as illustrated in Fig. 3.

$$WS_i^{t+1} = WS_i^t + 2\text{rand} \times \left(WS_{BL}^t - WS_i^t\right). \tag{16}$$

3.1.5 Death and succession

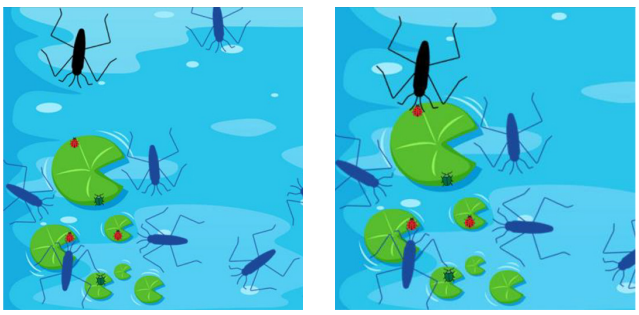
The outcome of the feeding process is determined by evaluating the objective function and comparing it with the previous position. If the new value is lower, indicating less food, the water strider perishes. A new larva replaces the deceased water strider, assuming the keystone role and being randomly placed in the territory, as defined by Eq. (17). However, if the new value is higher, the water strider survives.

$$WS_i^{t+1} = Lb_j^t + 2\text{rand} \times \left(Ub_j^t - Lb_j^t\right). \tag{17}$$

Ub_j^t and Lb_j^t are the maximum and minimum values of WS's position inside jth territory. They define the boundaries of the territory where the WS died.

3.1.6 Termination of WSA

The algorithm's final step is to verify the termination criterion. The algorithm terminates and returns the optimal position found if the criterion is satisfied. The algorithm repeats the mating step for a new cycle of life and region formation if the criterion is not satisfied. The termination criterion for all problems is the maximum number of function evaluations (*MaxNFE*) in this paper. Alternatively, other criteria such as the maximum number of cycles (*MaxCycle*) can be applied as a termination criterion for WSA. Algorithm 1 shows the WSA's pseudo-code.



(a) (b)

Fig. 3 Foraging behavior (a) Foraging: When a water strider doesn't find sufficient food at its new position, it seeks out food resources, (b) Movement towards the best strider: The water strider moves towards the best-performing water strider on the lake to find optimal feeding grounds

Algorithm 1 WSA

Inputs: *nws*, *nt*, *MaxCycle*
Outputs: The best position of WSs and their objective values
Randomly generate the initial population by Eq. (13)
Evaluate the fitness value of WSs
while (t < *MaxCycle*) **do**
Divide the WSs intot number of territories and assign them
for (each territory) do
The male keystone sends mating ripples, and the chosen female responds with an attraction or repulsion signal.
Update the position of keystone based on the female's response and Eq. (14)
Evaluate the new position to search for food to recover the energy spent during mating
if (keystone does not find food) **then**
Forage for food resource and move to the food-rich territory by Eq. (16).
if (keystone does not find food again) **then**
The starving keystone will die from hunger or be killed by the resident keystone of the new territory.
A matured larva will take the place of the dead keystone as the successor defined by Eq. (17).
end
end
end While
Return WS optimal

3.2 NSWSA

This section describes the Non-Dominated Sorting Water Strider Algorithm (NSWSA). The main steps of NSWSA are as follows:

Step 1: Similar to other meta-heuristics, the initial positions of the water striders are randomly generated.

Step 2: The fitness value of each agent is evaluated using the objective functions in each iteration.

Step 3: The external archive stores non-dominated solutions and guides agent updates. New, non-dominated solutions are added to the archive, while dominated members are removed. The archive size is limited. To maintain a limited archive size, a secondary mechanism is employed. If a new solution dominates an existing archive member, the dominated member is replaced. To assess solution diversity, the objective function space is discretized into grids. Solutions are sorted by objective function value, and distances between consecutive points are calculated.

Step 4: Choose the elite water striders using the Roulette wheel selection mechanism.

Step 5: Updating the positions of the agents using the Eqs (14–17).

Step 6: The optimization process terminates after a fixed number of iterations (*MaxCycle*). If this condition is not met, Steps 2 to 4 are repeated. The final Pareto front approximation is represented by the positions and objective values of the "bestsol" solutions.

3.3 GMOWSA

This section extends the basic WSA method to handle multi-objective problems by introducing a Grid-based multi-objective water strider algorithm (GMOWSA). The following steps outline the GMOWSA algorithm.

Step 1: GMOWSA initializes parameters and randomly generates initial agent positions (solutions) in the search space.

Step 2: According to each agent, the process starts calculating the objective function value $\left(f_1(x), f_2(x), \ldots, f_m(x)\right)$.

Step 3: Saving in the external archive and selection of leader.

The external archive stores non-dominated solutions to guide water strider updates. New, non-dominated solutions are added to the archive, while dominated solutions are removed. To maintain a limited archive size, a secondary mechanism is employed. The objective function space is divided into hypercubes. Non-dominated solutions are assigned to their corresponding hypercubes based on their objective function values. In a minimization problem, only the solution closest to the lowest left corner of each hypercube is retained. A leader selection identifies the best water strider.

Step 4: Eqs (14–17) evaluate the new position of each agent.

Step 5: The optimization process stops after a fixed number of iterations (MaxCycle). Otherwise, Steps 2–4 are repeated.

4 Validation of the proposed algorithms

The proposed algorithms was evaluated against well-known optimization algorithms on nine benchmark functions and two structural design problems. These problems are typically tackled as single-objective, but this research employs a multi-objective version with two conflicting objectives.

The algorithms were coded in MATLAB 9.7 (R2019b) [21]. Experiments ran on a Quad-core Intel CORE i7-1065G7 CPU (2 GHz) with 16 GB RAM on Windows 10.

4.1 The evaluation metrics

Multi-objective optimization algorithms generate an approximate Pareto front of non-dominated solutions. To assess the quality of these solutions, evaluation metrics are essential. However, due to the conflicting and incomparable nature of objectives, evaluating solution quality is challenging. While visual inspection is suitable for problems with 2 or 3 objectives [22], quantitative indexes are necessary for general assessment. This paper employs the following five indexes:

4.1.1 Inverted generational distance (IGD)

The IGD metric evaluates how well an approximate Pareto front approximates the true Pareto front. It calculates the average distance from each point on the true front to the nearest point on the approximate front [23]. Lower IGD values indicate better approximation quality.

$$\mathrm{IGD} = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n}. \tag{18}$$

Here, n represents the number of true Pareto optimal solutions, and d_i denotes the Euclidean distance between a true Pareto optimal solution and its nearest obtained Pareto optimal solution in the reference set. Unlike the standard

IGD, where distance is calculated for every true solution, here, the distance is calculated for every obtained solution relative to the reference set.

4.1.2 Spacing (S)

This metric indicates the evenness of solution distribution along the known Pareto front.

$$S = \sqrt{\frac{\sum_{i=1}^{n}\left(d_i - \bar{d}\right)^2}{n}},\ \bar{d} = \frac{\sum_{i=1}^{n} d_i}{n}, \tag{19}$$

where d_i represents the Euclidean distance from the ith solution to its nearest neighbor on the known Pareto front. A smaller S indicates a more uniform distribution of non-dominated solutions.

4.1.3 Maximum spread (MS)

The MS measures the distance between the boundary solutions of the known and true Pareto fronts:

$$\mathrm{MS} = \sqrt{\frac{\sum_{i=1}^{m}\left[\frac{\min\left(f_i^{\max} - F_i^{\max}\right) - \max\left(f_i^{\min} - F_i^{\min}\right)}{F_i^{\max} - F_i^{\min}}\right]^2}{m}}, \tag{20}$$

where m is the number of objectives, $f_i^{\max}$ and $f_i^{\min}$ are maximum and minimum values of the ith objective in known pareto front, respectively. Similarly, $F_i^{\max}$ and $F_i^{\min}$ are maximum and minimum values of the ith objective in true pareto front. A larger MS value indicates a better spread of solutions.

4.2 Mathematical test functions

The performance of GMOWSA and NSWSA was validated on nine benchmark functions with known Pareto fronts (Table 1). These functions include bi-objective and tri-objective problems with varying complexities: convex, disconnected convex, non-convex, and non-uniformly disconnected non-convex. Table 1 also specifies the number of variables (Dim) and their bounds for each function.

GMOWSA and NSWSA were compared to stat-of-the-art algorithms in (MOALO, NSGA-II, and MOPSO). The same parameter settings as in the original papers were used for all algorithms. Table 2 summarizes these parameter settings.

All algorithms used a population size of 50 and 200 iterations. All algorithms were limited to 1,000 fitness function evaluations. To ensure statistical significance, 30 independent runs were conducted for each problem. Statistical tests were employed to assess differences in algorithm performance. For each problem size and algorithm, data normality was evaluated using the Kolmogorov–Smirnov test. Because all p-values were below 0.05 – indicating that the datasets are not normally distributed – the Friedman test was used to compare the mean ranks of the five algorithms (with the null hypothesis stating that all algorithms perform equally and the alternative hypothesis stating that at least one algorithm performs differently). A significance level of 0.05 was applied, and the computed H-value was compared against the critical value from the chi-square (χ^2) distribution, based on the appropriate degrees of freedom. With five algorithms (k = 5), the critical value from the χ^2 table is 9.49.

Table 3 summarizes the statistical results of, S, and MS for all algorithms, including GMOWSA and NSWSA. The Friedman test confirms that the differences among the five algorithms are statistically significant across all nine problems.

Boldface indicates the best result for each metric and problem. Looking at ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 as well as the DTLZ problems, it will notice that no single algorithm dominates every metric. For example, while NSWSA is often among the best for IGD on several problems (e.g., ZDT3 and ZDT6), other metrics on different problems favor MOALO or GMOWSA. This variability indicates that an algorithm's strength can depend on the problem's characteristics and the aspect of performance being emphasized.

4.3 Truss structure instances

4.3.1 A 120-bar dome truss

To evaluate the performance of GMOWSA and NSWSA on structural optimization problems, two structural problems with two conflicting objectives were considered. All algorithms used a population size of 50 and 300 iterations. The primary objectives in structural multi-objective optimization are minimizing weight and displacement under design constraints [24]. Fig. 4 illustrates a 120-bar truss with element groups.

Material properties: elasticity = 30,450 ksi (210 GPa), density = 0.288 lb/in^3 (7971.81 kg/m^3), yield stress = 58.0 ksi (400 MPa). The dome is subjected

Table 1 Mathematical test functions

Function	Shift position	Dim	Objective functions
ZDT1	[0, 1]	30	$F_1 = x_1, F_2 = g\left(1-\sqrt{\frac{x_1}{g(x)}}\right),\ g = 1+\left(9\sum_{i=2}^{d}\frac{x_i}{(d-1)}\right)$
ZDT2	[0, 1]	30	$F_1 = x_1, F_2 = g\left(1-\left(\frac{x_1}{g(x)}\right)^2\right),\ g = 1+\left(9\sum_{i=2}^{d}\frac{x_i}{(d-1)}\right)$
ZDT3	[0, 1]	30	$F_1 = x_1, F_2 = g\left(1-\sqrt{\frac{F_1}{g}}\right),\ g = 1+10(d-1)+\sum_{i=2}^{d}\left(x_i^2 10\cos(4\pi x_1)\right)$
ZDT4		10	$F_1 = x_1, F_2 = g\left(1-\sqrt{\frac{F_1}{g}}\right),\ g = 1+\left(9\sum_{i=2}^{d}\frac{x_i}{(d-1)}\right)$
ZDT6	[0, 1]	10	$F_1 = 1-\exp(-4x_1)\sin^6(6\pi x_1), F_2 = g\left(1-\left(\frac{F_1}{g}\right)^2\right),\ g = 1+9\left(\sum_{i=2}^{d}\frac{x_i}{(d-1)}\right)^{0.25}$
DTLZ2	[0, 1]	12	$F_1 = (1+g)\cos\left(x_1\left(\frac{\pi}{2}\right)\right)\cos\left(x_2\left(\frac{\pi}{2}\right)\right), F_2 = (1+g)\cos\left(x_1\left(\frac{\pi}{2}\right)\right)\sin\left(x_2\left(\frac{\pi}{2}\right)\right),$ $F_3 = (1+g)\sin\left(x_1\left(\frac{\pi}{2}\right)\right),\ g = \sum_{i=3}^{d}(x_i-0.5)^2$
DTLZ4	[0, 1]	12	$F_1 = (1+g)\cos\left(x_1^{\pi}\left(\frac{\pi}{2}\right)\right)\sin\left(x_2^{\pi}\left(\frac{\pi}{2}\right)\right), F_2 = (1+g)\cos\left(x_1^{\pi}\left(\frac{\pi}{2}\right)\right)\cos\left(x_2^{\pi}\left(\frac{\pi}{2}\right)\right),$ $F_3 = (1+g)\sin\left(x_1^{\pi}\left(\frac{\pi}{2}\right)\right),\ g = \sum_{i=3}^{d}(x_i-0.5)^2$
DTLZ5	[0, 1]	12	$F_1 = (1+g)\cos\left(\frac{\pi}{2}\theta_1\right)\cos\left(\frac{\pi}{2}\theta_2\right), F_2 = (1+g)\cos\left(\frac{\pi}{2}\theta_1\right)\sin\left(\frac{\pi}{2}\theta_2\right),$ $F_3 = (1+g)\sin\left(\frac{\pi}{2}\theta_1\right),\ g = \sum_{i=3}^{d}(x_i-0.5)^2,\ \theta_1 = x_1\left(\frac{\pi}{2}\right)\ \theta_1 = \frac{\pi}{4(1+g)}(1+2x_2\times g)$
DTLZ6	[0, 1]	12	$F_1 = (1+g)\cos\left(\frac{\pi}{2}\theta_1\right)\cos\left(\frac{\pi}{2}\theta_2\right), F_2 = (1+g)\cos\left(\frac{\pi}{2}\theta_1\right)\sin\left(\frac{\pi}{2}\theta_2\right),$ $F_3 = (1+g)\sin\left(\frac{\pi}{2}\theta_1\right),\ g = \sum_{i=3}^{d}(x_i)^{0.1},\ \theta_1 = x_1\left(\frac{\pi}{2}\right)\ \theta_1 = \frac{\pi}{4(1+g)}(1+2x_2\times g)$

Table 2 Parameter settings of the algorithms

Algorithm	Parameter	value
NSGA-II	Mutation probability	0.1/dim
	Crossover probability	0.9
	External archive size	100
MOPSO	Inertia weight	0.4
	External archive size	100
	nGrids	30
	Leader selection pressure	4
GMOWSA	nGrid	12
	pt	3
	pro	0.5
NSWSA	pt	3
	pro	0.5

to vertical loads at: –13.49 kips (–60 kN) at node 1, –6.744 kips (–30 kN) at nodes 2–14, –2.248 kips (–10 kN) elsewhere. Element areas range from 0.775 in^2 (5 cm^2) to 20.0 in^2 (129 cm^2). Pipe sections (a = 0.4993, b = 0.6777) are used for radius of gyration (r_i) calculation $\left(r_i = ar_i^b\right)$ [2]. AISC provisions [25] are used to impose stress constraints on the members.

The allowable tensile stresses for tension members are calculated as:

$$\begin{cases}\sigma_i^+ = 0.6F_y & \text{for} \quad \sigma_i \geq 0\\ \sigma_i^- & \text{for} \quad \sigma_i < 0\end{cases}, \tag{21}$$

where F_y is the yield strength. Compression members can fail through elastic or inelastic buckling. The allowable stress limits for these failure modes are:

Table 3 Statistical results of the mathematical test functions

Function	Metric	Algorithm	Average	Srandard deviation	Rank	P-value	χ^2	Significant difference
		MOPSO	0.0397	0.0041	4.70			
		MOALO	**0.0110**	**0.0015**	1.00			
	IGD	NSGA-II	0.0343	0.0026	3.50	1E-4	104	Yes
		GMOWSA	0.0358	0.0066	3.77			
		NSWSA	0.0201	0.0044	2.03			
		MOPSO	0.0109	0.0004	3.43			
		MOALO	0.0115	0.0012	3.57			
ZDT1	Spacing	NSGA-II	0.0337	0.0027	5.00	1E-17	111	Yes
		GMOWSA	0.0071	0.0009	1.87			
		NSWSA	**0.0060**	**0.0003**	1.13			
		MOPSO	0.7789	0.1105	4.73			
		MOALO	0.8499	0.1077	2.97			
	MS	NSGA-II	0.7902	0.1460	4.20	6E-18	111	Yes
		GMOWSA	**0.9188**	**0.0702**	1.05			
		NSWSA	0.8901	0.0896	2.05			
		MOPSO	0.0235	0.0031	4.72			
		MOALO	0.0201	0.0049	4.05			
	IGD	NSGA-II	0.0123	0.0020	2.27	1E-5	82	Yes
		GMOWSA	0.0125	0.0015	2.30			
		NSWSA	**0.0113**	**0.0014**	1.67			
		MOPSO	**0.0074**	**0.0006**	1.18			
		MOALO	0.0100	0.0005	4.00			
ZDT2	Spacing	NSGA-II	0.0101	0.0005	5.00	2E-18	113	Yes
		GMOWSA	0.0081	0.0004	1.95			
		NSWSA	0.0087	0.0002	2.87			
		MOPSO	0.9121	0.0987	3.30			
		MOALO	**0.9329**	**0.0898**	1.63			
	MS	NSGA-II	0.9420	0.0898	3.30	2E-19	120	Yes
		GMOWSA	0.9720	0.0695	2.13			
		NSWSA	0.9320	0.0898	4.63			
		MOPSO	0.0622	0.0052	4.87			
		MOALO	0.0427	0.0074	2.43			
	IGD	NSGA-II	0.0500	0.0074	3.38	2E-16	80	Yes
		GMOWSA	0.0481	0.0080	2.95			
		NSWSA	**0.0357**	**0.0026**	1.37			
		MOPSO	0.0110	0.0012	4.85			
		MOALO	0.0086	0.0008	2.47			
ZDT3	Spacing	NSGA-II	0.0106	0.0008	4.15	2E-16	113	Yes
		GMOWSA	0.0087	0.0006	2.53			
		NSWSA	**0.0078**	**0.0005**	1.00			
		MOPSO	0.9320	0.0898	2.17			
		MOALO	0.9020	0.0908	4.83			
	MS	NSGA-II	0.9120	0.1010	4.17	0	113	Yes
		GMOWSA	**0.9420**	**0.0807**	1.00			
		NSWSA	0.9320	0.0813	2.83			

Headline	Metric	Algorithm	Average	Srandard deviation	Rank	P-value	χ^2	Significant difference
		MOPSO	0.0679	0.0086	3.85			
		MOALO	0.0609	0.0058	2.83			
	IGD	NSGA-II	0.0639	0.0092	3.33	3E-6	31	Yes
		GMOWSA	0.0638	0.0055	3.27			
		NSWSA	**0.0529**	**0.0087**	1.72			
		MOPSO	0.0168	0.0011	5.00			
		MOALO	0.0107	0.0014	2.97			
ZDT4	Spacing	NSGA-II	0.0137	0.0010	4.00	2E-19	114	Yes
		GMOWSA	0.0096	0.0013	1.55			
		NSWSA	**0.0095**	**0.0007**	1.48			
		MOPSO	0.9820	0.0711	3.00			
		MOALO	0.9720	0.0698	4.00			
	MS	NSGA-II	0.9639	0.0693	5.00	0	120	Yes
		GMOWSA	**0.9939**	**0.0324**	1.00			
		NSWSA	0.9901	0.0396	2.00			
		MOPSO	0.0699	0.0061	3.85			
		MOALO	0.0671	0.0067	2.83			
	IGD	NSGA-II	0.0689	0.0061	3.33	3E-6	31	Yes
		GMOWSA	0.0620	0.0080	2.27			
		NSWSA	**0.0596**	**0.0060**	1.72			
		MOPSO	0.0198	0.0022	4.67			
		MOALO	0.0191	0.0019	3.43			
ZDT6	Spacing	NSGA-II	0.0193	0.0017	3.90	3E-10	108	Yes
		GMOWSA	0.0170	0.0008	2.00			
		NSWSA	**0.0150**	**0.0007**	1.00			
		MOPSO	0.9801	0.0294	3.97			
		MOALO	0.9701	0.0281	4.97			
	MS	NSGA-II	0.9801	0.0362	3.07	0	118	Yes
		GMOWSA	0.9901	0.0199	2.00			
		NSWSA	**0.9984**	**0.0082**	1.00			
		MOPSO	0.0423	0.0044	4.97			
		MOALO	**0.0183**	**0.0021**	1.43			
	IGD	NSGA-II	0.0295	0.0052	3.80	1E-9	95	Yes
		GMOWSA	0.0226	0.0028	2.73			
		NSWSA	0.0207	0.0029	2.07			
		MOPSO	0.0071	0.0004	1.32			
		MOALO	**0.0072**	**0.0006**	1.68			
DTLZ2	Spacing	NSGA-II	0.0111	0.0006	5.00	1E-5	43	Yes
		GMOWSA	0.0082	0.0006	4.00			
		NSWSA	0.0078	0.0006	3.00			
		MOPSO	0.8581	0.0591	5.00			
		MOALO	0.8981	0.0489	4.00			
	MS	NSGA-II	0.9101	0.0408	2.03	0	43	Yes
		GMOWSA	0.9100	0.0306	2.97			
		NSWSA	**0.9300**	**0.0305**	1.00			

Headline	Metric	Algorithm	Average	Srandard deviation	Rank	P-value	χ^2	Significant difference
		MOPSO	0.0431	0.0082	4.67			
		MOALO	0.0245	0.0045	1.67			
	IGD	NSGA-II	0.0395	0.0097	4.40	1E-9	103	Yes
		GMOWSA	0.0308	0.0064	3.10			
		NSWSA	**0.0225**	**0.0018**	1.37			
		MOPSO	0.0117	0.0016	4.10			
		MOALO	0.0104	0.0012	2.75			
DTLZ4	Spacing	NSGA-II	0.0125	0.0023	4.90	0	116	Yes
		GMOWSA	0.0103	0.0011	2.25			
		NSWSA	**0.0092**	**0.0009**	1.00			
		MOPSO	0.9305	0.0314	4.97			
		MOALO	0.9315	0.0327	4.03			
	MS	NSGA-II	0.9365	0.0225	3.00	0	119	Yes
		GMOWSA	0.9666	0.0124	2.00			
		NSWSA	**0.9965**	**0.0111**	1.00			
		MOPSO	0.0564	0.0144	4.63			
		MOALO	0.0389	0.0083	2.85			
	IGD	NSGA-II	0.0534	0.0089	4.33	0	109	Yes
		GMOWSA	0.0341	0.0038	2.18			
		NSWSA	**0.0273**	**0.0028**	1.00			
		MOPSO	0.0294	0.0041	4.92			
		MOALO	0.0278	0.0030	4.08			
DTLZ5	Spacing	NSGA-II	0.0268	0.0029	2.85	0	115	Yes
		GMOWSA	0.0267	0.0028	2.15			
		NSWSA	**0.0243**	**0.0028**	1.00			
		MOPSO	0.8965	0.0314	4.90			
		MOALO	0.9112	0.0365	2.10			
	MS	NSGA-II	**0.9152**	**0.0362**	1.10	2E-16	110	Yes
		GMOWSA	0.9112	0.0353	3.00			
		NSWSA	0.9082	0.0333	3.90			
		MOPSO	0.1227	0.0139	5.00			
		MOALO	0.0941	0.0125	3.93			
	IGD	NSGA-II	0.0840	0.0060	3.07	1E11	119	Yes
		GMOWSA	0.0583	0.0027	2.00			
		NSWSA	**0.0535**	**0.0039**	1.00			
		MOPSO	0.0352	0.0053	3.85			
		MOALO	0.0369	0.0063	4.87			
DTLZ6	Spacing	NSGA-II	0.0339	0.0050	2.72	1E10	88	Yes
		GMOWSA	**0.0330**	**0.0046**	1.58			
		NSWSA	0.0324	0.0030	1.98			
		MOPSO	**0.9882**	**0.0121**	1.00			
		MOALO	0.9893	0.0110	2.07			
	MS	NSGA-II	0.9893	0.0103	2.93	0	119	Yes
		GMOWSA	0.9999	0.0001	4.00			
		NSWSA	0.9999	0.0001	5.00			

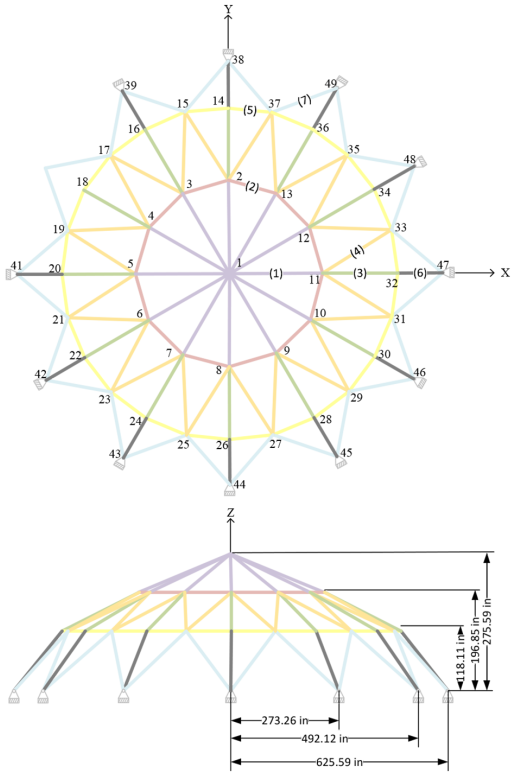


Fig. 4 120-bar dome truss schematic

$$\sigma_i^- = \begin{cases} \dfrac{\left(1-\dfrac{\lambda_i^2}{2C_c^2}\right)\times F_y}{\dfrac{5}{3}+\dfrac{3\lambda_i}{8C_c}-\dfrac{\lambda_i^3}{8C_c^3}} & \text{for} \quad \lambda_i < C_c \\ \dfrac{12\pi^2 E}{12\lambda_i^2} & \text{for} \quad \lambda_i \geq C_c \end{cases}, \tag{22}$$

where E is the modulus of elasticity, λ_i is the slenderness ratio $\lambda_i = \dfrac{kL_i}{r_i}$, C_c is the critical slenderness ratio $\left(C_c = \sqrt{\dfrac{2\pi^2 E}{F_y}}\right)$, k is the effective length factor (here is 1 for all members), L_i is the member's length; and r_i is the minimum radius of gyration.

Two primary objectives was minimized: structural weight and maximum nodal displacements. Since the true Pareto front is unknown, the performance metrics defined previously are inapplicable. Each algorithm was run 20 times, and the best result is presented graphically. Fig. 5 shows that the non-dominated solutions obtained by GMOWSA and NSWSA dominate other metaheuristic algorithms. The average computational time per run was 171 s for GMOWSA, 173 s for NSWSA, 185 s for MOALO, 186 s for MOPSO, and 198 s for NSGA-II.

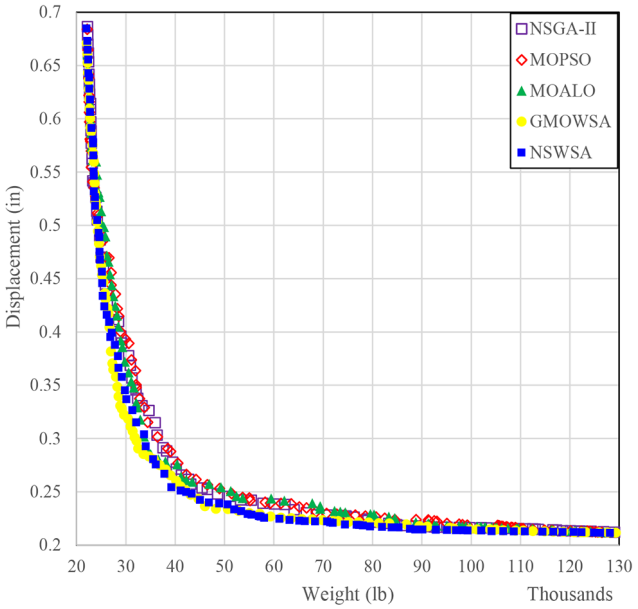


Fig. 5 Pareto front for 120-bar dome truss

4.3.2 A 582-bar truss

A 582-bar tower truss (Fig. 6), symmetrically divided into 32 groups is considered [9, 26]. Cross-sectional areas of elements (variables) selected from a discrete set of *W*-shaped standard steel sections based on the area and

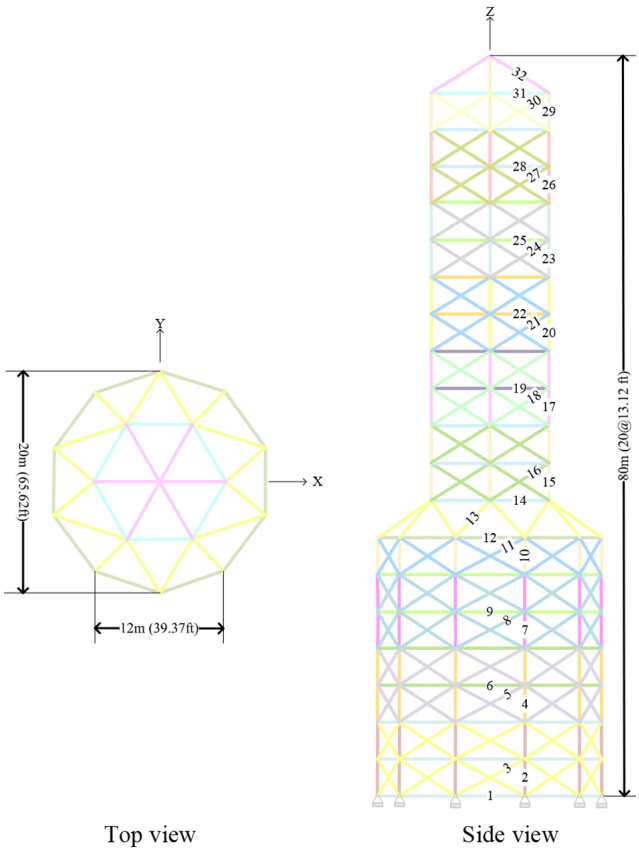


Fig. 6 Spatial 582-bar tower truss schematic

radii of gyration properties, range from 6.16 to 215 in^2. (i.e., between 39.74 and 1387.09 cm^2). The structure is subjected to lateral loads of 1.12 kips (5.0 kN) in both x and y directions, and a vertical load of –6.74 kips (–30 kN) at all nodes. Design adheres to AISC specifications [25], with maximum slenderness ratios of 300 for tension members and 200 for compression members. The modulus of elasticity is E = 29,000 ksi (203893.6 MPa) and the steel's yield stress of F_y = 36 ksi (253.1 MPa).

The two primary objectives are to minimize the total structural volume and maximum nodal displacements. Fig. 7 compares the Pareto fronts obtained by MOALO, NSGA-II, MOPSO, GMOWSA, and NSWSA. The Pareto fronts obtained by GMOWSA and NSWSA are competitive with, and even GMOWSA dominate, those of MOALO, MOPSO, and NSGA-II in many regions. The average computational time per run was 3619 s for MOALO, 3757 s for MOPSO, 3914 s for NSGA-II, 3427 s for NSWSA, and 3397 s for GMOWSA.

4.4 Project Scheduling Problem

4.4.1 Time, Cost And Risk trade-off (TCRT) project scheduling

In this problem, a project is represented as an activity-on-node network $G = (A)$, where A is the set of N activities. Activities 0 and $N + 1$ are dummy activities representing the project start and finish, respectively. P is the set of all paths in the network, and P_l is the set of activities in path $l \in P$. Each activity $i \in A$ has multiple execution methods, each with its own duration D_i, cost C_i, and resource requirement R_i. The TCRT problem aims to optimize the combination of execution methods and the slack time for non-critical activities to balance time, cost, and risk [27].

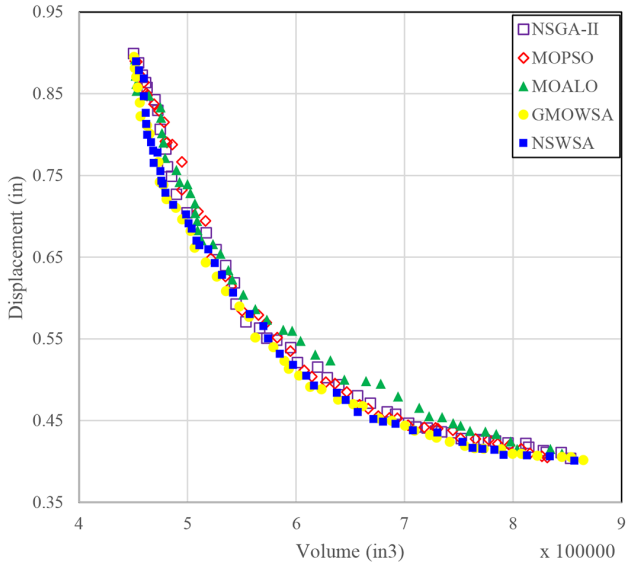


Fig. 7 Pareto front of algorithms on 582-bar tower truss problem

The first objective is to minimize the total project duration, Z_T:

$$\min Z_T = \max_{i=1,\ldots,N} \left(FT_i\right) = \min_{i=1,\ldots,N} \left(ST_i - D_i\right), \tag{23}$$

where ST_i and FT_i are the start and finish times of activity i, respectively. The ST_i values are decision variables. Project duration is determined by precedence constraints and activity durations. Precedence constraints are defined by the project information, while activity durations are determined by the selected execution methods.

The total project cost comprises direct costs (DCs), indirect costs (ICs), and tardiness costs (TCs). The second objective is to minimize the total project cost:

$$\min Z_C = \sum_{i=1}^{N} \left(\mathrm{DC}_i^{S_i} + \mathrm{IC}_i^{S_i} + \mathrm{TC}_i^{S_i}\right), \tag{24}$$

where $\mathrm{DC}_i^{S_i}$, $\mathrm{IC}_i^{S_i}$, and $\mathrm{TC}_i^{S_i}$ are the direct, indirect, and tardiness costs, respectively, of activity i for a chosen execution method (S_i) and N is the number of activities. IC is proportional to the project duration (Z_T), $IC = C_0 + b \times Z_T$, where C_0 is the initial cost and b is the daily indirect cost. TC is a penalty for project delays.

A "risk value" integrates total float and resource fluctuation. Total float, the maximum delay allowable without project delay, can mitigate risks associated with non-critical activity delays. A safe float range can minimize risks associated with non-critical activity delays [28–30]. However, construction schedules often suffer from resource fluctuations, hindering practical, efficient, and risk-free implementation. To mitigate these fluctuations, construction managers adjust schedules to optimize resource utilization. Non-critical activity float can be used to achieve this. The third objective is to minimize total project risk, calculated as:

$$\min Z_{\mathrm{Risk}} = \omega_1 \times \left(1 - \frac{TF_{\mathrm{current}} + 1}{TF_{\max} + 1}\right) + \omega_2 \times \frac{\sum_{t=1}^{P_d} \left(R_t - \overline{R}_t\right)^2}{P_d \times \left(R_t\right)^2}, \tag{25}$$

where TF_{current} is current total float; $TF_{\max}$ is maximum total float (which all activities' start at earliest STs); R_t is

required resource amount on day t of the project, $\bar{R}_t$ is uniform resource level, and w_1 and w_2 weights for the importance of total float and resource fluctuation, respectively.

A larger total float reduces the risk of schedule delays. A smoother resource profile, closer to the uniform resource level R, also reduces risk. This study focuses on a deterministic environment with known input data. It assumes all objective functions are quantified and uses average resource consumption.

This research studied a numerical construction project to analyze the efficacy of the proposed GMOWSA and NSWSA in addressing the TCRT problem. The outcomes were compared to MOALO, NSGA-II, and MOPSO. The project involved ten construction activities and two dummy activities, each with multiple execution methods. The project involved ten construction activities and two dummy activities, each with multiple execution methods. This case was presented by [27] and includes 10 activities with different execution modes. Table 4 outlines the precedence relationships and the resource usage, durations, and costs of each option (mode). Each combination of methods influences project performance, requiring a comprehensive search for optimal trade-offs among duration, cost, and risk.

To ensure statistical statistical significance, 20 independent optimization runs were conducted. The weights for the risk factors were set as $w_1 = 0.6$ and $w_2 = 0.4$, reflecting their relative importance.

This case study was conducted using NSWSA, GMOWSA, MOALO, MOPSO, and NSGA-II. Fig. 8 illustrates the Pareto fronts achieved by NSWSA, GMOWSA, MOALO, MOPSO, and NSGA-II. The time-cost Pareto

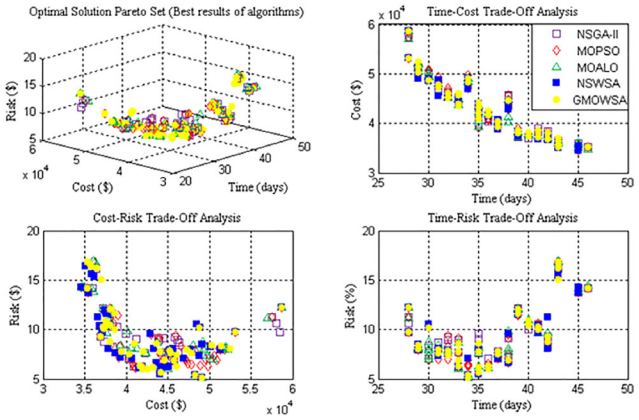


Fig. 8 Pareto fronts for the project time-cost-risk trade-off problem

highlights the trade-off between project funding and duration. However, projecting the three-dimensional Pareto onto two dimensions can obscure some non-dominated solutions. To ensure a fair comparison, all algorithms used a population size of 50, an iterations number of 300, and the same number of function evaluations. Both GMOWSA and NSWSA demonstrate competitive performance compared to the other algorithms. The range of time, cost, and risk deviations for GMOWSA (28–45 days, 34,600$–58,700$, and 5.17%–16.41%) and NSWSA (28–48 days, 35,300$–58,700$, and 5.17%–16.86%) is comparable to that of MOPSO (28–46 days, 34,600$–58,700$, and 6.11%–16.94%), MOALO (28–46 days, 34,700$–58,700$, and 5.17%–16.94%), and NSGA-II (28–46 days, 35,300$–58,500$, and 6.26%–16.16%). These results highlight the effectiveness of GMOWSA and NSWSA in optimizing the three conflicting objectives of project time, cost, and risk.

Table 4 Activities information for the scheduling problem [27]

activity	predecessor	Option I			Option II			Option II		
		duration (days)	cost ($)	resourse (worker)	duration (days)	cost ($)	resourse (worker)	duration (days)	cost ($)	resourse (worker)
1	-	7	1,900	6	3	5,900	8	-	-	-
2	-	9	5,000	5	7	6,000	7	5	10,000	8
3	1	8	2,900	7	7	3,500	9	5	4,400	12
4	2	12	1,700	2	10	3,500	4	9	4,700	7
5	3, 4	4	1,300	3	3	2,000	4	2	2,800	6
6	2	9	3,200	4	7	5,800	6	5	6,200	7
7	6	7	2,600	2	6	4,000	3	4	5,800	5
8	5, 7	10	5,400	2	6	7,400	4	-	-	-
9	6, 7	6	4,200	3	5	5,000	5	4	6,200	6
10	8, 9	10	6,400	3	6	8,400	4	-	-	-

4.5 Construction site layout problem

4.5.1 Cost and safety factor trade-off construction site layout problem

Site layout planning is a critical managerial aspect in the construction industry, significantly impacting productivity, safety, and health conditions [31, 32]. This case study, adapted from [33], focuses on optimizing the placement of temporary facilities for a multistory garage building site, considering sustainability. Tables 5 and 6 provide details on temporary and fixed facilities, while Fig. 9 shows the site plan.

Table 5 Temporary facilities information [33]

Index	Temporary facilities	Sensitivity	Length (m)	Width (m)
F1	Parking lot	Medium	20	20
F2	Office 1	High	20	5
F3	Office 2	High	20	5
F4	Office 3	High	20	5
F5	Office 4	High	20	5
F6	Workshop	High	5	4
F7	Storage 1	Medium	6	5
F8	Storage 2	Medium	4	5
F9	Electric generator	Medium	2	2
F10	Toilets	Low	5	6
F11	Fire station	Medium	3	3
F12	Flammable materials storage	Medium	3	3

Table 6 Fixed facilities information [33]

Index	Fixed facilities	Length (m)	Width (m)	X (m)	Y (m)
CI	Multistory garage building	120	95	75	67.5
K2	Tower crane	15	15	75	10
G3	Entrance gate	-	-	155	10

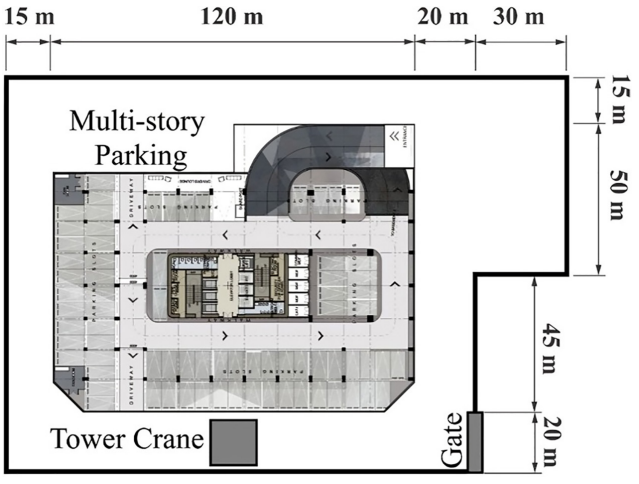


Fig. 9 Multistory garage building site plan [33]

The first objective is minimizing transportation costs between facilities, calculated as:

$$\min \mathrm{TC} = \sum_{i=1}^{I-1} \sum_{j=i+1}^{I} C_{ij} + D_{ij}, \tag{26}$$

where I is the total number of facilities, D_{ij} is the Euclidean distance between facilities i and j, and C_{ij} represents the corresponding transportation cost (Table 7).

The second objective is maximizing safety, quantified by the safety index (SI):

$$\max \mathrm{SI} = 0.6\ \mathrm{CSC} + 0.2\ \mathrm{NHCC} + 0.2\ \mathrm{IPC}\ . \tag{27}$$

Crane Safety Criterion (CSC) evaluates facility layout based on proximity to cranes and sensitivity to falling objects using Eq. (28) and Table 8:

$$\mathrm{CSC} = \left(\frac{1}{I}\right) \sum_{i=1}^{I} CS_i, \tag{28}$$

where CS_i depends on the distance between facility i and the crane, crane dimensions, and facility sensitivity (Table 5).

Normalized Hazard Control Criterion (NHCC) assesses hazard control:

$$\mathrm{NHCC} = \frac{\left(\mathrm{HCC} - \mathrm{HCC}_{\min}\right)}{\left(\mathrm{HCC}_{\max} - \mathrm{HCC}_{\min}\right)}, \tag{29}$$

where $\mathrm{HCC} = \sum_{i=1}^{I-1} \sum_{j=i+1}^{I} \mathrm{HCW}_{ij} \times D_{ij}$ and HCW_{ij} is the hazard control weight between facilities (Table 9).

Intersection Point Criterion (IPC) reduces accident risks by minimizing crowded route intersections:

$$\mathrm{IPC} = \left(\frac{1 - \mathrm{IP}}{\mathrm{IP}_{\max}}\right) \times 100\%, \tag{30}$$

$$\mathrm{IP}_{\max} = \frac{\mathrm{NR} \times \left(\mathrm{NR} - 1\right)}{2}, \tag{31}$$

where IP is the number of intersection points of crowded routes, NR is the number of the crowded routes marked in Table 7 by "c", and $\mathrm{IP}_{\max}$ is the maximum possible intersections.

The problem involves 24 continuous variables (facility coordinates) constrained within site boundaries without overlaps. Minimizing transportation costs and maximizing safety are conflicting objectives, as closer facilities reduce costs but compromise safety.

Table 7 Transportation cost between facilities [33]

Facility *i*	Facility *j*														
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	C1	K1	G1
F1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-
F2	4	0	-	-	-	-	-	-	-	-	-	-	-	-	-
F3	4	7.5[c]	0	-	-	-	-	-	-	-	-	-	-	-	-
F4	4	7.5[c]	7.5[c]	0	-	-	-	-	-	-	-	-	-	-	-
F5	4	5.5	5.5	2.5	0	-	-	-	-	-	-	-	-	-	-
F6	1.5	1	1	1	1	0	-	-	-	-	-	-	-	-	-
F7	1.5	1	1	1	1	9.5[c]	0	-	-	-	-	-	-	-	-
F8	1.5	1	1	1	1	9.5[c]	6.5	0	-	-	-	-	-	-	-
F9	1.5	2	1	3	3	3	3	3	0	-	-	-	-	-	-
F10	1.5	7.5[c]	7.5[c]	7.5[c]	7.5[c]	6.5	6.5	6.5	1	0	-	-	-	-	-
F11	1.5	1	1	1	1	1	1	1	1	I	0	-	-	-	-
F12	1.5	1	1	1	1	3.5	1	1	3.5	1	1	0	-	-	-
C1	1.5	3.5	3.5	3.5	3.5	6.5	4.5	4.5	5.5	3	1	4.5	0	-	-
K1	0	7.5[c]	5.5	7.5[c]	7.5[c]	9.5[c]	9.5[c]	9.5[c]	0	0	1	4.5	5	0	-
G1	1.5	0	0	0	0	3	7[c]	7[c]	0	0	0	1	0	0	0

[c] Crowded routes

Table 8 Crane safety indicator for each zone [33]

Zone	Range	Crane safety indicator
Zone 1	$D_{i,\text{crane}} < J + \frac{M}{2}, \theta_i \int \theta_f$	$CS_i^1 = \begin{cases} 0\%\left(V_i = \text{high}\right) \\ 25\%\left(V_i = \text{medium}\right) \\ 50\%\left(V_i = \text{low}\right) \end{cases}$
Zone 2*	$J + \frac{M}{2} \le D_{i,\text{crane}} < H + J + \frac{M}{2}$	$CS_i^{2^*} = 100 - \frac{100 - CS_i^1}{m}$
Zone 2	$D_{i,\text{crane}} < J + \frac{M}{2}, \theta_i \int \theta_m$	$CS_i^2 = \left(100 - CS_i^{2^*}\right) \times \left(\frac{D_{i,\text{crane}} - J - \frac{M}{2}}{H} \right) + CS_i^{2^*}$
Zone 3	$D_{i,\text{crane}} \ge H + J + \frac{M}{2}$	$CS_i^3 = 100\%\left(V_i = \text{high, medium, low}\right)$

Using a population size of 50, an iterations number of 300, and the same number of function evaluations, NSWSA, GMOWSA, MOALO, MOPSO, and NSGA-II were applied. Fig. 10 illustrates the Pareto fronts achieved by NSWSA, GMOWSA, MOALO, MOPSO, and NSGA-II. The cost-safety Pareto highlights the trade-off between safety index and transportation costs between facilities. Both GMOWSA and NSWSA demonstrate competitive performance and convergence compared to the other algorithms.

The range of safety index and transportation costs for GMOWSA (33–81 and 8,980–29,470) and NSWSA (34–83 and 8,980–29,530) is competitive to that of MOPSO (32–85 and 8,340–28,720), MOALO (33–85 and 8,570–29,680), and NSGA-II (34–82 and 8,430–28,657). These results highlight the effectiveness of GMOWSA and NSWSA in optimizing the three conflicting objectives of construction site safety index and transportation costs. Also, GMOWSA dominates other algorithms, as shown in Fig. 10.

5 Concluding remarks

This research compares two new multi-objective algorithms, multi-objective water strider algorithms, GMOWSA and NSWSA. These algorithms were tested on nine mathematical benchmark functions and three real-world engineering problems (120-bar and 582-bar dome-shaped trusses, a time-cost-risk trade-off project scheduling problem, and a construction site layout

Table 9 Hazard control weight between the facilities [33]

Facility *i*	Facility *j*												
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	C1
F1	0	-	-	-	-	-	-	-	-	-	-	-	-
F2	0.25	0	-	-	-	-	-	-	-	-	-	-	-
F3	0.25	0	0	-	-	-	-	-	-	-	-	-	-
F4	0.25	0	0	0	-	-	-	-	-	-	-	-	-
F5	0.25	0	0	0	0	-	-	-	-	-	-	-	-
F6	0.50	0.50	0.50	0.50	0.50	0	-	-	-	-	-	-	-
FT	0.50	0.50	0.50	0.50	0.50	0.25	0	-	-	-	-	-	-
F8	0.50	0.50	0.50	0.50	0.50	0.25	0	0	-	-	-	-	-
F9	0.50	0.75	0.75	0.75	0.75	0.75	0.25	0.25	0	-	-	-	-
F10	0	0	0	0	0	0	0.25	0.25	0.25	0	-	-	-
F11	0	0	0	0	0	0	0	0	0	0	0	-	-
F12	0.75	0.50	0.50	0.50	0.50	0.25	0	0	1	0	0	0	-
C1	0.75	1	1	1	1	0.50	0.50	0.50	0.75	0.50	0	0.50	0

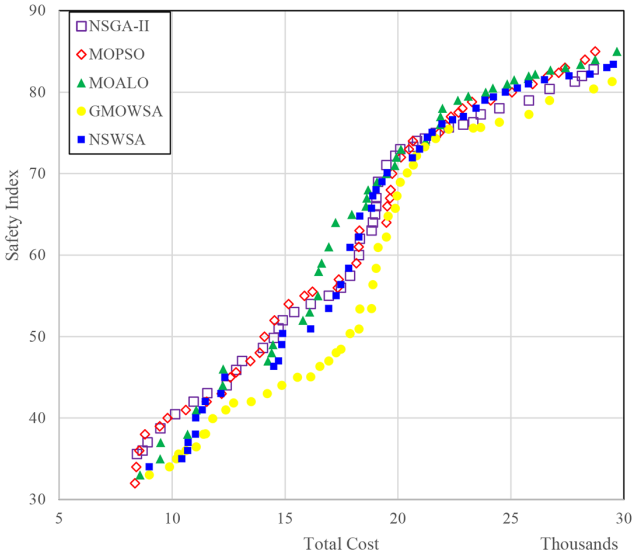


Fig. 10 Pareto fronts for the construction site layout problem

planning problem). The performance of GMOWSA and NSWSA was compared to three state-of-the-art algorithms: MOALO, MOPSO, and NSGA-II. The comparison was based on three standard performance indicators: IGD, *S*, and MS. The results demonstrate that NSWSA and especially GMOWSA exhibit competitive performance and uniform diversity compared to the other algorithms, even for complex real-world problems. It is concluded that GMOWSA and NSWSA are recommendable for solving various multi-objective optimization problems across different engineering fields.

Compliance with ethical standards

Conflict of interest: No potential conflict of interest was reported by the authors.

References

[1] Sharifi, A. "Trade-offs and conflicts between urban climate change mitigation and adaptation measures: A literature review", Journal of Cleaner Production, 276, 122813, 2020. https://doi.org/10.1016/j.jclepro.2020.122813

[2] Kaveh, A., Ilchi Ghazaan, M. "A new VPS-based algorithm for multi-objective optimization problems", Engineering with Computers, 36, pp. 1029–1040, 2020. https://doi.org/10.1007/s00366-019-00747-8

[3] Mirjalili, S., Saremi, S., Mirjalili, S. M., Coelho, L. D. S. "Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization", Expert Systems with Applications, 47, pp. 106–119, 2016. https://doi.org/10.1016/j.eswa.2015.10.039

[4] Habashneh, M., Rad, M. M. "An investigation of the recent developments in reliability-based structural topology optimization", Periodica Polytechnica Civil Engineering, 67(3), pp. 765–774, 2023. https://doi.org/10.3311/PPci.22107

[5] Kaveh, A., Javid, A. A. S., Vazirinia, Y. "Physics-inspired metaheuristics for construction site layout planning problem", Periodica Polytechnica Civil Engineering, 68(1), pp. 68–87, 2024. https://doi.org/10.3311/PPci.22902

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

[7] Coello, C. C. A., Pulido, G. T., Lechuga, M. S. "Handling multiple objectives with particle swarm optimization", IEEE Transactions on Evolutionary Computation, 8(3), pp. 256–279, 2004. https://doi.org/10.1109/TEVC.2004.826067

[8] Mirjalili, S., Jangir, P., Saremi, S. "Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems", Applied Intelligence, 46(1), pp. 79–95, 2017. https://doi.org/10.1007/s10489-016-0825-8

[9] Kaveh, A., Mahdavi, V. R. "Multi-objective colliding bodies optimization algorithm for design of trusses", Journal of Computational Design and Engineering, 6(1), pp. 49–59, 2019. https://doi.org/10.1016/j.jcde.2018.04.001

[10] Kaveh, A., Vazirinia, Y. "Smart-home electrical energy scheduling system using multi-objective antlion optimizer and evidential reasoning", Scientia Iranica, 27(1), pp. 177–201, 2020. https://doi.org/10.24200/sci.2019.53783.3412

[11] Wang, M., Wang, J.-S., Song, H.-M., Zhang, M., Zhang, X. Y., Zheng, Y., Zhu, J.-H. "Hybrid multi-objective Harris Hawk optimization algorithm based on elite non-dominated sorting and grid index mechanism", Advances in Engineering Software, 172, 103218, 2022. https://doi.org/10.1016/j.advengsoft.2022.103218

[12] Kaveh, A., Dadras Eslamlou, A. "Water strider algorithm: A new metaheuristic and applications", Structures, 25, pp. 520–541, 2020. https://doi.org/10.1016/j.istruc.2020.03.033

[13] Kaveh, A., Ghazaan, M. I., Asadi, A. "An improved water strider algorithm for optimal design of skeletal structures", Periodica Polytechnica Civil Engineering, 64(4), pp. 1284–1305, 2020. https://doi.org/10.3311/PPci.16872

[14] Kaveh, A., Dadras Eslamlou, A., Rahmani, P., Amirsoleimani, P. "Optimal sensor placement in large-scale dome trusses via Q-learning-based water strider algorithm", Structural Control and Health Monitoring, 29(7), e2949, 2022. https://doi.org/10.1002/stc.2949

[15] Ren, W., Hasanzade Bashkandi, A., Afshar Jahanshahi, J., Qasim Mohammad AlHamad, A., Javaheri, D., Mohammadi, M. "Brain tumor diagnosis using a step-by-step methodology based on courtship learning-based water strider algorithm", Biomedical Signal Processing and Control, 83, 104614, 2023. https://doi.org/10.1016/j.bspc.2023.104614

[16] Kalita, K., Ganesh, N., Narayanan, R. C., Jangir, P. Oliva, D. "Multi-objective water strider algorithm for complex structural optimization: A comprehensive performance analysis", IEEE Access, 12, pp. 55157–55183, 2024. https://doi.org/10.1109/ACCESS.2024.3386560

[17] Shafaie, V., Movahedi Rad, M. "Multi-objective genetic algorithm calibration of colored self-compacting concrete using DEM: an integrated parallel approach", Scientific Reports, 14(1), 4126, 2024. https://doi.org/10.1038/s41598-024-54715-4

[18] Ngatchou, P., Zarei, A., El-Sharkawi, A., "Pareto multi objective optimization", In: Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems, Arlington, VA, USA, 2005, pp. 84–91. ISBN 1-59975-174-7 https://doi.org/10.1109/ISAP.2005.1599245

[19] Pareto, V., "Cours d'economie politique", Librairie Droz, 1964. ISBN 9782600040143

[20] Zhang, H., Lei, X., Wang, C., Yue, D., Xie, X. "Adaptive grid based multi-objective Cauchy differential evolution for stochastic dynamic economic emission dispatch with wind power uncertainty", PLoS ONE, 12(9), e0185454, 2017. https://doi.org/10.1371/journal.pone.0185454

[21] The MathWorks Inc. "Statistics and machine learning toolbox documentation, (2019b)", [computer program], Available at: https://www.mathworks.com/help/stats/index.html

[22] Behnamian, J., Ghomi, S. M. T. F., Zandieh, M. "A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic", Expert Systems with Applications, 36(8), pp. 11057–11069, 2009. https://doi.org/10.1016/j.eswa.2009.02.080

[23] Sierra, M. R., Coello Coello, C. A. "Improving PSO-based multi-objective optimization using crowding, mutation and ε-dominance", Lecture Notes in Computer Science, 3410, pp. 505–519, 2005. https://doi.org/10.1007/978-3-540-31880-4_35

[24] Kaveh, A., Laknejadi, K. "A new multi-swarm multi-objective optimization method for structural design", Advances in Engineering Software, 58, pp. 54–69, 2013. https://doi.org/10.1016/j.advengsoft.2013.01.004

[25] AISC Manual Committee "Manual of steel construction: allowable stress design", American Institute of Steel Construction (AISC), 1989. ISBN 978-1564240002

[26] Hasançebi, O., Çarbaş, S., Doğan, E., Erdal, F., Saka, M. P. "Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures", Computers and Structures, 87(5–6), pp. 284–302, 2009. https://doi.org/10.1016/j.compstruc.2009.01.002

[27] Tran, D. H., Long, L. D. "Project scheduling with time, cost and risk trade-off using adaptive multiple objective differential evolution", Engineering, Construction and Architectural Management, 25(5), pp. 623–638, 2018. https://doi.org/10.1108/ECAM-05-2017-0085

[28] Al-Gahtani, K. S. "Float allocation using the total risk approach", Journal of Construction Engineering and Management, 135(2), pp. 88–95, 2009. https://doi.org/10.1061/(asce)0733-9364(2009)135:2(88)

[29] de la Garza Jesús, M., Prateapusanond, A., Ambani, N. "Preallocation of total float in the application of a critical path method based construction contract", Journal of Construction Engineering and Management, 133(11), pp. 836–845, 2007. https://doi.org/10.1061/(asce)0733-9364(2007)133:11(836)

[30] Gong, D., Rowings, J. E. "Calculation of safe float use in risk-analysis-oriented network scheduling", International Journal of Project Management, 13(3), pp. 187–194, 1995. https://doi.org/10.1016/0263-7863(94)00004-V

[31] Kaveh, A., Rastegar Moghaddam, M., Khanzadi, M. "Efficient multi-objective optimization algorithms for construction site layout problem", Scientia Iranica, 25(4), pp. 2051–2062, 2018. https://doi.org/10.24200/sci.2017.4216

[32] Rastegar-Moghaddam, M., Khanzadi, M., Kaveh, A. "Multi – objective billiards – inspired optimization algorithm for construction management problems", Iranian Journal of Science and Technology, Transactions of Civil Engineering, 45(4), pp. 2177–2200, 2021.
https://doi.org/10.1007/s40996-020-00467-w

[33] El-Rayes, K., Khalafallah, A. "Trade-off between safety and cost in planning construction site layouts", Journal of Construction Engineering and Management, 131(11), pp. 1186–1195, 2005.
https://doi.org/10.1061/(ASCE)0733-9364(2005)131:11(1186)