1284 | https://doi.org/10.3311/PPci.16872
Creative Commons Attribution

Periodica Polytechnica Civil Engineering, 64(4), pp. 1284–1305, 2020

An Improved Water Strider Algorithm for Optimal Design of Skeletal Structures

Ali Kaveh*[1], Majid Ilchi Ghazaan[1], Arash Asadi[1]

[1] School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran
* Corresponding author, e-mail: alikaveh@iust.ac.ir

Received: 17 July 2020, Accepted: 04 September 2020, Published online: 05 October 2020

Abstract

Water Strider Algorithm (WSA) is a new metaheuristic method that is inspired by the life cycle of water striders. This study attempts to enhance the performance of the WSA in order to improve solution accuracy, reliability, and convergence speed. The new method, called improved water strider algorithm (IWSA), is tested in benchmark mathematical functions and some structural optimization problems. In the proposed algorithm, the standard WSA is augmented by utilizing an opposition-based learning method for the initial population as well as a mutation technique borrowed from the genetic algorithm. By employing Generalized Space Transformation Search (GSTS) as an opposition-based learning method, more promising regions of the search space are explored; therefore, the precision of the results is enhanced. By adding a mutation to the WSA, the method is helped to escape from local optimums which is essential for engineering design problems as well as complex mathematical optimization problems. First, the viability of IWSA is demonstrated by optimizing benchmark mathematical functions, and then it is applied to three skeletal structures to investigate its efficiency in structural design problems. IWSA is compared to the standard WSA and some other state-of-the-art metaheuristic algorithms. The results show the competence and robustness of the IWSA as an optimization algorithm in mathematical functions as well as in the field of structural optimization.

Keywords

Improved Water Strider Algorithm, structural optimization, skeletal structures, opposition-based learning, generalized space transformation search

1 Introduction

Optimization methods can generally be categorized into two distinct classes: 1) Gradient-based methods and 2) Metaheuristic methods. In the past, the most commonly used optimization techniques were gradient-based algorithm which utilized gradient information to search the solution space near an initial starting point. The objective functions are usually complex and non-convex in engineering design problems and obtaining the gradient or a starting point could be difficult or even impossible in some cases. Metaheuristic algorithms do not need gradient information to solve optimization problems and this is one of the reasons for attraction toward these methods in the last two decades amongst engineers [1].

Developing metaheuristic algorithms could be tracked down in John Holland works in 1975 [2] and what became known as the genetic algorithm. Probably the most well-known metaheuristic algorithm is Particle Swarm Optimization (PSO) developed by Kennedy and Eberhart in 1995 [3]. After these pioneering works, many metaheuristic algorithms have been developed by researchers to solve various kinds of optimization problems. Many of these algorithms are inspired by nature. For instance, Ant Colony Optimization (ACO) [4] is inspired by the behavior of ants for finding food, Artificial Bee Colony (ABC) [5] simulates foraging behavior of honey bees, and Simulated Annealing (SA) [6] is inspired by a heat treatment method in metallurgy. Some of the recently introduced metaheuristic methods employed by many researchers are Grey Wolf Optimization (GWO) [7], Whale Optimization Algorithm (WOA) [8], Charged System Search (CSS) [9], Colliding Bodies Optimization (CBO) [10], Teaching Learning Based Optimization (TLBO) [11]. Besides, many metaheuristic algorithms have been modified to make them suitable for more complex problems such as structural optimization

Cite this article as: Kaveh, K., Ilchi Ghazaan, M., 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

problems or any other branches of science. In this regard, there is usually a chance to enhance a particular metaheuristic algorithm so as to make it suitable for special purposes [12–14]. According to the NFL theory [15], developing new optimization algorithms is an open problem since there is not one optimization algorithm which can successfully solve all optimization problems.

The branch of structural optimization has been extensively developed in the last three decades and could be classified as follows: (1) obtaining optimal size of structural members (size optimization); (2) finding the optimal form for the structure (shape optimization); and (3) achieving optimal size and connectivity between structural members (topology optimization). Metaheuristic algorithms have been widely used in the field of structural optimization [16–22]. Most of the studies focused on the size optimization of structures that its purpose is to design structures with minimum weight or to minimize a target function corresponding to the minimal cost of construction while the design constraints are met simultaneously.

Water Strider Algorithm (WSA) is a new metaheuristic algorithm and its performance has been shown in obtaining near-optimum solutions for mathematical functions as well as structural engineering problems [23]. However, it can still be enhanced in terms of exploration and accuracy of the results and achieving a better solution faster. In this paper, an improved version of this newly developed algorithm is proposed which is called IWSA. The Generalized Space Transformation Search as an opposition-based learning method and a mutation technique is added to the WSA. The opposition-based learning has been previously applied to some metaheuristic algorithms. For instance, this idea has been employed in moth-flame optimization [24], grasshopper algorithm [25], particle swarm optimization [26–29], differential evolution [30], firefly algorithm [31], and sine cosine algorithm [32].

The rest of the paper is organized as follows. In Section 2, a brief overview of the WSA is presented and the IWSA is proposed in Section 3. In Section 4, some numerical examples are studied and the performance of the IWSA is compared with WSA and some other algorithms in the literature. Finally, the conclusions and future works are explained in Section 5.

2 A brief introduction to Water Strider Algorithm (WSA)

The WSA is a population-based algorithm mimics territorial behavior, intelligent ripple communication, mating style, feeding mechanism; and succession of water strider bugs [23]. The steps of this method are briefly described as follows:

2.1 Initial birth

The water striders (WSs) or the candidate solutions are generated randomly in the search space as follows:

$$WS_i^0 = Lb + rand.\left(Ub - Lb\right);\ \ i = 1,2,\ldots,nws\ , \tag{1}$$

where WS_i^0 is the initial positions of the ith WS in the lake (search space). Lb and Ub denote lower and upper bound of variables, respectively. $rand$ is a random number between [0,1] and nws is the number of WS_s (population size). The initial positions of WS_s are evaluated by an objective function to calculate their fitness.

2.2 Territory establishment

To establish nt number of territories, the WS_s are sorted according to their fitness and $\frac{nws}{nt}$ number of groups are orderly created. The jth member of each group is assigned to the jth territory, where j = 1, 2, …, nt. Therefore, the number of WS_s living in each territory is equal to $\frac{nws}{nt}$. The positions in each territory with the worst and best fitness are considered as male (keystone) and female, respectively.

2.3 Mating

The male WS sends ripple to the female WS in order to mate. Since the response of the female is not known, a probability like p is defined for attraction or else repulsion. The p is set to 0.5. The position of the male WS is updated:

$$\begin{cases} WS_i^{t+1} = WS_i^t + R.rand;\ \ if\ mating\ happens\ (with\ probability\ of\ p) \\ WS_i^{t+1} = WS_i^t + R.\left(1 + rand\right);\ \ otherwise \end{cases}. \tag{2}$$

The length of R is calculated by the following formula:

$$R = WS_F^{t-1} - WS_i^{t-1}, \tag{3}$$

where WS_i^{t-1} and WS_F^{t-1} denote the male and female WS in the $(t-1)^{th}$ cycle, respectively.

2.4 Feeding

Mating consumes a lot of energy for water striders and the male WS forages for food after mating. We assess the objective function for food availability. If the fitness is better than the previous fitness the male WS has found food in the new position, and otherwise, it has not. In the latter condition, the male WS moves toward the best WS of the lake (WS_{BL}) to find food according to the following formula:

$$WS_i^{t+1} = WS_i^t + 2rand * (WS_{BL}^t - WS_i^t), \tag{4}$$

2.5 Death and succession

If in the new position, the male *WS* cannot find food, it will die and a new *WS* will replace it:

$$WS_i^{t+1} = Lb_j^t + rand * \left(Ub_j^t - Lb_j^t\right), \quad (5)$$

where Ub_j^t and Lb_j^t denote the maximum and minimum values of the *WS*'s position inside the *j*th territory.

2.6 WSA termination

If the termination condition is not met, the algorithm will return to the mating step for a new loop. Here, the maximum number of function evaluation (MaxNFEs) is considered as the termination condition here.

3 Improved Water Strider Algorithm (IWSA)

Based on the knowledge from opposition-based initialization of population and mutation technique borrowed from the genetic algorithm, the standard WSA was modified to be a more suitable global optimization algorithm. These two features are introduced in the following and the flowchart of the IWSA as shown in Fig. 1.

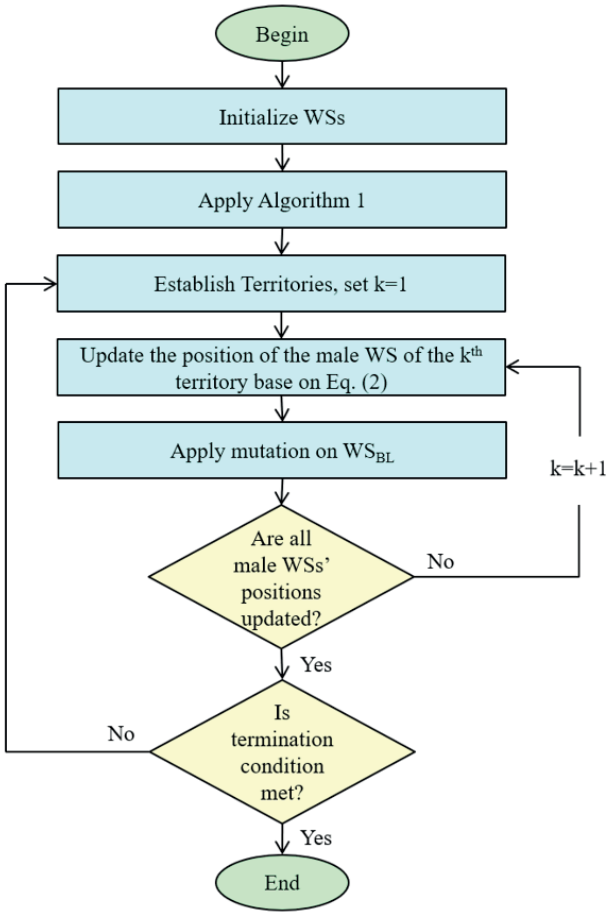


Fig. 1 The flowchart of the proposed IWSA

3.1 Opposition-based learning

The opposition-based learning (OBL) was initially introduced by Tizhoosh [33] for machine learning. By utilizing OBL, the whole search space is searched efficiently by considering the corresponding opposite estimate simultaneously along with the estimate. So, the current estimate is searched in two directions and the search space is searched more efficiently. The opposition-based optimization helps the solution to converge faster hence reducing the time complexity [31].

The opposite of real number $x \in [l, u]$ is given by $\tilde{x}$:

$$\tilde{x} = l + u - x, \quad (6)$$

where l and u are the lower and upper bound of search space, respectively. In the multimodal space, the definition of $\tilde{x}$ can be generalized. Suppose $x = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n$ and $x_j \in [l_j, u_j]$. The opposite point $\tilde{x} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n]$ is defined by:

$$\tilde{x}_j = u_j + l_j - x_j; \;\; j = 1, 2, \ldots, n. \quad (7)$$

3.1.1 Generalized opposition-based learning (GOBL)

Let x be a solution in the current search space S, $x \in [a, b]$. According to GOBL the opposition of the x is calculated as follows:

$$\tilde{x} = k\left(a + b\right) - x, \quad (8)$$

where k is a random number in . The GOBL can also be used in a multi-dimensional search space similar to OBL [26]. When the limits of the variables are violated, the following formula is employed.

$$x_0^* = Lb + k \times \left(Ub - Lb\right), \text{ if } x_0^* \left\langle x_{min} \text{ or } x_0^* \right\rangle x_{max} \quad (9)$$

3.1.2 Generalized Space Transformation Search (GSTS)

Let $P = (x_1, x_2, \ldots, x_D)$ and $Q = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_D)$ denote two different points distributed in D-dimensional space, where $x_1, x_2, \ldots, x_D \in \mathbb{R}$ and $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_D \in \mathbb{R}$. Assume $l = (l_1, l_2, \ldots, l_D)$ and $u = (u_1, u_2, \ldots, u_D)$ are the lower and the upper bounds of the D-dimensional space, respectively. The opposite point $\breve{P} = (\breve{x}_1, \breve{x}_2, \ldots, \breve{x}_D)$ of the point P is defined as:

$$\breve{x}_i = \lambda\left(l_i + u_i\right) - \left(x_i - \hat{x}_i\right), i = 1, 2, \ldots, D, \quad (10)$$

where λ called elastic factor is a random number drawn from interval [0,1].

In comparison to OBL and GOBL, GSTS has higher potential to find the better opposite solution. More specifically, compared with OBL, GOBL and GSTS all can not only enhance the exploitation of the current search space

but also strengthen the exploration in the neighborhood of the current search space while GSTS does better [27]. In order to return the solutions that violate the side constraints into the feasible search space, Eq. (9) is utilized.

3.1.3 Opposition-based optimization

In this optimization strategy, the candidate solution and the corresponding opposite solution are evaluated simultaneously and the fitter solution is stored and the other one is omitted. Let $f(m)$ denote the fitness of a candidate solution $m = (m_1, m_2, \ldots, m_D)$ in D-dimensional space and $f(\tilde{m})$ denote the fitness of its opposite, $\tilde{m} = (\tilde{m}_1, \tilde{m}_2, \ldots, \tilde{m}_D)$ Replace m with $\tilde{m}$ if $(\tilde{m}) > f(m)$; otherwise, leave m unchanged.

Since both the current point and the opposite point are considered simultaneously for computation and evaluation, faster convergence toward a better solution is seen [24].

To strengthen the standard WSA we employ Generalized Space Transformation Search (GSTS) to it. The GSTS is a more general form of the OBL and GOBL. GSTS has also a more potential to find better solutions around optimal solution than OBL and GOBL. Here, GSTS is only applied in the initialization part of the algorithm, and it is not used in the main loop of the algorithm. There are similarly some examples in the literature in which OBL has merely been applied in the population initialization time and has produced a quite successful algorithm [28, 31]. However, OBL has been utilized in the main loop of the metaheuristic algorithms in the literature with a probability called jumping rate [34], too. Herein, utilizing OBL in the main loop might increase the exploration of the algorithm to an unnecessary level. Especially in structural optimization problems in which evaluation of the objective function could be a very costly action, and converging to a near optimal solution with a reasonable number of objective function evaluation is essential. Thus, the generalized space transformation search is applied to enhance the initial population quality of the algorithm.

Initial population plays an important role in any optimization algorithm. It has been shown that the random selection of solutions from a given solution space can result in exploiting the fruitless areas of the search space. Intelligent initialization methods based on realistic approaches are required for efficient results [28]. In fact, it has been proven mathematically and empirically that, in terms of convergence speed, utilizing random numbers and their opposite is more beneficial than using the pure randomness to generate initial estimates in absence of a priori knowledge about the solution [35]. For a better insight into GSTS application in IWSA, the pseudo code is provided in Algorithm 1.

Algorithm 1 Pseudo code of GSTS for initializing population

```
Input: the random initial candidate solutions
Output: the improved initial candidate solutions with GSTS
application
for (each initial candidate solution) do
calculate P̃ = (x̃_1, x̃_2, …, x̃_D) using Eq. (9)
    if f(P̃) > f(P)
        replace P with P̃
    else
        leave P unchanged
    end if
end for
```

3.2 Mutation

The exploration phase of the standard WSA is performed using a random solution created when the keystone cannot reach food after he moves toward the best water strider of the lake. In order to enhance the ability of the WSA to explore more promising regions of the search space and help it to escape from local optimums, a mutation technique is embedded in this metaheuristic. In evolutionary algorithms, mutation plays a significant role to provide diversity of the solutions.

After the last stage of the standard WSA, which is generating a random solution in the search space, mutation is utilized with a probability called *pro* to improve the exploration of the algorithm. In our study, the mutation was applied on the best-so-far solution of the algorithm, which is also called the best water strider of the lake (WS_{BL}). One of the components of the WS_{BL} is selected randomly and regenerated by the following formula:

$$x_j = x_{j,min} + rand.\left(x_{j,max} - x_{j,min}\right), \tag{11}$$

where x_j is the selected component, $x_{j,max}$ and $x_{j,min}$ are the maximum and minimum of all the components in the best territory, respectively. The regenerated *WS* is evaluated using the objective function. If the fitness is better than the previous fitness, this mutated *WS* is replaced with the WS_{BL}. After a number of trial-and-error experiments, the *pro* was set equal to 30 percent for our problems.

4 Numerical examples

In this section, the efficiency of IWSA is investigated through benchmark mathematical functions and structural

optimization problems and the results are compared with the standard WSA and some of the state-of-the-art metaheuristic algorithms.

4.1 Mathematical benchmark functions

In the first step, 23 mathematical functions from the literature (7 unimodal $F_1 - F_7$, 6 multimodal $F_8 - F_{13}$ and 10 fixed dimension multimodal functions $F_{14} - F_{23}$) are considered. These functions are presented in Tables 1–3. The number of territories and the population of WS_S are assumed as 25 and 50, respectively. For a fair comparison, in all algorithms, the maximum number of function evaluations (MaxNFEs) are predefined as 5000 multiples by dimension (Dim). IWSA is executed 30 times independently similar to other algorithms reported in [23]. The statistics results such as average and standard deviation of the results are reported in Tables 4–6. IWSA has outperformed WSA in most of the benchmark mathematical functions $F_1 - F_{23}$ in terms of the average and standard deviation. The results found by IWSA is also better than those of some of the classic well-known metaheuristic namely PSO, GA and ICA. Considering some modern well-established metaheuristic algorithms such as BBO, SSA, SCA, MFO, DA and MVO, the IWSA has also achieved better results for most of the functions in terms of mean and standard deviation values.

Table 1 The unimodal benchmark functions

Function	Dim	Range	f_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10,10]	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_i^2)^2$	30	[-100,100]	0
$F_4(x) = \max_i \{ \lvert x_i \rvert,\ 1 \le i \le n \}$	30	[-100,100]	0
$F_5(x) = \sum_{i=1}^{n} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2 \right]$	30	[-30,30]	0
$F_6(x) = \sum_{i=1}^{n} \left(x_i + 0.5\right)^2$	30	[-100,100]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	30	[-1.28,1.28]	0

To further investigate the performance of the IWSA, in the next step, 21 benchmark functions $F_{24} - F_{44}$ taken from [36] are tested for both IWSA and WSA. These bench-mark functions have been also solved by 13 different methods in [37]. Properties of these functions are

Table 2 The multimodal benchmark functions

Function	Dim	Range	f_{min}
$F_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	[-500,500]	-418.9829 × Dim
$F_9(x) = -\sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$	30	[-5.12,5.12]	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \right) - \exp(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)) + 20 + e$	30	[-32,32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0
$F_{12}(x) = \frac{\pi}{n}\left\{ 10\sin(\pi y_1) + \sum_{i=1}^{n} (x_i - 1)^2 [1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2 \left[1 + sin^2(2\pi x_n) \right] \right\}$ $+ y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} (x_i - a)^m x_i \\ x_i \\ (-x_i - a)^m x_i \end{cases}$	30	[-50,50]	0
$F_{13}(x) = 0.1\left\{ sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2 \left[1 + sin^2(2\pi x_n) \right] \right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[-50,50]	0

Table 3 Multimodal benchmark Functions with fixed dimension

Function	Dim	Range	f_{min}
$F_{14}(x)=(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6})^{-1}$	2	[-65,65]	1
$F_{15}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_i\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[-5,5]	0.00030
$F_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5,5]	-1.0316
$F_{17}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)cosx_1+10$	2	[-5,5]	0.398
$F_{18}(x)=\left[1+\left(x_1+x_2+1\right)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]$ $\times\left[30+\left(2x_1-3x_2\right)^2\times\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$	2	[-2,2]	3
$F_{19}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}\left(x_j-p_{ij}\right)^2);$ $a=\begin{bmatrix}3&10&30\\0.1&10&35\\3&10&30\\0.1&10&35\end{bmatrix},c=\begin{bmatrix}1\\1.2\\3\\3.2\end{bmatrix},and$ $p=\begin{bmatrix}0.3689&0.117&0.2673\\0.4699&0.4387&0.747\\0.1091&0.8732&0.5547\\0.03815&0.5743&0.8828\end{bmatrix}$	3	[0,1]	-3.86
$F_{20}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}\left(x_j-p_{ij}\right)^2)$ $a=\begin{bmatrix}10&3&17&3.5&1.7&8\\0.05&10&17&0.1&8&14\\3&3.5&17&10&17&8\\17&8&0.05&10&0.1&14\end{bmatrix},\ c=\begin{bmatrix}1\\1.2\\3\\3.2\end{bmatrix}$ $p=\begin{bmatrix}0.1312&0.1696&0.5569&0.0124&0.8283&0.5886\\0.2329&0.4135&0.8307&0.3736&0.1004&0.9991\\0.2348&0.1451&0.3522&0.2883&0.3047&0.6650\\0.4047&0.8828&0.8732&0.5743&0.1091&0.0381\end{bmatrix}$	6	[0,1]	-3.32
$F_{21}(x)=-\sum_{i=1}^{5}\left[\left(X-a_i\right)\left(x-a_i\right)^T+c_i\right]^{-1}$	4	[0,10]	-10.1532
$F_{22}(x)=-\sum_{i=1}^{7}\left[\left(X-a_i\right)\left(x-a_i\right)^T+c_i\right]^{-1}$	4	[0,10]	-10.4028
$F_{23}(x)=-\sum_{i=1}^{10}\left[\left(X-a_i\right)\left(x-a_i\right)^T+c_i\right]^{-1}$	4	[0,10]	-10.5363

Table 4 The statistical results of unimodal benchmark functions (F_1–F_7)

		IWSA	WSA	GA	PSO	ICA	BBO	SSA	SCA	MFO	DA	MVO
F_1	Ave	9.25E-195	1.09E-50	0.493651	474.0286	1.78E-27	0.631961	5.28E-09	1.94E-16	2000	101.3336	0.022462
	STD	0	3.99E-50	0.976457	266.9246	4.27E-27	0.665318	8.26E-10	9.45E-16	4068.381	96.58759	0.006160
F_2	Ave	1.04E-105	4.89E-28	0.027773	9.234122	4.24E-15	0.165913	0.535547	1.28E-18	28.66667	7.530338	8.877180
	STD	1.73E-105	1.94E-27	0.053506	2.396783	9.59E-15	0.057118	0.813057	4.59E-18	15.69831	6.143436	33.74294
F_3	Ave	1.80E-14	0.014089	4682.494	4441.754	1.307412	10032.46	6.70E-7	650.4789	16833.44	6410.172	1.916605
	STD	34.74E-14	0.011180	1974.782	1762.155	0.812703	3031.347	4.49E-7	1248.101	12520.75	5456.428	0.706664
F_4	Ave	1.22E-73	0.000490	9.282368	14.10854	0.069323	7.924755	1.328179	1.335270	45.80769	5.930244	0.218609
	STD	1.18E-73	0.000354	2.208485	2.390036	0.093936	1.155925	1.611482	1.828663	13.34950	7.065519	0.093077
F_5	Ave	25.9132	32.42146	481.1478	31370.23	107.5567	219.3536	67.11092	27.51712	18268.11	2890.214	176.5168
	STD	21.3821	29.52849	481.3205	24726.88	135.2055	151.6324	84.22198	0.556120	36497.23	3939.582	253.5243
F_6	Ave	0	0	0.398834	477.1967	1.65E-27	0.500071	5.44E-09	3.720126	1340.033	127.2942	0.018740
	STD	0	0	0.472675	265.9348	3.56E-27	0.650763	9.93E-10	0.334225	3474.982	101.2584	0.005324
F_7	Ave	0.000641	0.006433	0.011947	0.141906	0.026134	0.027428	0.017356	0.00598	1.375182	0.067759	0.005043
	STD	0.0002717	0.0018394	0.0063860	0.059233	0.010122	0.009938	0.006085	0.00590	3.577760	0.050227	0.0017982

Table 5 The statistical results of multimodal benchmark functions (F_8–F_{13})

		IWSA	WSA	GA	PSO	ICA	BBO	SSA	SCA	MFO	DA	MVO
F_8	Ave	-12569.49	−9354.74	−10348.4	−5693.364	-8087.05	-12566.2	-7529.63	-4314.06	-8732.63	-6793.96	-7918.0
	STD	1.85E-12	653.1757	389.5341	614.6499	486.5310	2.18026	705.6967	255.6938	1072.626	989.3598	782.706
F_9	Ave	0.0055	40.56002	9.177337	62.34716	100.6904	1.20078	51.96994	1.83186	148.5348	60.4525	112.474
	STD	0.0071	10.78416	2.255343	18.0979	17.71236	0.82409	16.85193	6.96691	40.87507	27.58083	35.2363
F_{10}	Ave	8.88E-16	1.88E-14	1.123709	6.63813	0.04571	0.23474	1.78994	12.38263	9.7306	5.09027	0.15472
	STD	0	4.52E-15	0.558145	1.00764	0.24526	0.1827	0.86588	9.10541	9.74539	2.14984	0.40645
F_{11}	Ave	0.0067	0.016042	0.293621	4.63638	0.02423	0.52837	0.01026	0.00786	21.14255	1.9369	0.10271
	STD	0.0095	0.020111	0.246095	1.83215	0.02844	0.22331	0.01217	0.02831	45.56648	1.57439	0.03706
F_{12}	Ave	1.57E-32	1.57E-32	0.128613	7.71879	0.03784	0.00879	1.80927	0.3975	0.25016	2.68162	0.20096
	STD	5.57E-48	5.57E-48	0.153503	3.60638	0.20661	0.02595	1.5772	0.13269	0.48244	5.13881	0.37664
F_{13}	Ave	1.35E-32	1.35E-32	0.177732	873.08203	1.99E-23	0.02846	0.00366	2.06867	1.37E + 7	0.19126	0.01012
	STD	5.57E-48	5.57E-48	0.139143	3390.035	1.08E-22	0.02109	0.00527	0.13673	7.49E + 7	11.45285	0.01326

described in Tables 7–9. These functions are more complicated than the first 23 functions and finding their optimum is more challenging for an optimization algorithm. They incorporate shifted unimodal, and shifted multimodal as well as hybrid composite functions. IWSA and WSA are run 30 times independently and the termination condition is 5000 multiplied by dimension which is set as 50 for all the functions.

The statistical results including average, best, worst, standard deviation as well as Friedman test [38] are provided in Table 10. The Friedman test is a non-parametric statistical test employed to detect the differences among the algorithms. The confidence level of 0.05 is used to assess the significance level of difference amongst the algorithms. Thus, if the p-value is less than 0.05, we can reject the null hypothesis. The methods utilized in [37] are NNA, RS, TLBO, ICA, CS, GSA, WCA, HS, PSO, GA, SA, DE, CMA-ES. According to Table 11 [37], the NNA (Neural Network Algorithm) was placed at the first rank and the DE and TLBO were located in the second and third place, respectively. Here, statistical results of the NNA are compared with IWSA and WSA because it has the best performance amongst the 13 mentioned methods for solving these 21 benchmark functions.

Table 6 The statistical results of fixed-dimension multimodal benchmark functions (F_{14}–F_{23})

		IWSA	WSA	GA	PSO	ICA	BBO	SSA	SCA	MFO	DA	MVO
F_{14}	Ave	0.998004	0.998004	1.130409	3.693964	1.330271	3.527829	1.592317	1.794415	1.525135	1.757204	1.560495
	STD	5.83E-17	1.13E-16	0.430993	2.446979	0.655267	3.634702	1.150621	1.892839	1.34095	1.289434	0.810885
F_{15}	Ave	0.0011	0.000549	0.001722	0.000848	0.000686	0.003912	0.002086	0.001134	0.00092	0.001832	0.003426
	STD	0.0036	0.00032	0.003555	0.000504	0.000158	0.004179	0.004974	0.00035	0.000284	0.001337	0.006762
F_{16}	Ave	-1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03082	−1.03163	−1.03156	−1.03163	−1.03163	−1.03163
	STD	5.22E-16	5.68E-16	1.27E-15	6.45E-16	5.05E-16	0.001666	4.95E-14	7.79E-05	6.78E-16	2.70E-14	1.47E-06
F_{17}	Ave	0.397887	0.397887	0.397887	0.397887	0.397887	0.40586	0.397887	0.403152	0.397887	0.397887	0.39789
	STD	0	0	0	0	0	0.011123	9.94E-14	0.007668	0	6.48E-15	3.98E-06
F_{18}	Ave	3	3	3	3	3	5.687621	3	3.000109	3	3	3.000014
	STD	2.49E-15	2.91E-15	1.69E-15	1.58E-15	4.15E-15	6.048765	4.75E-13	0.000124	2.04E-15	4.13E-09	1.23E-05
F_{19}	Ave	-3.86278	−3.86278	−3.86278	−3.86278	−3.86278	−3.86178	−3.86278	−3.85388	−3.86278	−3.86071	−3.86278
	STD	2.32E-15	2.46E-15	2.71E-15	2.68E-15	2.36E-15	0.001816	7.55E-10	0.002132	2.71E-15	0.003279	2.83E-06
F_{20}	Ave	-3.2705	−3.25066	−3.28633	−3.28826	−3.31011	−3.27638	−3.21634	−2.94575	−3.22824	−3.23633	−3.25038
	STD	0.059923	0.059241	0.055415	0.056989	0.036278	0.057813	0.042258	0.320805	0.053929	0.081325	0.059472
F_{21}	Ave	-6.8983	−6.72819	−6.99664	−5.82346	−6.97219	−5.92825	−8.47826	−3.37565	−7.30772	−6.01288	−7.80433
	STD	3.4182	3.378711	3.696605	3.645863	3.350957	3.306321	2.897832	2.046341	3.400748	2.150163	3.018852
F_{22}	Ave	-9.6191	−7.35819	−8.46037	−6.4398	−7.99872	−6.45539	−8.3773	-4.06593	−8.17415	−6.47641	−8.32628
	STD	2.0677	3.609873	3.285714	3.565778	3.259124	3.598817	3.203741	1.942808	3.250245	2.735904	3.062084
F_{23}	Ave	-9.7433	−8.30703	−8.57634	−5.42603	−6.49715	−5.52286	−9.49528	−4.667	−8.66814	−6.24959	−9.02186
	STD	2.0894	3.499898	3.324285	3.527142	3.647386	3.453432	2.700535	1.758723	3.183513	2.475969	2.584132

As seen in Table 12, considering the total average ranking by Friedman test, the IWSA is placed in the first rank, the NNA is placed in the second rank; and the WSA is located in the third rank. According to the Table 10, it is seen that for 14 benchmark functions, IWSA has obtained the first rank among the three methods (including two simultaneous first rank with another algorithm). The p-value in the last column of the table is lower than the confidence level of 0.05 for all the functions except for F_{26}. Therefore, the null hypothesis is rejected for almost all functions and there is a significant difference among the three algorithms.

From Fig. 2 it can be observed that the differences between the performance of the algorithms will be more obvious. IWSA's line is specified by a blue color and the red lines are the algorithms which differ from IWSA significantly while the gray lines are the algorithms that do not differ significantly. For instance, regarding the graph of F_{34}, IWSA has outperformed WSA and NNA, and its result is significantly better than WSA and NNA.

The average convergence history obtained by IWSA and WSA for F24 - F44 are depicted in Fig. 3. They demonstrate that the IWSA has a faster convergence rate in comparison with WSA in most of the cases.

4.2 Structural optimization problems

In this section, three benchmark structural optimization problems are solved by IWSA and its results are compared with the standard WSA and some of the well-established metaheuristic algorithms. Here, the objective is to minimize the weight of the structures to reduce the construction costs by selecting the best possible design variables from a given set of sections provided by valid codes while meeting the design constraints simultaneously (sizing optimization). The optimization problem can formally be stated as

$$\begin{cases} \textit{Find} & \{X\} = \left[x_1, x_2, \ldots, x_{ng} \right] \\ \textit{to minimize} & \mathrm{W}\left(\{X\}\right) = \sum_{i=1}^{ng} x_i \sum_{j=1}^{nm(i)} p_j L_j \\ \textit{subjected to} & \begin{cases} g_j\left(\{X\}\right) \le 0, \quad j = 1,2,\ldots,nc \\ x_i \min \le x_i \le x_i \max \end{cases} \end{cases}, \tag{12}$$

Table 7 The benchmark Functions F_{24} to F_{34} (dimension = 50)

Function	Name	Definition
F_{24}	Shifted Hyper Sphere	$\sum_{i=1}^{n} z_i^2 + \mathrm{f\_bias},\ \ \mathrm{z = x - o^*}$
F_{25}	Shifted Schwefel 2.21	$max_i\left\{\lvert z_i\rvert,\ \ 1 \le \mathrm{i} \le \mathrm{n}\right\} + \mathrm{f\_bias}, \mathrm{z = x - o}$
F_{26}	Shifted Rosenbrock	$\sum_{i=1}^{n}\left[100\left(z_i^2 + z_{i+1}\right)^2 + \left(z_i - 1\right)^2\right] + \mathrm{f\_bias},\ \ \mathrm{z = x - o}$
F_{27}	Shifted Rastrigin	$\sum_{i=1}^{n}\left[z_i^2 - 10\cos\left(2\pi z_i\right) + 10\right] + \mathrm{f\_bias},\ \ \mathrm{z = x - o}$
F_{28}	Shifted Griewank	$\frac{1}{4000}\sum_{i=1}^{n} z_i^2 - \prod_{i=1}^{n}\cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + \mathrm{f\_bias},\ \ \mathrm{z = x - o}$
F_{29}	Shifted Ackley	$-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi z_i\right)\right) + 20 + e + \mathrm{f\_bias},\ \ \mathrm{z = x - o}$
F_{30}	Schwefel 2.22	$\sum_{i=1}^{n}\lvert z_i\rvert + \prod_{i=1}^{n}\lvert z_i\rvert$
F_{31}	Schwefel 1.2	$\sum_{i=1}^{n}\left(\sum_{j-1}^{i} z_i^2\right)^2$
F_{32}	Extended	$\left(\sum_{i=1}^{n-1} f_{10}(z_i, z_{i+1})\right) + f_{10}\left(z_m, z_1\right), f_{10} = \left(x^2 + y^2\right)^{0.25}\left(sin^2\left(50\left(x^2 + y^2\right)^{0.1}\right) + 1\right)$
F_{33}	Bohachevsky	$\sum_{i=1}^{n-1}\left(z_i^2 + 2z_{i+1}^2 - 0.3\cos\left(3\pi z_i\right) - 0.4\cos\left(4\pi z_{i+1}\right) + 0.7\right)$
F_{34}	Schaffer	$\sum_{i=1}^{n-1}\left(z_i^2 + z_{i+1}^2\right)^{0.25}\left(sin^2\left(50\left(z_i^2 + z_{i+1}^2\right)^{0.1}\right) + 1\right)$

Table 8 The composition benchmark functions F_{35} to F_{44}.

Function	First Function	Second Function	Weight Factor
F_{35}	F_{32}	$+F_{24}$	0.25
F_{36}	F_{32}	$+F_{26}$	0.25
F_{37}	F_{32}	$+F_{27}$	0.25
F_{38}	F_{33}	$+F_{30}$	0.25
F_{39}	F_{28}	$+F_{24}$	0.50
F_{40}	F_{26}	$+F_{27}$	0.50
F_{41}	F_{32}	$+F_{24}$	0.75
F_{42}	F_{32}	$+F_{26}$	0.75
F_{43}	F_{32}	$+F_{27}$	0.75
F_{44}	F_{33}	$+F_{30}$	0.75

where $\{X\}$ is the vector containing the design variables; $W(\{X\})$ presents the weight of the structure; $nm(i)$ is the number of members for the ith group; ρ_j and L_j denote the material density and the length of the jth member, respectively. $x_{i\min}$ and $x_{i\max}$ are the lower and upper bounds of the design variable x_i, respectively. $g_j(\{X\})$ denotes the design constraints; and nc is the number of the constraints.

Table 9 The properties of F_{24} to F_{44}.

Function	Range	Optimum	U/M*	Separable	Shifted	
F_{24}	[-100,100]	0	U	Yes	Yes	-450
F_{25}	[-100,100]	0	U	No	Yes	-450
F_{26}	[-100,100]	0	M	Yes	Yes	390
F_{27}	[-5,5]	0	M	Yes	Yes	-330
F_{28}	[-600,600]	0	M	No	Yes	-180
F_{29}	[-32,32]	0	M	Yes	Yes	-140
F_{30}	[-10,10]	0	U	Yes	No	-
F_{31}	[-65.536,65.536]	0	U	No	No	-
F_{32}	[-100,100]	0	U	No	No	-
F_{33}	[-15,15]	0	U	Yes	No	-
F_{34}	[-100,100]	0	U	Yes	No	-
F_{35}	[-100,100]	0	U	No	Yes	-
F_{36}	[-100,100]	0	M	No	Yes	-
F_{37}	[-5,5]	0	M	No	Yes	-
F_{38}	[-10,10]	0	U	Yes	No	-
F_{39}	[-100,100]	0	M	No	Yes	-
F_{40}	[-10,10]	0	M	Yes	Yes	-
F_{41}	[-100,100]	0	U	No	Yes	-
F_{42}	[-100,100]	0	M	No	Yes	-
F_{43}	[-5,5]	0	M	No	Yes	-
F_{44}	[-10,10]	0	U	Yes	No	-

Table 10 Optimization results for benchmark functions F_{24}–F_{44}.

		WSA	IWSA	NNA	p-value
F_{24}	Ave	0	0	2.25e-10	
	Best	0	0	4.95e-12	
	Worst	0	0	1.39e-9	
	STD	0	0	3.45e-10	9.36e-14
	Ave. rank. Friedman test	1.5	1.5	3	
F_{25}	Ave	0.1397	0.1633	1.03	
	Best	0.0494	0.0812	0.452	
	Worst	0.3755	0.3929	1.65	
	STD	0.0791	0.0626	0.31	9.93e-11
	Ave. rank. Friedman test	1.37	1.63	3	
F_{26}	Ave	75.7416	106.6633	96.0	
	Best	4.0971	0.1268	0.539	
	Worst	377.2147	362.3666	335	0.0794
	Ave. rank. Friedman test	1.67	2.20	2.13	
F_{27}	Ave	112.5852	4.7691	6.60	
	Best	61.6874	0.5223	1.99	
	Worst	162.1779	17.2707	11.9	5.10e-11
	Ave. rank. Friedman test	3	1.3	1.7	
F_{28}	Ave	0.0079	0.0121	0.062	
	Best	0	0	6.76e-12	
	Worst	0.0443	0.0850	0.161	2.26e-07
	Ave. rank. Friedman test	1.62	1.58	2.8	
F_{29}	Ave	1.712e-11	5.3054e-14	1.12e-06	
	Best	2.8422e-14	2.8422e-14	1.21e-07	
	Worst	2.7256e-10	1.1369e-13	5.27e-06	5.15e-12
	Ave. rank. Friedman test	1.78	1.22	3	

		WSA	IWSA	NNA	p-value
F_{30}	Ave	5.8425e-62	5.9421e-106	3.99e-11	
	Best	1.3248e-71	5.4202e-109	1.78e-12	
	Worst	1.4217e-60	6.5974e-105	1.44e-10	9.36e-14
	Ave. rank. Friedman test	2	1	3	
F_{31}	Ave	3.4286e-96	2.5197e-176	5.18e-20	
	Best	2.6174e-104	1.7090e-183	6.80e-23	
	Worst	8.1214e-95	7.0225e-175	3.08e-19	9.36e-14
	Ave. rank. Friedman test	2	1	3	
F_{32}	Ave	72.0886	4.2070e-47	3.21	
	Best	19.2195	5.4523e-49	0.107	
	Worst	133.8062	1.7425e-46	11.0	9.36e-14
	Ave. rank. Friedman test	3	1	2	
F_{33}	Ave	1.7568	0	0	
	Best	0.4699	0	0	
	Worst	6.2987	0	0	9.36e-14
	Ave. rank. Friedman test	3	1.5	1.5	
F_{34}	Ave	78.9188	1.4532e-43	5.80	
	Best	24.0197	2.4220e-48	0.0647	
	Worst	154.2299	4.3454e-42	20.4	9.36e-14
	Ave. rank. Friedman test	3	1	2	
F_{35}	Ave	88.3081	9.9425	16.4	
	Best	54.0279	0.0069	0.175	
	Worst	118.2414	25.0593	52.9	1.23e-10
	Ave. rank. Friedman test	3	1.4	1.60	
F_{36}	Ave	186.9210	72.3444	146	
	Best	101.6506	1.8425	9.70	
	Worst	740.5083	432.3517	942	1.51e-05
	Ave. rank. Friedman test	2.7	1.6	1.7	
F_{37}	Ave	88.1138	8.1221	6.83	
	Best	51.0990	1.4281	1.81	
	Worst	127.4487	18.2882	11.5	1.25e-10
	Ave. rank. Friedman test	3	1.6	1.4	
F_{38}	Ave	2.9507e-35	5.5811e-29	3.48e-11	
	Best	2.6456e-37	1.5090e-31	1.48e-12	
	Worst	2.8271e-34	6.0541e-28	2.75e-10	9.36e-14
	Ave. rank. Friedman test	1	2	3	
F_{39}	Ave	0.5456	1.6829e-30	7.35e-11	
	Best	0	0	4.56e-18	
	Worst	1.9254	5.0487e-29	2.01e-09	3.29e-07
	Ave. rank. Friedman test	2.3	1.23	2.47	
F_{40}	Ave	24.0433	0.1303	1.39	
	Best	8.9552	1.2233e-14	4.63e-04	
	Worst	44.7731	2.9021	9.60	2.46e-13
	Ave. rank. Friedman test	3	1.03	1.97	
F_{41}	Ave	71.5098	16.0430	11.4	
	Best	34.7482	2.9948	1.90	
	Worst	115.7428	53.8301	33.9	7.35e-11
	Ave. rank. Friedman test	3	1.66	1.33	
F_{42}	Ave	213.7061	21.4102	4.21	
	Best	142.1433	3.9660	0.108	
	Worst	309.2074	85.8444	13.8	1.39e-12
	Ave. rank. Friedman test	3	1.90	1.10	
F_{43}	Ave	21.7265	3.8763	2.62	
	Best	14.9470	0.4509	0.00168	
	Worst	30.4789	7.9178	5.59	9.93e-11
	Ave. rank. Friedman test	3	1.63	1.36	
F_{44}	Ave	0.1325	3.5755e-62	8.85e-38	
	Best	0	0	4.39e-72	
	Worst	1.0498	1.0726e-60	2.66e-36	3.75e-08
	Ave. rank. Friedman test	2.1	1.2	2.7	

Table 11 Sum of average ranking using Friedman test for optimization algorithms used in [37]

Methods	Total Average Ranking by Friedman Test (Rank)
NNA	**84.41 (1)**
RS	271.74 (13)
TLBO	103.21 (3)
ICA	232.52 (12)
CS	139.35 (8)
GSA	107.27 (4)
WCA	163.17 (10)
HS	126.03 (7)
PSO	206.81 (11)
GA	116.06 (5)
SA	140.05 (9)
DE	85.33 (2)
CMA-ES	125.25 (6)

Table 12 Sum of average ranking using Friedman test for IWSA, WSA and NNA [37]

Methods	Total Average Ranking by Friedman Test (Rank)
IWSA	**30.18 (1)**
NNA	45.76 (2)
WSA	50.04 (3)

For constraints handling, a penalty approach is utilized. For this purpose, the objective function (Eq. (12)) is redefined as follows:

$$p\left(\{X\}\right)=\left(1+\varepsilon_1.\upsilon\right)^{\varepsilon_2}\times W\left(\{X\}\right), \tag{13}$$

where $P(\{X\})$ is the penalized cost function or the objective function to be minimized and υ denotes the sum of the violations of the design constraints. Here, ε_1 is set to unity and ε_2 is increased linearly from 1.5 to 3 during the optimization process.

In the following subsections the benchmark examples are examined. All our codes are implemented in MATLAB and the structures are analyzed using direct stiffness method with our own codes. In all examples, the population size for the IWSA and WSA are considered as *nws* = 50 and the number of territories is considered to be 25. Termination condition is the maximum number of structural analyses which is $MaxNFE_S$ = 20000 and twenty independent runs are considered for each example.

4.2.1 3-bay 15-story frame

Fig. 4 shows the 3-bay 15-story steel moment frame as the first design example. The applied loads and the numbering of the member groups are also shown in this figure. The modulus of elasticity is 29 Msi (200 GPa), and the yield stress is 36 ksi (248.2 MPa). The effective length factors of the members are calculated as $k_x \geq 0$ for a sway-permitted frame, and the out-of-plane effective length factor is specified as $k_y \geq 1$. Each column is considered as non-braced along its length, and the non-braced length for each beam member is specified as one-fifth of the span length. Limitation on displacement and strength are imposed according to the provisions of the AISC-LRFD [39]. The design variables are chosen from 267 W-shaped sections.

This benchmark problem has been studied by many researchers; therefore, it is a suitable example to investigate the performance of the proposed algorithm against some well-established metaheuristic algorithms. The results obtained by some metaheuristic methods reported in the literature [16, 40] as well as WSA and IWSA are provided in Table 13. The best designs found by IWSA, ECBO, and VPS have approximately the same weight. The second lightest design is reached by cuckoo search algorithm (CS) which is 0.6 % heavier than IWSA, ECBO and VPS. The mean and standard deviation of the obtained results show that the IWSA has performed very well. By comparing the IWSA and WSA results, it is seen that, the best and the mean weight obtained by the IWSA is about 0.6 % lighter than the WSA. The best design convergence curve as well as the mean convergence curve of the IWSA and WSA are depicted in Fig. 5.

4.2.2 384-bar double-layer barrel vault

The second design problem is the size optimization of a 384-bar double-layer barrel vault which is shown in Fig. 6. The span of the barrel vault is 24.82 m, its rise is 5.12 m, and its length is 26.67 m. The depth of the structure, i.e., the distance between the top and bottom layers, is equal to 1.35 m. The barrel vault consists of 111 pinned joints and 384 bar elements, which are grouped into 31 independent sizing variables as depicted in Fig. 6(b). The modulus of elasticity is considered to be 30450 ksi (210000 MPa), the yield stress of steel is taken as 58 ksi (400 MPa), and the density of steel is equal to 0.288 lb/in^3 (7833.413 kg/m^3). All connections are assumed as ball jointed and the supports are considered at the two external edges of the top layer of the barrel vault. Vertical concentrated loads of -20 kips (-88.964 kN) are applied to all free joints (non-support joints) of the top layer. Strength and slenderness limitations are according to AISC-ASD provision [41]. Displacement constraints of ±0.1969 *in* (5 mm) are imposed on all nodes in *x*, *y* and *z* directions.

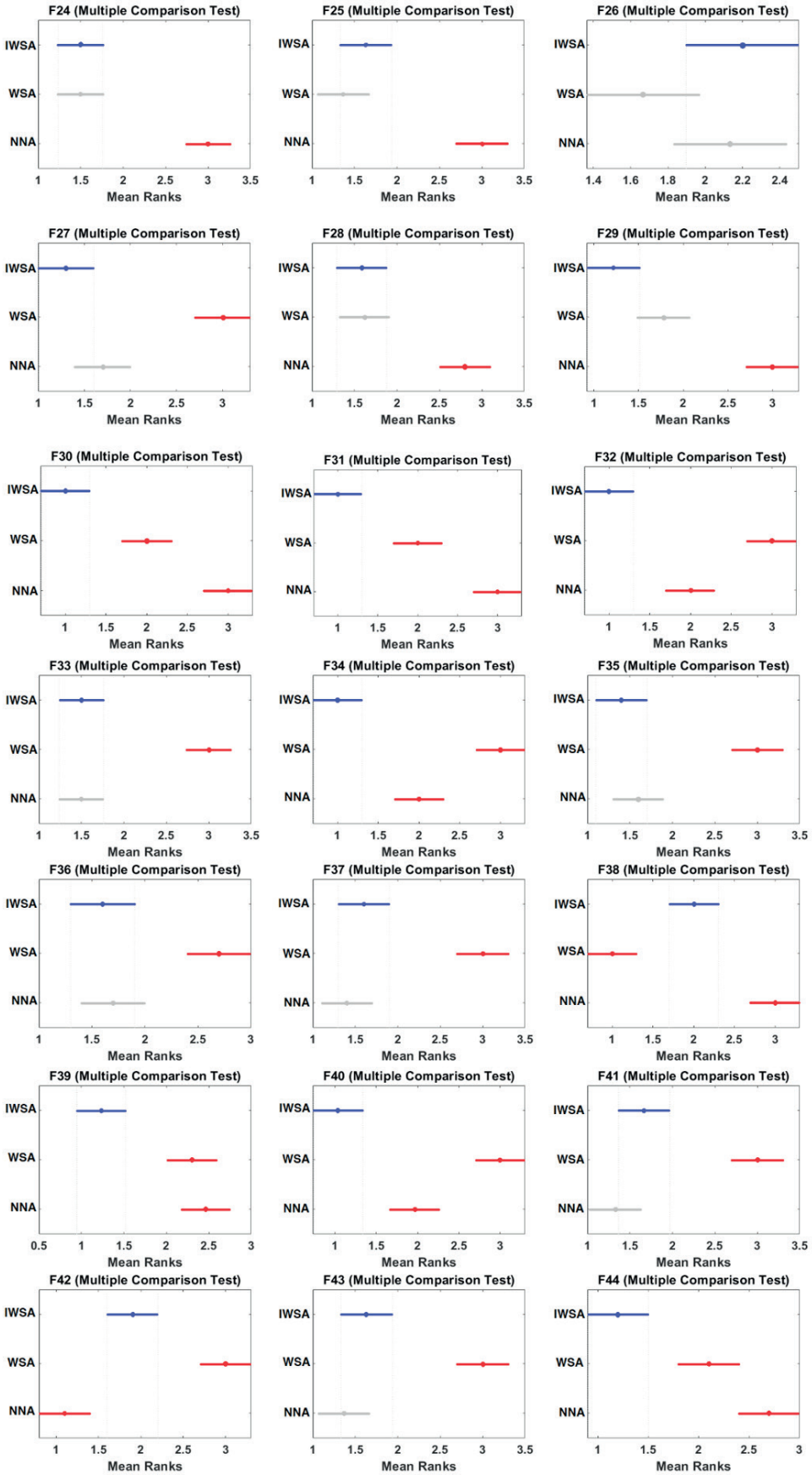


Fig. 2 Multiple comparison test for benchmark functions F_{24}–F_{44} obtained by IWSA, WSA and NNA

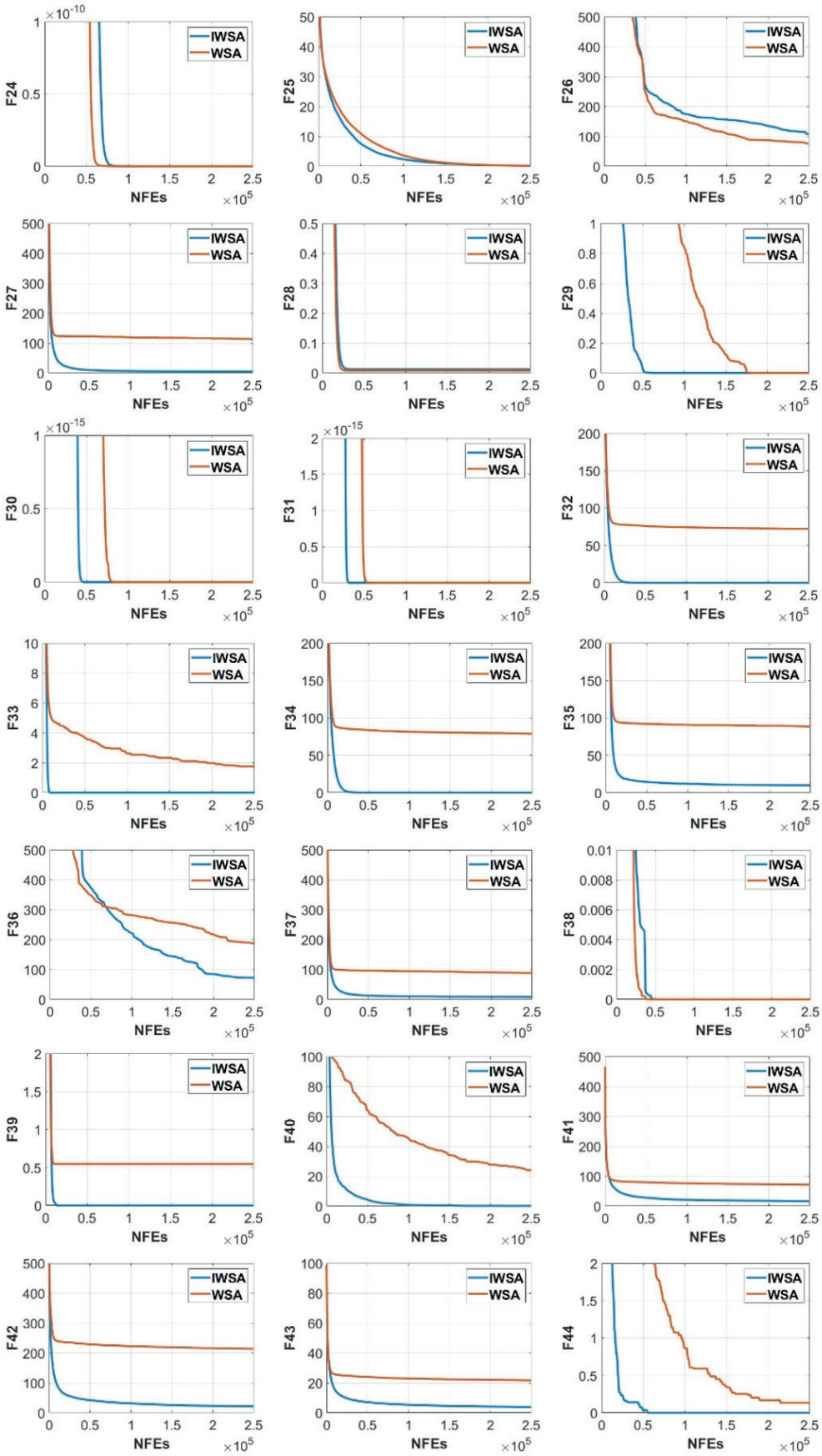


Fig. 3 Average convergence curves obtained by IWSA and WSA for the benchmark functions 24–44

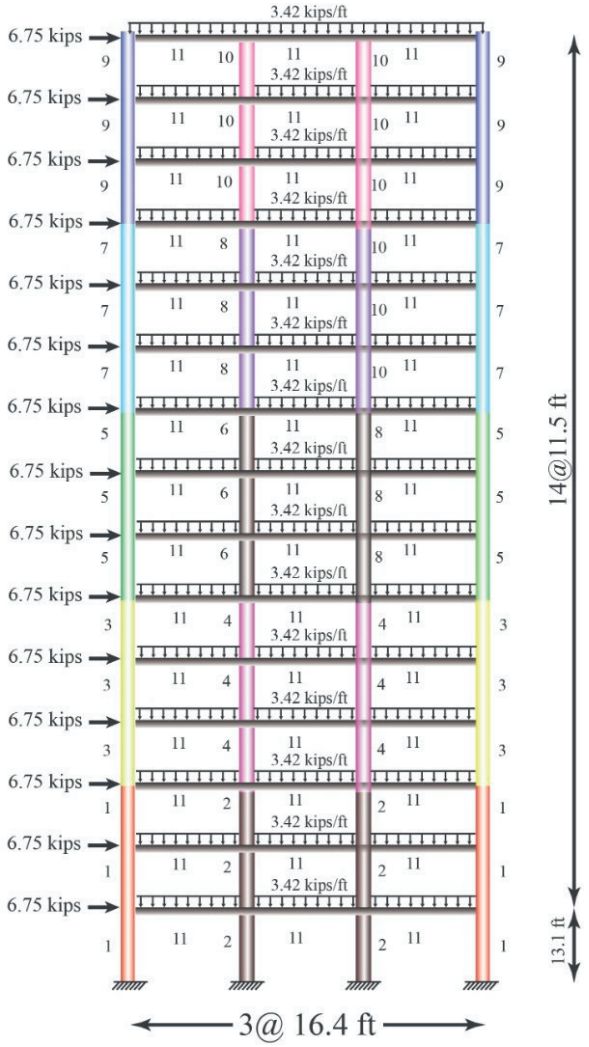


Fig. 4 The 3-bay 15-story frame

The statistical results achieved by different methods [16] are provided in Table 14. The IWSA has reached the lightest weight among the 6 algorithms in terms of best design and average optimized weight, and after that, WSA is placed in the second. The weight for the best design and the average optimized weight achieved by the proposed algorithm is 0.64 % and 0.75 % lighter in comparison to WSA, respectively. The standard deviation achieved by IWSA is lower than WSA and the other four algorithms which demonstrates that IWSA is a more reliable and robust algorithm.

The convergence curves of the algorithms are depicted in Fig. 7. The IWSA has a faster convergence rate than WSA in terms of average convergence curve. The stress ratio graph for the best design obtained by IWSA is illustrated in Fig. 8. The maximum stress ratio is 98.22 %.

4.2.3 800-bar grid

The last design problem deals with the size optimization of the 800-bar grid shown in Fig. 9. The structure has 800 members and 221 nodes and the bottom layer is simply supported at the nodes illustrated in Fig. 10. Each top layer joint is subjected to a concentrated vertical load of 30 kN. Cross-section areas of the members are categorized into 24 groups as shown in Fig. 10. The design variables are the cross-sectional areas of the bar elements which are selected from the list of steel pipe sections from AISC-LRFD [39]. The modulus of elasticity, the yield stress, and the density of steel are taken as 205 GPa, 248.2 MPa, and 7833.413 kg/m^3, respectively. Strength and slenderness

Table 13 Performance comparison for the 3-bay 15-story frame structure

Element group	Sections										
	HPSACO	ABC	ICA	CS	ECBO	BB-BC	TLBO	VPS	TEO	WSA	IWSA
1	W21×111	W14×99	W24×117	W14×109	W14×99	W21×122	W12×96	W14×90	W18×86	W14×99	W14×90
2	W18×158	W40×264	W21×147	W27×161	W27×161	W27×146	W27×161	W36×170	W36×182	W27×161	W36×170
3	W10×88	W14×82	W27×84	W27×84	W27×84	W27×84	W27×84	W14×82	W14×68	W27×84	W14×82
4	W30×116	W33×118	W27×114	W24×104	W24×104	W30×108	W24×104	W24×104	W36×182	W24×104	W24×104
5	W21×83	W21×68	W14×74	W14×61	W14×61	W24×68	W10×68	W21×68	W10×49	W21×68	W21×68
6	W24×103	W18×86	W18×86	W30×90	W30×90	W16×89	W30×90	W18×86	W30×99	W30×90	W18×86
7	W21×55	W21×93	W12×96	W14×48	W14×48	W16×57	W8×48	W21×48	W21×48	W14×48	W21×48
8	W27×114	W12×58	W24×68	W21×68	W14×61	W21×68	W24×68	W14×61	W14×68	W12×65	W14×61
9	W10×33	W40×149	W10×39	W6×25	W14×30	W12×50	W8×28	W12×30	W6×25	W8×28	W12×30
10	W18×46	W18×35	W12×40	W14×43	W12×40	W18×35	W10×39	W10×39	W10×45	W10×39	W10×39
11	W21×44	W18×46	W21×44	W21×44	W21×44	W14×48	W21×50	W21×44	W24×68	W21×44	W21×44
Weight (lb)	95,850	88,536	93,846	87,469	86,986	89,483	87,735	86,985	87,735	87,538	86,986
Mean weight (lb)	N/A	101,424	N/A	99,674	88,410	98,039	95,206	90,066	95,206	89,044	88,522
STD (lb)	N/A	31,734	N/A	24,308	N/A	19,215	11,346	2,533	11,346	1,576	1,490

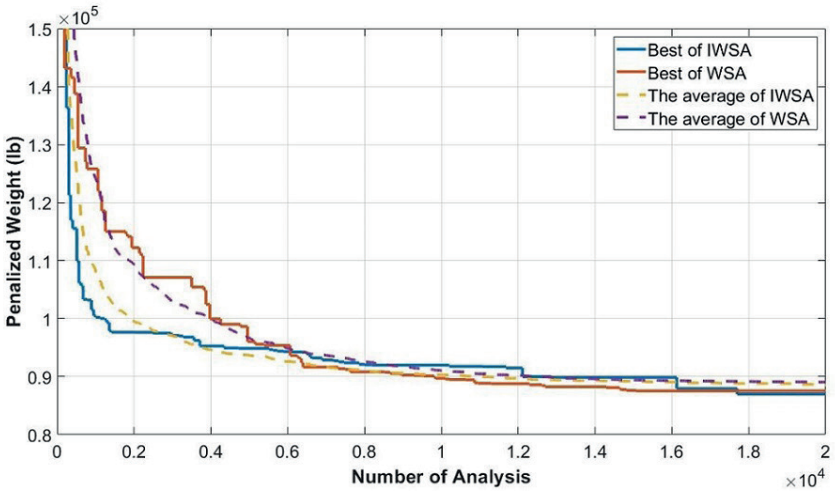


Fig. 5 Convergence curve of IWSA and WSA for the 3-bay 15-story frame

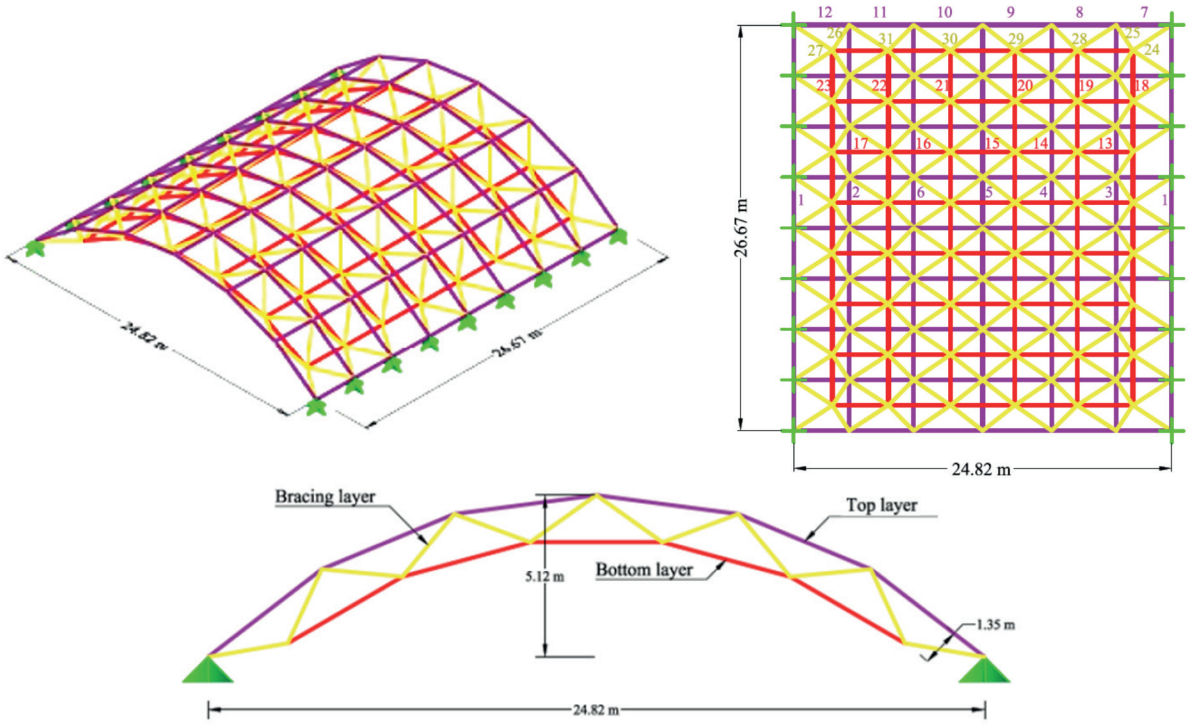


Fig. 6 (a) 3D view, (b) plan view with group numbers (c) flattened cross-sectional view of the 384-bar double-layer barrel vault

limitations are according to AISC-LRFD provisions, and displacement limitations of span/600 were imposed on all nodes in vertical direction.

The example has been optimized by CBO, ECBO, VPS and MDVC-UVPS algorithms [16] and comparison of optimized designs are provided in Table 15. The lightest design is achieved by the proposed method which is 53101 kg that shows the IWSA is more precise than other algorithms. In terms of average optimized weight and standard deviation, the proposed algorithm has outperformed the other methods.

The average convergence curve and the convergence curve for the best result of WSA and IWSA are depicted in Fig. 11. It can be seen that for the initial and middle part of the curve, both algorithms have approximately achieved similar results in nearly the same rate (with WSA doing a little faster). After the middle part, WSA cannot improve the obtained solution, but IWSA has still found the better designs. The stress ratio for the best design obtained by IWSA is depicted in Fig. 12. The maximum stress ratio is 90.79 %.

Table 14 Performance comparison for the 384-bar double-layer barrel-vault problem

	Sections					
Element group	CBO	ECBO	VPS	MDVC-UVPS	WSA	IWSA
1	ST 1/2	ST 1/2	ST 3/4	ST 1/2	ST 1/2	EST 1/2
2	EST 2	ST 2 1/2	EST 2 1/2	EST 2	EST 2	EST 2
3	EST 2	EST 2	EST 2 1/2	EST 2	EST 2	ST 2 1/2
4	ST 3	ST 1 1/2	EST 1 1/2	ST 1 1/2	ST 1 1/4	ST 1 1/4
5	DEST 2 1/2	EST 4	DEST 3	DEST 3	ST 5	EST 4
6	ST 2 1/2	ST 1 1/2	ST 1 1/2	ST 1 1/2	ST 1 1/4	ST 1 1/4
7	ST 12	ST 12	ST 12	ST 12	EST 8	ST 12
8	DEST 4	ST 10	EST 8	DEST 5	DEST 5	DEST 5
9	DEST 5	ST 12	EST 10	EST 10	EST 12	DEST 6
10	ST 12	DEST 8	EST 10	EST 10	EST 10	EST 10
11	DEST 5	DEST 5	DEST 5	DEST 5	DEST 5	ST 10
12	DEST 6	EST 8	DEST 5	ST 12	ST 10	EST 8
13	DEST 3	ST 6	ST 6	ST 6	DEST 3	DEST 3
14	EST 3 1/2	EST 3 1/2	DEST 3	ST 4	ST 5	EST 3 1/2
15	ST 2 1/2	ST 2 1/2	ST 2 1/2	EST 2 1/2	ST 2 1/2	ST 2 1/2
16	EST 6	ST 5	ST 5	ST 4	EST 3 1/2	ST 4
17	EST 6	EST 4	DEST 3	ST 6	DEST 3	EST 5
18	EST 2	EST 1 1/2	EST 1 1/2	EST 1 1/2	DEST 2	EST 1 1/2
19	EST 2	ST 1 1/4	ST 1 1/4	ST 1 1/4	ST 1 1/4	ST 1 1/4
20	EST 2 1/2	EST 1 1/2	EST 1 1/2	EST 1 1/2	EST 2	EST 1 1/2
21	EST 4	EST 1 1/2	EST 1 1/2	EST 1 1/2	EST 2	EST 2
22	ST 3 1/2	ST 1 1/4	EST 1 1/2	ST 1 1/4	ST 1 1/2	ST 1 1/4
23	EST 1 1/2	EST 1 1/2	EST 1 1/2	EST 1 1/2	EST 1 1/2	EST 1 1/2
24	ST 3 1/2	EST 2 1/2	EST 2 1/2	ST 3 1/2	DEST 2	DEST 2
25	ST 2 1/2	ST 2 1/2	EST 2 1/2	EST 2	EST 2	EST 1 1/2
26	DEST 4	ST 2 1/2	EST 1 1/2	EST 2	EST 2	ST 2 1/2
27	EST 3	DEST 2	ST 3	ST 3 1/2	DEST 2	DEST 2
28	EST 2	EST 1 1/2	EST 1 1/2	EST 2	EST 1 1/2	EST 1 1/2
29	ST 2 1/2	ST 2 1/2	EST 2	EST 2	ST 2 1/2	ST 2 1/2
30	ST 3	EST 1 1/2	EST 2	EST 2	EST 2	EST 2
31	ST 2 1/2	EST 1 1/2	EST 1 1/2	EST 2	EST 1 1/2	EST 2
Weight (lb)	69,448.52	62,486.02	62,455.30	62,735.42	61,962.70	61,564.72
Average (lb)	123,397	65,785	67,900	65,738	64,254	63,771
STD (lb)	103,837	3,386	2,913	2,882	1,670	1,495

5 Conclusions

In this study, an improved version of a newly developed metaheuristic algorithm, namely the water strider algorithm, is proposed by utilizing Generalized Space Transformation Search and a mutation technique. The GSTS was only applied to the initial population for the first time to direct the algorithm in more promising regions of the search space. The mutation was also used on the best-so-far-solution of the algorithm to help WSA jump out of the local optimums. The proposed algorithm is examined by solving benchmark mathematical functions as well as three structural optimization problems. Almost in all of the mathematical examples, IWSA outperformed the standard WSA in terms of the best result, mean result, standard deviation; and convergence rate. IWSA was also tested on two space truss and a steel moment frame design problem. The results demonstrated superiority of the IWSA to the standard version in terms of best design

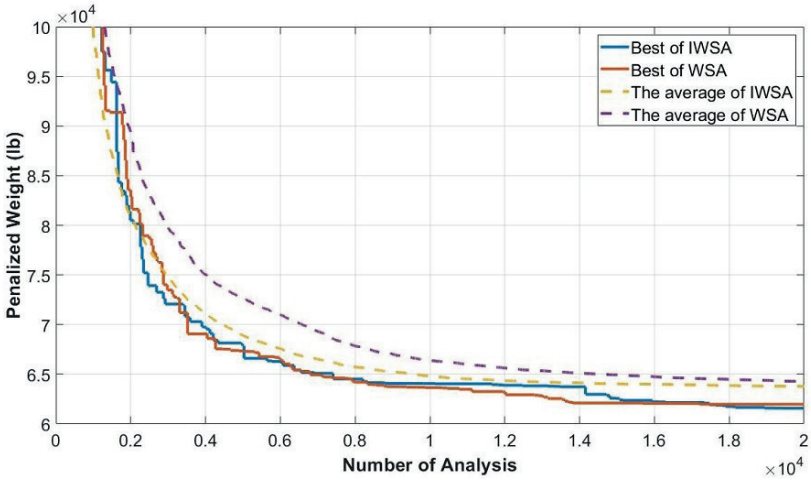


Fig. 7 Convergence curve of IWSA and WSA for the 384-bar barrel vault

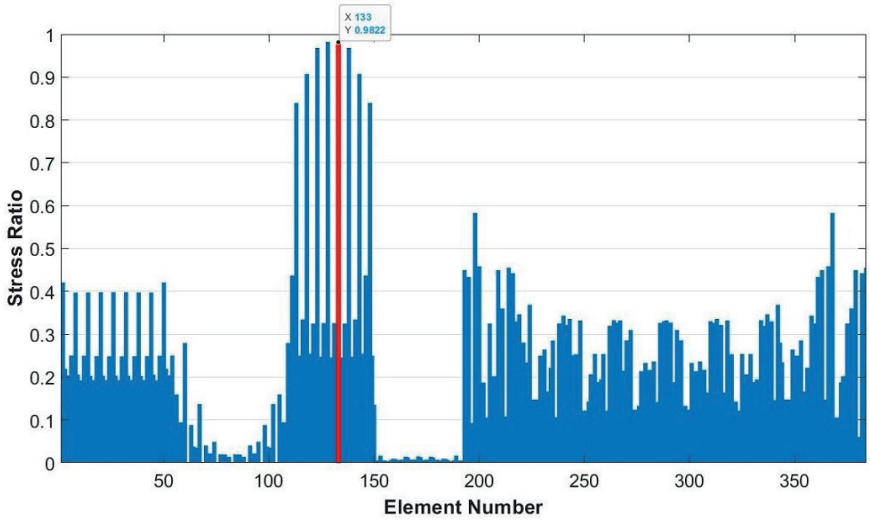


Fig. 8 Stress ratio graph for the best design obtained by IWSA for 384-bar barrel vault

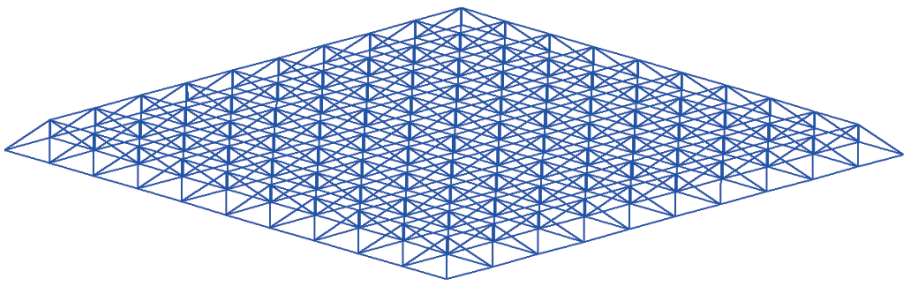
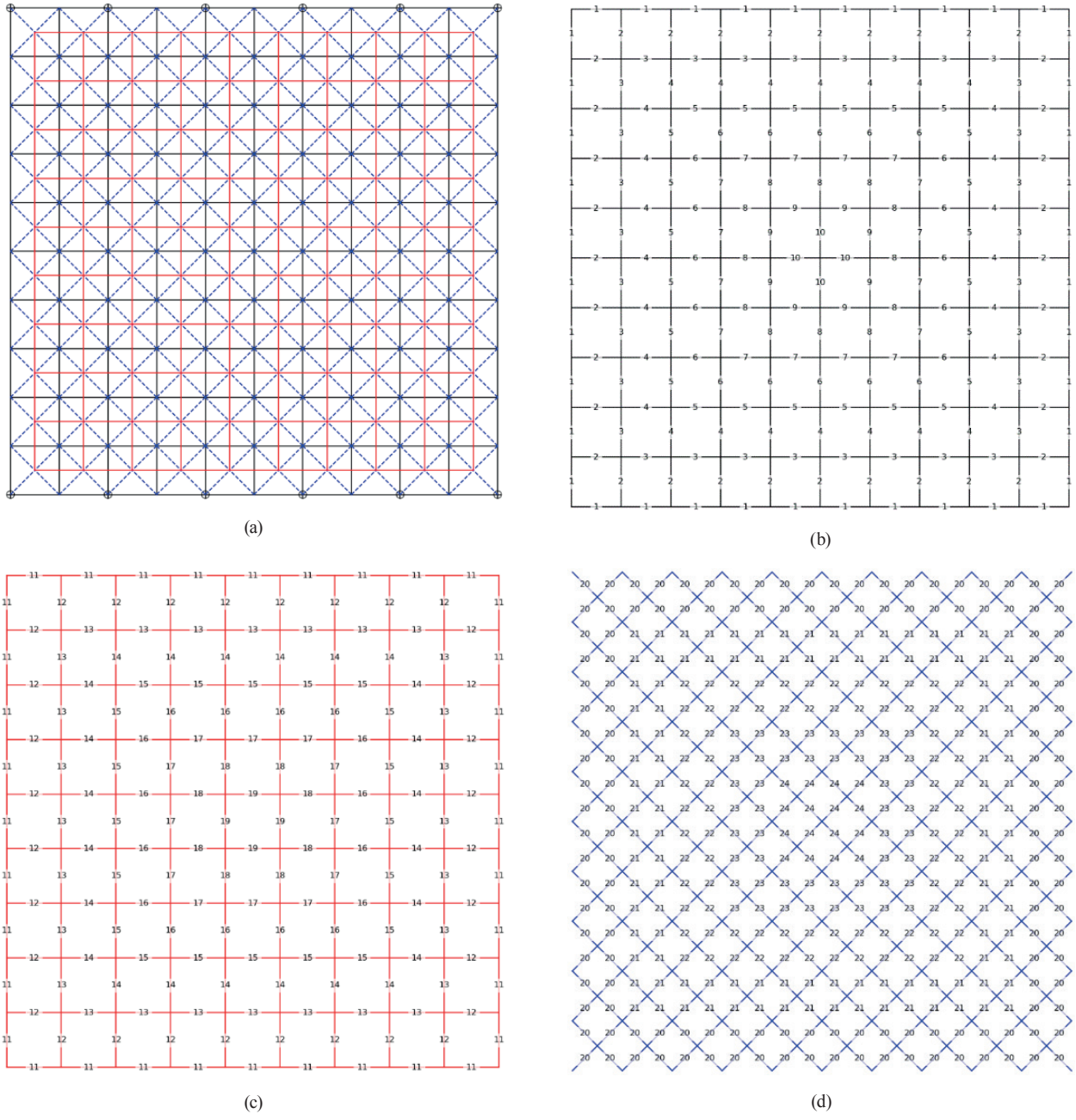


Fig. 9 3D view of the 800-bar double-layer grid problem

weight, average optimized weight, and standard deviation on average weight. This indicates that IWSA is a more reliable and robust algorithm in comparison to its standard version. This also shows that GSTS is a powerful tool in strengthening metaheuristic algorithms, and the proposed mutation technique was successful as the results of IWSA was better that the WSA. IWSA was also compared with some other state-of-the-art metaheuristic algorithms in mathematical function as well as in structural optimization examples. The results confirmed the competitiveness of IWSA in these examples.

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.



(a) (b) (c) (d)

Fig. 10 Top view of the 800-bar double-layer grid problem and member groups: a) all members with simple supports, b) bottom layer members, c) top layer members, and d) web members

Table 15 Performance comparison for the 800-bar double-layer grid problem

	Sections					
Element group	CBO	ECBO	VPS	MDVC-UVPS	WSA	IWSA
1	EST 3 1/2	ST 4	ST 4	ST 4	ST4	ST 4
2	ST 6	ST 5	ST 5	ST 5	ST 5	ST 5
3	ST 2	EST 2	EST 1 1/2	ST 1 1/2	EST 2	EST 1 1/2
4	ST 3 1/2	ST 3	EST 3	ST 3	ST 4	ST 3
5	ST 2 1/2	EST 2	ST 3 1/2	ST 2 1/2	ST 2 1/2	ST 3 1/2
6	ST 3	ST 2	EST 1 1/2	ST 2	ST 2 1/2	ST 2
7	EST 3	EST 3 1/2	ST 5	ST 3	DEST 2 1/2	EST 3
8	ST 2 1/2	ST 3	ST 4	DEST 2	EST 2	DEST 2 1/2
9	EST 3	EST 3 1/2	EST 3	ST 5	ST 5	ST 5
10	ST 5	ST 3	ST 2	DEST 3	EST 3	EST 3 1/2
11	ST 8	EST 5	ST 6	DEST 4	ST 5	EST 5
12	ST 3 1/2	ST 3 1/2	ST 3 1/2	ST 3 1/2	EST 3 1/2	ST 3 1/2
13	ST 4	ST 6	ST 6	ST 6	EST 6	ST 5
14	ST 5	ST 6	ST 6	ST 5	EST 4	ST 5
15	ST 6	ST 6	ST 6	ST 5	ST 6	DEST 4
16	ST 6	ST 6	ST 6	ST 6	ST 5	ST 6
17	DEST 4	EST 5	EST 6	DEST 4	EST 5	EST 5
18	EST 5	EST 6	EST 5	DEST 4	EST 5	DEST 4
19	EST 5	DEST 4	DEST 4	DEST 5	ST 8	ST 6
20	EST 3 1/2	ST 4	ST 4	ST 4	ST 4	ST 4
21	ST 3 1/2	ST 3 1/2	ST 3 1/2	ST 3 1/2	ST 3 1/2	ST 3 1/2
22	ST 3	ST 3 1/2	ST 3	ST 3	ST 3	ST 3
23	ST 2 1/2	ST 2 1/2	ST 2 1/2	ST 2 1/2	ST 2 1/2	ST 2 1/2
24	ST 2 1/2	ST 2 1/2	ST 2 1/2	ST 2 1/2	ST 2 1/2	ST 2 1/2
Weight (kg)	55,714	53,673	53,714	53,590	54,286	53,101
Average (kg)	61,464	58,953	57,912	57,679	56,120	54,764
STD (kg)	10,127	4,643	4,102	3,524	1,568	1,202

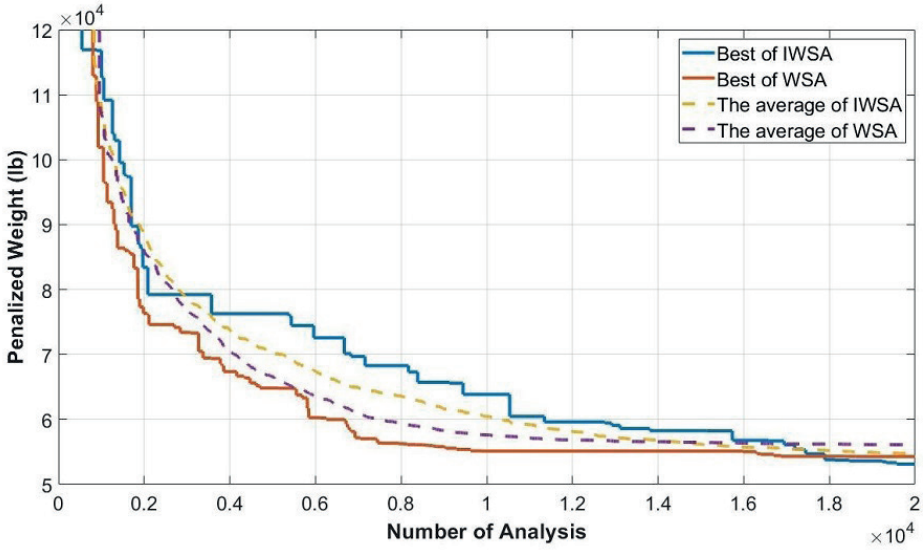


Fig. 11 Convergence curve for the 800-bar double-layer grid

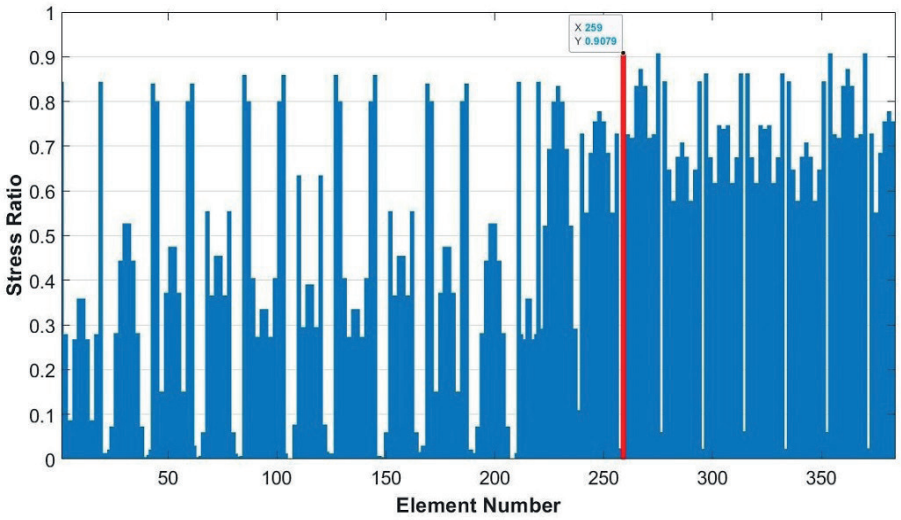


Fig. 12 Stress ratio graph for the best design obtained by IWSA for 800-bar grid

References

[1] Kaveh, A. "Advances in Metaheuristic Algorithms for Optimal Design of Structures", 2nd ed., Springer, Cham, Switzerland, 2017. https://doi.org/10.1007/978-3-319-05549-7

[2] Holland, J. H. "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, USA, 1975.

[3] Kennedy, J., Eberhart, R. "Particle swarm optimization", In: Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942–1948.

[4] Dorigo, M., Maniezzo, V., Colorni, A. "Ant System: Optimization by a Colony of Cooperating Agents", IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 26(1), pp. 29–41, 1996. https://doi.org/10.1109/3477.484436

[5] Karaboga, D. "An idea based on honey bee swarm for numerical optimization", Erciyes University, Kayseri, Türkiye, Rep. TR06, 2005.

[6] Kirkpatrick, S., Gelatt Jr., C. D., Vecchi, M. P. "Optimization by Simulated Annealing", Science, 220(4598), pp. 671–680, 1983. https://doi.org/10.1126/science.220.4598.671

[7] Mirjalili, S., Mirjalili, S. M., Lewis, A. "Grey Wolf Optimizer", Advances in Engineering Software, 69, pp. 46–61, 2014. https://doi.org/10.1016/j.advengsoft.2013.12.007

[8] Mirjalili, S., Lewis, A. "The Whale Optimization Algorithm", Advances in Engineering Software, 95, pp. 51–67, 2016. https://doi.org/10.1016/j.advengsoft.2016.01.008

[9] Kaveh, A., Talatahari, S. "A novel heuristic optimization method: charged system search", Acta Mechanica, 213, pp. 267–289, 2010. https://doi.org/10.1007/s00707-009-0270-4

[10] Kaveh, A., Mahdavi, V. R. "Colliding bodies optimization: A novel meta-heuristic method", Computers & Structures, 139, pp. 18–27, 2014. https://doi.org/10.1016/j.compstruc.2014.04.005

[11] Rao, R. V., Savsani, V. J., Vakharia, D. P. "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems", Computer-Aided Design, 43(3), pp. 303–315, 2011. https://doi.org/10.1016/j.cad.2010.12.015

[12] Guo, W., Liu, T., Dai, F., Xu, P. "An improved whale optimization algorithm for forecasting water resources demand", Applied Soft Computing, 86, Article number: 105925, 2020. https://doi.org/10.1016/j.asoc.2019.105925

[13] Kazemzadeh Azad, S. "Enhanced hybrid metaheuristic algorithms for optimal sizing of steel truss structures with numerous discrete variables", Structural and Multidisciplinary Optimization, 55, pp. 2159–2180, 2017. https://doi.org/10.1007/s00158-016-1634-8

[14] Kaveh, A., Ilchi Ghazaan, M. "Enhanced colliding bodies optimization for design problems with continuous and discrete variables", Advances in Engineering Software, 77, pp. 66–75, 2014. https://doi.org/10.1016/j.advengsoft.2014.08.003

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

[16] Kaveh, A., Ilchi Ghazaan, M. "Meta-heuristic Algorithms for Optimal Design of Real-Size Structures", Springer, Cham, Switzerland, 2018. https://doi.org/10.1007/978-3-319-78780-0

[17] Pham, H. A. "Truss optimization with frequency constraints using enhanced differential evolution based on adaptive directional mutation and nearest neighbor comparison", Advances in Engineering Software, 102, pp. 142–154, 2016. https://doi.org/10.1016/j.advengsoft.2016.10.004

[18] Kaveh, A., Ilchi Ghazaan, M. "A new meta-heuristic algorithm: vibrating particles system", Scientia Iranica, Transaction A, Civil Engineering A, 24(2), pp. 551–566, 2017. https://doi.org/10.24200/sci.2017.2417

[19] Tejani, G. G., Savsani, V. J., Patel, V. K., Savsani, P. V. "Size, shape, and topology optimization of planar and space trusses using mutation-based improved metaheuristics", Journal of Computational Design and Engineering, 5(2), pp. 198–214, 2018. https://doi.org/10.1016/j.jcde.2017.10.001

[20] Han, Z., Hu, Z., Ma, X., Chen, W. "Multimaterial layout optimization of truss structures via an improved particle swarm optimization algorithm", Computers & Structures, 222, pp. 10–24, 2019. https://doi.org/10.1016/j.compstruc.2019.06.004

[21] Kaveh, A., Dadras Eslamlou, A., Khodadadi, N. "Dynamic Water Strider Algorithm for Optimal Design of Skeletal Structures", Periodica Polytechnica Civil Engineering, 64(3), pp. 904–916, 2020. https://doi.org/10.3311/ppci.16401

[22] Kaveh, A., Fazam, M. F., Maroofiazar, R. "Comparing H2 and H∞ algorithms for optimum design of tuned mass dampers under near-fault and far-fault earthquake motions", Periodica Polytechnica Civil Engineering, 64(3), pp. 828–844, 2020. https://doi.org/10.3311/ppci.16389

[23] 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

[24] Sapre, S., Mini, S. "Opposition-based moth flame optimization with Cauchy mutation and evolutionary boundary constraint handling for global optimization", Soft Computing, 23, pp. 6023–6041, 2019. https://doi.org/10.1007/s00500-018-3586-y

[25] Raeesi, F., Farahmand Azar, B., Veladi, H., Talatahari, S. "An inverse TSK model of MR damper for vibration control of nonlinear structures using an improved grasshopper optimization algorithm", Structures, 26, pp. 406–416, 2020. https://doi.org/10.1016/j.istruc.2020.04.026

[26] Wang, H., Wu, Z., Rahnamayan, S., Liu, Y., Ventresca, M. "Enhancing particle swarm optimization using generalized opposition-based", Information Sciences, 181(20), pp. 4699–4714, 2011. https://doi.org/10.1016/j.ins.2011.03.016

[27] Zhang, Y., Jin, Z. "Quantum-behaved particle swarm optimization with generalized space transformation search", Soft Computing 24, pp. 14981–14997, 2020. https://doi.org/10.1007/s00500-020-04850-7

[28] Jabeen, H., Jalil, Z., Baig, A. R. "Opposition based initialization in particle swarm optimization (O-PSO)", In: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers (GECCO '09), Montreal, QC, Canada, 2009, pp. 2047–2052. https://doi.org/10.1145/1570256.1570274

[29] Dong, W., Kang, L., Zhang, W. "Opposition-based particle swarm optimization with adaptive mutation strategy", Soft Computing, 21, pp. 5081–5090, 2017. https://doi.org/10.1007/s00500-016-2102-5

[30] Rahnamayan, S., Tizhoosh, H. R., Salama, M. M. A. "Opposition-Based Differential Evolution", IEEE Transactions on Evolutionary Computation, 12(1), pp. 64–79, 2008. https://doi.org/10.1109/TEVC.2007.894200

[31] Verma, O. P., Aggarwal, D., Patodi, T. "Opposition and dimensional based modified firefly algorithm", Expert Systems with Applications, 44, pp. 168–176, 2016. https://doi.org/10.1016/j.eswa.2015.08.054

[32] Elaziz, M. A., Oliva, D., Xiong, S. "An improved Opposition-Based Sine Cosine Algorithm for global Optimization", Expert Systems with Applications, 90, pp. 484–500, 2017. https://doi.org/10.1016/j.eswa.2017.07.043

[33] Tizhoosh, H. R. "Opposition-based learning: a new scheme for machine intelligence", In: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vienna, Austria, 2005, pp. 695–701. https://doi.org/10.1109/CIMCA.2005.1631345

[34] Rahnamayan, S., Tizhoosh, H. R., Salama, M. M. A. "Opposition-based differential evolution (ODE) with variable jumping rate", In: Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence (FOCI 2007), Honolulu, HI, USA, 2007, pp. 81–88. https://doi.org/10.1109/FOCI.2007.372151

[35] Rahnamayan, S., Tizhoosh, H. R., Salama, M. M. A. "Opposition versus randomness in soft computing techniques", Applied Soft Computing, 8(2), pp. 906–918, 2008. https://doi.org/10.1016/j.asoc.2007.07.010

[36] Herrera, F., Lozano, M., Molina, D. "Test Suite for the Special Issue of Soft Computing on Scalability of Evolutionary Algorithms and other Metaheuristics for Large Scale Continuous Optimization Problems", University of Granada, Granada, Spain, Rep. SCI2S, 2010.

[37] Sadollah, A., Sayyaadi, H., Yadav, A. "A dynamic metaheuristic optimization model inspired by biological nervous systems: Neural network algorithm", Applied Soft Computing, 71, pp. 747–782, 2018. https://doi.org/10.1016/j.asoc.2018.07.039

[38] Friedman, M. "The use of ranks to avoid the assumption of normality implicit in the analysis of variance", Journal of the American Statistical Association, 32(200), pp. 675–701, 1937.

[39] AISC Manual Committee "Manual of steel construction: load resistance factor design", AISC, Chicago, IL, USA, 1994.

[40] Kaveh, A., Biabani Hamedani, K., Hosseini, S. M., Bakhshpoori, T. "Optimal design of planar steel frame structures utilizing metaheuristic optimization algorithms", Structures, 25, pp. 335–346, 2020. https://doi.org/10.1016/j.istruc.2020.03.032

[41] AISC Manual Committee "Manual of steel construction: allowable stress design", AISC, Chicago, IL, USA, 1989.