https://doi.org/10.3311/PPci.16401

Periodica Polytechnica Civil Engineering, 64(3), pp. 904–916, 2020

Dynamic Water Strider Algorithm for Optimal Design of Skeletal Structures

Ali Kaveh[1*], Armin Dadras Eslamlou[1], Nima Khodadadi[1]

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

Received: 07 May 2020, Accepted: 26 May 2020, Published online: 18 June 2020

Abstract

In the present paper, a dynamic version of Water Strider Algorithm (WSA) is proposed. The WSA as well as the Dynamic Water Strider Algorithm (DWSA) are applied to minimize the weight of several skeletal structures. WSA is a nature-inspired metaheuristic that mimics the territorial behavior, intelligent ripple communication, mating style, feeding mechanisms, and succession of water strider insects. The efficiency of these algorithms is tested by optimizing different truss and frame structures subject to multiple loading conditions and constraints. Comparing the results obtained by DWSA with those of other methods it becomes evident that DWSA is a suitable technique for optimizing the structural design and minimizing the weight of structures while fulfilling all constraints.

Keywords

Dynamic Water Strider Algorithm, optimal design, truss, steel frame, metaheuristic

1 Introduction

In engineering design, most of the optimal design problems can be formulated mathematically. To solve these problems, different techniques have been proposed among which the metaheuristic algorithms have drawn considerable attention from the scientific community. Metaheuristics usually offer a balanced trade-off between solution quality and computing time. Furthermore, these approximate techniques are more flexible than the exact ones. For example, unlike the classic methods, they do not require the derivability of functions or any specific demands on the formulation of the problems or constraints.

The metaheuristics are mainly inspired by nature that is the oldest and wisest teacher. It always shows the ingenious solutions and pathways toward intelligence. Nowadays, nature-inspired designs and technologies employ the principles that nature discovered after billion years of evolution. Metaheuristic algorithms utilize these rules for developing smart search strategies to tackle hard optimization problems. They are classified according to the source of inspiration into different categories. The main three categories are evolutionary algorithms, swarm intelligence techniques, and physics-based methods.

Evolutionary algorithms use biological evolution mechanisms, such as reproduction, mutation, and selection. For example, Genetic Algorithm (GA) [1], Differential Evolutionary Algorithm (DEA) [2], and Biogeography Based Optimizer (BBO) [3] are among the popular evolutionary techniques. Physics-based algorithms simulate the physical process to update the solutions iteratively. For instance, Simulated Annealing (SA) algorithm is based on the annealing process of metal [4]; Thermal Exchange Optimization (TEO) is inspired by Newton's law of cooling [5]. Swarm Intelligence methods refer to algorithms that imitate the intelligence behavior of the animal's population (swarms). For example, Particle Swarm Optimization (PSO) algorithm represents the movement of fishes or birds [6]; Ant Colony Optimization (ACO) is inspired by foraging behavior of ants [7] and Water Strider Algorithm (WSA) simulates the life cycle of water strider including territorial life, ripple communication, mating behavior, feeding mechanisms and succession [8]. Although, in recent years, several metaheuristics have been established and the optimization field progressed significantly, yet some problems cannot effectively be solved by these techniques. According to No-Free-Lunch (NFL) theorem, none of these algorithms can solve all optimization problems efficiently [9]. This theorem implies the importance of new and specific algorithms in different fields. Because a single algorithm does

Cite this article as: Kaveh, A., Eslamlou, A. D., 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

not guarantee its success in different sets of applications. For this purpose, various algorithms have been developed and tested focusing on the structural optimization.

Arora et al. [10] reviewed the methods for discrete-integer-continuous variable nonlinear optimization such as Branch and bound method (BBM), Simulated Annealing (SA), and Integer Programming. They concluded that Metaheuristics fit better with engineering applications. Hasançebi et al. [11] evaluated the performance of primary metaheuristics including the Genetic Algorithm, Simulated Annealing, Evolution Strategies, Particle Swarm Optimizer, Tabu Search, Ant Colony Optimization and Harmony Search in optimum design of truss structures. Their numerical experiments indicated the strong performance of SA among others. Saka and Geem [12] carried out an extensive review of the mathematical and metaheuristic methods in steel frame structures and concluded that the metaheuristics can be considered as the standard design optimization tools. However, they stated that the researchers have yet to search for finding better stochastic techniques. Stolpe [13] did a review of the optimal design of truss structures. He similarly showed that further research is required and the solution to this optimization problem has not yet touched.

In the present paper, a newly developed population-based metaheuristic so-called Water Strider Algorithm is applied to structural design problems. This new algorithm presents an efficient balance between exploration and exploitation strategies that yields to a good accuracy as well as fast convergence. The results obtained from optimizing different structures are carefully examined and discussed. In these problems, sections with discrete and continuous areas are considered as design variables and the structural weight with displacement and stress constraints are served as the objective function.

The paper is organized in the following manner. In Section 2, the WSA algorithm and its dynamic version are comprehensively explained, and the pseudocode and flowchart are presented. In Section 3, the problem is stated, and in its subsections, the numerical results are provided and discussed. Eventually, the concluding remarks and suggestions for future research are provided in Section 4.

2 Water Strider Algorithm

Water Strider is a bug genus in the family of Gerridae. These aquatic creatures usually reside on the ponds and rivers surface and can coast across the water surface without sinking. Their interesting capabilities inspired researchers and innovators in devising several innovations such as self-cleaning surfaces and anti-dew materials. The strider's legs do more than skating on the water; they're also configured to communicate with other striders through a complex system so-called "ripples communication" that is based on vibrating water surface. Furthermore, they evolved intelligent territorial, mating, and foraging behaviors. Kaveh and Dadras Eslamlou [8] recently developed a new metaheuristic optimization algorithm inspired by water striders' social behavior patterns.

2.1 Steps of WSA

The general flow of WSA is characterized by five main steps, namely, birth, establishing territory, mating, feeding, death, and succession [8]. In the following, each of these steps is explained and their mathematical interpretation is provided. It should be noted that water striders (WSs) position on the water represent the solutions of any arbitrary optimization problem

2.1.1 Birth

WSs are born with eggs that are laid on the lake by females. For the sake of simplicity, we assume that they are randomly distributed. Therefore, the initial population of the WSs are generated by the Eq. (1)

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

where WS_i^0 determines the initial position of *i*th water strider. U_b and L_b denote the upper and lower bounds corresponding to variables' maximum and minimum allowable values, respectively; *rand* is a vector with uniform random numbers between 0 and 1; *nws* is the number of WSs.

After generating the initial WSs, they are evaluated using the objective function of the optimization problem to calculate the fitness of their position on the lake. The value of the objective function is attributed to the abundance of food in their position.

2.1.2 Establishing territory

WSs live in the territories and display territorial behavior. Each territory includes at least one mature male (keystone) and a few female (optimal foraging-habitat user) bugs. In this algorithm, to establish *nt* number of territories by total *nws* number of water striders, the WSs are sorted based on their fitness. Then, they are divided into $\frac{nws}{nt}$ groups and each territory is assembled by taking one WS from each group in a sorted manner. This process is illustrated in Fig. 1 through a suitable example of establishing 4 territories out of 12 WSs.

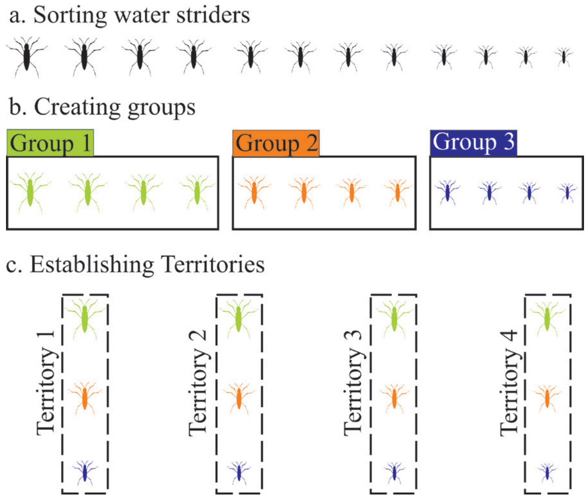


Fig. 1 An illustration of establishing territories

2.1.3 Mating

WSs spend a considerable portion of their life mating or attempting to do so. In this process, the keystone of each territory sends precopulatory courtship calling signals to a female through ripple signals. Then, the female responds to the request with either attraction or repelling signals. If she accepts to mate, they do and otherwise the male might mount her aggressively, but since the female is equipped with a hard shield, she can avoid the mating and repel him. The following equation gives an equal probability of mating and repelling and updates the position of keystone bug as Eq. (2) and Fig. 2.

$$\begin{cases} WS_i^{t+1} = WS_i^t + R.rand & \textit{if mating happens} \\ & (\textit{with probability of } p), \\ WS_i^{t+1} = WS_i^t + R.(1+rand) & \textit{otherwise} \end{cases} \quad (2)$$

where WS_i^t is the position of *i*th WS in the *i*th cycle; *rand* is a random vector between 0 and 1; *R* is a vector whose starting point is at the position of male (WS_i^{t-1}) and the endpoint is at the position of a female in the same territory (WS_F^{t-1}). This female can be selected by a fitness proportionate selection mechanism such as roulette wheel selection. The length of *R* is equal to the distance between male (WS_i^{t-1}) and female *WSs*(WS_F^{t-1}) (the radius of ripple wave) as Eq. (3) and Fig. 2(a).

$$R = WS_F^t - WS_i^t \quad (3)$$

2.1.4 Foraging

The mating process consumes a lot of energy, regardless of whether it happens successfully or not. Thus, in the new position, WSs forage for food resources to get energy. In this step, the new position is evaluated by the objective function. If the objective value is higher than the previous state, it means that it has found the food for recovery. But if the objective value is less than the former state, it should move toward the best habitat that contains the highest fitness. As illustrated in Fig. 3, Eq. (4) is formulated for the mathematical representation of moving forward to the new position around the best WS of lake (WS_{BL}^t) that has a good deal of food resources.

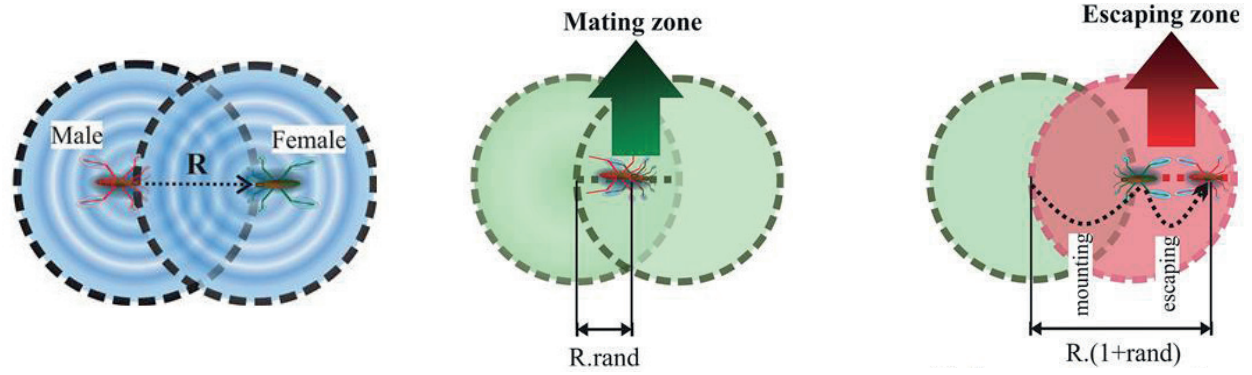
$$WS_i^{t+1} = WS_i^t + 2rand.\left(WS_{BL}^t - WS_i^t\right) \quad (4)$$

2.1.5 Death and succession

When the WS moves toward the new position, it is not guaranteed to find food. In most cases, when alien WS enters into new territory, males show aggressive territorial behavior for disposing of intruders. Aggression between residents and intruders is severe and might lead to murder. In this step, if the objective value is less than the previous position the WS will die and new WS will be replaced according to Eq. (5) and if it was otherwise, the keystone would remain alive.

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

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



(a) The ripple communications for mating (b) Successful mating and attraction (c) Unsuccessful mating and escaping

Fig. 2 The mating process of water striders and position updating based on two main scenarios

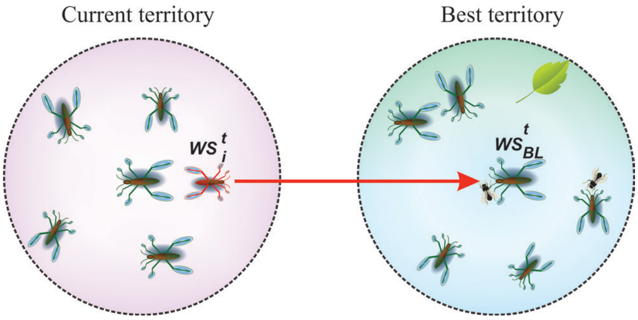


Fig. 3 Foraging for food and moving toward the best strider of lake

2.1.6 Termination condition of WSA

In the last step of the WSA, if the termination condition is met, the algorithm stops searching and reports the best position experienced so far. But if the condition is not fulfilled, it will return to the mating step for a new cycle of life. Here, the maximum number of function evaluations (*MaxNFE*) is considered as the termination condition in all problems.

2.2 Dynamic Water Strider Algorithm

In order to improve the performance of WSA, two dynamic features are successfully implemented in the basic version. In the dynamic version (DWSA), the number of territories (*nt*) and approaching distances change during the optimization process to regulate the exploration and exploitation behavior. The following sections detail the mentioned dynamic features.

2.2.1 Dynamic number of territories

At the beginning of the search, since the promising regions are not yet discovered, the population should follow several local optimums and improve their fitness. In this stage, it is reasonable to have several territories, while in the last stages, populations have probed the space and have found the most promising regions. Therefore, it is better to concentrate on a limited number of territories. In the DWSA, the number of territories starts with a maximum number and as the iterations increase, the number of territories tends toward the smallest prime divisor of the initial number. For example, if the maximum number of populations is considered as 200, the algorithm starts with 100 territories and during the optimization gradually reduces them to 50, 40, 25, 20, 10, 5, 4, and finally 2 territories. Fig. 4 illustrates the decrease in the number of territories for this example. It should be noted that the total number of populations is fixed throughout the search process.

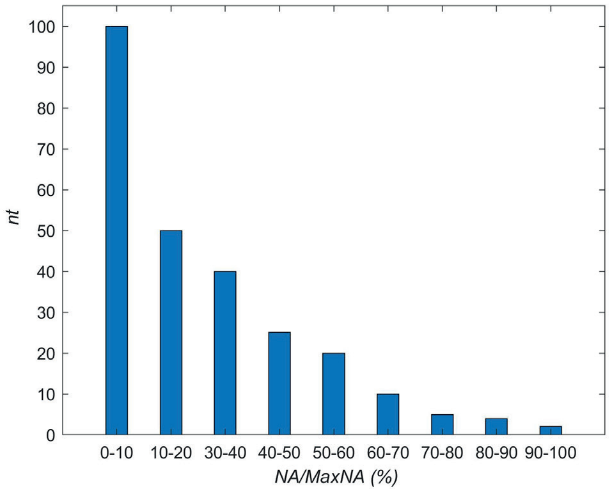


Fig. 4 Illustration of the number of territories () against the percentage of performed analyses

2.2.2 Dynamic approaching distance

For the same reason that as the algorithm draws to a close, the algorithm should intensify exploitations rather than diversification. In the final stages, the WSs should move closer to the local or global best member than in the initial steps. In the proposed dynamic version, the approaching distance for mating and foraging are redefined as Eq. (6) and Eq. (7) that are respectively used instead of Eq. (2) and Eq. (4) in the basic version.

$$\begin{cases} WS_i^{t+1} = WS_F^t + \left(1 - \dfrac{NA}{MaxNA}\right)^2 R.rand & \begin{array}{l} \textit{if mating happens} \\ (\textit{with probability of } p) \end{array} \\ WS_i^{t+1} = WS_F^t + \left(1 - \dfrac{NA}{MaxNA}\right)^2 R.(1 + rand) & \textit{otherwise} \end{cases}, \quad (6)$$

$$WS_i^{t+1} = WS_{BL}^t + \left(1 - \frac{NA}{MaxNA}\right)^2 (-1 + 2rand).\left(WS_{BL}^t - WS_i^t\right), \quad (7)$$

where, *NA* and *MaxNA* denote the current and maximum number of analyses, respectively.

As seen, the distance between best and keystone is decreased continuously in a parabolic manner from the former default value at the beginning to near-zero distance at the end of the algorithm.

The pseudo-code of WSA is provided in Algorithm 1 and its flowchart is illustrated in Fig. 5.

3 Numerical experiments

In this section, several well-known structural optimum design problems are investigated to demonstrate the performance of WSA. For the sake of comparison, the results of some other optimization algorithms are also reported.

Algorithm 1 Pseudo-code of WSA optimizer

```
Inputs: The population size nws, number of territories nt and the maximum number of analyses MaxNA
Outputs: The richest location of WS and its objective value
Create the initial population randomly, as Eq. (1)
Calculate the object value of WSs
while (terminating condition is not satisfied) do
    Establish nt number of territories and allocate the WSs according to Sections 2.1.2 or 2.2.1
    for (each of territories) do
    The male sends mating signals and the selected female responds which can be an attractive or repulsive signal.
    Update the position of keystone male based on the response of female and Eq. (2) or Eq. (6).
    Evaluate the objective value to find food for compensating the consumed energy
        If (keystone cannot find food) then
        Forage for food resource and approach the best territory by Eq. (4) or Eq. (7).
            if (keystone cannot find food again) then
                The hungry keystone will be killed by resident keystone of the new territory or be died because of starvation
                A larva will replace the killed keystone as the successor defined by Eq. (5).
            end
        end
    end
Return $WS_{optimal}$
```

The goal of optimizing these structures is to obtain minimum weight while satisfying their certain constraints. The mathematical expression for structural optimization can be expressed as follows:

$$\begin{aligned}
&\min imize\ W(\{x\}) = \sum_{i-1}^{n} \gamma_i . A_i . L_i(x) \\
&subjected\ to\ : \delta_{\min} \le \delta_i \le \delta_{\max},\quad i = 1,2,\ldots,m \\
&\sigma_{\min} \le \sigma_i \le \sigma_{\max},\quad i = 1,2,\ldots,n \\
&\sigma_i^{\ b} \le \sigma_i \le 0 \quad i = 1,2,\ldots,ns \\
&A_{\min} \le A_i \le A_{\max},\quad i = 1,2,\ldots,ng
\end{aligned} \tag{8}$$

where $\{x\}$ is a vector that contains design variables, $W\{x\}$ denotes the weight of the structure; n is the number of structural members; γ_i and L_i denote the density and length of the ith member, respectively. A_i is the cross-section area of ith truss member. δ_i is the nodal displacement and σ_i is the element stress.

To handle the design constraints, the penalty approach is utilized and a penalty term is added to the objective function as Eq. (9):

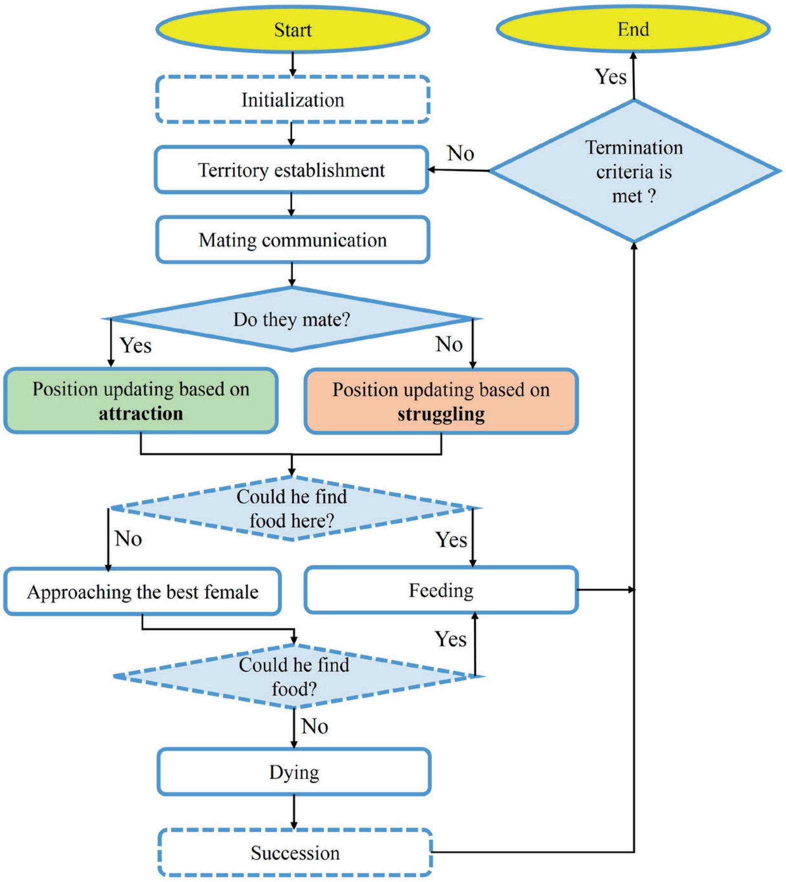


Fig. 5 Flowchart of the WSA

$$f\left(X\right)=\left(1+\varepsilon_1.\nu\right)^{\varepsilon_2}\times w\left(\left\{x\right\}\right),\ \nu=\sum_{i=1}^{nc}\max\left[0,g_j\left(\left\{X\right\}\right)\right], \quad (9)$$

where, ν is the sum of the violation ratios, the constant ε_1 is set equal to 1, and ε_2 starts at 1.5 and linearly increases to 3 at the end of the process. At the final stage, to ensure that the optimum design satisfies the constraints, the solutions are rechecked and the designs with violations are excluded.

In the following subsections the 25- and 72-bar spatial trusses as well as 15- and 24-story frames are examined. The trusses have continuous sections and the frames have discrete variables selected from standard W-sections. The population is considered as 50 and the number of territories is fixed as 2 in all problems as utilized in the source paper [8]. The computer codes have been prepared in MATLAB and all the runs for the truss problems have been implemented on a MacBook Pro with CPU 3.3 GHZ (an Intel Core i9 computer platform), Ram 16 GB and MATLAB 2020 running on a computer with Macintosh (macOS Catalina).

3.1 25-bar spatial transmission tower

The spatial 25-bar truss problem is a popular testing model that has been optimized by many algorithms. This space truss is shown in Fig. 6. The structural elements are divided into 8 groups, each of which has the same material and section. Table 1 represents element groups by members and their beginning and ending nodes. Table 2 shows two load cases that are exerted to the truss. The axial stress limitations differ for element groups as shown in Table 3, but all nodes have the maximum displacement limit of ±0.35 *in* each direction. The cross-section area size is considered between 0.01 to 3.4 in^2. The modulus of elasticity is 10000 *ksi* and the density of construction material is 0.1 lb/in^3.

Table 1 Element groups of 25- bar spatial truss

Element group number							
1	2	3	4	5	6	7	8
1:(1.2)	2:(1.4)	6:(2.4)	10:(6.3)	12:(3.4)	14:(3.10)	18:(4.7)	22:(10.6)
	3:(2.3)	7:(2.5)	11:(5.4)	13:(6.5)	15:(6.7)	19:(3.8)	23:(3.7)
	4:(1.5)	8:(1.3)			16:(4.9)	20:(5.10)	24:(4.8)
	5:(2.6)	9:(1.6)			17:(5.8)	21:(6.9)	25:(5.9)

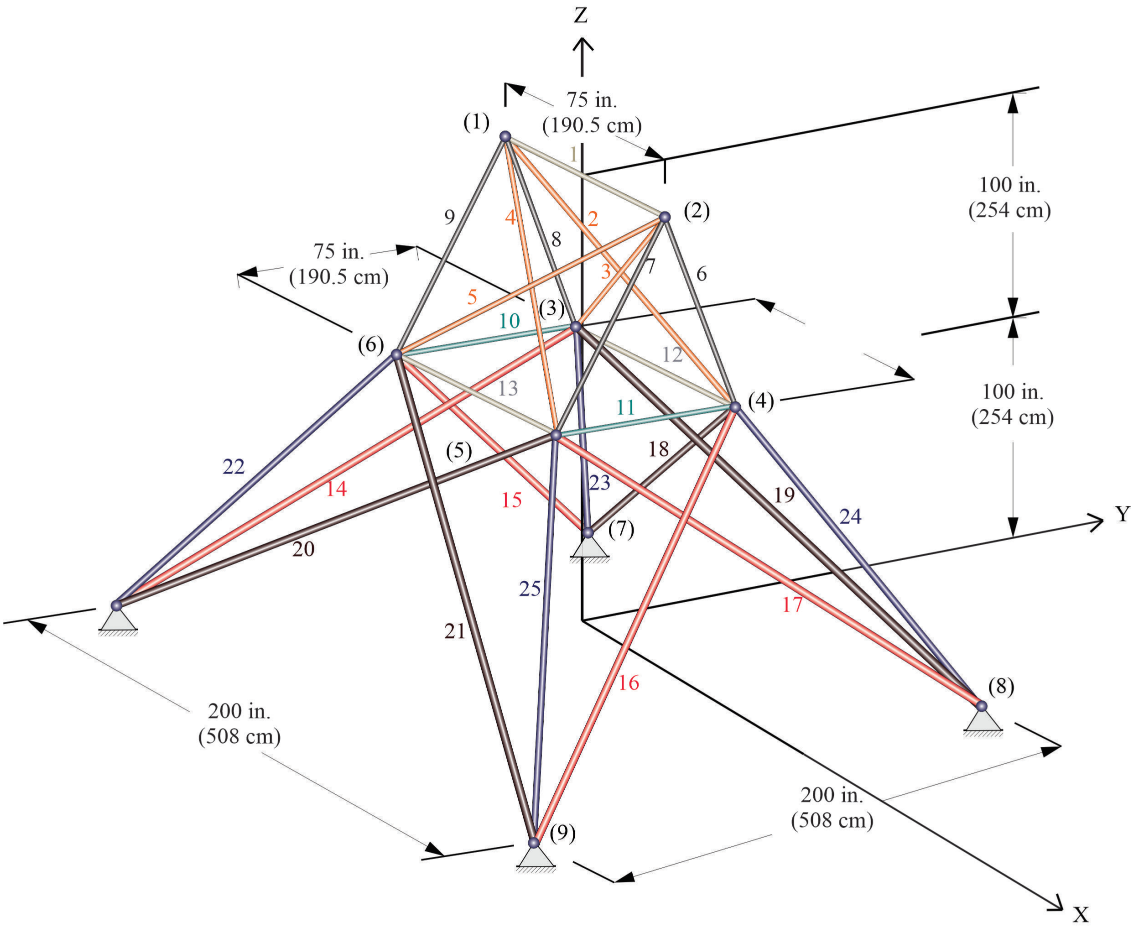


Fig. 6 Schematic of the 25-bar transmission tower

The optimization results are provided in Table 4. In addition to WSA and DWSA, the results of the CSS [14], TLBO [15], CBO [16] and BB-BC [17] algorithms are considered for comparison. As seen, after the CBO algorithm, DWSA and WSA stand in the second and third rank, respectively. But it should be noted that WSAs obtain these designs with a lower number of analyses. DWSA obtained the lowest average weight among all algorithms. The standard deviations of the results of WSA and DWSA are less than 0.1 that shows most of their designs are close to the average weight and since the average values are reasonably low, thus these algorithms in most cases reach acceptably light designs.

The convergence curve shown in Fig. 7 not only confirms the previous statement but also shows that the DWSA finds an acceptable design in almost 2000 analyses.

3.2 A 72-bar spatial truss

As shown in Fig. 8, the second problem is a spatial truss with 72-bar. The modulus of elasticity is 10000 *ksi* and the material density is assumed as 0.1 lb/in^3. The structural bar members are divided into 16 groups and their maximum allowable tension and compression stresses are equal to 25 *ksi*. The displacement of the uppermost nodes in both *x* and *y* directions shall be less than 0.25. The maximum and minimum permitted cross-sectional area is 4.00 in^2 and 0.10 in^2. This spatial truss is under two load cases that are represented in Table 5.

Table 2 Load cases of 25- bar spatial truss

Node number	Load (kips)					
	Case 1			Case 2		
	P_x	P_y	P_z	P_x	P_y	P_z
1	0	20	–5	1	10	–5
2	0	–20	–5	0	10	–5
3	0	0	0	0.5	0	0
6	0	0	0	0.5	0	0

Table 3 Allowable stress of members for 25-bar truss

Element group		Compressive stress Limitations *ksi* (MPa)	Tensile stress Limitations *ksi* (MPa)
1	A_1	35.092 (241.96)	40.0 (275.80)
2	$A_2 \sim A_5$	11.590 (79.0913)	40.0 (275.80)
3	$A_6 \sim A_9$	17.305 (119.31)	40.0 (275.80)
4	$A_{10} \sim A_{11}$	35.092 (241.96)	40.0 (275.80)
5	$A_{12} \sim A_{13}$	35.092 (241.96)	40.0 (275.80)
6	$A_{14} \sim A_{17}$	6.759 (46.603)	40.0 (275.80)
7	$A_{18} \sim A_{21}$	6.959 (47.982)	40.0 (275.80)
8	$A_{22} \sim A_{25}$	11.082 (76.410)	40.0 (275.80)

Table 4 Optimized design for the 25-bar truss

	BB-BC [17]	CBO [16]	TLBO [15]	CSS [14]	WSA	DWSA
1 (A1)	0.010	0.010	0.010	0.010	0.010	0.012
2 (A2 ~ A5)	2.092	2.129	2.071	2.003	2.008	1.996
3 (A6 ~ A9)	2.964	2.886	2.957	3.007	2.962	2.979
4 (A10 ~ A11)	0.010	0.010	0.010	0.010	0.010	0.010
5 (A12 ~ A13)	0.010	0.010	0.010	0.010	0.010	0.010
6 (A14 ~ A17)	0.689	0.679	0.689	0.687	0.683	0.684
7 (A18 ~ A21)	1.601	1.607	1.620	1.655	1.674	1.674
8 (A22 ~ A25)	2.686	2.692	2.676	2.660	2.666	2.661
Best weight (lb.)	545.38	544.31	545.09	545.10	544.84	544.75
Average weight (lb.)	545.78	545.25	545.41	545.58	545.25	545.02
Standard deviation	0.49	0.29	0.42	0.41	0.085	0.046
Number of analyses	10000	9090	15318	7000	6000	6000

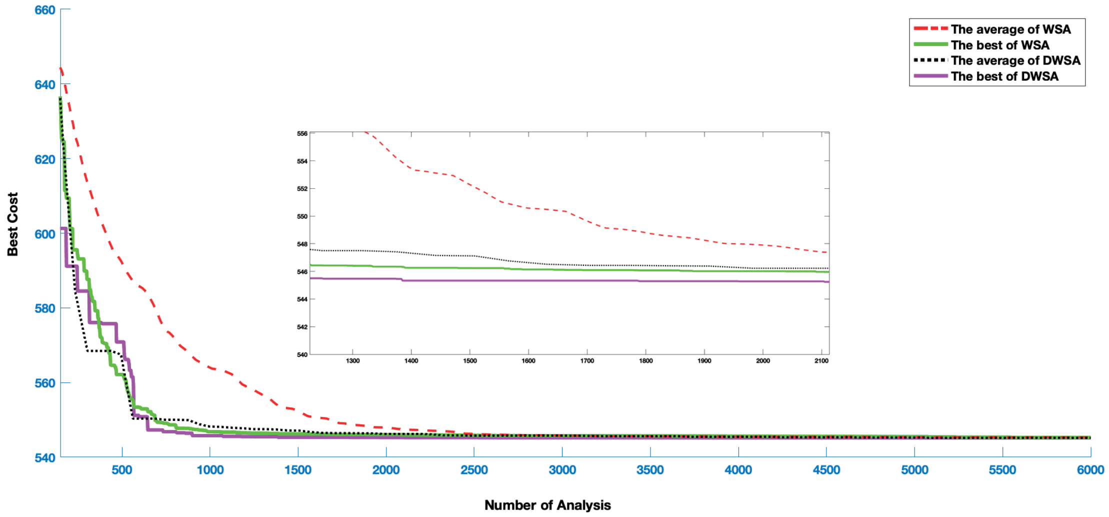


Fig. 7 Convergence curve of the average and best results for 25-bar truss problem

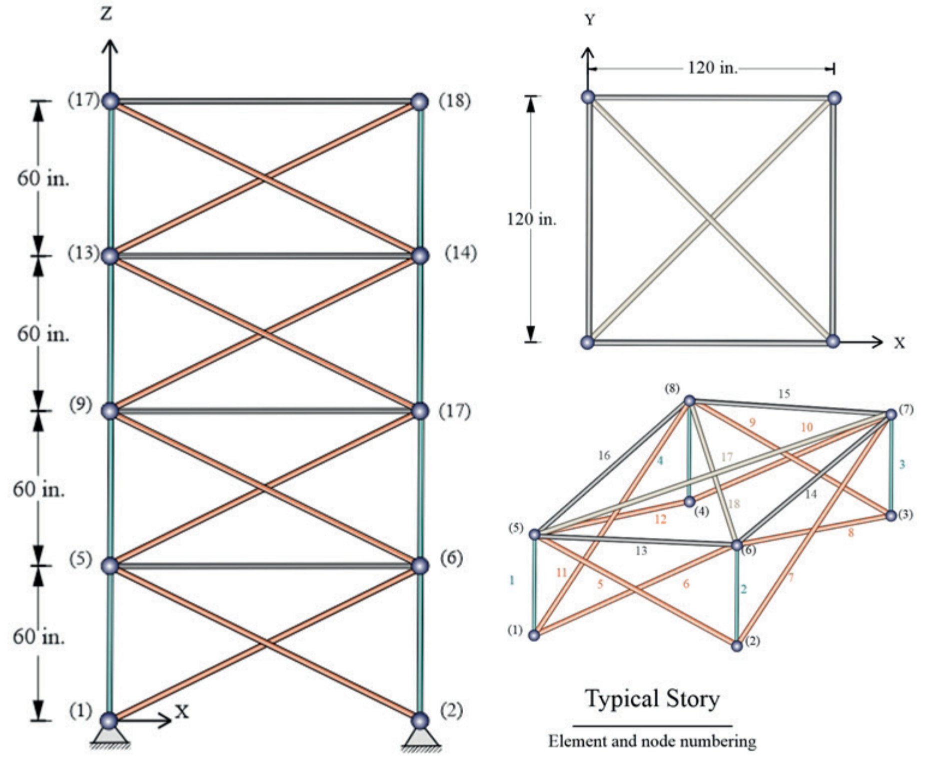


Fig. 8 Schematic of 72-bar truss

Table 5 Load cases of 72- bar spatial truss

Node number	Load (kips)					
	Case1			Case 2		
	P_x	P_y	P_z	P_x	P_y	P_z
17	5	5	–5	0	0	–5
18	0	0	0	0	0	–5
19	0	0	0	0	0	–5
20	0	0	0	0	0	–5

Table 6 Optimized designs for the 72-bar truss problem

Member group	ACO [19]	CS [18]	TLBO [15]	CBO [16]	WSA	DWSA
1 (A1-A4)	1.948	1.912	1.906	1.917	1.8821	1.8796
2 (A5-A12)	0.508	0.510	0.506	0.503	0.5106	0.5147
3 (A13-A16)	0.101	0.100	0.100	0.100	0.1000	0.1000
4 (A17-A18)	0.102	0.100	0.100	0.100	0.1000	0.1000
5 (A19-A22)	1.303	1.257	1.261	1.272	1.2701	1.2645
6 (A23-A30)	0.511	0.512	0.511	0.505	0.5094	0.5084
7 (A31-A34)	0.101	0.100	0.100	0.100	0.1000	0.1000
8 (A35-A36)	0.100	0.100	0.100	0.100	0.1000	0.1000
9 (A37-A40)	0.561	0.522	0.531	0.518	0.5270	0.5175
10 (A41-A48)	0.492	0.517	0.515	0.536	0.5177	0.5186
11 (A49-A52)	0.100	0.100	0.100	0.100	0.1000	0.1000
12 (A53-A54)	0.107	0.100	0.100	0.100	0.1000	0.1000
13 (A55-A58)	0.156	0.156	0.156	0.156	0.1563	0.1564
14 (A59-A66)	0.550	0.540	0.549	0.537	0.5480	0.5497
15 (A67-A70)	0.390	0.415	0.409	0.406	0.4118	0.4072
16 (A71-A72)	0.592	0.570	0.569	0.574	0.5694	0.5699
Best weight (lb.)	380.24	379.63	379.69	379.75	379.62	379.57
Average weight (lb.)	383.16	379.73	380.86	380.03	379.66	379.60
Standard deviation	3.66	N/A	1.85	0.37	0.23	0.14
Number of analyses	18500	10600	18460	18000	10000	10000

The optimized designs of this example are provided in Table 6. The results of TLBO [15], CBO [16], CS [18] and ACO [19] algorithms are considered for comparison. As seen, DWSA obtained the best design among algorithms in terms of best weight, average weight, and standard deviation in a less or equal number of analyses. According to small standard deviations, like the previous example, both WSA and DWSA exhibit a reliable performance. The convergence curves of the dynamic and basic versions of the presented algorithms are shown in Fig. 9 that demonstrates the fast convergence rate of both the best and average results of DWSA in comparison with WSA.

3.3 The 3-bay 15-story frame example

This example is a steel frame with 105 members divided into 10 groups. The configuration and loads of this 15-story frame are plotted in Fig. 10. The modulus of elasticity is 200 GPa,

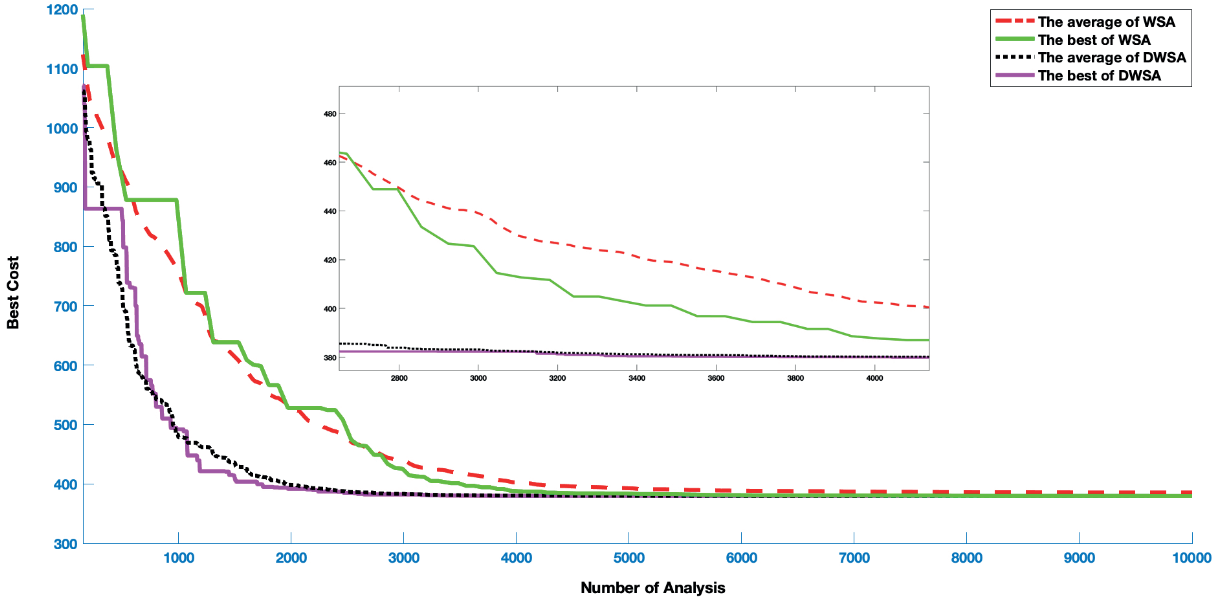


Fig. 9 Convergence curve of the average and best results for 72-bar truss example

the yield stress is 248.2 MPa and the unit weight of its constructional steel is 7.85 Ton/m^3. 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 assumed as $K_y = 1$. For columns, the members are considered non-braced along their length, and for beams, the unbraced length is defined as one-fifth of the span length.

In Table 7, the optimum designs obtained by DWSA, WSA, CSS [14], DE [15], CBO [16] and ICA [20] are compared. As can be seen, DWSA has obtained the lightest design, and CSS reached a design with 0.0 8% lighter design than WSA. WSA and DWSA respectively converged to an average 401.67 kN and 393.67 kN weight with 3.67 % and 2.48 % coefficients of variation. The average and best convergence histories of WSA and DWSA are plotted in Fig. 11. As seen, DWSA has a higher convergence rate than WSA. The convergence curves of best designs have sharp edges that are mainly because of the discrete variables of this problem.

3.4 The 3-bay 24-story frame

This example is a tall frame structure that consists of 100 joints and 168 members. As can be seen in Fig. 12, the gravity loads and lateral loads are applied to this structure. All members are constructed of steel sections and the yield stress, material density, and modulus of elasticity are considered as 230.3 MPa, 0.283 lb/in^3 (7933.41 kg/m^3) and 30 Msi (205 GPa), respectively. Like the previous example, 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 $K_x = 1$. The columns

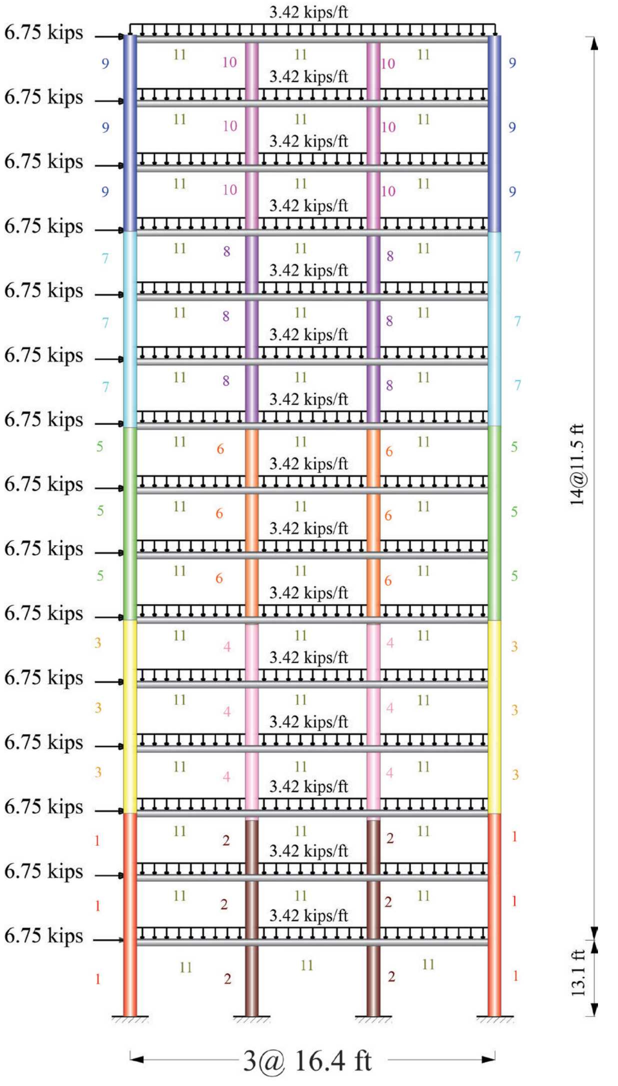


Fig. 10 Schematic of 3-bay 15-story frame

Table 7 Optimized designs for the 15-story frame

	ICA [20]	DE [15]	CSS [14]	CBO [16]	WSA	DWSA
1	W24 × 117	W21 × 44	W12 × 87	W14 × 90	W14 × 109	W24 × 104
2	W21 × 147	W12 × 106	W36 × 182	W26 × 146	W26 × 146	W26 × 146
3	W27 × 84	W27 × 161	W21 × 93	W18 × 76	W14 × 82	W18 × 76
4	W27 × 114	W27 × 84	W18 × 106	W24 × 104	W24 × 104	W24 × 104
5	W14 × 74	W27 × 114	W18 × 65	W12 × 72	W18 × 76	W14 × 74
6	W18 × 86	W16 × 67	W14 × 90	W18 × 86	W12 × 87	W18 × 86
7	W12 × 96	W18 × 86	W10 × 45	W12 × 58	W18 × 65	W18 × 65
8	W24 × 68	W24 × 55	W12 × 65	W14 × 61	W18 × 65	W18 × 65
9	W10 × 39	W16 × 67	W6 × 25	W6 × 25	W10 × 33	W8 × 24
10	W12 × 40	W8 × 24	W10 × 45	W16 × 36	W18 × 40	W16 × 36
11	W21 × 44	W16 × 45	W21 × 44	W21 × 44	W21 × 44	W21 × 44
Weight (kN)	417.46	412.62	395.34	416.96	395.67	387.55

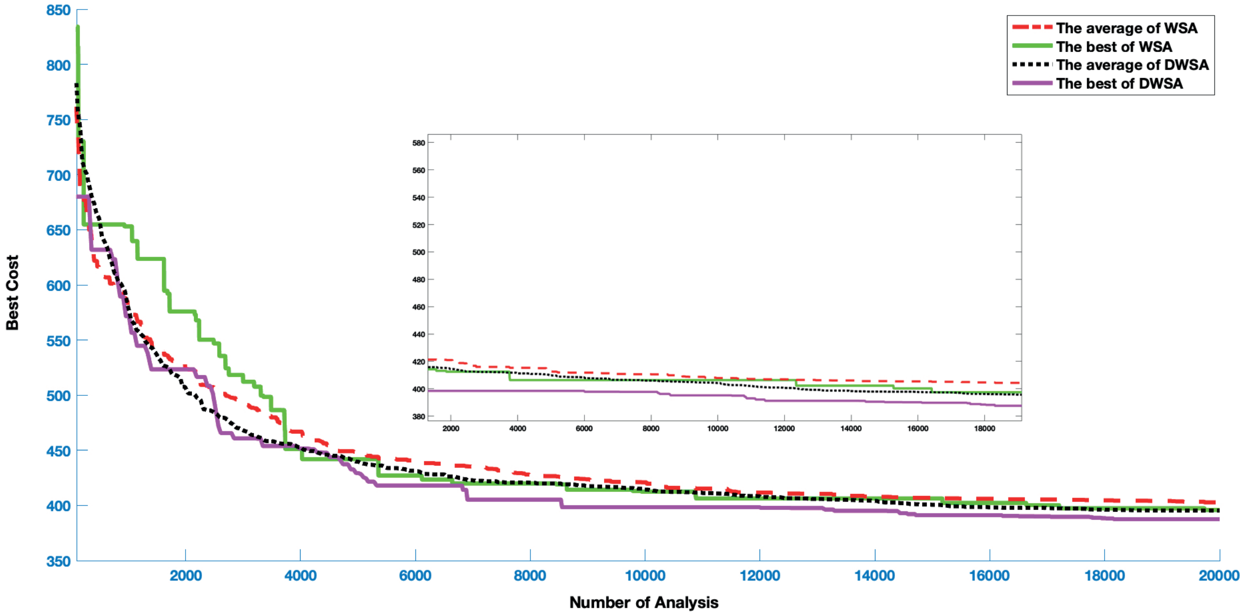


Fig. 11 Convergence curve of the average and best results for 15-story frame

are considered as non-braced along their length, and the unbraced length for beam members are considered as one-fifth of the span length. To impose fabrication conditions, the beams of the first and third bay except the roof are categorized in one group, which results in four beam groups. The exterior columns are categorized into one group and the interior columns are considered together in another group that changes in every three stories. The grouping finally results in 16 column groups chosen from 267 W-shape sections and 4 beam groups chosen from 37 W14 sections.

The optimum sections obtained by DWSA, WSA, CSS [14], DE [15], CBO [16] and ICA [20] are provided in Table 8. As can be seen, both versions obtained a competitive performance for this frame example. DWSA obtained the minimum weight among the compared algorithms. From the convergence curves shown in Fig. 13, it is observed that after 20000 number of structural analyses, both WSA and DWSA averagely reach designs with less than 1000 kN weight. In this example, the mean weights obtained by DWSA and WSA are 933.86 and 934.18 kN, respectively. Moreover, DWSA with 3.1 % coefficient of variation has low variability than WSA with 3.66 % coefficient of variation.

4 Conclusions

The main objective of the present study is to investigate the performance of WSA metaheuristic and its dynamic version for continuous and discrete design optimization

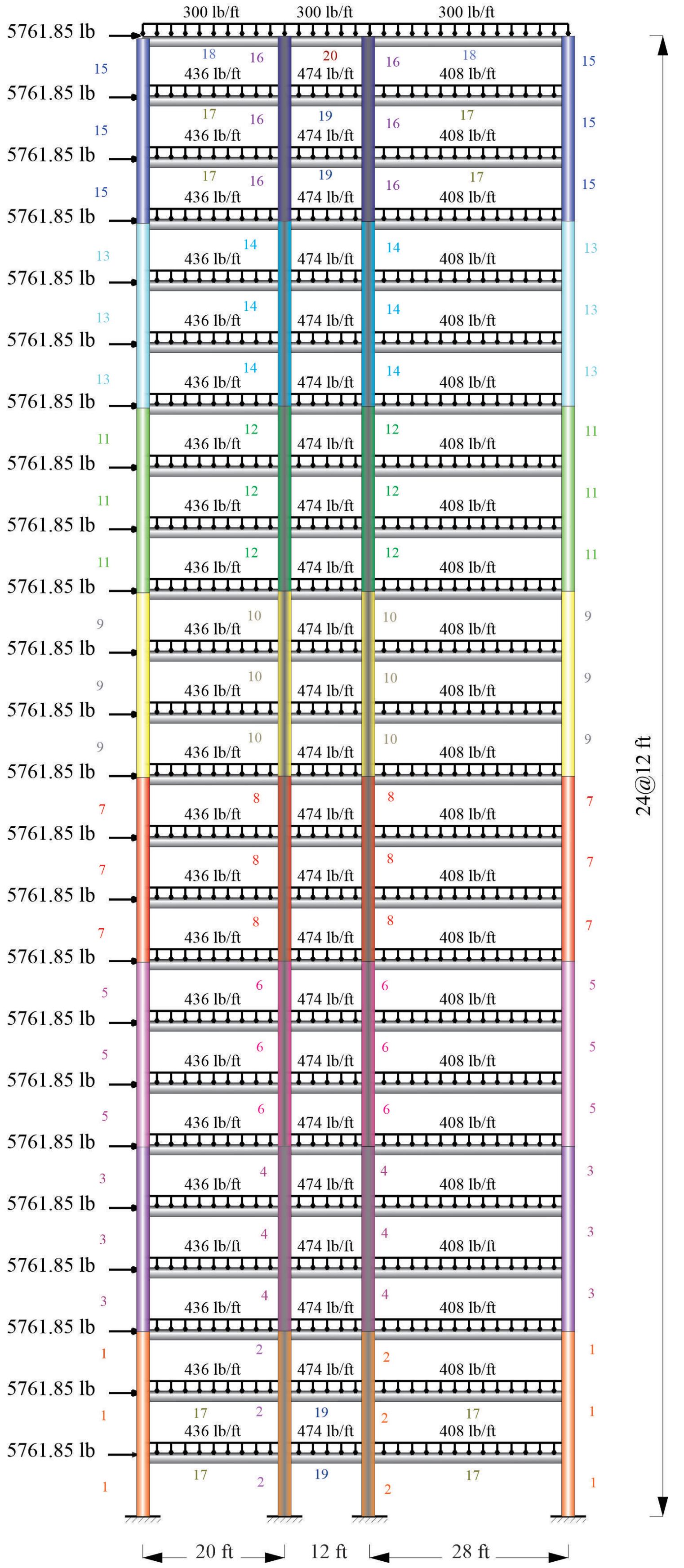


Fig. 12 Schematic of 3-bay 24-story frame

Table 8 Optimized design for the 24-story frame

	ICA [20]	DE [15]	CSS [14]	CBO [16]	WSA	DWSA
1	W30 × 90	W30 × 90	W30 × 90	W27 × 102	W30 × 90	W30 × 90
2	W21 × 50	W21 × 50	W6 × 20	W8 × 18	W21 × 50	W21 × 50
3	W24 × 55	W21 × 48	W21 × 44	W24 × 55	W21 × 48	W21 × 44
4	W8 × 24	W12 × 19	W6 × 9	W6 × 8.5	W8 × 24	W21 × 48
5	W14 × 109	W14 × 176	W14 × 159	W14 × 132	W14 × 159	W14 × 176
6	W14 × 159	W14 × 145	W14 × 145	W14 × 120	W14 × 120	W14 × 132
7	W14 × 120	W14 × 109	W14 × 132	W14 × 145	W14 × 120	W14 × 109
8	W14 × 90	W14 × 90	W14 × 99	W14 × 82	W14 × 109	W14 × 82
9	W14 × 74	W14 × 74	W14 × 68	W14 × 61	W14 × 82	W14 × 61
10	W14 × 68	W14 × 61	W14 × 61	W14 × 43	W14 × 53	W14 × 38
11	W14 × 30	W14 × 34	W14 × 43	W14 × 38	W14 × 30	W14 × 34
12	W14 × 38	W14 × 34	W14 × 22	W14 × 22	W14 × 30	W14 × 34
13	W14 × 159	W14 × 145	W14 × 109	W14 × 99	W14 × 99	W14 × 90
14	W14 × 132	W14 × 132	W14 × 109	W14 × 109	W14 × 109	W14 × 109
15	W14 × 99	W14 × 109	W14 × 90	W14 × 82	W14 × 109	W14 × 99
16	W14 × 82	W14 × 82	W14 × 82	W14 × 90	W14 × 74	W14 × 90
17	W14 × 68	W14 × 68	W14 × 74	W14 × 74	W14 × 61	W14 × 74
18	W14 × 48	W14 × 43	W14 × 43	W14 × 61	W14 × 48	W14 × 61
19	W14 × 34	W14 × 34	W14 × 30	W14 × 30	W14 × 38	W14 × 34
20	W14 × 22	W14 × 22	W14 × 26	W14 × 22	W14 × 22	W14 × 22
Weight (kN)	946.25	945.02	912.25	960.25	913.03	905.760

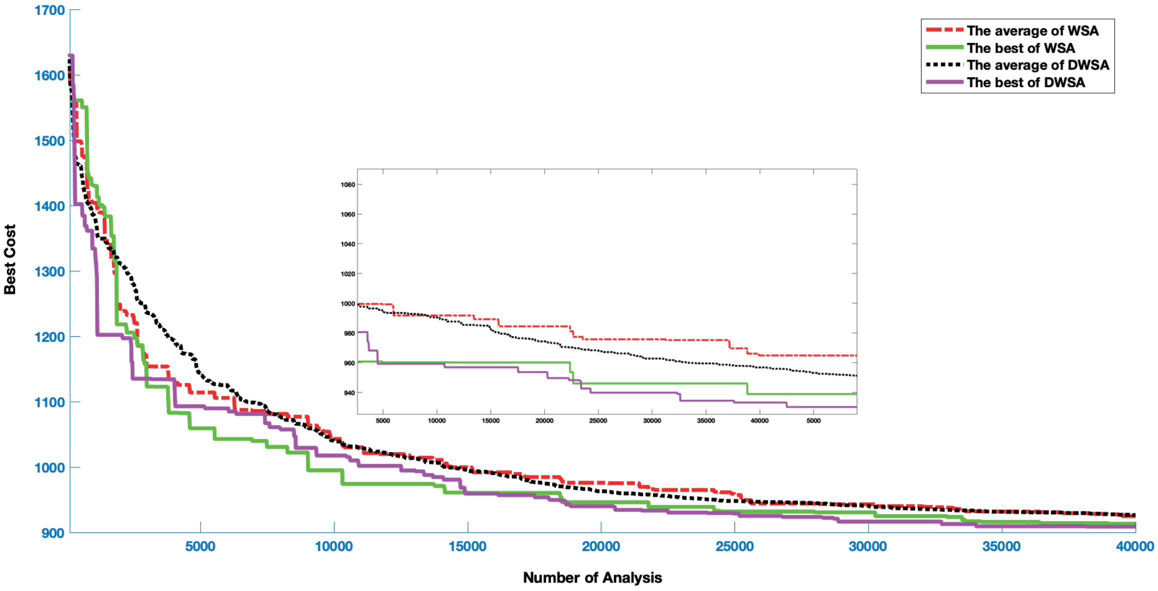


Fig. 13 Convergence curve of the average and best results for 24-story frame

of trusses and steel moment frames taken from [21]. Two benchmark truss examples and two steel frames are examined, and the optimization results are provided. The obtained designs indicate the efficiency and competitiveness of the proposed algorithms compared to other algorithms in terms of the best and average optimal weight. According to the results, the DWSA with obtaining small variability for different trials shows constantly reliable performance that is not very sensitive to the initial search point. In all examples, the proposed dynamic version outperformed the basic WSA. Furthermore, DWSA has a fast convergence rate than WSA by which in a low number of iterations can reach acceptably light designs. In future research, the authors are going to hybridize the algorithm with other state-of-the-art algorithms for enhancing the performance of WSA.

References

[1] Goldberg, D. E. "Genetic Algorithms", Pearson India Education Services, Chennai, India, 2006.

[2] Storn, R., Price, K. "Differential Evolution -A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization, 11, pp. 341–359, 1997. https://doi.org/10.1023/A:1008202821328

[3] Simon, D. "Biogeography-Based Optimization", IEEE Transactions on Evolutionary Computation, 12(6), pp. 702–713, 2008. https://doi.org/10.1109/TEVC.2008.919004

[4] Kirkpatrick, S., Gelatt, 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

[5] Kaveh, A., Dadras, A. "A novel meta-heuristic optimization algorithm: Thermal exchange optimization", Advances in Engineering Software, 110, pp. 69–84, 2017. https://doi.org/10.1016/j.advengsoft.2017.03.014

[6] Eberhart, R. C., Yuhui, S. "Particle swarm optimization: developments, applications and resources", In: Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, South Korea, 2001, pp. 81–86. https://doi.org/10.1109/CEC.2001.934374

[7] Dorigo, M., Birattari, M., Stutzle, T. "Ant colony optimization", IEEE Computational Intelligence Magazine, 1(4), pp. 28–39, 2006. https://doi.org/10.1109/MCI.2006.329691

[8] 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

[9] Wolpert, D. H., Macready, W. G. "No Free Lunch Theorems for Search", Santa Fe Institute, Santa Fe, NM, USA, Rep. 95-02-010, 1995. [online] Available at: https://www.santafe.edu/research/results/working-papers/no-free-lunch-theorems-for-search

[10] Arora, J. S., Huang, M. W., Hsieh, C. C. "Methods for optimization of nonlinear problems with discrete variables: A review", Structural Optimization, 8, pp. 69–85, 1994. https://doi.org/10.1007/BF01743302

[11] 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 & Structures, 87(5), pp. 284–302, 2009. https://doi.org/10.1016/j.compstruc.2009.01.002

[12] Saka, M. P., Geem, Z. W. "Mathematical and Metaheuristic Applications in Design Optimization of Steel Frame Structures: An Extensive Review", Computational Intelligence in Civil and Hydraulic Engineering, 2013, Article ID:271031, 2013. https://doi.org/10.1155/2013/271031

[13] Stolpe, M. "Truss optimization with discrete design variables: a critical review", Structural and Multidisciplinary Optimization, 53, pp. 349–374, 2016. https://doi.org/10.1007/s00158-015-1333-x

[14] Kaveh, A., Talatahari, S. "Optimal design of skeletal structures via the charged system search algorithm", Structural and Multidisciplinary Optimization, 41, pp. 893–911, 2010. https://doi.org/10.1007/s00158-009-0462-5

[15] Degertekin, S. O., Hayalioglu, M. S. "Sizing truss structures using teaching-learning-based optimization", Computers & Structures, 119, pp. 177–188, 2013. https://doi.org/10.1016/j.compstruc.2012.12.011

[16] Kaveh, A., Mahdavi, V. R. "Colliding Bodies Optimization: Extensions and Applications", Springer, Cham, Switzerland, 2015. https://doi.org/10.1007/978-3-319-19659-6

[17] Camp, C. V. "Design of Space Trusses Using Big Bang-Big Crunch Optimization", Journal of Structural Engineering, 133(7), pp. 999–1008, 2007. https://doi.org/10.1061/(ASCE)0733-9445(2007)133:7(999)

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

[19] Camp, C. V., Bichon, B. J. "Design of Space Trusses Using Ant Colony Optimization", Journal of Structural Engineering, 130(5), pp. 741–751, 2004. https://doi.org/10.1061/(ASCE)0733-9445(2004)130:5(741)

[20] Kaveh, A., Talatahari, S. "Imperialist competitive algorithm for engineering design problems", Asian Journal of Civil Engineering, 11(6), pp. 675–697, 2010.

[21] 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