



Water strider algorithm: A new metaheuristic and applications

A. Kaveh*, A. Dadras Eslamlou

School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran, P.O. Box 16846, 13114, Iran

ARTICLE INFO

Keywords:

Optimization
Metaheuristic algorithm
Water strider algorithm
Mathematical functions
Engineering problems

ABSTRACT

The present paper proposes a novel nature-inspired optimization paradigm, which is called the Water Strider Algorithm (WSA). The WSA is a population-based optimizer inspired by the life cycle of water strider bugs. The WSA mimics territorial behavior, intelligent ripple communication, mating style, feeding mechanisms and succession of water striders. The algorithm is mathematically formulated and benchmarked on forty-four numerical problems, four classic examples, two large scale structural size optimization and one structural damage identifying applications. In two phases, the algorithm is investigated to prove its promising performance against possible biases. Different parametric and nonparametric tests, as well as comparison with well-established and state of the art algorithms, indicate the efficient performance of the proposed WSA. The algorithm is applied to classical constrained, unconstrained, continuous and discrete structural design problems confirming its capability of tackling various challenging problems. The paper also presents a real application of the proposed method in structural health monitoring, in which the WSA effectively identified the damage scenarios.

1. Introduction

Metaheuristic methods normally are nature-inspired techniques for stochastic global optimization [1]. Within an affordable computational time, they can find optimal or near-optimal solutions to the tough and even NP-hard problems. Unlike mathematical methods, they are much flexible and simple, which popularized them among both researchers and practitioners [2]. They generally mimic a simple or complicated strategy to search the space of solutions without involving the time-consuming derivative information. The mentioned advantages encouraged engineers to apply metaheuristics to different hard optimization problems such as the optimum design of structures, optimum control, etc. [3–6]. As aforementioned, these techniques rely on the rules observed in nature, such as the evolutionary process, physical phenomenon, or animals' behavior.

Evolutionary Algorithms (EAs) employ the concepts inspired by biological evolution, such as crossover, mutation, and selection. Genetic Algorithm (GA) is a well-known EA that represents the solutions as chromosomes consisting of different genes [7]. It attempts to achieve the fittest chromosome by performing reiterated reproduction, mutation, and selection mechanisms.

Animals' behavior is another source of inspiration for searching in swarm intelligent algorithms. Their hunting, flocking, migrating and foraging strategies are studied and imitated as intelligent rules to develop various efficient algorithms. For instance, the well-established

Particle Swarm Optimization (PSO) inspired by the social behavior of bird flocking or fish schooling [8]. In this algorithm, a population of search agents shares their information for finding the optimal solution. Salp Swarm Algorithm (SSA) mimicking the swarm behavior of leader and followers groups in salp chains [9]. The political behavior of humans for extending the rule or authority constitutes the framework of the Imperialist Competitive Algorithm (ICA) [10].

Insects display complex and intelligent behaviors such as smart navigations, communication, displaying courtship, finding food, avoiding predators and care for their young. The biologists studied their complex behavior for centuries. The small bodies of insects helped researchers to perform many experiments and observations associated with their instincts. Their intelligent behaviors despite their tiny brain size imply this fact that their intelligence stems from the strategies which evolution equipped them with. This hypothesis has attracted interests in the metaheuristic's community because in this field we generally seek for smart strategies rather than depending on the processing power.

For example, the Ant Colony Optimization (ACO) algorithm is developed according to the pheromone-based communication and food-finding strategies of biological ants in nature [11]. Firefly Algorithm (FA) is established based on the luminary flashing activities of fireflies for attracting the partners, communication, and risk warning [12]. Artificial Bee Colony (ABC) algorithm looks for the optimum solution imitating to the foraging behavior of honey bee colonies [13]. Social

* Corresponding author.

E-mail address: alikaveh@iust.ac.ir (A. Kaveh).

Spider Optimization (SSO) assumes the search space as a communal spider web and models the cooperative behavior of the social spider [14]. The Grasshopper Optimization algorithm is inspired by social interactions of grasshopper swarms [15] and The Fruit Fly Optimization (FFO) algorithm is inspired by the food-finding behavior of the fruit flies [16]. Moth Flame Optimization (MFO) simulates the transverse orientation and navigation method of moth flies [17]. Dragonfly Algorithm (DA) is established based on the intelligent static and dynamic swarming behavior of animals [18].

Physics-based algorithms usually simulate physical laws of nature, such as gravity, annealing, collision, thermal exchange, etc. [19]. For instance, Multi-Verse Optimizer (MVO), originating from concepts in cosmology [20], derived exploitative and exploration mechanisms which imitate the white hole, black hole, and wormhole regions. Additional physics-based algorithms can be found in [21,22].

Although the metaheuristics are commonly inspired by the phenomena in nature, some algorithms have no clear origin. For example, Sine Cosine Algorithm (SCA) has a purely mathematical model based on sine and cosine functions [23]. Moreover, some algorithms like Biogeography-Based Optimization (BBO) has featured in is with more than one category [24]. BBO gradually evolves the fitness of species (solutions) through probabilistic immigration and emigration models.

As seen, various algorithms have been developed as metaheuristics. Each algorithm takes advantage of one or several search patterns that are provided by nature as a high-level strategy. According to the No Free Lunch (NFL) theorem, there is not a single universal robust algorithm for all problems [25]. The implication of this theorem, the progress made in computer technology, and the growing need for optimizers (especially in engineering problems) are the main reasons for the development of these algorithms. The modern optimization algorithms, though always do not perform as what we would like, but have led to brilliant results in some problems.

Studying the biological and social life of water strider insects, we found interesting exploratory and exploitative patterns which can be interpreted as a metaheuristic. In the proposed algorithm, the life cycle of waters striders, including territory establishment, mating, feeding, and their intrinsic syndromes, are mathematically modeled as a real-valued optimization algorithm. In Water Strider Algorithm (WSA), the solutions gradually get optimum step-by-step and cycle-by-cycle of life of water striders. The proposed WSA can be classified as a swarm intelligent algorithm.

The rest of this paper is organized as follows. The source of inspiration for the algorithm and the formulations are outlined in Section 2. The proposed method is tested by several numerical examples in Section 3. Some applications of the suggested algorithm in engineering problems are provided Section 4. The numerical results are discussed in Section 5. Finally, the conclusions and some suggestions for future directions are provided in Section 6.

2. Water strider optimizer

In the following subsections, the interesting characteristics of water strider insects and their inspiration for proposing a new metaheuristic algorithm are stated. The formulations, flowchart, and pseudo-code are clearly provided for the implementation of the algorithm.

2.1. Inspiration

Water striders (Gerridae) are a class of insects belonging to the order of Hemiptera [26]. Their astounding ability to live on top of the water surface has drawn the attention of humans for many years [27,28]. Indeed, using surface tension as well as hydrophobic legs enabled them to stay above the surface film. The mechanism of locomotion, communication, and behavioral syndromes are some other unique aspects of these insects [29].

Fossil records imply that water striders have been lived on earth for

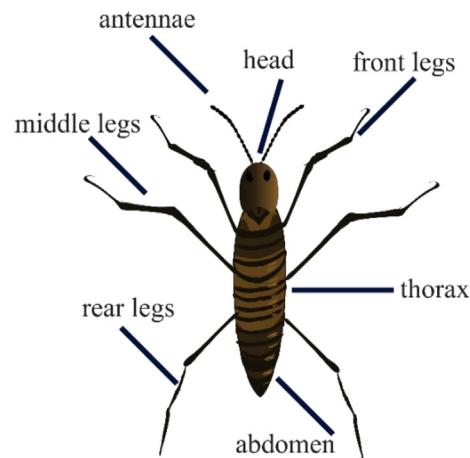


Fig. 1. Water strider anatomy.

more than 50 million years [26]. Their length and weight are about 1cm and 10dynes, respectively. The body size of males are typically smaller than that of females [26]. A schematic of an individual water strider is depicted in Fig. 1 showing different parts of its body.

2.1.1. Territorial behavior

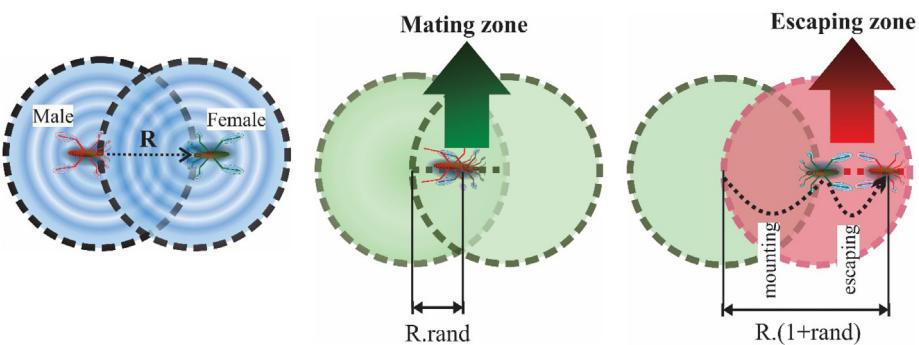
They establish territories to defend their assets consisting of food and mating resources [30]. Entomologists observed that the food resources are predominantly demanded by females, while mating resources are mainly desired by men [30]. Biologically speaking, mating is the main factor in the reproductive success of men, while food resources have a direct influence on the egg numbers and survival of females [31]. Each territory is usually resided by one mature male and a few female bugs. The resident male so-called “keystone” show vigorous mating activity [32]. If alien water striders enter into territories, males will show territorial behavior to dispose of intruders. Aggression between intruder and resident territorial is severe and may lead to murder [30].

2.1.2. Communicating system

Water striders use ripples communication for conveying various information, as illustrated in Fig. 2. This system is also used for detecting probable predators and prey [33]. They produce ripples with different amplitudes, durations, and frequencies by oscillating legs on the water surface. Each of the specific signals is ritualized for different purposes, such as courtship, repelling, sex discrimination, prey locating, etc. [34,35]. They receive the communication ripples through sensory receptors on their legs and react to the situation.

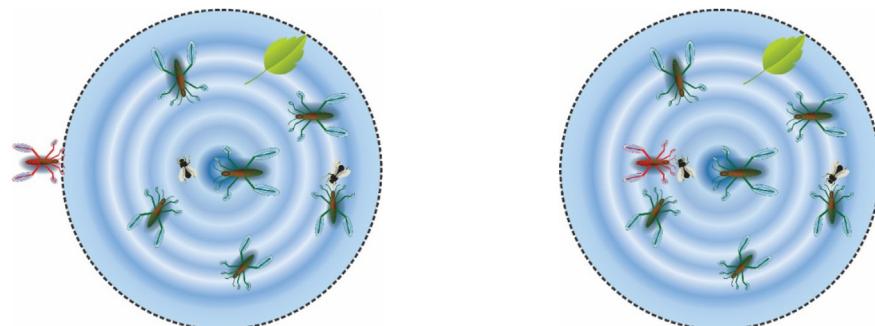


Fig. 2. The ripple communication between two water striders.



(a). The signal exchange (b). Successful mating of WSs (c). Unsuccessful mating of
between male and female WSs after attraction WSs consisting of mating and
for mating escaping levels

Fig. 3. The mating behavior of water striders and position updating.



(a). Foraging for food if the new position is not (b). Moved toward the best strider
enough rich enough

Fig. 4. The foraging process after that the water strider could not find enough food in the new position after mating behavior.

2.1.3. Mating process

As aforementioned, in contrary to reluctant females, males are much eager to mate. The males send precopulatory courtship calling signals, and females respond to them involving attraction or repulsive signals [33]. If they attract each other, they will mate. Otherwise, the male will mount her for a forceful mating. The unwilling females use a hard shield to obstruct male grip and get rid of him [36]. Although the male may win the struggle and begin mating, after a few attempts, he usually desists from harassment and gets away. Because their struggle vibrates water and puts their life in increased danger by attracting predators [37]. It has been observed that mating and pre mating processes, including struggles and display behavior, consume a considerable amount of energy [36].

2.1.4. Foraging behavior

Water striders eat various foods for growing and energy recovery. For instance, they eat dead insects, floating hatched larvae, tadpoles, planets, spiders, and other protein resources. When a land-dwelling insect drops into the water and struggles to survive, the water striders will receive the ripple signals and will capture it [38]. Some species collect and consume sediment or deposit surface. They sometimes involve cannibalism behavior, killing, and feeding off their species [39]. It has been observed that females kill more than males [40]. As aforementioned, food takes priority over any other resource for females, so that they are called “optimal foraging-habitat users,” who live in food-rich positions [41]. Males are tended to skate near the most productive areas occupied by females. In these areas, the preys are more accessible, but they may enter a competition resulting in a battle with potential males [42].

2.1.5. Life cycle of water striders

After successful mating, females lay gelatinous eggs on submerged cliffs or plants. It takes about two months for eggs to become mature water striders. Since then, they generally live for more than one month. They are agile insects; however, it has been observed that they are sometimes hunted by sea birds, lizards, turtles and fishes [43]. They also may die because of severe territorial behavior, cannibalism and harsh temperature [39,44].

2.2. Mathematical model and algorithm

To outline the WSO algorithm, five main steps, including birth, territory establishment, mating, feeding, and death, are mathematically modeled. Throughout the algorithm, search-space is defined as a lake containing different territories of solutions, and food serves as a metaphor for the objective function. In the following steps, the optimization problems assumed as a maximization problem. Thus, the more the objective value, the more optimal the solution. For the sake of brevity, here, a one-dimensional problem is supposed, which can be generalized to any problem with a higher dimension. It is noteworthy that the minimization problems can easily be converted into maximization problems with the same optimal decision variables via simple techniques [1].

2.2.1. Initial birth

At birth, the water striders (WSs) are born with eggs distributed on the lake. Here a random distribution is assumed as Eq. (1)

$$WS_i^0 = Ub + rand(Ub - Lb), i = 1, 2, \dots, nws \quad (1)$$

where WS_i^0 determines the initial position of i th water strider. Ub and

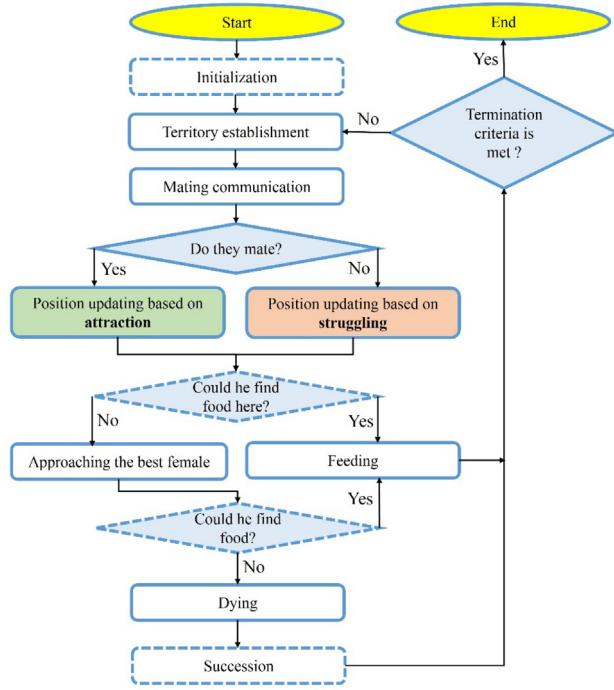


Fig. 5. The flowchart of the proposed WSA.

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 x_i + \prod_{i=1}^n x_i $	30	[-10,10]	0
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j^2)^2$	30	[-100,100]	0
$F_4(x) = \max\{ x_i \}, 1 \leq i \leq n\}$	30	[-100,100]	0
$F_5(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30,30]	0
$F_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	30	[-100,100]	0
$F_7(x) = \sum_{i=1}^n i x_i^4 + \text{random}[0, 1)$	30	[-1.28,1.28]	0

Lb denote the upper and lower bounds corresponding to maximum and minimum allowable values, respectively; $rand$ is a random number between 0 and 1; nws is the number of WSs. The initialized WSs are evaluated using an objective function to calculate the fitness of their position on the lake.

2.2.2. Territory establishment

As mentioned in Section 2.1.1, WSs maintain territories to live, mate, and feed. To establish nt number of territories, the following technique is utilized for assigning the WSs to the territories. Firstly, WSs are sorted based on their fitness and are divided into $\frac{nws}{nt}$ groups orderly. The j^{th} member of each group is assigned to j^{th} territory, where $j = 1, 2, \dots, nt$. Hence $\frac{nws}{nt}$ -number of WSs live inside each territory. As stated in Section 2.1.4, since the females usually find the best place of each territory for feeding, the positions in each territory with the best and worst fitness are considered as the female (optimal foraging-habitat) and male (keystone) positions, respectively.

2.2.3. Mating

Mating is a remarkable process in the life of water striders. As aforementioned, the keystone sends courtship calling ripples and the female respond by sending attraction or repulsive ripple signals. The probability of sending attraction response is considered equal to p , and therefore $(1 - p)$ probability is left for the repulsive response. Since the

Table 2
The multimodal benchmark functions.

Function	Dim	Range	f_{\min}
$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[-500,500]	-418.9829 × 5
$F_9(x) = -\sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5,12.5,12]	0
$F_{10}(x) = -20 \exp(-0.2^2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	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
$F_{12}(x) = \frac{\pi}{n} \{10 \sin(3\pi x_i) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	[-50,50]	0
$F_{13}(x) = 0.1 \{\sin^2(3\pi x_i) \cdot (\sin(\frac{3\pi x_i}{\pi}))^{2m}\}, m = 10$	30	[0,π]	-4.687
$F_{14}(x) = 0.1 \{\sin^2(x_i) \cdot (\sin(\frac{3\pi x_i}{\pi}))^{2m}\}, m = 10$	30	[-20,20]	-1
$F_{15}(x) = [e^{-\sum_{i=1}^n (x_i/\beta)^{2m}} - 2e^{-\sum_{i=1}^n x_i^2}] \cdot \prod_{i=1}^n \cos^2 x_i, m = 5$	30	[-10,10]	-1
$F_{16}(x) = [\{\sum_{i=1}^n \sin^2(x_i)\} - \exp(-\sum_{i=1}^n x_i^2)] \cdot \exp[-\sum_{i=1}^n \sin^2 \sqrt{ x_i }]$	30	[-10,10]	-1

Table 3

Multimodal benchmark functions with fixed-dimension.

Function	Dim	Range	f_{\min}
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{l=1}^2 (x_l - a_{lj})^6} \right)^{-1}$	2	[65,65]	1
$F_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_i(b_i^2 + b_i x_3 + x_4)}{b_i^2 + b_i x_3 + x_4}]^2$	4	[-5,5]	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	[-5,5]	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	2	[-2,2]	3
$F_{19}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij}((x_j - p_{ij})^2))a = \begin{bmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}, c = \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix} \text{ and } p = \begin{bmatrix} 0.3689 & 0.117 & 0.2673 \\ 0.4699 & 0.4387 & 0.747 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{bmatrix}$	3	[1,3]	-3.86
$F_{20}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij}((x_j - p_{ij})^2))a = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 17 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}, c = \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix} \text{ and } p = \begin{bmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{bmatrix}$	6	[0,1]	-3.32
$F_{21}(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.1532
$F_{22}(x) = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.5363

Table 4

The statistical results of unimodal benchmark functions 1–7.

	WSA	GA	PSO	ICA	BBO	SSA	SCA	MFO	DA	MVO
F1	Ave	1.09E-50	0.493651	474.0286	1.78E-27	0.631961	5.28E-09	1.94E-16	2000	101.3336
	Std	3.99E-50	0.976457	266.9246	4.27E-27	0.665318	8.26E-10	9.45E-16	4068.381	96.58759
F2	Ave	4.89E-28	0.027773	9.234122	4.24E-15	0.165913	0.535547	1.28E-18	28.66667	7.530338
	Std	1.94E-27	0.053506	2.396783	9.59E-15	0.057118	0.813057	4.59E-18	15.69831	6.143436
F3	Ave	0.014089	4682.494	4441.754	1.307412	10032.46	6.70E-07	650.4789	16833.44	6410.172
	Std	0.011180	1974.782	1762.155	0.812703	3031.347	4.49E-07	1248.101	12520.75	5456.428
F4	Ave	0.000490	9.282368	14.10854	0.069323	7.924755	1.328179	1.335270	45.80769	5.930244
	Std	0.000354	2.208485	2.390036	0.093936	1.155925	1.611482	1.828663	13.34950	7.065519
F5	Ave	32.42146	481.1478	31370.23	107.5567	219.3536	67.11092	27.51712	18268.11	2890.214
	Std	29.52849	481.3205	24726.88	135.2055	151.6324	84.22198	0.556120	36497.23	3939.582
F6	Ave	0	0.398834	477.1967	1.65E-27	0.500071	5.44E-09	3.720126	1340.033	127.2942
	Std	0	0.472675	265.9348	3.56E-27	0.650763	9.93E-10	0.334225	3474.982	101.2584
F7	Ave	0.006433	0.011947	0.141906	0.026134	0.027428	0.017356	0.00598	1.375182	0.067759
	Std	0.0018394	0.0063860	0.059233	0.010122	0.009938	0.006085	0.00590	3.577760	0.050227

Table 5

The statistical results of multimodal benchmark functions 8–13.

	WSA	GA	PSO	ICA	BBO	SSA	SCA	MFO	DA	MVO
F8	Ave	-9354.74	-10348.4	-5693.364	-8087.05	-12566.2	-7529.63	-4314.06	-8732.63	-6793.96
	Std	653.1757	389.5341	614.6499	486.5310	2.18026	705.6967	255.6938	1072.626	989.3598
F9	Ave	40.56002	9.177337	62.34716	100.6904	1.20078	51.96994	1.83186	148.5348	60.4525
	Std	10.78416	2.255343	18.0979	17.71236	0.82409	16.85193	6.96691	40.87507	27.58083
F10	Ave	1.88E-14	1.123709	6.63813	0.04571	0.23474	1.78994	12.38263	9.7306	5.09027
	Std	4.52E-15	0.558145	1.00764	0.24526	0.1827	0.86588	9.10541	9.74539	0.15472
F11	Ave	0.016042	0.293621	4.63638	0.02423	0.52837	0.01026	0.00786	21.14255	1.9369
	Std	0.020111	0.246095	1.83215	0.02844	0.22331	0.01217	0.02831	45.56648	1.57439
F12	Ave	1.57E-32	0.128613	7.71789	0.03784	0.00879	1.80927	0.3975	0.25016	2.68162
	Std	5.57E-48	0.153503	3.60638	0.20661	0.02595	1.5772	0.13269	0.48244	5.13881
F13	Ave	1.35E-32	0.177732	873.08203	1.99E-23	0.02846	0.00366	2.06867	1.37E + 7	10.19126
	Std	5.57E-48	0.139143	3390.035	1.08E-22	0.02109	0.00527	0.13673	7.49E + 7	11.45285

response of females is not determined, for simplicity, we assumed p as 50%. If the female sends an attraction signal, they will move toward each other and will mate. Considering a circle wave as illustrated in Fig. 3.a, after mating, the new position will be updated to a location between them as Fig. 3.b. As mentioned in Section 2.1.3, if the female rejects the request, the male will mount her, then female dismount him and get him away, as shown in Fig. 3.c. The keystone may mate or be repelled, either way, the new position of keystone will be calculated by Eq. (2)

$$\left\{ \begin{array}{l} WS_i^{t+1} = WS_i^t + R \cdot rand \quad \text{if mating happens (with probability of } p) \\ WS_i^{t+1} = WS_i^t + R \cdot (1 + rand) \quad \text{otherwise} \end{array} \right. \quad (2)$$

where WS_i^t is the position of i^{th} WS in the t^{th} cycle; $rand$ is a random number between 0 and 1; R is a vector whose initial 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 fitness proportionate selection mechanisms such as roulette wheel selection [45]. The length of R is equal to the Euclidean distance between male (WS_i^{t-1}) and female WSs (WS_F^{t-1}) (the radius of ripple wave) as Eq. (3)

Table 6

The statistical results of fixed dimension multimodal benchmark functions 14–23.

		WSA	GA	PSO	ICA	BBO	SSA	SCA	MFO	DA	MVO
F14	Ave	0.998004	1.130409	3.693964	1.330271	3.527829	1.592317	1.794415	1.525135	1.757204	1.560495
	Std	1.13E-16	0.430993	2.446979	0.655267	3.634702	1.150621	1.892839	1.34095	1.289434	0.810885
F15	Ave	0.000549	0.001722	0.000848	0.000686	0.003912	0.002086	0.001134	0.00092	0.001832	0.003426
	Std	0.00032	0.003555	0.000504	0.000158	0.004179	0.004974	0.00035	0.000284	0.001337	0.006762
F16	Ave	-1.03163	-1.03163	-1.03163	-1.03163	-1.03082	-1.03163	-1.03156	-1.03163	-1.03163	-1.03163
	Std	5.68E-16	1.27E-15	6.45E-16	5.05E-16	0.01666	4.95E-14	7.79E-05	6.78E-16	2.70E-14	1.47E-06
F17	Ave	0.397887	0.397887	0.397887	0.397887	0.40586	0.397887	0.403152	0.397887	0.39789	
	Std	0	0	0	0	0.011123	9.94E-14	0.007668	0	6.48E-15	3.98E-06
F18	Ave	3	3	3	3	5.687621	3	3.000109	3	3	3.000014
	Std	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
F19	Ave	-3.86278	-3.86278	-3.86278	-3.86278	-3.86178	-3.86278	-3.85388	-3.86278	-3.86071	-3.86278
	Std	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
F20	Ave	-3.25066	-3.28633	-3.28826	-3.31011	-3.27638	-3.21634	-2.94575	-3.22824	-3.23633	-3.25038
	Std	0.059241	0.055415	0.056989	0.036278	0.057813	0.042258	0.320805	0.053929	0.081325	0.059472
F21	Ave	-6.72819	-6.99664	-5.82346	-6.97219	-5.92825	-8.47826	-3.37565	-7.30772	-6.01288	-7.80433
	Std	3.378711	3.696605	3.645863	3.350957	3.306321	2.897832	2.046341	3.400748	2.150163	3.018852
F22	Ave	-7.35819	-8.46037	-6.4398	-7.99872	-6.45539	-8.3773	-4.06593	-8.17415	-6.47641	-8.32628
	Std	3.609873	3.285714	3.565778	3.259124	3.598817	3.203741	1.942808	3.250245	2.735904	3.062084
F23	Ave	-8.30703	-8.57634	-5.42603	-6.49715	-5.52286	-9.49528	-4.667	-8.66814	-6.24959	-9.02186
	Std	3.499898	3.324285	3.527142	3.647386	3.453432	2.700535	1.758723	3.183513	2.475969	2.584132

and Fig. 3.a.

$$R = WS_F^{t-1} - WS_i^{t-1} \quad (3)$$

2.2.4. Feeding

As stated in Section 2.1.3, whether the mating happens successfully or not, the process consumes a lot of energy. Therefore, in the new position, WSs forage for food resources. As aforesaid, to assess the position for food availability, it is evaluated by the objective function. If the value of the objective function 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 containing the highest fitness. To this end, Eq. (4) is defined for transferring to the new position around the best WS of lake (WS_{BL}^t) containing a good deal of food resources, as shown in Fig. 4.

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

2.2.5. Death and succession

In order to determine the result of the food attainment process, the objective function is evaluated and is compared with that of the prior position. If the new fitness is lesser, the WS will die, because it not only could not find food but also increased the battle danger with the WSs of the destination territory. In this case, the newly matured larva will succeed the dead WS as the keystone, and the position of him is randomly initialized inside the territory as Eq. (5). If it were otherwise, the keystone would remain alive.

$$WS_i^{t+1} = Lb_j^t + 2rand.(Ub_j^t - Lb_j^t) \quad (5)$$

Ub_j^t and Lb_j^t denote the maximum and minimum values of WS's position inside j th territory. In other words, they determine the boundaries of died WS's territory.

2.2.6. Termination of WSA

In the last step of the algorithm, if the termination condition is met, the algorithm stops and reports the best-experienced position. But if the condition is not satisfied, it will return to the mating step for a new loop of the life cycle and territory establishment. In this paper, the maximum number of function evaluations ($MaxNFE$) is considered as the termination condition in all problems. However, other conditions such as the maximum number of life cycles ($MaxCycle$) can be implemented as a stopping condition of WSA.

The pseudo-code and flowchart of the proposed WSA are provided

in Algorithm 1 and Fig. 5, respectively. As seen, the algorithm can be simply implemented in programming languages.

Algorithm 1 Pseudo-code of WSA optimizer

```

Inputs: The population size  $nws$ , number of territories  $nt$  and the maximum number of iterations  $MaxCycle$ 
Outputs: The richest location of WS and its objective value
Initialize the random population as Eq. (1)
Calculate the fitness value of WSs
while (terminating condition is not met) do
    Establish  $nt$  number of territories and allocate the WSs as described in Section 2.2.2
    for (each territory) do
        The male keystone sends mating ripples, and the selected female decides about the response which can be an attractive or repulsive signal.
        Update the position of keystone based on the response of female and Eq. (2)
        Evaluate the new position to find food for compensating the consumed energy during the mating
        If (keystone could not find food) then
            Forage for food resource and approach the food-rich territory by Eq. (4).
            if (keystone could not find food again) then
                The hungry keystone will be died because of starvation or killed by resident keystone of the new territory.
                A matured larva will replace the killed keystone as the successor defined by Eq. (5).
            end
        end
    end
Return  $WS_{optimal}$ 

```

2.3. Computational complexity of the WSA

For calculating the computational complexity of the WSA, three main processes should be taken into account. In the beginning of the algorithm, all N water striders must be randomly initialized and evaluated. So, the computational complexity of the initialization process is $O(N)$. In the second process, the attraction or struggling position updating occurs inside all T number of territories which corresponds to $O(T)$ complexity. After updating the position, the algorithm might require from 0 to 2 number of evaluations for each of the keystones. These evaluations are performed for finding food or succession. Therefore, the computational complexity of the third process is between $O(0)$ and $O(2T)$, regarding the quality of solutions. It is noteworthy that the first process is employed only once while the second and third processes are repeated for several cycles (C). Putting these together, the computational complexity of the proposed algorithm is between $O(N + C \times T)$ and at most $O(N + C \times 3T)$.

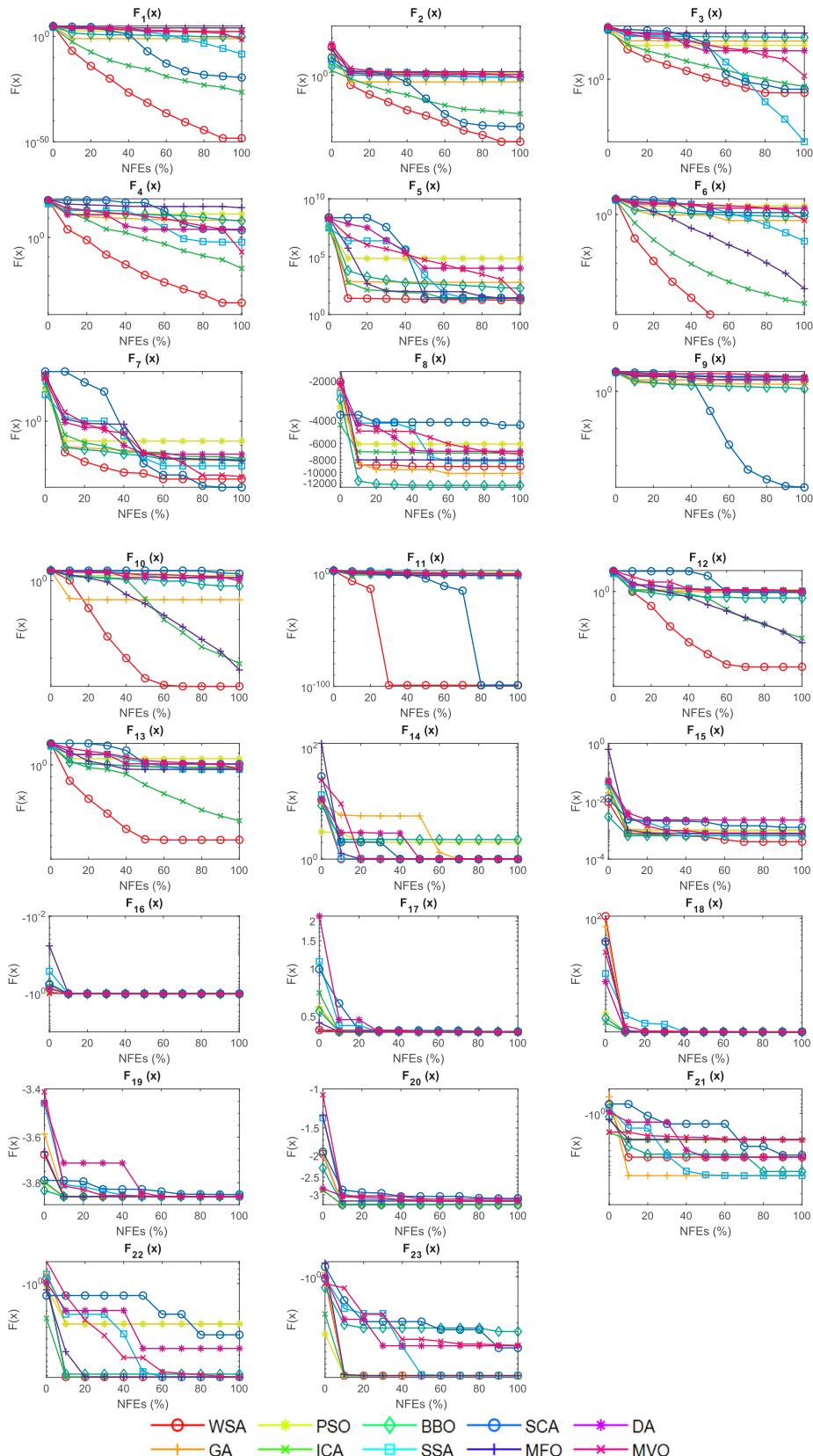


Fig. 6. The comparison of the convergence rate of algorithms.

3. Numerical experiments

In this section, the proposed algorithm is tested by several

numerical and engineering problems. Its results are compared with those of well-known and state of the art methods. The following subsections discuss the results to provide an insight into the search

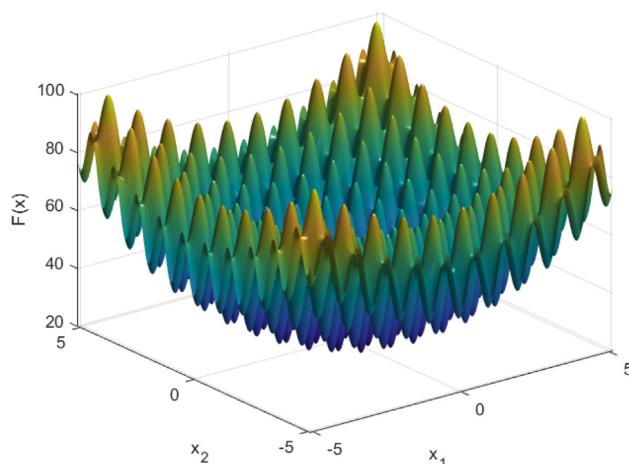


Fig 7. The 3D representation of Rastrigin function.

behaviors of the algorithm, such as exploration, exploitation capabilities as well as its convergence rate in a comparative manner.

The experiments are arranged in two phases. In the first phase, the exploitation and exploration capabilities of the algorithms are tested by unimodal and multimodal function, and statistical measures such as averages and standard deviations are examined. Furthermore, the convergence rate of algorithms is discussed in the first phase. This phase provides the overall assessment of the algorithm. In the second phase, complicated problems such as shifted, biased and composite functions with different dimensions are selected, and non-parametric the Kruskal-Wallis Test is carried out to ensure unbiased performance and investigation of the WSA.

The internal parameters of algorithms are tuned according to the literature [46,47], and the number of territories and the population of WSs are assumed as 25 and 50, respectively. These parameters lead to 2 WSs inside each territory that is the minimum possible number of territories' population. For a fair comparison, in all algorithms, the NFEs are predefined as 5000 multiples by dimension (D) [48]. For a reliable assessment, the algorithms are executed 30 times independently for each task.

3.1. Phase I: Exploration and exploitation behavior of WSA

The exploitation (intensification) and exploration (diversification) are the most important characteristics of metaheuristic optimization algorithms that determine their performance. The exploitation component tries to search spaces near the solutions with higher fitness. In contrast, the exploration is the component of the search which focuses on the unexploited regions of the search space. Without good explorative behavior, the algorithm will trap in local optima. On the other hand, the lack of exploitation can make the algorithm work slowly, and it never can converge to the optimal solution. Therefore, a good balance between exploration and exploitation will result in a brilliant performance. However, some problems require intensive exploitation and some others need extreme exploitation for finding their optimum. Hence balancing these mechanisms, itself is a hard optimization problem to solve.

In WSA, to make a balance between intensification and diversification, a few mechanisms are devised, such as attraction for intense exploitation, repulsion, and foraging for exploitation with moderate intensity and the death and succession for exploration. The roulette wheel selection makes it possible for lower females to get involved in mating. The repulsive behavior prevents excessive proximity of the WSs. The foraging mechanism makes more chances of approaching the best WS. In succession, the low-quality WSs are removed, and chances for creating random solutions are provided. It is noteworthy that the

boundaries for successor larvae, by involving the so-far-acquired knowledge of the problem, prevent excessive randomization. The territory establishment layout also distributes the search around the space and makes the algorithm suitable for problems with several local optima.

To examine the performance of the exploitation and exploration operators of the algorithms, a set of twenty-three popular unimodal, multimodal and fixed-dimension multimodal benchmark problems are investigated [49,50]. These problems are described in Tables 1–3. In these tables, Dim is the dimension of functions, Boundary denotes the upper and lower bounds of search space, and F_{\min} stands for the optimum values of the functions.

The results are given in Tables 4–6, where the average values and the standard deviation for 30 independent trials are compared with those of the current algorithms. To this end, the well-known algorithms, namely GA, PSO, ICA, and BBO, as well as state of the art methods, namely SSA, SCA, MFO, DA, MVO, are selected for comparison. Their internal parameters are tuned according to [46,47].

As reported in Tables 4–6, WSA obtained the minimum average values in twelve problems and the lowest standard deviations in nine problems. After WSA, the SSA stands in second place with the lowest average in two problems and ICA and SCA with the lowest standard deviation in four problems. These results indicate that the proposed method has an overall competitive performance. Table 4 indicates that the WSA is capable of exploiting the space efficiently. Because WSA has the first place in four functions out of seven unimodal functions and in the rest of the problems, it has the second or third rank. Additionally, the high capability of exploration of WSA can be implied from Tables 5 and 6, where it has the highest number of mean values with the first rank in eight functions and the second or third rank in five functions.

3.2. Phase I: convergence rate

Convergence speed or rate of convergence is a substantial feature of metaheuristics. Since the number of function evaluations (NFEs) is the most effective factor in the performance of algorithms, the convergence histories are compared with drawing best-so-far optimum value against the ratio of the evaluated functions. Fig. 6 demonstrates the average convergence curves for all 23 problems mentioned in Tables 1–3. As seen, the new algorithm has the highest convergence rate in most of the problems. Moreover, the ICA and MFO have a relatively quick convergence among the previous classic modern metaheuristics, respectively.

3.3. Phase I: Monitoring the position of WSs

In this section, to provide an insight into the movement of WSs in search-space (lake), their positions in different cycles are monitored. As shown in Fig. 7, the non-linear multimodal Rastrigin function is selected as the test function, which has a lot of local optima. This function is provided in Table 2 as $F_9(x)$. Rastrigin's global minimum has a value of 0 at point (0, 0).

As shown in Fig. 8, for a better view of the positions three territories, are deployed and distinguished with black, yellow, and red WSs. As seen, at the beginning of the optimization, the WSs are randomly dispersed throughout the search-space (lake). After 20 cycles, the WSs focused on the central parts which have near optimum objectives. In cycle 40, some of the WSs disclosed several local optima shown with contour rings; however, some WSs yet explore the lake. In cycle 60, most of the WSs occupied the local optima. The territorial behavior of WSs help algorithm to retest various local optima before final exploitation. This feature prevents the premature convergence of WSA. Cycle 80 shows that they all aggregated in the near-global optimum location. Finally, in Cycle 90, the optimum point is exploited by the WSs.

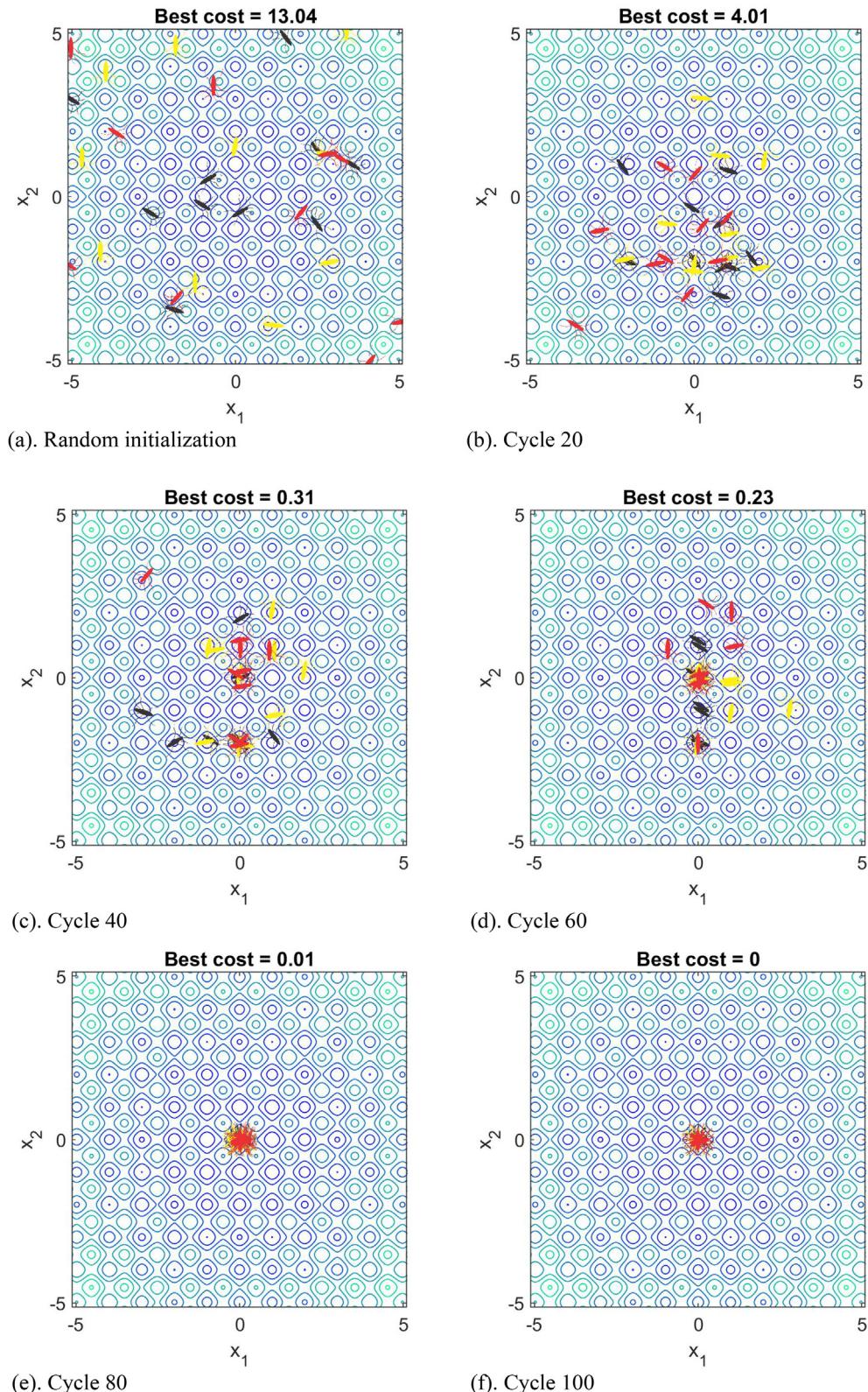


Fig. 8. The position of water striders in different cycles.

3.4. Phase II: verification against possible biases

Each of the metaheuristics usually proves superior performance in some problems and demonstrate average performance on the other kind of problems. Indeed, finding a metaheuristic that performs well in all problems is almost impossible. Among them, some algorithms solve

some of the benchmark optimization problems efficiently but demonstrate extremely poor performance in other problems. For instance, if a weak algorithm searches only point zero on space and the optimal points of testing functions are on zero, this algorithm will be successful in a test, but it is deficient in both exploration and exploitation. Technically speaking, the latter type of algorithms may have an intense

Table 7The benchmark functions F₂₄-F₃₄.

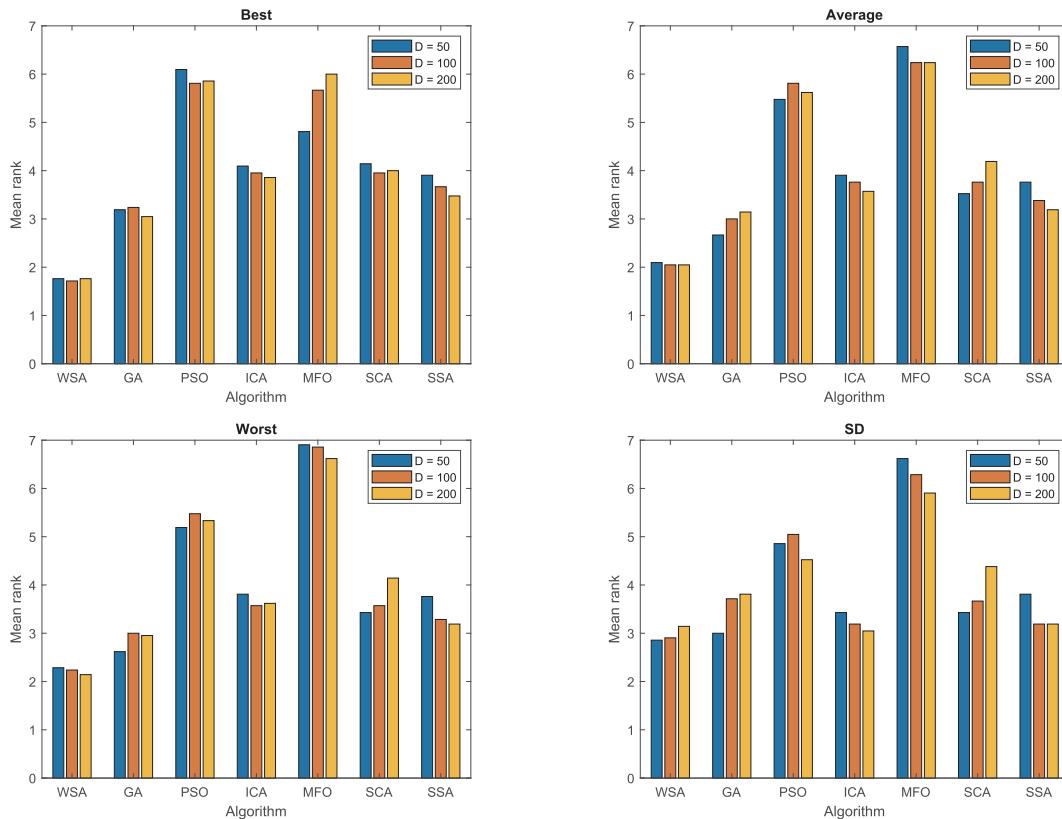
Function	Dims.	Range	F(x*)	U/M	Separable	Shifted	f _{bias}
$F_{24} = \sum_{i=1}^n z_i^2 + f_{bias}, z = x - o^1$	50, 100, 200	[−100,100]	0	U	No	Yes	−450
$F_{25} = \max_i\{ z_i \}, 1 \leq i \leq n\} + f_{bias}, z = x - o$	50, 100, 200	[−100,100]	0	M	Yes	Yes	390
$F_{26} = \sum_{i=1}^n (100(z_i^2 + z_{i+1})^2 + (z_i - 1)^2) + , z = x - o$	50, 100, 200	[−5,5]	0	M	Yes	Yes	−330
$F_{27} = \sum_{i=1}^n (z_i^2 - 10\cos(2\pi z_i) + 10) + f_{bias}, z = x - o$	50, 100, 200	[−600,600]	0	M	No	Yes	−180
$F_{28} = \sum_{i=1}^n \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f_{bias}, z = x - o$	50, 100, 200	[−32,32]	0	M	Yes	Yes	−140
$F_{29} = -2\exp(-0.2\sqrt{\frac{\sum_{i=1}^n z_i^2}{n}}) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi z_i)\right) + 20 + e + f_{bias}, z = x - o$	50, 100, 200	[−10,10]	0	U	Yes	No	−
$F_{30} = \sum_{i=1}^n z_i + \prod_{i=1}^n z_i $	50, 100, 200	[−65.536,65.536]	0	U	No	No	−
$F_{31} = \sum_{i=1}^n (\sum_{j=1}^n z_j)^2$	50, 100, 200	[−100,100]	0	U	No	No	−
$F_{32} = (\sum_{i=1}^{n-1} f_{10}(z_i, z_{i+1}) + f_{10}(z_m, z_1), f_{10} = (x^2 + y^2)^{0.25}(\sin^2(50(x^2 + y^2)^{0.1}) + 1)$	50, 100, 200	[−15,15]	0	U	Yes	No	−
$F_{33} = \sum_{i=1}^{n-1} (z_i^2 + 2z_{i+1}^2 - 0.3\cos(3\pi z_i) - 0.4\cos(4\pi z_{i+1}) + 0.7)$	50, 100, 200	[−100,100]	0	U	Yes	No	−
$F_{34} = \sum_{i=1}^{n-1} (z_i^2 + z_{i+1}^2)^{0.25}(\sin^2(50(z_i^2 + z_{i+1}^2)^{0.1}) + 1)$	50, 100, 200	[−100,100]	0	U	No	Yes	−

Table 8The composite benchmark functions F₃₅-F₄₄.

Function	First function	Second Function	Weight factor of first function
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

systematic bias in favor of the structure of some specific problems. These biases can be observed in the poor performance of the algorithms when computing variable-shifted, objective-biased, or hybrid problems. Moreover, they may show disparate performance in the same problems with various dimensions, while they perform well on the problems without the mentioned characteristics, e.g. low dimension or shift-less functions. To verify the performance of the proposed algorithm against the stated biases, in this section, an assortment of twenty-one additional functions with a wide range of characteristics is studied [48]. Furthermore, to ensure that the proposed algorithm does not have size bias, three sets of dimensions with 50, 100 and 200 variables are examined. The functions are described in Tables 7 and 8.

Thirty independent runs are performed for each of the functions, and the statistical measures such as mean, best, worst, and standard deviation of the results as well as Kruskal-Wallis nonparametric test are

**Fig. 9.** Comparison of statistical measures for algorithms.

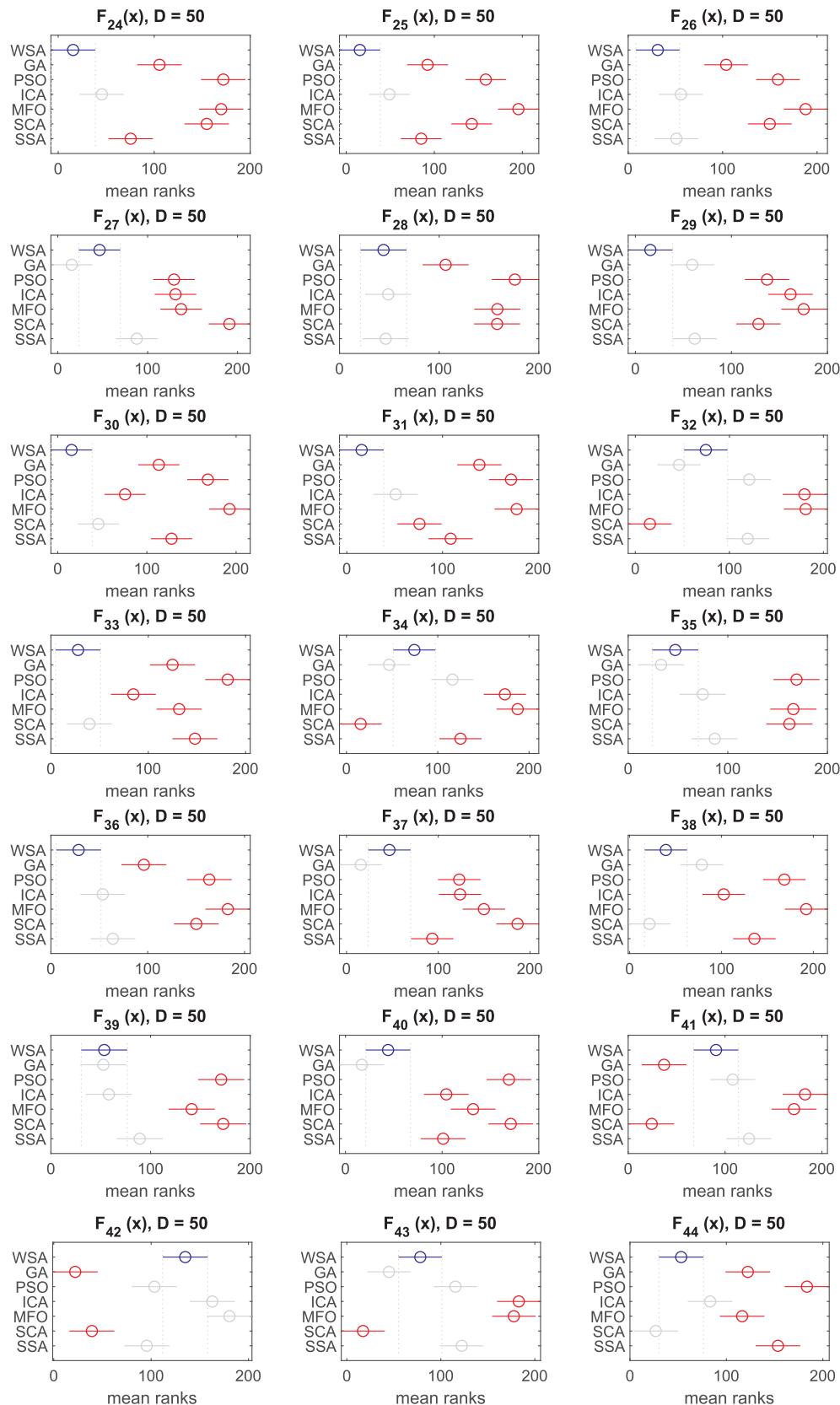


Fig. 10. Comparison of mean ranks of for 50 dimension problems.

investigated. For the sake of brevity, the best of implemented metaheuristics comprising GA, PSO, and ICA as well-established classic metaheuristics and MFO, SCA and SCA as modern algorithms are

examined for this new set.

The mean ranks of algorithms for statistical measurements (e.g. best, average, worst and standard deviation (SD)) are summarized in

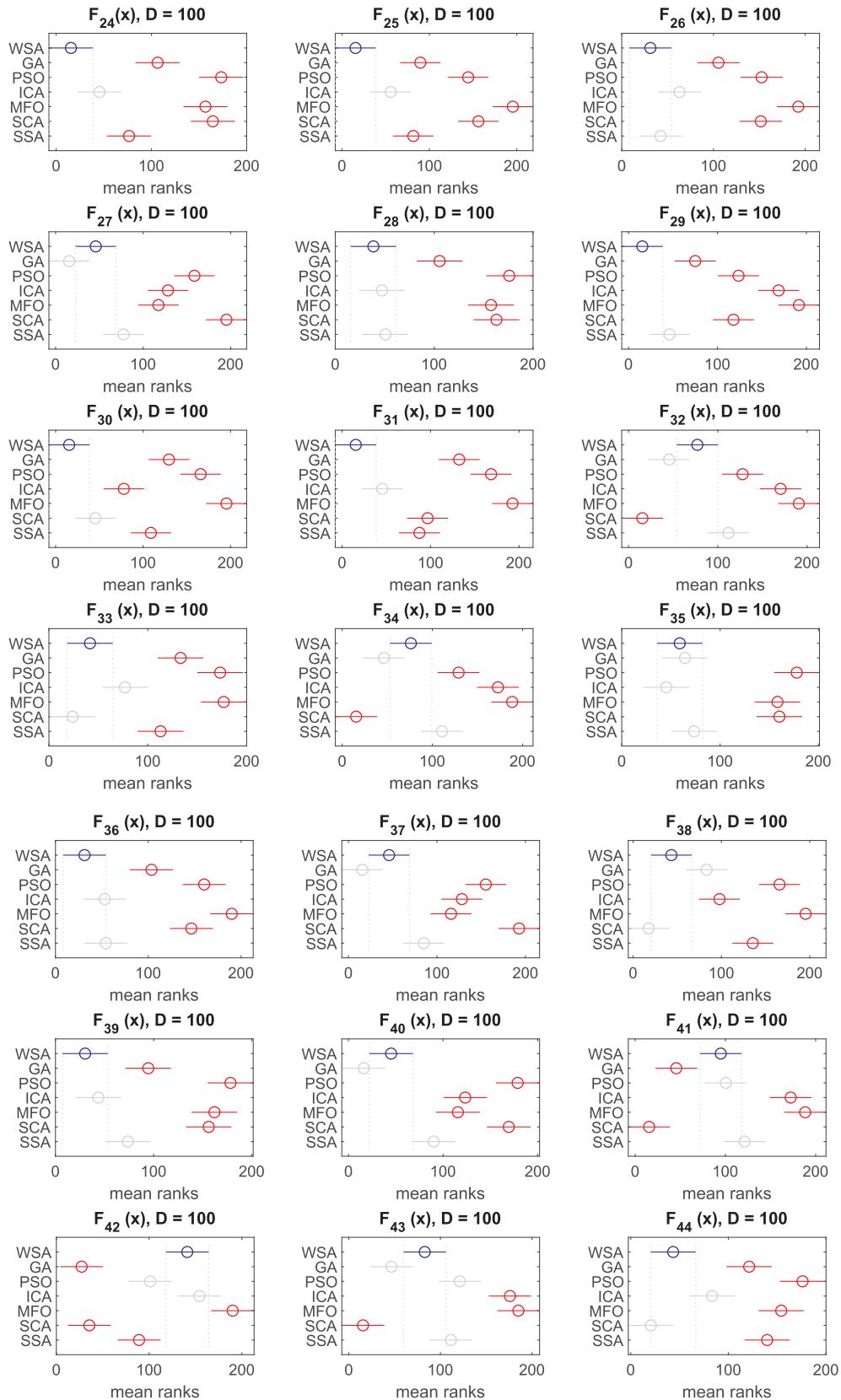


Fig. 11. Comparison of mean ranks of for 100 dimension problems.

Fig. 9. As seen, considering the rank, WSA is better than other algorithms almost for all criteria. Moreover, the dimension changing has the least influence on the performance of WSA for all measurements.

In this section, possible biases of statistical measurements are

resolved considering nonparametric tests. Hence the Kruskal-Wallis with test 5% degree is carried out and the results are reported in Figs. 10–12. The blue shapes represent the results of the WSA. The red shapes stand for the algorithms with significant difference and rejection

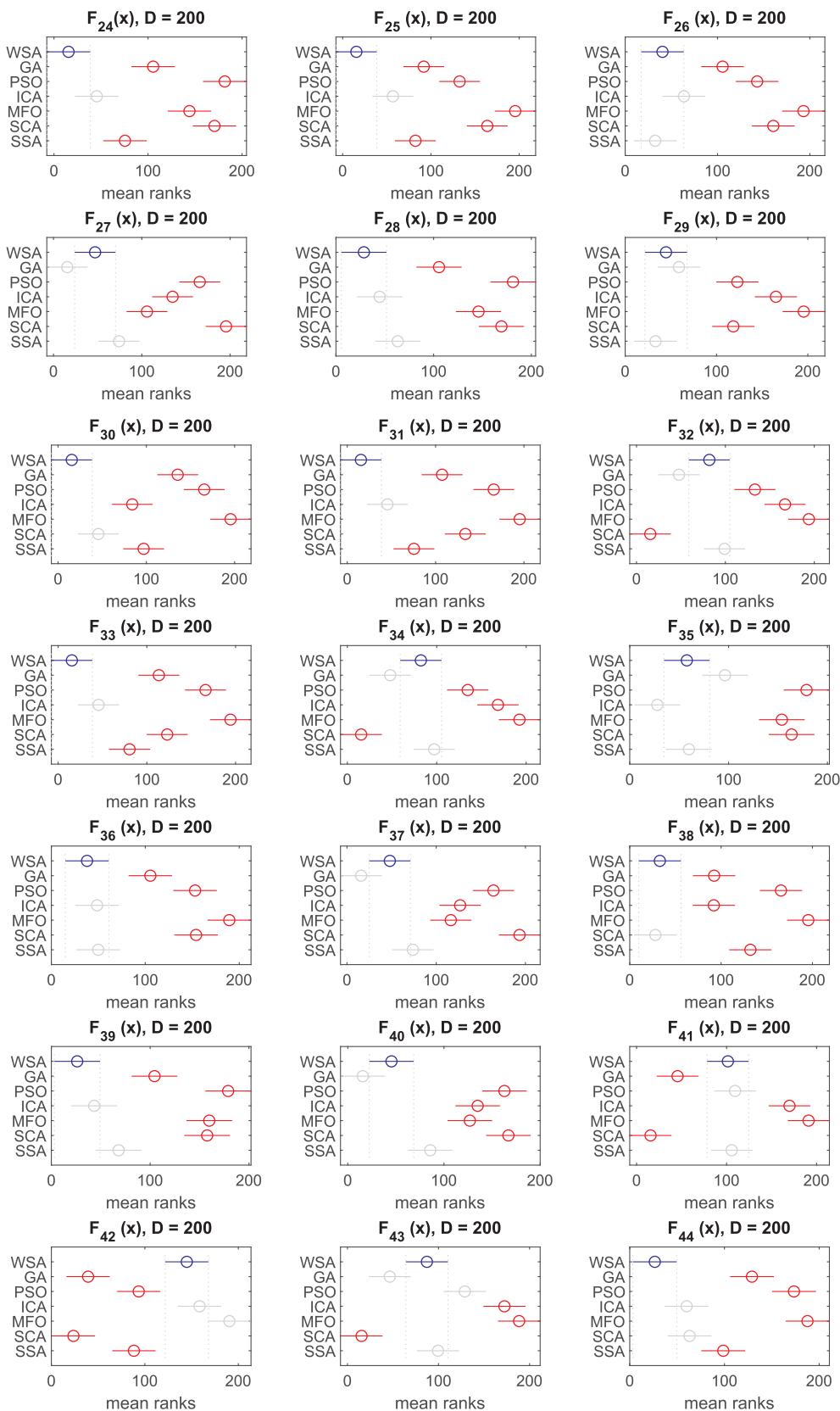


Fig. 12. Comparison of mean ranks of for 200 dimension problems.

of the null hypothesis, while, the gray shapes denote the algorithms without significant difference. According to the results, WSA obtained the first rank for twenty-seven experiments (nine experiments for each dimensional set). The equal number of first ranks in different

dimensional sets, once again confirms that WSA is free from size bias. In the rest of the problems, apart from f_{19} , WSA obtained the second or third place.

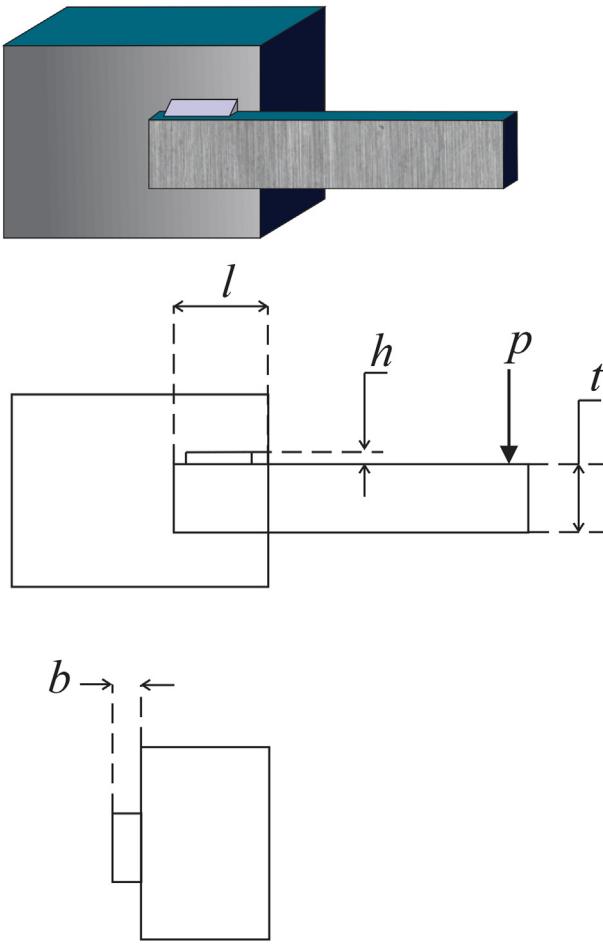


Fig. 13. Welded beam design problem.

Table 9

Comparison of statistical results obtained by various algorithms for welded beam design.

Algorithm	Best	Mean	Worst	SD
WSA	1.724852	1.724908	1.725068	4.15E-05
GA	1.778087	2.331286	3.23984	0.345193
PSO	1.85272	2.613785	3.841845	0.470875
ICA	1.725135	1.79433	2.237755	0.109935
MFO	1.724852	1.732109	1.950241	0.03398
SCA	1.786863	1.849364	1.925162	0.034689
SSA	1.725886	1.823426	2.246638	0.128283
BBO	1.918055	2.630412	3.606933	0.410914
CBO	1.724862	1.855919	2.514803	0.208853
NNA	1.724852	1.727439	1.766896	0.006565
GWO	1.725232	1.72631	1.728487	0.000771
WOA	1.777113	2.098289	3.507098	0.327464
GA [54]	1.748309	1.771973	1.785835	1.12E-02
GA4 [55]	1.728226	1.792654	1.993408	7.47E-02
CAEP [56]	1.724852	1.971809	3.179709	4.43E-01
CPSO [57]	1.728024	1.748831	1.782143	1.29E-02
HPSO [58]	1.724852	1.749040	1.814295	4.01E-02
NM-PSO [59]	1.724717	1.726373	1.733393	3.50E-03
MGA [60]	1.82452	1.9190	1.9950	5.37E-02
DE [61]	1.733461	1.768158	1.824105	2.21E-02
UPSO [62]	1.921990	2.83721	N.A	N.A
CDE [61]	1.73346	1.76815	N.A	N.A

Table 10
Comparison of the best design for welded beam example.

Algorithm	Optimal values for variables				Optimal cost
	x_1	x_2	x_3	x_4	
WSA	0.20573	3.470489	9.036624	0.20573	1.724852
GA	0.220448	3.282171	8.750954	0.219381	1.778087
PSO	0.219292	3.430416	8.433559	0.236204	1.85272
ICA	0.205799	3.469634	9.03495	0.205806	1.725135
MFO	0.20573	3.470489	9.036624	0.20573	1.724852
SCA	0.205977	3.805191	8.99305	0.208803	1.786863
SSA	0.205672	3.467299	9.04818	0.205672	1.725886
BBO	0.185486	4.3129	8.439903	0.235902	1.918055
CBO	0.205728	3.47052	9.036627	0.20573	1.724862
NNA	0.20573	3.470489	9.036624	0.20573	1.724852
GWO	0.205677	3.470894	9.038558	0.205739	1.725232
TEO	0.205681	3.472305	9.035133	0.205796	1.725284
WOA	0.204883	3.519003	8.959591	0.213723	1.777113
GSA [63]	0.182129	3.856979	10	0.202376	1.879952
HHO [47]	0.204039	3.531061	9.024763	0.206147	1.731991
HS [64]	0.2442	6.2231	8.2915	0.2443	2.3807
ESs [65]	0.199742	3.61206	9.0375	0.206082	1.7373
CDE [66]	0.203137	3.542998	9.033498	0.206179	1.733462
CPSO [57]	0.202369	3.544214	9.04821	0.205723	1.728024
FGA [67]	0.205986	3.471328	9.020224	0.20648	1.728226
GA 1 [68]	0.2489	6.173	8.1789	0.2533	2.433116
GA 2 [54]	0.2088	3.4205	8.9975	0.21	1.74831
GA 3 [54]	0.205986	3.471328	9.020224	0.20648	1.728226
Coello and Montes [70]	0.205986	3.471328	9.020224	0.20648	1.72822
Siddall [71]	0.2444	6.2189	8.2915	0.2444	2.38154
Ragsdell [72]	0.2455	6.196	8.2915	0.2444	2.38154
Random [72]	0.4575	4.7313	5.0853	0.66	4.1185
Simplex [72]	0.2792	5.6256	7.7512	0.2796	2.5307
David [72]	0.2434	6.2552	8.2915	0.2444	2.3841
APPROX [72]	0.2444	6.2189	8.2915	0.2444	2.3815

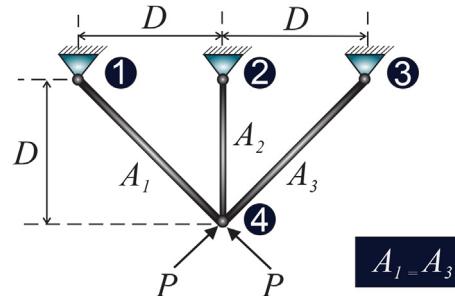


Fig. 14. Three-bar truss design problem.

Table 11

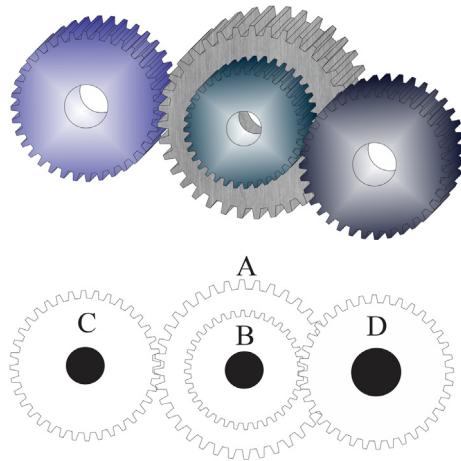
Comparison of statistical results obtained by various algorithms for three bar truss design.

Algorithm	Best	Mean	Worst	SD
WSA	263.89584340	263.89606687	263.89743217	0.00031196
GA	263.89588573	263.96803663	264.82080546	0.16686174
PSO	263.89584341	263.95741428	264.58490296	0.13689749
ICA	263.89584519	263.89932689	263.91413326	0.00411693
MFO	263.89584742	263.91806578	264.16411946	0.04703090
SCA	263.89778646	264.75204098	282.84271247	3.73097657
SSA	263.89584346	263.89635079	263.90188334	0.00093128
BBO	263.89665288	264.50214578	267.74884623	0.88058464
CBO	263.89588209	263.91023329	264.04724926	0.02779265
NNA	263.89584432	263.89739164	263.90537191	0.00245495
GWO	263.89600631	263.89795501	263.90421778	0.00161422
WOA	263.89609706	264.18659579	265.08263623	0.35863425

Table 12

The best designs obtained for three-bar truss design problem.

Algorithm	Optimal values for variables		Optimal cost
	x_1	x_2	
WSA	0.788683	0,408,227	263.89584340
GA	0.788915	0.407569	263.89588573
PSO	0.788669	0.408265	263.89584341
ICA	0.788625	0.408389	263.89584519
MFO	0.788601	0.408458	263.89584742
SCA	0.787854	0.41059	263.89778646
SSA	0.788683	0.408265	263.89584346
BBO	0.789066	0.408227	263.89665288
CBO	0.788504	0.408733	263.89588209
NNA	0.788639	0.40835	263.89584432
GWO	0.788648	0.408325	263.89600631
WOA	0.788091	0.409903	263.89609706
DEDS [74]	0.788675	0.408248	263.8958434
MBA [61]	0.788565	0.40856	263.8958522
HHO [47]	0.788663	0.408283	263.8958434
MVO [20]	0.788603	0.408453	263.8958499
GOA [75]	0.788898	0.40762	263.8958814
Tsai [76]	0.788	0.408	265.6245
CS [77]	0.78867	0.40902	263.9716
Ray and Sain [73]	0.795	0.395	264.3

**Fig. 15.** Compound gear design problem.**Table 13**

Comparison of statistical results obtained by various algorithms for gear train design.

Algorithm	Best	Mean	Worst	SD
WSA	2.7009E-12	1.6800E-10	1.3616E-09	3.8265E-10
GA	2.7009E-12	1.6212E-09	1.5247E-08	3.2174E-09
PSO	2.3078E-11	7.9383E-08	1.0222E-06	1.8147E-07
ICA	2.7009E-12	8.0417E-10	2.3576E-09	7.7862E-10
MFO	2.3078E-11	7.5337E-09	2.7265E-08	9.3539E-09
SCA	2.7009E-12	8.8113E-10	2.3576E-09	6.4529E-10
SSA	2.7009E-12	1.9822E-09	2.7265E-08	4.5748E-09
BBO	2.3078E-11	4.5418E-08	4.2018E-07	7.2953E-08
CBO	2.3078E-11	2.1032E-09	1.1173E-08	2.4025E-09
NNA	2.7009E-12	6.1128E-10	2.3576E-09	6.1167E-10
GWO	2.7009E-12	3.3777E-10	9.9216E-10	4.0956E-10
WOA	2.7009E-12	9.6633E-10	6.5123E-09	1.1296E-09

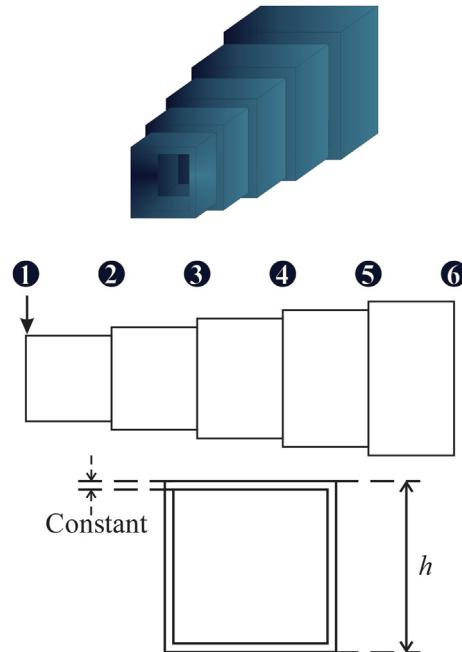
4. Engineering design problems

In this section, five engineering problems are solved using WSA, GA, PSO, SCA, SSA, BBO, Colliding Bodies Optimization (CBO) [22], Neural Network Algorithm (NAA) [46], Grey Wolf Algorithm (GWO) [51] and Whale Optimization Algorithm (WOA) [52]. The statistical results and

Table 14

Comparison of the best designs for gear design problem.

Algorithm	Optimal values for variables				Optimal cost
	x_1	x_2	x_3	x_4	
WSA	43	16	19	49	2.7009E-12
GA	49	19	16	43	2.7009E-12
PSO	34	13	20	53	2.3078E-11
ICA	43	16	19	49	2.7009E-12
MFO	51	30	13	53	2.3078E-11
SCA	43	16	19	49	2.7009E-12
SSA	49	16	19	43	2.7009E-12
BBO	53	26	15	51	2.3078E-11
CBO	53	13	20	34	2.3078E-11
NNA	49	16	19	43	2.7009E-12
GWO	49	19	16	43	2.7009E-12
WOA	43	19	16	49	2.7009E-12
ABC [61]	49	16	19	43	2.7009e-12
MBA [61]	43	16	19	49	2.7009e-12
GA [79]	49	16	19	43	2.7009E-12
CS [80]	43	16	19	49	2.7009E-12
ISA [81]	NA	NA	NA	NA	2.7009E-12
Kannan and Kramer [78]	33	15	13	41	2.1469E-8

**Fig. 16.** Cantilever beam design problem.**Table 15**

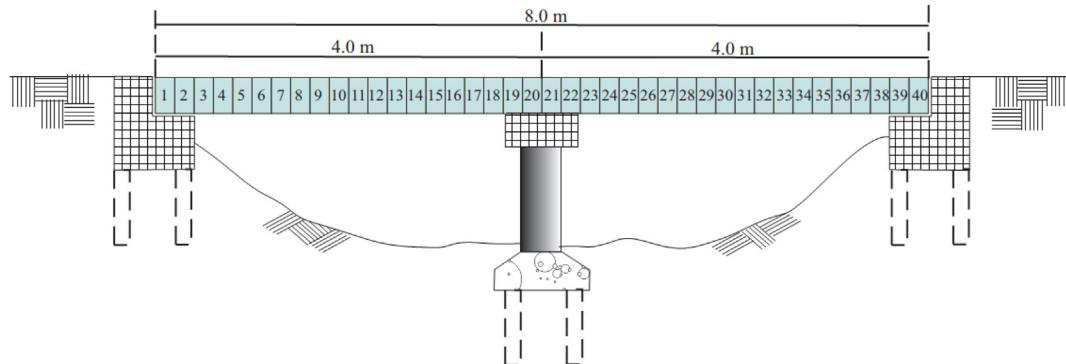
Comparison of statistical results obtained by various algorithms for cantilever beam design.

Algorithm	Best	Mean	Worst	SD
WSA	1.303252	1.303282	1.30333	0.0000146
GA	1.303294	1.30658	1.33795	0.006465
PSO	1.33559	2.930889	5.558154	1.033809
ICA	1.303361	1.303692	1.304437	0.0002296
MFO	1.30327	1.303538	1.304206	0.0001867
SCA	1.308902	1.34238	1.385756	0.0181865
SSA	1.303252	1.30326	1.30328	6.95E-06
BBO	1.313233	1.496408	1.949636	0.1307896
CBO	1.303259	1.303358	1.303695	0.0000962
NNA	1.303257	1.303282	1.303411	0.0000335
GWO	1.303256	1.303294	1.30343	0.0000365
WOA	1.319416	1.38278	1.542142	0.0537963

Table 16

The best designs for cantilever beam problem.

Algorithm	Optimal values for variables					Optimal cost
	x_1	x_2	x_3	x_4	x_5	
WSA	5.975921	4.880778	4.465427	3.477822	2.139194	1,303,252
GA	6.005599	4.851472	4.457484	3.482148	2.143109	1.303294
PSO	5.225227	5.386369	5.079555	3.381079	2.386473	1.33559
ICA	5.994105	4.904079	4.433163	3.497313	2.112234	1.303361
MFO	5.997765	4.874964	4.454762	3.483629	2.128316	1.30327
SCA	6.147025	4.652444	4.498136	3.624997	2.107317	1.308902
SSA	5.977432	4.874808	4.469891	3.478552	2.138456	1.303252
BBO	6.056668	5.261233	4.397537	3.416558	1.9675	1.313233
CBO	5.96968	4.878843	4.460389	3.486275	2.144072	1.303259
NNA	5.969628	4.88762	4.46466	3.477317	2.139987	1.303257
GWO	5.971406	4.881808	4.473385	3.476943	2.135661	1.303256
WOA	6.611188	4.889725	3.978585	3.637147	2.082194	1.319416
MMA [82]	6.01	5.3	4.49	2.15	2.15	1.34
GCA I [82]	6.01	5.3	4.49	2.15	2.15	1.34
GCA II [82]	6.01	5.3	4.49	2.15	2.15	1.33999
CS [80]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33996
SOS [83]	6.01878	5.30344	4.49587	3.49896	2.15564	1.34

**Fig. 17.** Schematic of the two-span bridge for damage detection.**Table 17**

Damage scenarios for the bridge structure.

	Damaged elements	Damage severity
Scenario #1	7, 20, 37	0.35, 0.05, 0.60
Scenario #2	2, 6, 8, 26, 32	0.45, 0.55, 0.05, 0.55, 0.05

Table 18

The mean of the results obtained by different algorithms.

Algorithms	Scenario #1		Scenario #2	
	Without noise	With noise	Without noise	With noise
WSA	2.09E-15	0.00016	8.79E-15	0.00014
GA	0.010701	0.002097	0.002541	0.002249
PSO	0.044475	0.040784	0.038388	0.038899
ICA	2.28E-06	0.000168	4.77E-06	0.000141
MFO	0.003849	0.000167	1.16E-05	0.001771
SCA	0.000924	0.007484	0.025267	0.033892
SSA	4.37E-06	0.000178	6.28E-05	0.000312

optimum designs of WSA are compared with those of the other implemented algorithms as well as reported results in the literature. The problems include classical constrained, unconstrained, continuous, and discrete design examples. Furthermore, the fifth example belongs to structural damage identification of a two-span bridge structure as a modern application in structural engineering. In all problems, the NFE is limited to 50,000 evaluations as the stopping criteria, the statistical

results are based on 50 independent trials, and the internal parameters are set as the previous section or according to the recommendations of their source paper. The constraints are implemented using penalty function, and for solving the discrete problem, the position of WS is rounded in each iteration [53]. The examples are clarified in the following sections.

4.1. Welded beam design problem

This benchmark design problem was proposed by Coello [54] and has been tackled by many researchers. As illustrated in Fig. 13, the beam is under a vertical force. The intention is to attain a design with the minimum manufacturing cost. The problem is subject to seven constraints of stress, deflection, welding, and geometry. The variables are weld thickness (h), height (l), length (t), and bar thickness (b) as Fig. 13. In the following, the formulation of this problem is stated.

$$\text{Consider } \vec{X} = [x_1, x_2, x_3, x_4] = [h, l, t, b]$$

$$\text{Minimize } f_{\text{cost}}(\vec{X}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

Subject to

$$g_1(\vec{X}) = \tau(\vec{X}) - \tau_{\max}$$

$$g_2(\vec{X}) = \sigma(\vec{X}) - \sigma_{\max}$$

$$g_3(\vec{X}) = x_1 - x_4 \leq 0$$

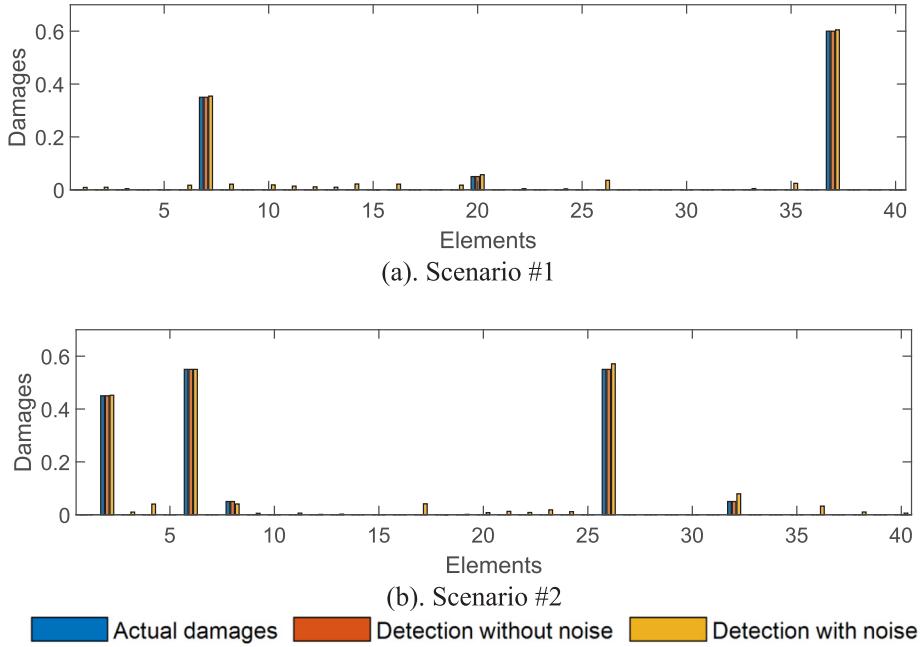


Fig. 18. The best detections obtained by WSA algorithm.

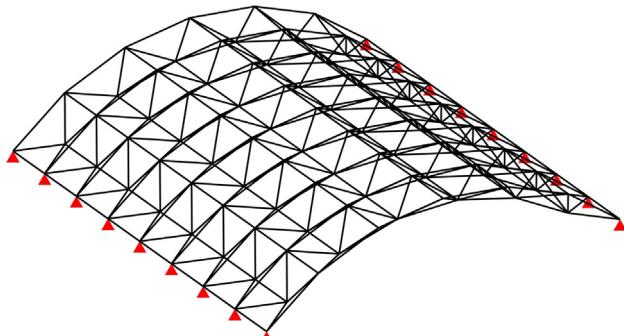


Fig. 19. Schematic of the 384-bar barrel vault.

$$g_4(\vec{X}) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0$$

$$g_5(\vec{X}) = 0.125 - x_1 \leq 0$$

$$g_6(\vec{X}) = \delta(\vec{X}) - \delta_{max}$$

$$g_7(\vec{X}) = P - P_c(\vec{X}) \leq 0$$

where

$$\tau(\vec{X}) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}\tau'' = \frac{MR}{J}$$

$$M = P\left(L + \frac{x_2}{2}\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$J = 2 \left\{ \sqrt{2}x_1x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2} \right)^2 \right] \right\}$$

$$\sigma(\vec{X}) = \frac{6PL}{x_4x_3^2}, \delta(\vec{X}) = \frac{4PL^3}{Ex_3^3x_4}$$

$$P_c(\vec{X}) = \frac{4.013E\sqrt{\frac{x_2^2x_4^6}{36}}}{L^2} \left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}} \right)$$

$$P = 6000lb, L = 14in, E = 30 \times 10^6psi, G = 12 \times 10^6psi$$

Variable range

$$0.1 \leq x_1, x_4 \leq 2$$

$$0.1 \leq x_2, x_3 \leq 10$$

The statistical results of the algorithms are provided in Table 9. As seen in this table, the present algorithm obtained the best results in terms of best, mean, worst and standard deviation (SD) of the solutions. The optimum result of WSA, as well as those of other methods, are listed in Table 10. As shown, the best design is discovered by WSA as well as MFO, NNA, CAEP, and HPSO. Since the optimum designs are the same, it sounds that this design is the global optimum of the problem.

4.2. Three-bar truss design problem

This benchmark structural engineering problem was firstly proposed by [73]. As demonstrated in Fig. 14, the truss structure does have three bar elements with symmetric configuration. The cross-section areas are considered as the design variables. The objective is to obtain the minimum weight of the structure while satisfying the stress constraints. The optimization problem is formulated in the following.

Consider $\vec{X} = [x_1, x_2] = [A_1, A_2]$

$$\text{Minimize } f(\vec{X}) = (2\sqrt{2}X_1 + X_2) \times 1$$

$$\text{Subject to } g_1(\vec{X}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0$$

$$g_1(\vec{X}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0$$

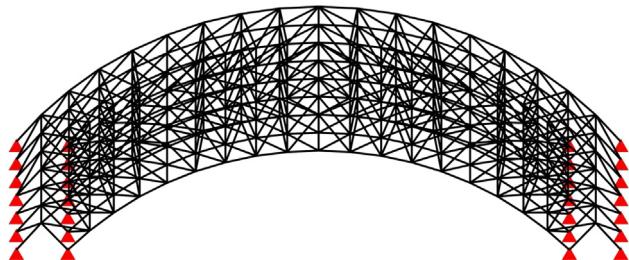
$$g_2(\vec{X}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0$$

$$g_3(\vec{X}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0$$

Table 19

Optimal design of 384-bar barrel vault.

Group number	Optimum sections								Kaveh and Moradveisi [87]		
	WSA	GA	PSO	ICA	MFO	SCA	SSA	CBO	ECBO	Engineering design	
1	S ½	S ¾	E 1	S ¾	S ¾	S 2 ½	E 4	S 1 ¼	S 1 ¼	S 1 ¼	
2	S 1	S 1	E 1 ½	E 4	S 1	D 2 ½	S 1	E 2	S 2	E 2	
3	S 1 ¼	S 1 ½	S 1 ½	S 3	S 1 ¼	D 4	S 1 ¼	S 1 ¼	S 1 ¼	S 1 ¼	
4	E 2 ½	E 2 ½	E 4	S 2 ½	D 3	E 4	E 10	S 3 ½	S 3 ½	D 2	
5	E 2	S 2 ½	E 4	E 1 ½	S 2	S 4	E 2	E 2	E 2	E 2	
6	S 2	S 4	D 2 ½	E 2	S 2	S 3	E 1 ½	E 2	S 2	E 2	
7	E 5	D 4	S 8	E 5	E 5	E 4	E 6	E 5	E 6	E 5	
8	E 5	D 3	S 6	S 6	E 6	D 3	D 5	S 8	S 4	E 5	
9	E 4	E 4	E 5	E 5	E 3	E 6	D 5	D 3	E 3	E 5	
10	E 3	S 3 ½	E 3 ½	S 3 ½	S 3	S 4	E 4	S 3 ½	S 3	S 5	
11	S 3	S 3 ½	E 5	S 3 ½	S 3 ½	D 4	S 3 ½	S 3 ½	S 3	S 5	
12	E 3 ½	S 5	E 4	S 4	S 3 ½	E 6	S 4	E 3 ½	S 4	S 5	
13	E 2	E 2	D 2	E 2	E 2	D 4	E 5	E 1 ½	E 1 ½	D 2	
14	E 1 ½	S 1 ¼	S 5	S 2 ½	E 1 ¼	E 2	D 2 ½	S 3 ½	S 2 ½	S 2 ½	
15	E 5	E 4	S 3 ½	S 6	E 4	D 3	E 6	S 4	S 4	E 3	
16	E 4	S 5	E 5	E 6	S 6	E 4	E 3	E 3 ½	S 5	D 2 ½	
17	E 4	D 2 ½	D 2 ½	S 3 ½	S 4	D 4	D 5	S 4	E 3 ½	S 5	
18	S ½	E ½	S ½	S 3	S ½	S 1 ½	S ½	S 1 ½	S 1 ½	E 1 ½	
19	S ¾	E ¾	S ½	S ½	E 2	S ¾	S ½	E 2	S 2	E 2	
20	S ½	S ½	S 3	S 1 ½	S ¾	E 2	E 8	E 2	S 2	E 2	
21	E ¾	S 1	E 5	S 1 ¼	S 1 ½	E 6	E 12	E 2	S 2	E 2	
22	S ¾	S ¾	E 6	E ¾	S ¾	E 1 ½	E 2 ½	E 2	S 2	S 2 ½	
23	S ½	S ½	E 3 ½	E 6	S ¾	S 1	D 3	E 2	S 1 ¼	S 1 ½	
24	S 1 ¼	S 1 ¼	E 2	S 1 ¼	S 1 ¼	E 4	S 2	E 1	S 1	E 1 ¼	
25	E 1	S 1 ¼	S 1 ½	S 2 ½	E 1 ¼	E 1 ¼	E 2	E 2	S 1 ½	E 2	
26	S 2	E 2	E 3 ½	S 2	E 1 ½	S 2 ½	S 4	S 1 ½	S 1 ½	E 1 ½	
27	S 2 ½	E 2	E 3	S 2 ½	E 2	E 4	S 3	E 2 ½	S 2	E 2	
28	S 1 ½	S 1 ¼	E 1 ¼	S 1 ½	E 1 ¼	S 1 ½	S 1 ¼	E 2	S 2	E 2	
29	S 2 ½	S 3	S 2 ½	S 3	S 2 ½	S 2 ½	S 2 ½	E 2	S 2	E 2 ½	
30	E 2	E 2	S 2 ½	S 2	S 2	D 2 ½	S 2 ½	S 2 ½	S 3	D 2	
31	E 1 ½	E 1 ½	E 1 ½	E 2	E 1 ½	E 1 ½	S 4	S 1 ½	E 2	S 2 ½	
Max stress ratio	0.9803	0.9959	0.8560	0.9760	0.9627	0.8926	1.0000	0.7176	0.9372	0.888	
Max displacement ratio	1.0000	0.9975	0.9932	1.0000	0.9980	0.9989	0.7646	0.9991	0.9996	0.9962	
Best weight (kg)	13150.08	13785.62	20390.87	165291.66	13165.50	25609.69	27700.36	14940.13	13345.92	16617.81	
Mean weight (kg)	17491.85	15979.16	23691.00	20139.83	14914.45	31369.38	35172.65	18602.01	15856.61	–	

**Fig. 20.** Schematic of the 910-barrel vault.

$$g_4(\vec{X}) = \frac{1}{\sqrt{2x_2 + x_1}}P - \sigma \leq 0$$

$$l = 100\text{cm}, P = 2\text{KN/cm}^3, \sigma = 2\text{KN/cm}^3$$

$$\text{Variable range } 0 \leq x_1, x_1 \leq 1.$$

Table 11, presents the statistical measurements of the algorithms. As seen, the WSA obtained the first rank among other algorithms in terms of the best, mean, worst, and standard deviation. The optimum designs are reported in **Table 12**. The best design provided by WSA, DEDS [74], and HHO [47] have the same weight, but their section areas are different.

4.3. Compound gear design problem

This example is a discrete design problem in mechanical

engineering [78]. The purpose of this benchmark task is to minimize the gear ratio defined as the ratio of the angular velocity of the output shaft to the angular velocity of the input shaft. As illustrated in **Fig. 15**, the number of teeth of the gears are considered as the discrete variables. This number of teeth must be between 12 and 60. The mathematical formulation is provided in the following.

Consider $\vec{X} = [x_1, x_2, x_3, x_4] = [n_A, n_B, n_C, n_D]$,

$$\text{Minimize}_f(\vec{X}) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4} \right)^2,$$

$$\text{Discrete variable range } 12 \leq x_1, x_2, x_3, x_4 \leq 60.$$

The statistical results are compared in **Table 13**. The proposed method outperformed other algorithms in terms of the mean and standard deviation of the final results. The WSA, GA, ICA, SCA, NNA, GWO and GOA could find the optimum number of teeth. The optimal designs are listed in **Table 14**, where four different permutations of variables are acquired as the optima. As one of the first or fourth variables must be 43 or 49 and the second or third variables must be 16 or 19.

4.4. Cantilever beam design problem

This problem is another structural engineering design example proposed in [82]. The purpose of this case is to optimize the material consumption of the rigidly supported beam in **Fig. 16**. The beam consists of five hollow square blocks with constant thickness, whose height

Table 20

Optimal design of 910-bar barrel vault.

Group number	Optimum sections							Kaveh and Moradveisi [87]		
	WSA	GA	PSO	ICA	MFO	SCA	SSA	CBO	ECBO	Engineering design
1	S 3 ½	E 2	S 3 ½	S 3	S 2 ½	S 3	E 2 ½	E 2	S 3	D 2
2	D 4	S 10	D 4	E 6	S 10	S 12	S 12	S 10	D 6	D 5
3	D 5	D 4	D 5	D 5	S 8	S 12	D 5	S 10	S 10	S 8
4	E 4	S 6	E 4	S 6	S 8	E 3 ½	D 2 ½	D 2 ½	S 8	S 8
5	S 5	E 4	S 5	S 4	S 2 ½	E 4	E 2 ½	S 3 ½	S 2	D 2
6	S 1 ¼	S ¾	S 1 ¼	S ¾	S 1	E 2 ½	E 1 ½	E ¾	S 1 ½	E ¾
7	E 2 ½	E 3	E 2 ½	S 3	E 3	E 2	S 1 ½	E 3	E 1 ½	S 4
8	E 5	E 5	E 5	S 5	D 2 ½	E 3 ½	D 2	D 3	E 3	S 8
9	E 8	S 10	E 8	D 5	D 5	E 6	D 5	D 5	D 4	S 10
10	E 10	S 12	E 10	S 10	S 12	E 12	S 12	D 6	E 12	S 12
11	S 1	S ½	S 1	S ½	S ½	S 2	E 2	S 1 ¼	S 1	S 1
12	S 1	S ½	S 1	S ¾	S ½	D 2	S ½	S 1 ¼	S 1	S 1
13	E 1 ¼	S 1	E 1 ¼	S 1	S 1	E 2	S 1 ¼	S 1 ¼	S 1 ¼	E 1
14	S ¾	S ½	S ¾	S ½	S ½	S 1	S ¾	S 1	S 1	S 1
15	S ½	S ½	S ½	S ½	S ½	S ¾	S ¾	S 1	S 1	E 2
16	E 1 ½	E 2	E 1 ½	E 2	S 2	S 1 ½	E 1 ¼	S 3	E 2	S 1
17	E 1 ¼	E 1	E 1 ¼	S 1 ½	E 1 ¼	E 2	S 1 ½	E 1 ½	E 2	E 1 ½
18	S 3	S 2 ½	S 3	E 2	E 1 ½	S 3 ½	S 2 ½	E 1	E 1 ½	E 1 ½
19	D 2	S 2 ½	D 2	E 2 ½	S 3	S 3	E 2	E 3	E 3	S 1 ¼
20	S 3	S 3	S 3	S 4	E 2 ½	D 3	S 4	S 2 ½	E 2	E 2
21	E 2	S 3	E 2	S 2 ½	S 2 ½	D 4	S 4	S 2 ½	E 2	E 3 ½
22	S 4	S 4	S 4	E 3 ½	S 2 ½	D 2 ½	E 2	S 2 ½	E 2	D 2 ½
23	E 5	E 5	E 5	S 10	E 6	E 6	E 12	E 4	D 3	D 5
24	D 5	E 5	D 5	E 5	S 6	D 8	D 5	S 8	E 5	E 5
25	S ½	S ½	S ½	S ½	E ½	S 1 ½	S ½	S ¾	S ¾	S ¾
26	S ½	S ½	S ½	S ½	S ½	S ¾	E 1 ¼	S ¾	S ¾	S ¾
27	S ½	S ½	S ½	S ½	S ½	S 3	S ½	S ½	S ¾	S ¾
28	S ½	S ½	S ½	S ½	S ½	S ½	E 6	S ¾	E ¾	S ¾
29	S ½	S ½	S ½	S ½	S ½	S ½	S 2 ½	S ¾	S ¾	E 2
30	E 1 ½	S 1 ½	E 1 ½	S 3	E 1 ¼	S ¾	E 2	E 1 ½	E 1	
Max stress ratio	0.9973	0.9967	0.7454	0.9913	0.8815	0.7569	0.9135	0.9767	0.9818	0.95
Max displacement ratio	0.9998	0.9994	0.9881	1.0000	0.9993	0.9704	0.9999	0.9990	0.9978	0.9993
Best weight (kg)	17734.57	18665.46	20639.27	19261.74	17867.98	27932.00	23477.23	18635.67	18615.32	19894.44
Mean weight (kg)	18613.55	23266.09	21880.78	22626.02	20285.43	31590.65	30881.73	23806.75	22442.64	–

is the decision variables. As provided below, using the classical beam theory, the deflection constraint makes the objective function a non-convex problem.

Consider $\vec{X} = [x_1, x_2, x_3, x_4, x_5] = [h_1, h_2, h_3, h_4, h_5]$,

$$\text{Minimize } f(\vec{X}) = 0.06224(x_1 + x_2 + x_3 + x_4 + x_5),$$

$$\text{subject to } g(\vec{X}) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0,$$

Variable range $0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100$.

The comparison of the statistical results of the present algorithm, as well as other methods, are reported in Table 15. As seen, while the WSA obtained the optimum design but it is in the second place considering the mean, worst and standard deviation of the results. It should be mentioned that both the mean and worst result of WSA are less than 0.01% heavier than those of SSA. The optimum design obtained by different algorithms is reported in Table 16. The table shows that the best design found by WSA and SSA consists of different heights for hollow sections.

4.5. Application of WSA in structural health monitoring

The existing structures are prone to different actions such as earthquakes, corrosion, thermal effects, or overloads, which can lead to structural damages, as well as economic or life losses. To ensure safety and structural health, the probable damages should be identified and repaired as soon as they occur. The process of identification of damages

is known as Structural Health Monitoring (SHM). The damage identification of a structure can be defined as an inverse optimization problem [84]. In this section, the WSA is applied to detect the damage scenarios of a two-span steel bridge shown in Fig. 17. Two damage scenarios are implemented, and the noise effects are considered according to Table 17 [85].

The damages are defined as a reduction in member's modulus of elasticity as Eq. (6)

$$E_e^d = (1 - d_e)E_e, (0 \leq d_e \leq 1) \quad (6)$$

d_e is the damage variable where 0 represents the intact element, and 1 represents the fully-damaged situation; and E_e stands for modulus of elasticity.

The location and severity of the damages can be detected by minimizing the following function [86].

Consider $\vec{X} = [x_1, x_2, \dots, x_{ne}] = [d_1, d_2, \dots, d_{ne}]$,

Minimize $f(\vec{X})$

$$= \sum_{i=1}^n \left[W_f \left| \frac{f_i^{(e)} - f_i^{(a)}}{f_i^{(e)}} \right|^2 \right] + \sum_{i=1}^n \left[W_\phi \left(1 - \frac{|\phi_i^{(e)T} \phi_i^{(a)}|}{(\phi_i^{(e)T} \phi_i^{(e)}) (\phi_i^{(a)T} \phi_i^{(a)})} \right) \right]$$

With variable range $0 \leq x_1, x_2, \dots, x_{nm} \leq 1$.

where n stands for the number of considered modal properties; ne is the number of candidate damage elements assumed as 40; f_i and ϕ_i are the i th frequency in Hz and normalized modal shape vector, respectively; W_f and W_ϕ are the weight factors assigned to the frequency and modal shape terms which are assumed as 0.1 and 1, respectively; (e) and

(a) superscripts denote the existing and analytic quantities in the inverse problem, and $| \cdot |$ stands for the absolute value. The number of mode shapes and frequencies is limited to 5 and the data are polluted with 1% noise. Data pollution is implemented according to Eq. (7)

$$\bar{\text{input}} = \text{input} \times (1 + \sigma \times \text{rand}) \quad (7)$$

where input denotes the component of the mode shapes and frequencies; σ is the noise ratio that is assumed as 1%, and rand is a uniform random number between 0 and 1.

The frequency and mode shapes are calculated using the eigenvalue and eigenvector analyses, respectively. Further details about damage detection inverse problems can be found in [85].

The comparison among the examined optimization algorithms is provided in Table 18. As seen, the WSA attained the best mean for both with and without noise scenarios. SSA for the noise-free case of the first scenario and the ICA for the other cases are placed in the second rank. Moreover, the noise influenced the results and increased the mean values of all algorithms. The optimum results of the proposed WSA, as well as the actual damages, are depicted in Fig. 18. As shown, the WSA accurately predicted the damages in noiseless cases, but in the with noise cases, some elements are wrongly detected with negligible damages. It should be noticed that the sensitivity of the optimization problem to polluted input parameters caused the errors. The results indicate the capability of WSA to detect damages in the existing buildings and bridges.

4.6. Application of WSA in optimal design of double-layer barrel vaults

In this section, two large-scale barrel vaults are designed and optimized using different algorithms. Also, the engineering design obtained by fully stressed design approach is included to show the efficiency of the methods. The steel pipe sections with standard weight (S), extra strong (E) and double extra strong (D) taken from AISC-LRFD are considered as the design sections. In the following, the structures are briefly explained and the results are provided. Further detailed information about their design constraints and geometry can be found in [87].

4.6.1. The 384-Bar double-layer barrel vault

The first structure has 384 bars divided into 31 discrete variables [87]. As shown in Fig. 19, in this structure two top and bottom layers (chord members) which are connected with braced members. The concentrated downward loads of 10 kips (44.482 kN) are applied at both top and bottom nodes and the 6 kips (26.689 kN) loads are exerted on top-layer joints.

The acquired designs by various algorithms are provided in Table 19. The numbers written in front of sections denote the diameter of pipe sections (in inch). As shown, the best designs of WSA, GA, ICA, MFO, CBO and ECBO algorithms are lighter than that of the engineering design obtained by SAP2000 software, while the optimum design of WSA is the best design among all optimizers. Its best weight is more than 20% lighter than engineering design. However, the MFO algorithm has the lightest designs averagely. As reported in Table 19, the best designs which are optimized by WSA and MFO do not use any double extra strong (D) section. As can be seen, using the penalty approach explained in [87], the final designs satisfied both stress and displacement constraints.

4.6.2. The 910-bar double-layer braced barrel vault

As shown in Fig. 20, this structure consists of two circular top and bottom layers with a total of 910 members that are categorized into 31 groups [87]. At the nodes of central arc, the arcs adjacent to the center, external arcs, and those of arcs adjacent to the external arcs, 15 kips (66.72 kN), 10 kips (44.48 kN), 5 kips (22.24 kN) and 2 kips (8.90 kN) downward loads are exerted, respectively.

The results of optimizations are reported in Table 20 which shows

that WSA outperformed the other algorithms in terms of best and mean weight. Furthermore, WSA, MFO and PSO respectively have the best average weights that are lighter than those reported in [87]. Like the previous example, in this problem, none of the optimum designs does violate the constraints demanded by design codes. The WSA's best design is almost 11% lighter than that of engineering design.

5. Discussion of the results

In the previous section, several mathematical and engineering problems were employed in examining the proposed algorithm. In this section, all the results are summarized to provide a tidy insight into the performance of the WSA.

Table 4 reports the results for unimodal functions. The good results of WSA for these problems shows that it can exploit the search region and probe the local optimums effectively. Metaheuristics usually control exploiting mechanisms by improving promising solutions. In WSA this operation is supplied by attraction and food-finding steps.

Tables 5 and 6 indicate the competitive capability of the WSA in exploring the spaces with multimodal topography with several local optimums and finding the near-global optimum. The exploration behavior relates to discovering new promising solutions and avoiding getting trapped in local optima. The death and succession operators remove the probably trapped solutions and prob the new regions.

The cost histories depicted in Fig. 6 shows the high convergence rate of the WSA. Since the algorithm always initializes with a random solution, this proves that it can find low-cost solutions rapidly independent from the starting point.

Section 3.4 experimentally investigates the algorithm for possible biases in measurements and internal mechanisms. In this section, different complex mathematical functions are successfully optimized and nonparametric tests are done. In all problem sets, except one function (f_{19}), the WSA stands in the first three ranks that shows the mentioned biases do not exist in it.

Moreover, in Section 4, the algorithm is applied to four classical engineering problems, two large scale structural design and a damage detection problem with different damage scenarios. In all of the classic problems, the WSA acquired the optimal cost. In the damage detection, WSA identified the damage of bridge structure with the best accuracy in comparison to other compared algorithms even when the input data is polluted with noise. Besides, the WSA obtained the optimum designs of the barrel vault structures with light weights, so that the best weights are at least 11% lighter than those of engineering full stressed design.

6. Conclusion and future directions

In this paper, a novel nature-inspired SI optimizer is proposed. The algorithm inspired by the life cycle of water striders and is called WSA. The WSA mimics the territorial behavior, mating manner, ripple communication, foraging and succession of water striders. We put an attempt to utilize mathematically simple formulations for implementing the technique. The exploratory, exploitative, local optima avoidance, convergence, and other features of the proposed WSA were investigated using 44 unimodal, multimodal, composite, shifted, and biased functions. Moreover, the WSA is successfully applied to four classical constrained, unconstrained, continuous, and discrete engineering design problems, two structural optimizations of double-layer barrel vaults and a challenging bridge damage detection problem. The results showed that the proposed WSA is an efficient yet simple algorithm for solving optimization problems.

Although the WSA showed an acceptable performance in the tested problems it has some limitations. In the mating step, the attraction probability can be converted into a dynamic or adaptive form instead of being static. Because, the search individuals should have different interactions (attraction/struggling) with other water striders regarding their fitness, distance and the portion of the remained cycles. The steps

size can also be adjusted because of the same mentioned reasons. As another issue regarding this algorithm, its population must be a factor of the number of territories. Moreover, as mentioned in Section 2.3, the number of evaluations is not a deterministic function of the internal parameters, hence it can be potential difficulty in programming languages that do not support global variable for controlling the total number of function evaluations.

For future work, we are going to develop different versions of WSA, such as multi-objective and adaptive versions of this algorithm, and optimize large-scale real-world problems. It is possible to utilize different chaotic maps, constraint handling strategies or hybrid versions for enhancement of the present algorithm.

Compliance with ethical standards

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.

References

- [1] Talbi E-G. Metaheuristics: from design to implementation 74. John Wiley & Sons; 2009.
- [2] Kaveh A. Advances in metaheuristic algorithms for optimal design of structures. Springer; 2014.
- [3] Kalemci EN, İkizler SB, Dede T, Angin Z. Design of reinforced concrete cantilever retaining wall using Grey wolf optimization algorithm. *Structures* 2020;23:245–53. <https://doi.org/10.1016/j.istruc.2019.09.013>.
- [4] Kaveh A, Mahjoubi S. Optimum design of double-layer barrel vaults by lion pride optimization algorithm and a comparative study. *Structures* 2018;13:213–29. <https://doi.org/10.1016/j.istruc.2018.01.002>.
- [5] Hoseini Vaez SR, Shahmoradi Qomi H. Bar layout and weight optimization of special RC shear wall. *Structures* 2018;14:153–63. <https://doi.org/10.1016/j.istruc.2018.03.005>.
- [6] Ha M-H, Vu Q-A, Truong V-H. Optimum design of stay cables of steel cable-stayed bridges using nonlinear inelastic analysis and genetic algorithm. *Structures* 2018;16:288–302. <https://doi.org/10.1016/j.istruc.2018.10.007>.
- [7] Goldberg DE. Genetic algorithms. Pearson Education India 2006.
- [8] Eberhart R., Kennedy J. Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks 1995, pp. 1942–1948. Citeseer.
- [9] Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 2017;114:163–91. <https://doi.org/10.1016/j.advengsoft.2017.07.002>.
- [10] Atashpaz-Gargari, E., Lucas, C.: Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In: 2007 IEEE Congress on Evolutionary Computation, 25–28 Sept. 2007; 2007, pp. 4661–4667.
- [11] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. *IEEE Comput Intell Mag* 2006;1(4):28–39. <https://doi.org/10.1109/MCI.2006.329691>.
- [12] Yang, X.-S.: Firefly Algorithms for Multimodal Optimization. Berlin, Heidelberg. Stochastic Algorithms: Foundations and Applications; 2009: Springer, Berlin Heidelberg. pp. 169–178.
- [13] Karaboga D, Akay B. A comparative study of Artificial Bee Colony algorithm. *Appl Math Comput* 2009;214(1):108–32. <https://doi.org/10.1016/j.amc.2009.03.090>.
- [14] Yu JJQ, Li VOK. A social spider algorithm for global optimization. *Appl Soft Comput* 2015;30:614–27. <https://doi.org/10.1016/j.asoc.2015.02.014>.
- [15] Mirjalili SZ, Mirjalili S, Saremi S, Faris H, Aljarah I. Grasshopper optimization algorithm for multi-objective optimization problems. *Appl Intell* 2018;48(4):805–20. <https://doi.org/10.1007/s10439-017-1019-8>.
- [16] Pan W-T. A new Fruit Fly Optimization Algorithm: taking the financial distress model as an example. *Knowl-Based Syst* 2012;26:69–74. <https://doi.org/10.1016/j.knosys.2011.07.001>.
- [17] Mirjalili S. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl-Based Syst* 2015;89:228–49. <https://doi.org/10.1016/j.knosys.2015.07.006>.
- [18] Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Appl* 2016;27(4):1053–73. <https://doi.org/10.1007/s00521-015-1920-1>.
- [19] Kaveh, A., Bakhshpoori, T. Metaheuristics: Outlines, MATLAB Codes and Examples. Springer; 2019.
- [20] Mirjalili S, Mirjalili SM, Hatamlou A. Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Comput Appl* 2016;27(2):495–513.
- [21] Kaveh A, Dadras A. A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Adv Eng Softw* 2017;110:69–84.
- [22] Kaveh A, Mahdavi VR. Colliding bodies optimization: A novel meta-heuristic method. *Comput Struct* 2014;139:18–27. <https://doi.org/10.1016/j.compstruc.2014.04.005>.
- [23] Mirjalili S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl-Based Syst* 2016;96:120–33. <https://doi.org/10.1016/j.knosys.2015.12.022>.
- [24] Simon D. Biogeography-based optimization. *IEEE Trans Evol Comput* 2008;12(6):702–13.
- [25] Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1997;1(1):67–82.
- [26] Andersen EA. The semiaquatic bugs (Hemiptera, Gerromorpha): phylogeny, adaptations, biogeography and classification. BRILL; 1982.
- [27] Zhang X, Zhao J, Zhu Q, Chen N, Zhang M, Pan Q. Bioinspired aquatic microrobot capable of walking on water surface like a water strider. *ACS Appl Mater Interfaces* 2011;3(7):2630–6.
- [28] Wei PJ, Chen SC, Lin JF. Adhesion forces and contact angles of water strider legs. *Langmuir* 2008;25(3):1526–8.
- [29] Hu DL, Bush JW. The hydrodynamics of water-walking arthropods. *J Fluid Mech* 2010;644:5–33.
- [30] Koga T, Hayashi K. Territorial behavior of both sexes in the water strider *Metrocoris histrio* (Hemiptera: Gerridae) during the mating season. *J Insect Behav* 1993;6(1):65–77.
- [31] Thornhill R, Alcock J. The evolution of insect mating systems. Harvard University Press; 1983.
- [32] Sih A, Watters JV. The mix matters: behavioural types and group dynamics in water striders. *Behaviour* 2005;142(9):1423.
- [33] Wilcox RS. Ripple communication in aquatic and semiaquatic insects. *Ecoscience* 1995;2(2):109–15.
- [34] Spence JR, Wilcox RS. The mating system of two hybridizing species of water striders (Gerridae). *Behav Ecol Sociobiol* 1986;19(2):87–95.
- [35] Spence JR, Anderson N. Biology of water striders: interactions between systematics and ecology. *Annu Rev Entomol* 1994;39(1):101–28.
- [36] Watson PJ, Stallmann RR, Arnqvist G. Sexual conflict and the energetic costs of mating and mate choice in water striders. *Am Nat* 1998;151(1):46–58.
- [37] Han CS, Jablonski PG. Female genitalia concealment promotes intimate male courtship in a water strider. *PLoS ONE* 2009;4(6). e5793.
- [38] Williams DD, Feltmate BW. Aquatic insects. Cab International; 1992.
- [39] Nummelin M. Cannibalism in waterstriders (Heteroptera: Gerridae): is there kin recognition? *Oikos* 1989;87–90.
- [40] Robert T. Parental investment and sexual selection. *Sexual Selection & the Descent of Man*, Aldine de Gruyter, New York, p. 136–179; 1972.
- [41] Wilcox RS, Ruckdeschel T. Food threshold territoriality in a water strider (*Gerris remigis*). *Behav Ecol Sociobiol* 1982;11(2):85–90.
- [42] Rubenstein DI. Resource acquisition and alternative mating strategies in water striders. *Am Zool* 1984;24(2):345–53.
- [43] Andersen NM, Cheng L. The marine insect Halobates (Heteroptera: Gerridae): biology, adaptations, distribution, and phylogeny. *Oceanogr Mar Biol Annu Rev* 2004;42:119–80.
- [44] Denny MW. Air and water: the biology and physics of life's media. Princeton University Press; 1993.
- [45] Lipowski A, Lipowska D. Roulette-wheel selection via stochastic acceptance. *Physica A* 2012;391(6):2193–6.
- [46] Sadollah A, Sayyadi H, Yadav A. A dynamic metaheuristic optimization model inspired by biological nervous systems: neural network algorithm. *Appl Soft Comput* 2018;71:747–82. <https://doi.org/10.1016/j.asoc.2018.07.039>.
- [47] Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: algorithm and applications. *Future Gener Comput Syst* 2019;97:849–72. <https://doi.org/10.1016/j.future.2019.02.028>.
- [48] 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. Last accessed: July; 2010.
- [49] Yao X, Liu Y, Lin G. Evolutionary programming made faster. *IEEE Trans Evol Comput* 1999;3(2):82–102.
- [50] Digalakis JG, Margaritis KG. On benchmarking functions for genetic algorithms. *Int J Comput Mathemat* 2001;77(4):481–506.
- [51] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Adv Eng Softw* 2014;69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>.
- [52] Mirjalili S, Lewis A. The whale optimization algorithm. *Adv Eng Softw* 2016;95:51–67.
- [53] Kaveh A, Dadras A, Bakhshpoori T. Improved thermal exchange optimization algorithm for optimal design of skeletal structures. *Smart Struct Syst* 2018;21(3):263–78.
- [54] Coello CAC. Use of a self-adaptive penalty approach for engineering optimization problems. *Comput Ind* 2000;41(2):113–27.
- [55] Coello CAC, Montes EM. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Adv Eng Inf* 2002;16(3):193–203.
- [56] Coello Coello CA, Becerra RL. Efficient evolutionary optimization through the use of a cultural algorithm. *Eng Optim* 2004;36(2):219–36.
- [57] He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng Appl Artif Intell* 2007;20(1):89–99.
- [58] He Q, Wang L. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Appl Math Comput* 2007;186(2):1407–22.
- [59] Zahara E, Kao Y-T. Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems. *Expert Syst Appl* 2009;36(2):3880–6.
- [60] Coello Coello CA. Constraint-handling using an evolutionary multiobjective optimization technique. *Civil Eng Syst* 2000;17(4):319–46.
- [61] Sadollah A, Bahreininejad A, Eskandar H, Hamdi M. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Appl Soft Comput* 2013;13(5):2592–612.
- [62] Parsopoulos, K.E., Vrahatis, M.N. Unified particle swarm optimization for solving constrained engineering optimization problems. In: International conference on

- natural computation; 2005, pp. 582–591. Springer.
- [63] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. *Inf Sci* 2009;179(13):2232–48.
- [64] Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm. *Comput Struct* 2004;82(9–10):781–98.
- [65] Mezura-Montes E, Coello CAC. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Trans Evol Comput* 2005;9(1):1–17.
- [66] Huang F-Z, Wang L, He Q. An effective co-evolutionary differential evolution for constrained optimization. *Appl Math Comput* 2007;186(1):340–56.
- [67] Deb K. An efficient constraint handling method for genetic algorithms. *Comput Methods Appl Mech Eng* 2000;186(2):311–38. [https://doi.org/10.1016/S0045-7825\(99\)00389-8](https://doi.org/10.1016/S0045-7825(99)00389-8).
- [68] Deb K. Optimal design of a welded beam via genetic algorithms. *AIAA J* 1991;29(11):2013–5.
- [69] Coello Coello CA. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comput Methods Appl Mech Eng* 2002;191(11):1245–87. [https://doi.org/10.1016/S0045-7825\(01\)00323-1](https://doi.org/10.1016/S0045-7825(01)00323-1).
- [70] Coello Coello CA, Mezura Montes E. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Adv Eng Inf* 2002;16(3):193–203. [https://doi.org/10.1016/S1474-0346\(02\)00011-3](https://doi.org/10.1016/S1474-0346(02)00011-3).
- [71] Siddall JN. Analytical decision-making in engineering design. Prentice Hall; 1972.
- [72] Ragsdell, K., Phillips, D. Optimal design of a class of welded structures using geometric programming; 1976.
- [73] Ray T, Saini P. Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng Optim* 2001;33(6):735–48. <https://doi.org/10.1080/03052150108940941>.
- [74] Zhang M, Luo W, Wang X. Differential evolution with dynamic stochastic selection for constrained optimization. *Inf Sci* 2008;178(15):3043–74. <https://doi.org/10.1016/j.ins.2008.02.014>.
- [75] Saremi S, Mirjalili S, Lewis A. Grasshopper Optimisation Algorithm: Theory and application. *Adv Eng Softw* 2017;105:30–47. <https://doi.org/10.1016/j.advengsoft.2017.01.004>.
- [76] Tsai J-F. Global optimization of nonlinear fractional programming problems in engineering design. *Eng Optim* 2005;37(4):399–409.
- [77] Arora JS. Introduction to optimum design. Elsevier; 2004.
- [78] Kannan BK, Kramer SN. An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *J Mech Des* 1994;116(2):405–11. <https://doi.org/10.1115/1.2919393>.
- [79] Deb K, Goyal M. A combined genetic adaptive search (GeneAS) for engineering design. *Comput Sci Inform* 1996;26:30–45.
- [80] Gandomi AH, Yang X-S, Alavi AH. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng Comput* 2013;29(1):17–35. <https://doi.org/10.1007/s00366-011-0241-y>.
- [81] Gandomi AH. Interior search algorithm (ISA): a novel approach for global optimization. *ISA Trans* 2014;53(4):1168–83.
- [82] Chickermane H, Gea H. Structural optimization using a new local approximation method. *Int J Numer Meth Eng* 1996;39(5):829–46.
- [83] Cheng M-Y, Prayogo D. Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput Struct* 2014;139:98–112. <https://doi.org/10.1016/j.compstruc.2014.03.007>.
- [84] Gerist S, Maher MR. Structural damage detection using imperialist competitive algorithm and damage function. *Appl Soft Comput* 2019;77:1–23. <https://doi.org/10.1016/j.asoc.2018.12.032>.
- [85] Kaveh A, Dadras A. Structural damage identification using an enhanced thermal exchange optimization algorithm. *Eng Optim* 2018;50(3):430–51. <https://doi.org/10.1080/0305215X.2017.1318872>.
- [86] Du D-C, Vinh H-H, Trung V-D, Hong Quyen N-T, Trung N-T. Efficiency of Jaya algorithm for solving the optimization-based structural damage identification problem based on a hybrid objective function. *Eng Optim* 2018;50(8):1233–51. <https://doi.org/10.1080/0305215X.2017.1367392>.
- [87] Kaveh A, Moradveisi M. Optimal design of double-layer barrel vaults using CBO and ECBO algorithms. *Iran J Sci Technol Trans Civil Eng* 2016;40(3):167–78. <https://doi.org/10.1007/s40996-016-0021-4>.