Applied Mathematics and Computation 389 (2021) 125535

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean

Jui-Sheng Chou*, Dinh-Nhat Truong

Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

ARTICLE INFO

Article history:
Received 5 February 2020
Revised 22 May 2020
Accepted 12 July 2020
Available online 7 August 2020

Keywords:
Design of metaheuristic algorithm
Bio-inspired swarm intelligence
Jellyfish search optimizer
Numerical computation
Benchmark functions
Engineering design

ABSTRACT

This study develops a novel metaheuristic algorithm that is motivated by the behavior of jellyfish in the ocean and is called artificial Jellyfish Search (JS) optimizer. The simulation of the search behavior of jellyfish involves their following the ocean current, their motions inside a jellyfish swarm (active motions and passive motions), a time control mechanism for switching among these movements, and their convergences into jellyfish bloom. JS optimizer is tested using a comprehensive set of mathematical benchmark functions and applied to a series of structural engineering problems. Fifty small/average-scale and twenty-five large-scale functions involving various dimensions were used to validate JS optimizer, which was compared with ten well-known metaheuristic algorithms. JS optimizer was found to outperform those algorithms in solving mathematical benchmark functions. The JS algorithm was then used to solve structural optimization problems, including 25-bar tower design, 52-bar tower design and 582-bar tower design problems. In those cases, JS not only performed best but also required the fewest evaluations of objective functions. Therefore, JS is potentially an excellent metaheuristic algorithm for solving optimization problems.

1. Introduction

Meta-heuristic optimization algorithms are becoming more popular for solving complex problems in various domains for the following reasons. (i) They rely on simple concepts and are easy to implement; (ii) they do not require information about the gradient of the objective function; (iii) they can bypass local minima; and (iv) they can be utilized to solve a wide range of problems in various fields [1,2]. Since the application of metaheuristics depends on computers, advances in the processing power of computers had accelerated the development of metaheuristics [3].

Exploration (diversification) and exploitation (intensification) are the two main phases of a metaheuristic algorithm. The main differences among metaheuristics are in the particular ways in which they balance those two processes [4-6]. The distinction between them is a main concept in relation to metaheuristic algorithms, such as single-solution-based metaheuristics or population-based metaheuristics, and is often taken to be fundamental. Basic single-solution-based metaheuristics are more exploitation-oriented than exploration-oriented, whereas population-based metaheuristics are more exploration-oriented than exploitation-oriented [7].

* Corresponding author.
E-mail addresses: jschou@mail.ntust.edu.tw (J.-S. Chou), D10605806@mail.ntust.edu.tw (D.-N. Truong).

https://doi.org/10.1016/j.amc.2020.125535
0096-3003/

Nomenclature

JS	Jellyfish Search
TSA	Tree-Seed Algorithm
ABC	Artificial Bee Colony
DE	Differential Evolution
FA	Firefly Algorithm
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
TLBO	Teaching-Learning-Based Optimization
WOA	Whale Optimization Algorithm
GSA	Gravitational Search Algorithm
SOS	Symbiotic Organisms Search
ES	Evolution Strategy
BBO	Biogeography-Based Optimizer
BBBC	Big-Bang Big-Crunch
CFO	Central Force Optimization
SA	Simulated Annealing
ACO	Ant Colony Optimization
GSO	Group Search Optimizer
FBI	Forensic-Based Investigation
HS	Harmony Search
WSO	Wasp Swarm Optimization
$\overrightarrow{\text{trend}}$	Direction of the ocean current
n_{Pop}	Number of jellyfish
Max_{iter}	Maximum number of iterations
f(X)	Objective function
X^*	Jellyfish currently with the best location in the swarm
e_c	Factor that governs the attraction
μ	Mean location of all jellyfish
df	Difference between the location of currently best jellyfish and the mean location of all jellyfish
α	Significance level of Wilcoxon rank sum test
β	Distribution coefficient
γ	Motion coefficient
U_b	Upper bound of search spaces
L_b	Lower bound of search spaces
$X_i(t)$	Location of i^{th} jellyfish at time t
X_i	Logistic chaotic value of location of the i^{th} jellyfish.
$\overrightarrow{\text{Direction}}$	Direction of motion in a swarm.
c(t)	Time control function
C_o	Constant of time control function
CEC2005	IEEE CEC-2005 special session on real-parameter optimization
W(A)	Weight of the structure
A	Vector of the sizing variables
γ_i	Material density of member i
A_i	Cross-sectional area of member i
F_p	Penalty function
λ_i	Slenderness ratio of i^{th} member
l_i	Length of the i^{th} member
r_i	Radius of gyration of the i^{th} member
E	Modulus of elasticity
F_y	Yield stress

Properly balancing the two phases is challenging owing to the stochastic nature of metaheuristics algorithms [4,7]. This work develops a new algorithm that is inspired by the social movement of jellyfish in the ocean, including their following the ocean current and their motion inside the swarm.

The rest of the paper is organized as follows. Section 2 reviews the most relevant literature on metaheuristic optimization. Section 3 outlines the proposed artificial JS optimizer, and explains its implementation in detail. Section 4 describes

the corresponding mathematical functions to verify the efficiency of the proposed algorithm. In Section 5, three structural optimization problems are solved using JS. The final section draws conclusions.

2. Literature review

Hussain surveyed 1,222 publications on metaheuristics from 1983 to 2016 (33 years), in which, birds, humans, plants, water, the ecosystem, electromagnetic forces, and gravitation were used as metaphors in metaheuristic methods [8]. These methods fall into two main categories. The first consists of those that mimic biological or physical phenomena, and can be grouped into three sub-categories (Fig.1) of evolution-based, physics-based, and swarm-based methods [9]. The second comprises those that are inspired by human phenomena [10].

Evolution-based methods are based on the law of natural selection. They begin with the initialization of a random population in a search space, and then subsequent generations evolve. Such methods identify the best individuals in a generation and combine them to form the next generation of individuals. Therefore, the population improves over generations. Genetic algorithms (GAs), which simulate Darwinian evolution, are the most popular evolution-inspired methods [11]. In a GA, each solution is represented as a chromosome. In every generation, chromosomes with higher fitness values are more likely to be crossed with other chromosomes. Therefore, the overall fitness of all chromosomes increases over the generations. Other algorithms in this class include Differential Evolution (DE) [12], the Evolution Strategy (ES) [13], and the Biogeography-Based Optimizer (BBO) [14].

The second group comprises physics-based methods, and involved simulations in which physical rules are applied. For example, the Simulated Annealing (SA) algorithm is inspired by annealing in metallurgy, which involves the heating and controlled cooling of a material to increase the size of its crystals and reduce their defect density [15,16]. Big-Bang Big-Crunch (BBBC) relies on the two major theories of the evolution of the universe to which its name refers [17]. The Gravitational Search Algorithm (GSA) is based on the law of gravity and mass interactions [18]. Other well-known algorithms include the Gravitational Local Search Algorithm (GLSA) [19], Charged System Search (CSS) [20], Central Force Optimization (CFO) [21], the Small-World Optimization Algorithm (SWOA) [22], and the Galaxy-based Search Algorithm (GbSA) [23].

The third group of nature-inspired methods comprises swarm-based techniques, which simulate the behavior of animal groups. Of this category, Particle Swarm Optimization (PSO), developed by Kennedy and Eberhart [24] is the most popular. PSO is based on the properties of moving particles (candidate solutions). Particles move in a search space, ultimately finding the best solution (optimal position). The path of each particle is affected by its own best position yet obtained and the best position so far obtained by the swarm.

Another popular algorithm is Ant Colony Optimization (ACO), which was originally proposed by Dorigo et al. [25]. This algorithm is inspired by the behavior of ants in an ant colony. Ants deposit their pheromones along favorable paths that should be followed by other members of the colony. Over time (iterations), more ants follow the better paths and reject the worse paths, until the best path is followed by most ants. Other swarm-based techniques include the Marriage in Honey Bees Optimization Algorithm (MBO) [26], the Artificial Fish-Swarm Algorithm (AFSA) [27], the Termite Algorithm [27], the Artificial Bee Colony (ABC) [28,29], the Wasp Swarm Optimization (WSO) [30].

Apart from nature-inspired algorithms are another set of meta-heuristic methods that are inspired by human behaviors. The most popular algorithm in this group is Teaching Learning-Based Optimization (TLBO), which is based on processes that are involved in teaching and learning, in which the teacher influences the output of students in a class, and the results of the students are influenced by both teaching and self-learning [31]. Harmony Search (HS) [32,33] is a method for finding solutions to optimization problems by analogy with the musical process of searching for a perfect state of harmony. Forensic-Based Investigation (FBI) is inspired by the suspect investigation–location–pursuit process of police officers engaged in criminal investigations [34]. Other popular algorithms include Tabu (Taboo) Search (TS) [35], the Group Search Optimizer (GSO) [36], and the League Championship Algorithm (LCA) [37].

Many of the aforementioned metaheuristic optimization algorithms are inspired by nature and humans. To the best of our knowledge, however, no algorithm in the literature mimics the search behavior of jellyfish in the ocean. This fact motivates the present attempt to model mathematically a society of jellyfish; to develop a metaheuristic optimization algorithm that is inspired by jellyfish, called artificial Jellyfish Search (JS) optimizer. In the newly proposed JS algorithm, both exploration and exploitation are considered. At the beginning, jellyfish sense the ocean current to explore for planktonic food. Over time, jellyfish's movements switch to passive and active motions inside swarms for exploitation.

3. Artificial jellyfish search optimizer

This section describes the inspiration for, and the mathematical simulation that is involved in, the artificial jellyfish search algorithm.

3.1. Inspiration

Jellyfish live in the water of various depths and temperatures around the world. They are shaped like bells; some are smaller than a centimeter in diameter while others are very large [38]. They have wide varieties of colors, sizes, and shapes. All of the many specifies exhibit particular adaptations to the oceanic environment. Their methods of feeding vary: some

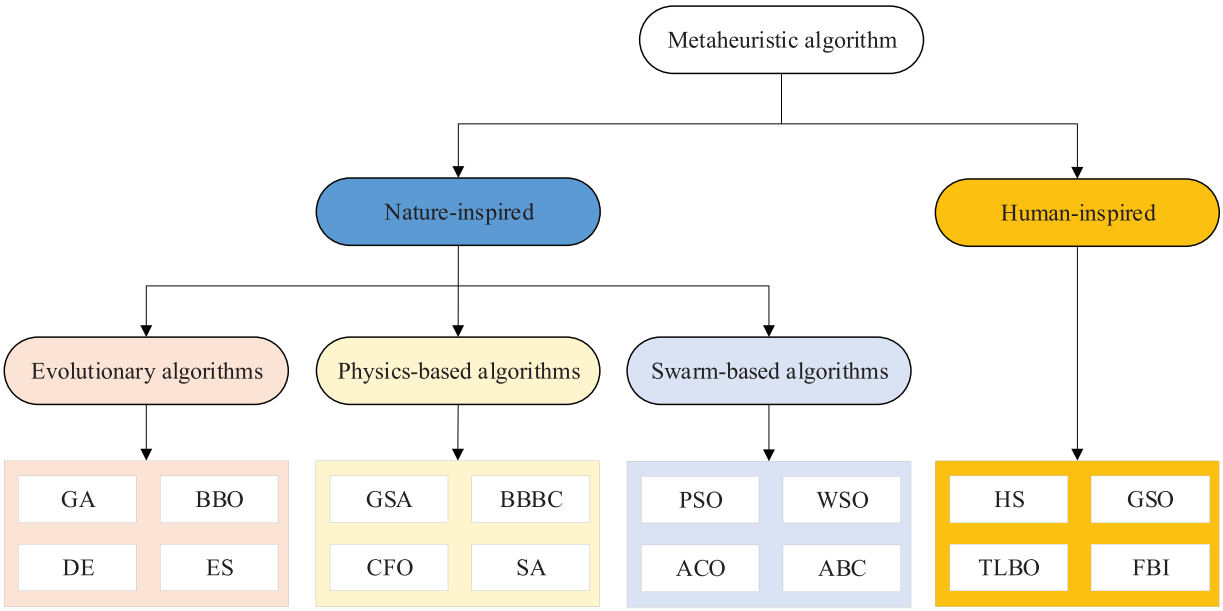


Fig. 1. Classification of meta-heuristic algorithms.

jellyfish use tentacles to bring food to their mouths while others use filter-feeding, so they eat whatever the current brings them. Other jellyfish actively hunt prey and immobilize them by stinging them with their tentacles [25,38].

Jellyfish use their tentacles to sting their prey, and then release a venom that paralyzes it. They do not attack creatures, but those who swim up against or touch them may be stung to death. Some jellyfish stings are painful, but not lethal. A typical sting can cause pain, red marks, itching, numbness, or tingling. However, stings from some specifies of jellyfish, such as the box jellyfish (also called sea wasp), are very dangerous and may even be deadly. Most such jellyfish found in the coastal waters of Australia, the coastal waters of the Philippines, the Indian Ocean, and the central Pacific Ocean. Jellyfish are most dangerous when gathered together in a jellyfish bloom [39-41].

Jellyfish have features that enable them to control their movements. Their underside closes like an umbrella, pushing water out to propel their body forward. Despite this ability, they mostly drift in the water, depending on currents and tides [42]. When conditions are favorable, jellyfish can form a swarm, and a large mass of jellyfish is called a Jellyfish bloom [43,44]. In particular, jellyfish are weak swimming organisms and their orientations with respect to currents is key to the maintenance of blooms and ensuring that they do not become stranded [43].

Numerous factors govern the formation of swarm, including ocean currents, available nutrients, oxygen availability, predation, and temperature. Among these factors, ocean currents are the most important as they can collect jellyfish into a swarm [43,45,46]. Apparently, jellyfish can live in areas of relatively high salinity and low oxygen concentration, so they can thrive on plankton without having to compete with other specifies. Saltier waters contain more iodine, which favors jellyfish polyp production. A rising sea temperature favors the formation of swarms because jellyfish can survive better than other sea animals under such conditions [45]. Briefly, the impact of the ecosystem on jellyfish swarms is significant [47,48] and current-oriented swimming is essential to their maintenance [43].

Ocean currents have a huge range of scales. For example, in the open ocean, currents may move around small submesoscale features; they may have a size of only a few hundred meters around mesoscale features, or a few tens of kilometers. They may flow across or around entire ocean basins, as do the Gulf Stream (North Atlantic), Kuroshio Current (North Pacific) and Agulhas Current (Indian Ocean). The primary sources of energy for these currents are solar heating and wind over the ocean surface [42,49].

This phenomenon, along with each jellyfish's own movements inside the swarm and following ocean current to form jellyfish bloom, has given these species the ability to appear almost everywhere in the ocean. The quantity of food at sites that are visited by a jellyfish varies; thus, when food proportions are compared, the best location would be identified. Therefore, a new algorithm that is inspired by search behavior and movement of jellyfish in the ocean is developed herein. It is named jellyfish search (JS) optimizer. Fig. 2 presents the steps of the algorithm. In the next subsection, the behavior and movements of jellyfish in the ocean are mathematically modeled, and an optimization algorithm that is based on the mathematical model is then developed.

3.2. Mathematical model for the optimization algorithm

The proposed optimization algorithm is based on three idealized rules.

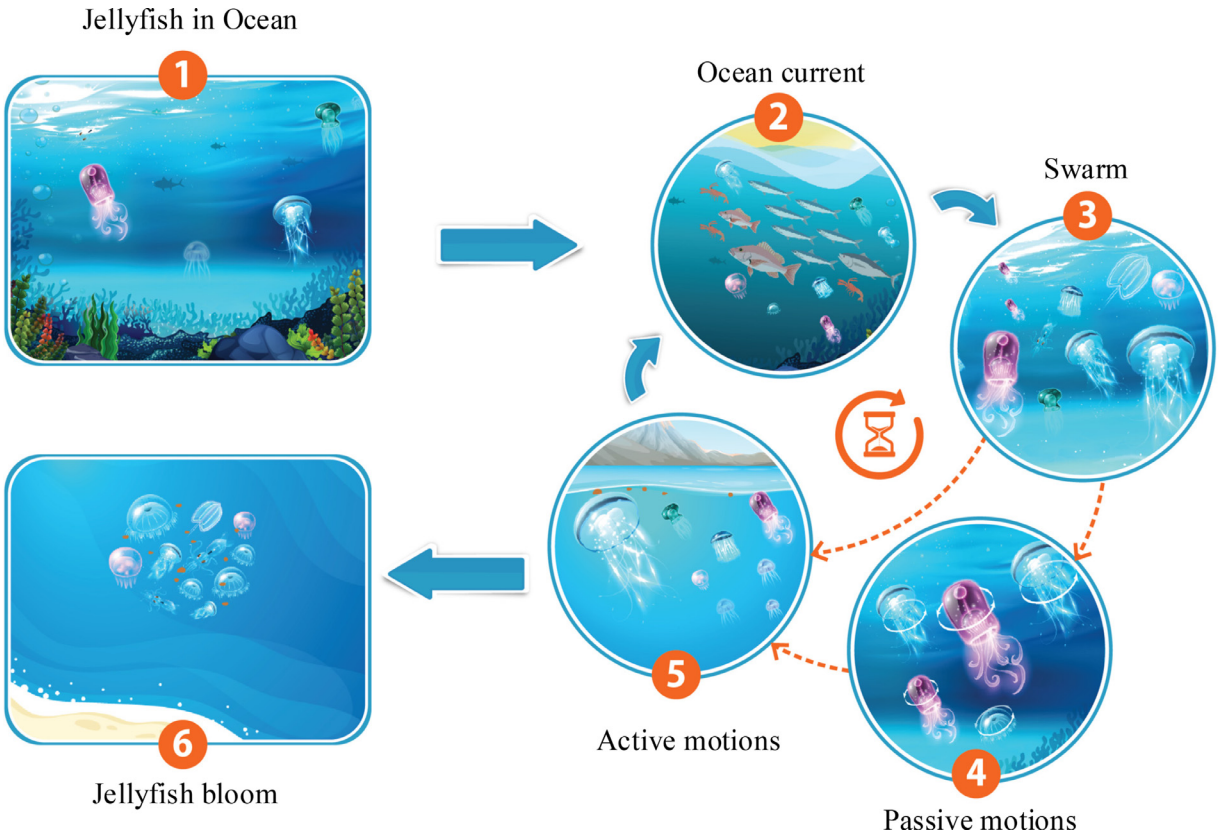


Fig. 2. Jellyfish's behavior in the ocean.

a. Jellyfish either follow the ocean current or move inside the swarm, and a "time control mechanism" governs the switching between these types of movement.
b. Jellyfish move in the ocean in search of food. They are more attracted to locations where the available quantity of food is greater.
c. The quantity of food found is determined by the location and its corresponding objective function.

3.2.1. Ocean current

The ocean current contains large amounts of nutrients, so the jellyfish are attracted to it. The direction of the ocean current ($\overrightarrow{\text{trend}}$) is determined by averaging all the vectors from each jellyfish in the ocean to jellyfish that is currently in the best location, as shown in Fig. 3a. Eq. (1) simulates the ocean current.

$$\overrightarrow{\text{trend}} = \frac{1}{n_{Pop}} \sum \overrightarrow{\text{trend}_i} = \frac{1}{n_{Pop}} \sum (X^* - e_c X_i) = X^* - e_c \frac{\sum X_i}{n_{Pop}} = X^* - e_c \mu \tag{1}$$

$$\text{Set } df = e_c \mu \tag{2}$$

Therefore, $\overrightarrow{\text{trend}}$ is determined by:

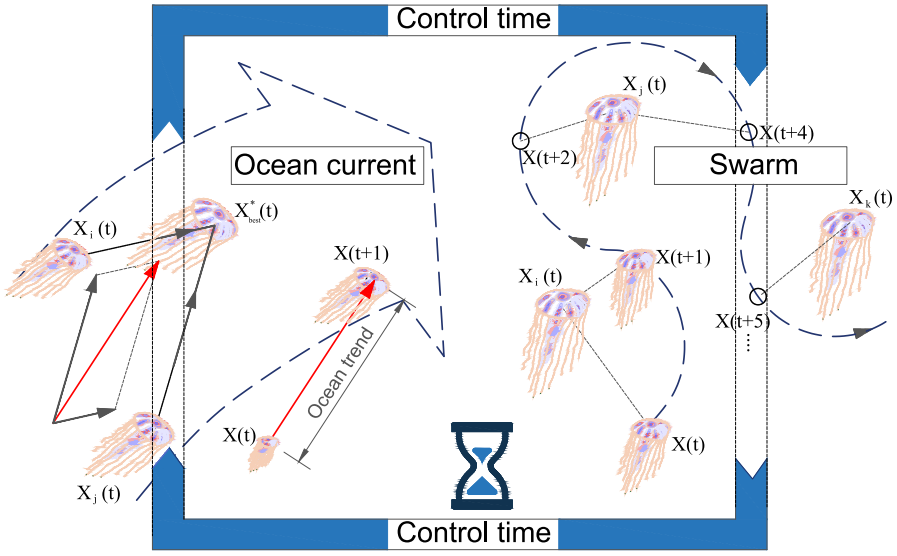
$$\overrightarrow{\text{trend}} = X^* - df \tag{3}$$

where n_{Pop} is the number of jellyfish; X^* is the jellyfish currently with the best location in the swarm; e_c is the factor that governs the attraction; μ is the mean location of all jellyfish; df is the difference between the current best location of the jellyfish and the mean location of all jellyfish.

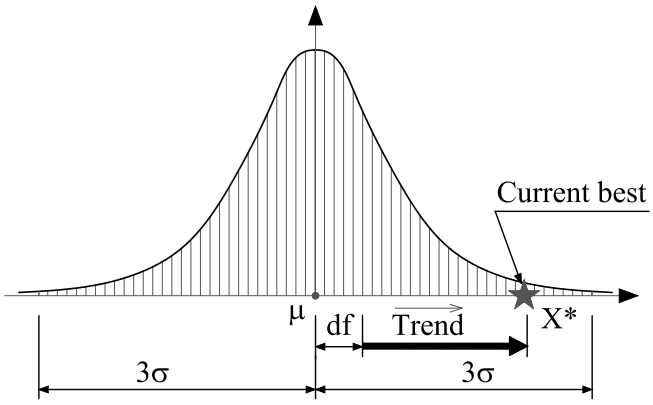
Based on the assumption of a normal spatial distribution of jellyfish in all dimensions, a distance of $\pm \beta\sigma$ around the mean location contains certain likelihood of all jellyfish, where σ is the standard deviation of the distribution (Fig. 3b). Therefore,

$$df = \beta \times \sigma \times \text{rand}^f(0, 1) \tag{4}$$

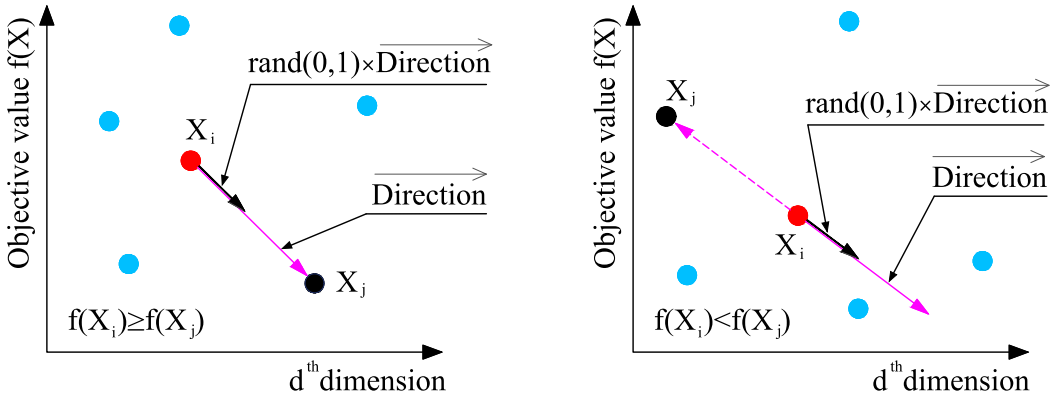
$$\text{Set } \sigma = \text{rand}^\alpha(0, 1) \times \mu \tag{5}$$



a) Simulation of ocean current, swarm and time control.



b) Jellyfish distribution in ocean.



c) Direction of movement in a swarm.

Fig. 3. Simulation of jellyfish behavior.

$$\text{Hence, } df = \beta \times rand^{f}(0,1) \times rand^{\alpha}(0,1) \times \mu \tag{6}$$

To simplify, Eq. (6) is rewritten as

$$df = \beta \times rand(0,1) \times \mu \tag{7}$$

$$\text{where } e_c = \beta \times rand(0,1) \tag{8}$$

Thus,

$$\overrightarrow{trend} = X^* - \beta \times rand(0,1) \times \mu \tag{9}$$

Now, the new location of each jellyfish is given by

$$X_i(t+1) = X_i(t) + rand(0,1) \times \overrightarrow{trend} \tag{10}$$

Eq. (10) can be given by

$$X_i(t+1) = X_i(t) + rand(0,1) \times (X^* - \beta \times rand(0,1) \times \mu \tag{11}$$

where $\beta > 0$ is a distribution coefficient, related to the length of $\overrightarrow{trend}$. Based on the results of sensitivity analysis in numerical experiments (ref. Section 4.5), $\beta = 3$ is obtained.

3.2.2. Jellyfish swarm

In swarm, jellyfish are passive (type A) and active (type B) motions, respectively [44,50]. Initially, when the swarm has just been formed, most jellyfish exhibit type A motion. Over time, they increasingly exhibit type B motion.

Type A motion is the motion of jellyfish around their own locations and the corresponding updated location of each jellyfish is given by.

$$X_i(t+1) = X_i(t) + \gamma \times rand(0,1) \times (U_b - L_b) \tag{12}$$

where U_b and L_b are the upper bound and lower bound of search spaces, respectively; $\gamma > 0$ is a motion coefficient, related to the length of motion around jellyfish's locations. Based on the results of sensitivity analysis in numerical experiments (ref. Section 4.5), $\gamma = 0.1$ is obtained.

To simulate type B motion, a jellyfish (j) other than the one of interest is selected at random and a vector from the jellyfish of interest (i) to the selected jellyfish (j) is used to determine the direction of movement. When the quantity of food at the location of the selected jellyfish (j) exceeds that at the location of the jellyfish (i) of interest, the latter moves toward the former; it moves directly away from it if the quantity of food available to the selected jellyfish (j) is lower than that available to the jellyfish of interest (i). So, each jellyfish moves toward the better direction to find the food in a swarm. Eqs. (15) and (16) simulate the direction of motion and the updated location of a jellyfish, respectively, whose movement is depicted in Fig. 3c. This movement is considered as an effective exploitation of the local search space [51].

$$\overrightarrow{Step} = X_i(t+1) - X_i(t) \tag{13}$$

$$\text{where } \overrightarrow{Step} = rand(0,1) \times \overrightarrow{Direction} \tag{14}$$

$$\overrightarrow{Direction} = \begin{cases} X_j(t) - X_i(t) \text{ if } f(X_i) \geq f\left(X_j\right) \\ X_i(t) - X_j(t) \text{ if } f(X_i) < f\left(X_j\right) \end{cases} \tag{15}$$

where f is an objective function of location X

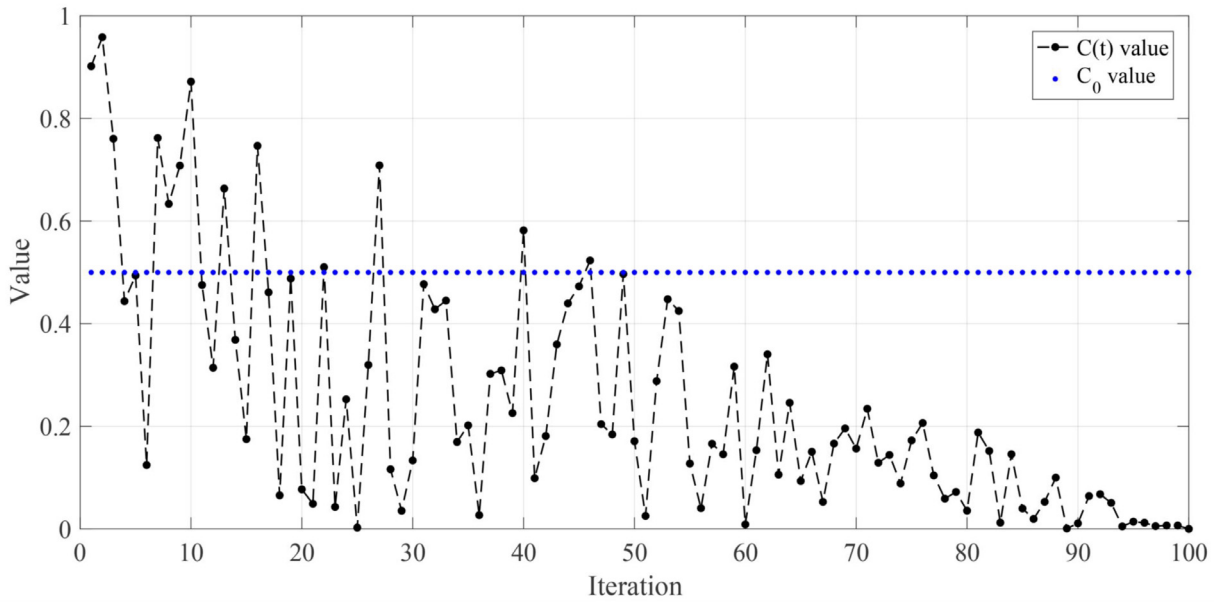
$$\text{Hence, } X_i(t+1) = X_i(t) + \overrightarrow{Step} \tag{16}$$

To determine the type of motion over time, a time control mechanism is used. It controls not only type A and type B motions in a swarm but also the movements of jellyfish toward an ocean current. The next subsection presents the time control mechanism in detail.

3.2.3. Time control mechanism

The ocean current contains large amounts of nutritious food so jellyfish are attracted by it [42]. Over time, more jellyfish gather together and swarm is formed. As the temperature or wind changes the ocean current, the jellyfish in the swarm move toward another ocean current and another jellyfish swarm is formed. The movements of jellyfish inside a jellyfish swarm are type A (passive motions) and type B (active motions), between which the jellyfish switch. Type A is favored in the beginning; as time goes by, type B is preferred.

The time control mechanism is introduced to simulate this situation. To regulate the movement of jellyfish between following the ocean current and moving inside the jellyfish swarm, the time control mechanism includes a time control function c(t) and a constant C_o. The time control function is a random value that fluctuates from 0 to 1 over time.



a) Time control function.

```
Begin
  For i=1: nPop do
    Calculate the time control c(t) using Eq. (17)
    If c(t)≥0.5:
      Jellyfish follows ocean current
    Else: Jellyfish moves inside swarm
      If rand(0,1)>(1-c(t)):
        Jellyfish exhibits type A motion (Passive motions)
      Else:
        Jellyfish exhibits type B motion (Active motions)
      End if
    End if
  End for
End
```

b) Pseudocode of time control mechanism.

Fig. 4. Simulation of time control mechanism.

Eq. (17) formulates the time control function and Fig. 4a plots the time control function over time. When its value exceeds C_o, the jellyfish follow the ocean current. When its value is less than C_o, they move inside the swarm. An exact C_o value is not known and the time control varies randomly from zero to one. Hence, C_o is set to 0.5, which is the mean of zero and one.

$$c(t) = \left|\left(1 - \frac{t}{Max_{iter}}\right) \times (2 \times rand(0,1) - 1)\right| \tag{17}$$

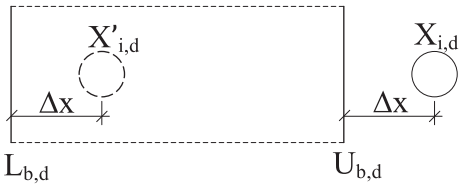


Fig. 5. Re-entering method.

where t is the time specified as the iteration number and Max_{iter} is the maximum number of iterations, which is an initialized parameter.

Fig. 4b presents pseudo-code of time control mechanism. Similar to c(t), the function $(1-c(t))$ is used to simulate the movement inside a swarm (type A or B). When rand(0, 1) exceeds $(1-c(t))$, the jellyfish exhibits type A motion. When rand(0, 1) is lower than $(1-c(t))$, jellyfish exhibits type B motion. Since $(1-c(t))$ increases from zero to one over time, the probability that rand(0, 1) $>$ $(1-c(t))$ initially exceeds the probability that $(1-c(t))$ $>$ rand(0, 1). Therefore, type A motion is preferred type B. As time goes by, $(1-c(t))$ approaches one, and the probability that $(1-c(t))$ $>$ rand(0, 1) ultimately exceeds rand(0, 1) $>$ $(1-c(t))$. Thus, type B motion is favored.

3.3. Population initialization

The population of jellyfish is normally initialized with random. The disadvantages of this approach are its slow convergence and its tendency to become trapped at local optima as a result of low population diversity. To improve the diversity of the initial population, many chaotic maps have been developed, including the logistic map, tent map, and Liebovitch map [52-54]. May developed the logistic map, which is one of the simplest chaotic maps [55]. This map provides more diverse initial populations than does random selection and it provides a lower probability of premature convergence [53,56]. Therefore, this map is used in this study (ref. Section 4.4):

$$X_{i+1} = \eta X_i (1 - X_i), \quad 0 \le X_0 \le 1 \tag{18}$$

X_i is the logistic chaotic value of location of the i^{th} jellyfish; X_0 is used for generating initial population of jellyfish, $X_0 \in (0, 1)$, $X_0 \notin \{0.0, 0.25, 0.75, 0.5, 1.0\}$, and parameter η is set to 4.0.

3.4. Boundary conditions

Oceans are located around the world. The earth is approximately spherical, so when a jellyfish moves outside the bounded search area, it will return to the opposite bound. Fig. 5 and Eq. (19) present this re-entering process.

$$\begin{cases} X'_{i,d} = \left(X_{i,d} - U_{b,d}\right) + L_b(d) \text{ if } X_{i,d} > U_{b,d} \\ X'_{i,d} = \left(X_{i,d} - L_{b,d}\right) + U_b(d) \text{ if } X_{i,d} < L_{b,d} \end{cases} \tag{19}$$

$X_{i,d}$ is the location of the i^{th} jellyfish in d^{th} dimension; $X'_{i,d}$ is the updated location after checking boundary constraints. $U_{b,d}$ and $L_{b,d}$ are upper bound and lower of d^{th} dimension in search spaces, respectively.

3.5. Schematic representation of artificial JS optimizer

The two main phases of a metaheuristic algorithm are exploration and exploitation [57]. In the artificial JS optimizer, movement toward an ocean current is exploration and movements within a jellyfish swarm is the exploitation, and a time control mechanism switches between them. Initially, the probability of exploration exceeds that of exploitation to find areas that contain promising optimal positions; over time, the probability of exploitation becomes much higher than that of exploration, and the jellyfish identify the best location inside the identified areas. Figs. 6 and 7 present the pseudocode and flowchart of the artificial JS optimizer, respectively.

4. Optimization of benchmark mathematical functions

Examples of mathematical optimization are presented to evaluate the artificial JS using small/average- and large-scale functions. Fig. 8 plots 3D view of selected mathematical functions. First, ten well-known metaheuristic algorithms (WOA [59], TSA [7], SOS [60], TLBO [31], FA [58], GSA [18], ABC [28], DE [12], PSO [24] and GA [11]) are compared with artificial JS optimizer using small/average scale functions. They are subsequently analyzed using the large-scale functions. All of the algorithms may yield optima that vary among runs owing to their stochastic characteristics, so 30 independent runs were performed for each benchmark function to eliminate any stochastic discrepancy [53]. To simplify the presentation, optimized costs that are less than 1E-12 are assumed to be zero [53,61].

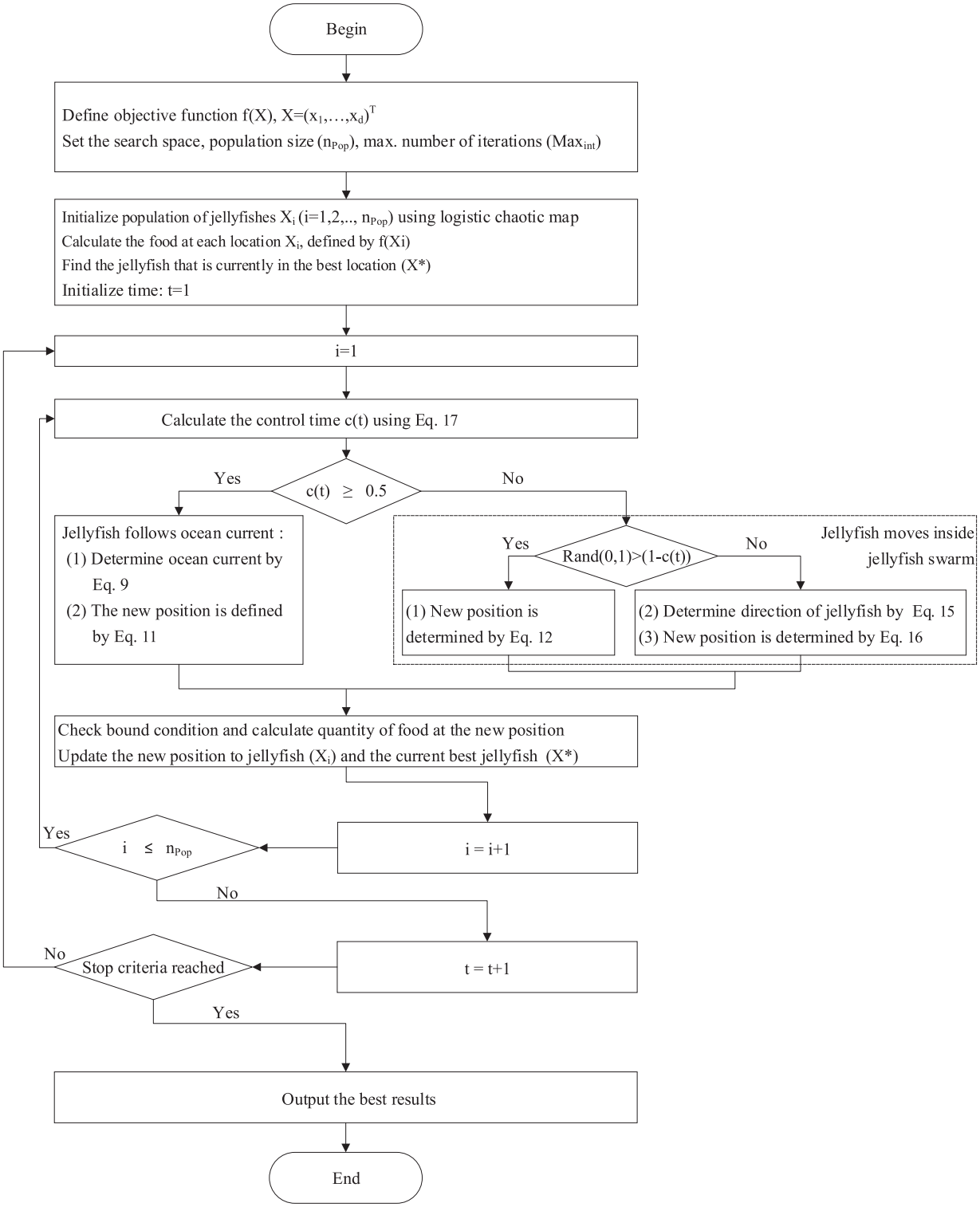


Fig. 6. Schematic flowchart of JS algorithm.

Begin

Define objective function f(X), $X=(x_1,\ldots,x_d)^T$

Set the search space, population size (n_{Pop}), and maximum iteration (Max_{int})

Initialize population of jellyfish X_i (i=1,2,.., n_{Pop}) using logistic chaotic map

Calculate quantity of food at each X_i, $f(X_i)$

Find the jellyfish at location currently with most food (X)*

Initialize time: t=1

Repeat

For i=1: n_{Pop} do

Calculate the time control c(t) using Eq. (17)

If *c(t)≥0.5: Jellyfish follows ocean current*

(1) Determine ocean current using Eq. (9)

(2) New location of jellyfish is defined by Eq. (11)

Else: *Jellyfish moves inside a swarm*

If *rand(0,1)>(1-c(t)): jellyfish exhibits type A motion (Passive motions)*

(1) New location of jellyfish is defined by Eq. (12)

Else: *Jellyfish exhibits type B motion (Active motions)*

(2) Determine direction of jellyfish using Eq. (15)

(3) New location of jellyfish is defined by Eq. (16)

End if

End if

Check boundary conditions and calculate quantity of food at new location

Update the location of jellyfish (X_i) and location of jellyfish currently with the most food (X)*

End for i

Update the time: t=t+1

Until *stop criterion is met (e.g., t> (Max_{int}))*

Output the best results and visualization (Jellyfish bloom)

End

Fig. 7. Pseudocode of JS optimizer algorithm.

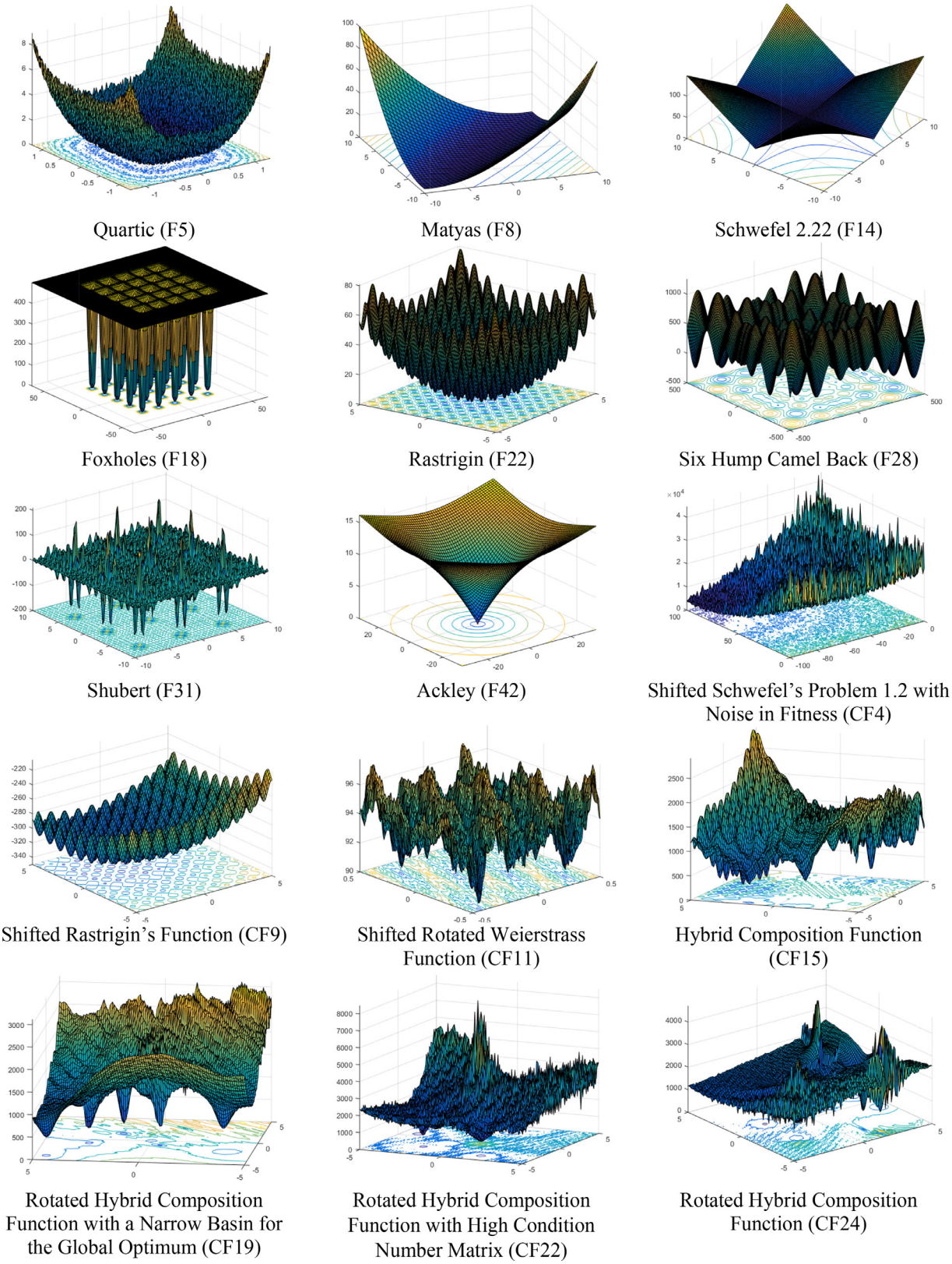


Fig. 8. 3D view of selected mathematical functions.

Table 1
Description of small/average-scale mathematical optimization problems.

Function	D	Opt.	Range	C
(F1) Stepint	5	0	[−5.12, 5.12]	US
(F2) Step	30	0	[−100, 100]	US
(F3) Sphere	30	0	[−100, 100]	US
(F4) SumSquares	30	0	[−10, 10]	US
(F5) Quartic	30	0	[−1.28, 1.28]	US
(F6) Beale	2	0	[−4.5, 4.5]	UN
(F7) Easom	2	−1	[−100, 100]	UN
(F8) Matyas	2	0	[−10, 10]	UN
(F9) Colville	4	0	[−10, 10]	UN
(F10) Trid6	6	−50	[$-D^2$, D^2]	UN
(F11) Trid10	10	−210	[$-D^2$, D^2]	UN
(F12) Zakharov	10	0	[−5, 10]	UN
(F13) Powell	24	0	[−4, 5]	UN
(F14) Schwefel 2.22	30	0	[−10, 10]	UN
(F15) Schwefel 1.2	30	0	[−100, 100]	UN
(F16) Rosenbrock	30	0	[−30, 30]	UN
(F17) Dixon-Price	30	0	[−10, 10]	UN
(F18) Foxholes	2	0.998	[−65.536, 65.536]	MS
(F19) Branin	2	0.398	[−5, 10]×[0, 15]	MS
(F20) Bohachevsky1	2	0	[−100, 100]	MS
(F21) Booth	2	0	[−10, 10]	MS
(F22) Rastrigin	30	0	[−5.12, 5.12]	MS
(F23) Schwefel	30	−12,569.5	[−500, 500]	MS
(F24) Michalewicz2	2	−1.8013	[0, π]	MS
(F25) Michalewicz5	5	−4.6877	[0, π]	MS
(F26) Michalewicz 10	10	−9.6602	[0, π]	MS
(F27) Schaffer	2	0	[−100, 100]	MN
(F28) Six Hump Camel Back	2	−1.03163	[−5, 5]	MN
(F29) Bohachevsky2	2	0	[−100, 100]	MN
(F30) Bohachevsky3	2	0	[−100, 100]	MN
(F31) Shubert	2	−186.73	[−10, 10]	MN
(F32) GoldStein-Price	2	3	[−2, 2]	MN
(F33) Kowalik	4	0.00031	[−5, 5]	MN
(F34) Shekel5	4	−10.15	[0, 10]	MN
(F35) Shekel7	4	−10.4	[0, 10]	MN
(F36) Shekel10	4	−10.53	[0, 10]	MN
(F37) Perm	4	0	[−D, D]	MN
(F38) Powersum	4	0	[0, 1]	MN
(F39) Hartman3	3	−3.86	[0, D]	MN
(F40) Hartman6	6	−3.32	[0, 1]	MN
(F41) Griewank	30	0	[−600, 600]	MN
(F42) Ackley	30	0	[−32, 32]	MN
(F43) Penalized	30	0	[−50, 50]	MN
(F44) Penalized2	30	0	[−50, 50]	MN
(F45) Langermann2	2	−1.08	[0, 10]	MN
(F46) Langermann5	5	−1.5	[0, 10]	MN
(F47) Langermann 10	10	NA	[0, 10]	MN
(F48) Fletcher Powell2	2	0	[$-\pi$, π]	MN
(F49) Fletcher Powell5	5	0	[$-\pi$, π]	MN
(F50) FletcherPowell10	10	0	[$-\pi$, π]	MN

Nomenclature: characteristics (C), dimension (D), optimal value (Opt.); separable (S), non-separable (N), multimodal (M) and unimodal (U) functions.

4.1. Small/Average-scale functions

In this experiment, 50 mathematical functions (F1-F50) that are taken from the literature [61] are used to evaluate JS. Examples include separable, non-separable, multimodal and unimodal functions, with two to 30 dimensions. Table 1 presents those functions in detail. Fig. 9, for example, presents the convergence of the Rastrigin function in two dimensions with a population size of 10 and 50 iterations.

4.2. Large-scale functions (CEC2005)

In this experiment, 25 functions with 10, 30 and 50 dimensions (taken from IEEE CEC-2005 special section on real-parameter optimization) [62] are used to evaluate JS (Table 2). These functions have various characteristics, such as unimodal, non-separable, shifted, scalable, noise in fitness, a huge number of local optima. Specifically, CF1–CF5 are unimodal,

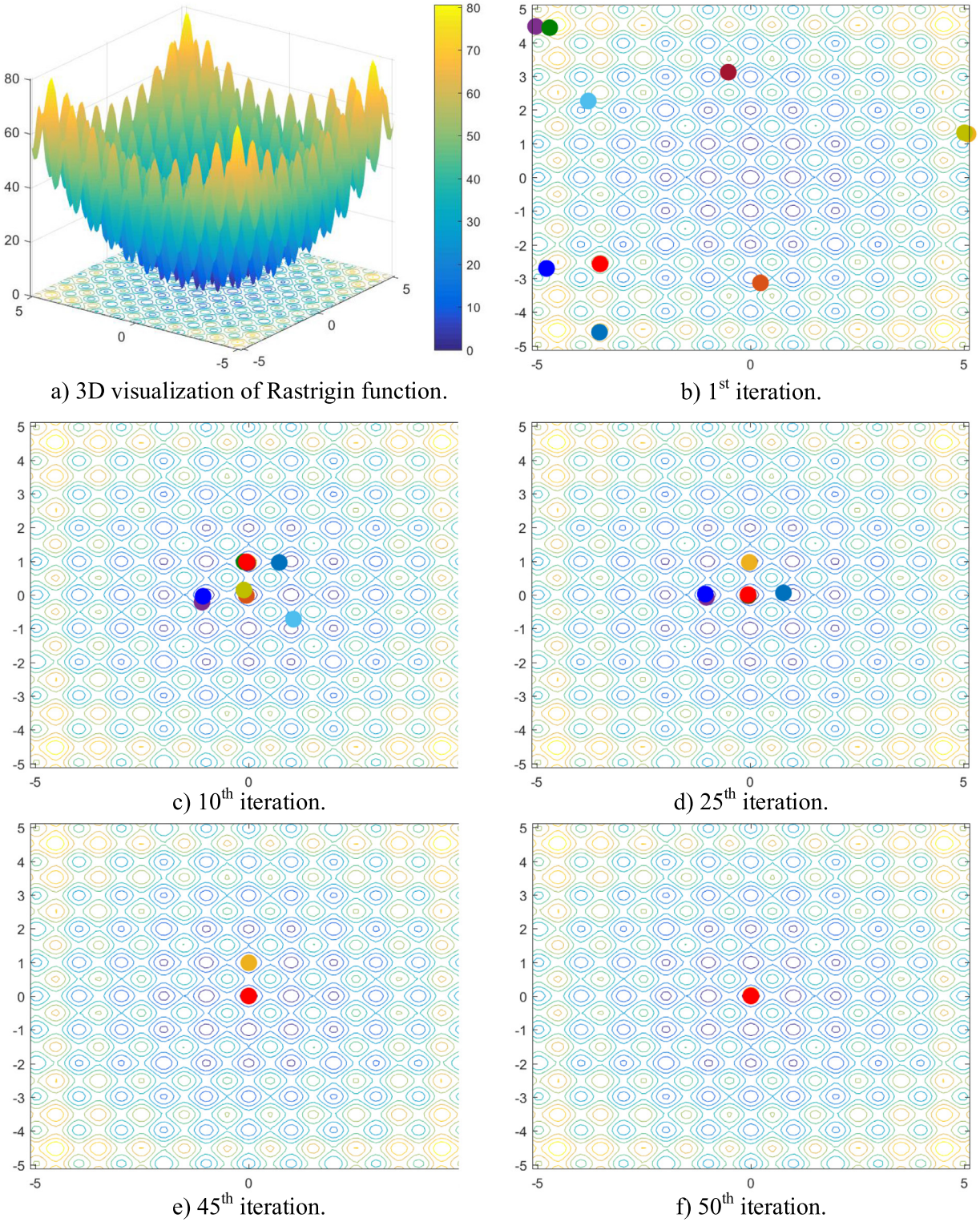


Fig. 9. Visualization and convergence of Rastrigin function (F22).

Table 2
Descriptions of CEC2005 functions.

Test function	Range	Opt.
Unimodal functions		
(CF1) Shifted Sphere Function	[−100, 100]	−450
(CF2) Shifted Schwefel's Problem 1.2	[−100, 100]	−450
(CF3) Shifted Rotated High Conditioned Elliptic Function	[−100, 100]	−450
(CF4) Shifted Schwefel's Problem 1.2 with Noise in Fitness	[−100, 100]	−450
(CF5) Schwefel's Problem 2.6 with Global Optimum on Bounds	[−100, 100]	−310
Multimodal functions		
(CF6) Shifted Rosenbrock's Function	[−100, 100]	390
(CF7) Shifted Rotated Griewank's Function without Bounds	[0, 600]	−180
(CF8) Shifted Rotated Ackley's Function with Global Optimum on Bounds	[−32, 32]	−140
(CF9) Shifted Rastrigin's Function	[−5, 5]	−330
(CF10) Shifted Rotated Rastrigin's Function	[−5, 5]	−330
(CF11) Shifted Rotated Weierstrass Function	[−0.5, 0.5]	90
(CF12) Schwefel's Problem 2.13	$[-\pi, \pi]$	−460
Expanded functions		
(CF13) Expanded Extended Griewank's plus Rosenbrock's Function (CF8CF2)	[−3, 1]	−130
(CF14) Shifted Rotated Expanded Scaffer's CF6	[−100, 100]	−300
Hybrid composite functions		
(CF15) Hybrid Composition Function	[−5, 5]	120
(CF16) Rotated Hybrid Composition Function	[−5, 5]	120
(CF17) Rotated Hybrid Composition Function with Noise in Fitness	[−5, 5]	120
(CF18) Rotated Hybrid Composition Function	[−5, 5]	10
(CF19) Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum	[−5, 5]	10
(CF20) Rotated Hybrid Composition Function with the Global Optimum on the Bounds	[−5, 5]	10
(CF21) Rotated Hybrid Composition Function	[−5, 5]	360
(CF22) Rotated Hybrid Composition Function with High Condition Number Matrix	[−5, 5]	360
(CF23) Non-Continuous Rotated Hybrid Composition Function	[−5, 5]	360
(CF24) Rotated Hybrid Composition Function	[−5, 5]	260
(CF25) Rotated Hybrid Composition Function without Bounds	[2, 5]	260

CF6-CF12 are multimodal, CF13-CF14 are expanded and CF15–CF25 are hybrid composition functions. Table 2 shows these functions in detail.

4.3. Evaluation methods

4.3.1. Hit rate

In this study, the hit rate is defined as the ratio of the number of times an algorithm reaches the global optimum to the number of independent optimization runs [63,64].

$$\text{Hit rate} = \frac{\text{Number of times an algorithm reaches the global optimum}}{\text{Number of independent optimization runs}} \tag{20}$$

4.3.2. Wilcoxon rank-sum test

The non-parametric statistical "Wilcoxon rank sum test" [65] was used to compare the JS with the other algorithms. Given a significance level of α, this test yields the significance of the results of any algorithm by confirming the null hypothesis H_o (that no significant difference or improvement exists between the results obtained using a pair of algorithms, where $p_{value} > \alpha$), or the alternative hypothesis H_1 (that a significant difference exists between the results obtained using a pair of algorithms, where $p_{value} \leq \alpha$).

The performance of JS and each peer algorithm is compared on the results of the evaluation values at a significance level of 5%. If the performance of JS is μ_1 and that of each peer algorithm is μ_2. The hypotheses for performance metrics are defined as

$$\begin{aligned} H_o &: \ \mu_1 \text{ is equivalent to } \mu_2 \\ H_1 &: \ \mu_1 \text{ is better than } \mu_2 \end{aligned} \tag{21}$$

4.3.3. Computation time

Computation time is the time taken by a computer to solve the problem. In this study, the JS and compared algorithms were executed in MATLAB R2016a on a PC with an Intel Core i5–7500 CPU, a clock speed of 3.40 GHz, 8 GB of RAM and Windows 10.

Table 3
Set values of internal parameters for compared metaheuristic algorithms.

Year	Algorithm	Parameters
2020	Jellyfish Search (JS)	NP = 50; NI = 10,000
2016	Whale Optimization Algorithm (WOA)	NP = 50; NI = 10,000; Fluctuation range: decreased from 2 to 0; Coefficient of the logarithmic spiral shape = 1
2015	Tree-Seed Algorithm (TSA)	NP = 50; NI = 10,000; MAX FE = 500,000; Search tendency: ST = 0.1
2014	Symbiotic Organisms Search (SOS)	NP = 50; MAX FE = 500,000
2011	Teaching-Learning-Based Optimization (TLBO)	NP = 50; MAX FE = 500,000
2009	Firefly Algorithm (FA)	NP = 50; NI = 10,000; Randomness of firefly movement = 0.25; Minimum value of attractiveness parameter = 0.2; Absorption coefficient = 1
2009	Gravitation Search Algorithm (GSA)	NP = 50; NI = 10,000; Gravity initial value = 100; Gravity coefficient value = 20
2007	Artificial Bee Colony (ABC)	NP = 50; NI = 10,000; Abandonment Limit Parameter (Trial Limit): round = (0,6 × Population size × Dimension); Acceleration coefficient upper bound = 1
1997	Differential Evolution (DE)	NP = 50; NI = 10,000; Lower bound of scaling factor = 0.2; Lower bound of scaling factor = 0.8; Crossover probability = 0.5
1995	Particle Swarm Optimization (PSO)	NP = 50; NI = 10,000; Inertia weight = 0.5; Inertia weight damping ratio = 0.99; Personal learning coefficient = 2.05; Global learning coefficient = 2.05
1975	Genetic Algorithm (GA)	NP = 50; NI = 10,000; Crossover percentage = 0.7; Mutation rate = 0.1

Note: NP is population size; NI is maximum number of iterations; MAX FE is maximum number of evaluations.

4.4. Effect of population initialization techniques on solution quality

Many chaotic maps have been developed to improve the diversity of the initial population so as to enhance the quality of global solution. To analyze the effect of chaotic maps on global optima, thirteen well-known chaotic maps (Chebyshev map (M1) [66], Circle map (M2) [66], Gauss map (M3) [66], Intermittency map (M4) [66], Iterative map (M5) [66], Kent map (M6) [67], Liebovitch map (M7) [66], Logistic map (M8) [66], Piecewise map (M9) [66], Sine map (M10) [66], Singer map (M11) [66], Sinusoidal map (M12) [66] and Tent map (M13) [66]), and typical random method (M14) were used to solve four mathematical functions (F4, F13, F22 and F41 in Table 1).

The number of jellyfish and iterations are set to 30 and 20, respectively. Statistical box plot is a standardized way to demonstrate the distribution of the data [68,69]. The narrower box plots show the higher level of agreement with each other. After running 30 times of optimization by using distinct maps and random process for population initialization, Fig. 10 shows that the JS algorithm with logistic map (M8) is better than the other chaotic maps and typical random method. The interquartile range and median of the applying logistic map are also superior to the other maps. Therefore, the logistic map is selected to generate initial population for operations of JS algorithm.

4.5. Spatial distribution ($\boldsymbol{\beta}$) and motion ($\boldsymbol{\gamma}$) coefficients of jellyfish search algorithm

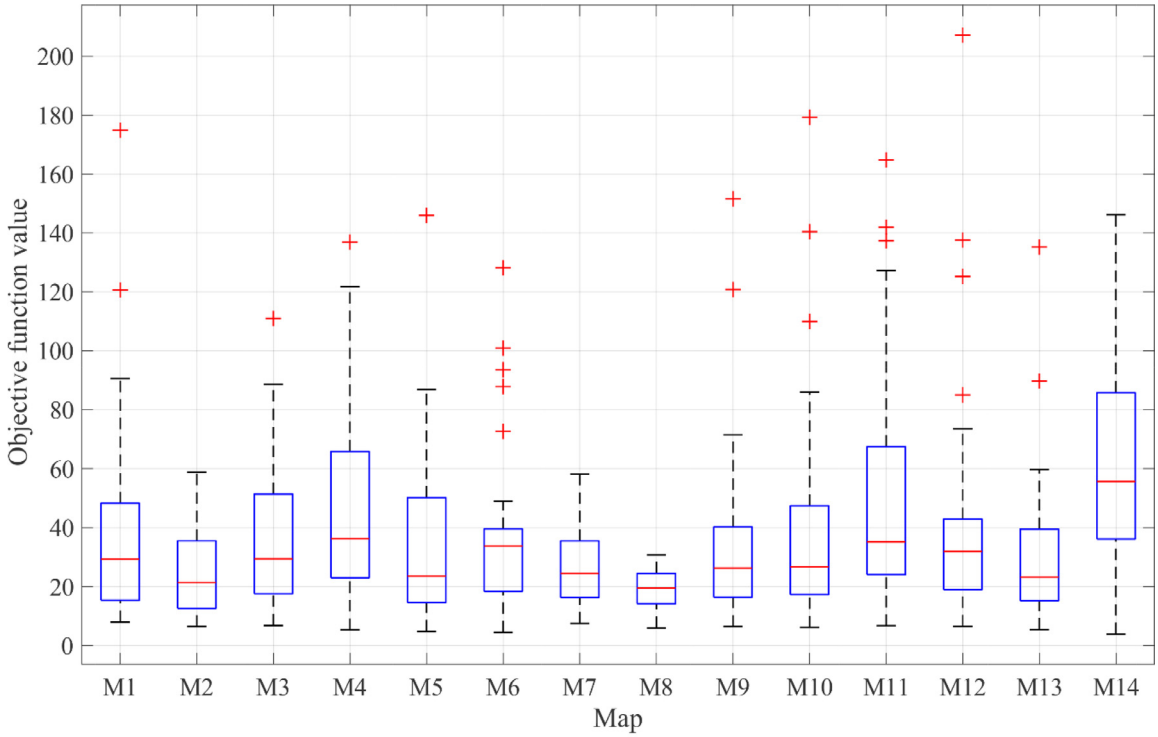
Prior to implementing the JS algorithm for optimization problems, the effectiveness of operating parameters, β (Eq. (11)) and γ (Eq. (12)), related to spatial distribution and movements of JS algorithm has to be evaluated. Four mathematical functions (F4, F13, F22 and F41 in Table 1), which are representations of US, UN, MS and MN characteristics, were used for this purpose. The number of jellyfish and iterations are set to 30 and 50, respectively, the range of β is [0.5 10], and γ is [0.05 1]. Fig. 11 plots the views (3D and 2D) of the effect of β and γ to the objective function values. Fluctuations of the objective function values show that the JS can reach the best optimum value when β =3 and γ = 0.1 in all F4, F13, F22 and F41. Therefore, β =3 and γ = 0.1 thus obtained were adopted to determine the influence of ocean current on, and passive motions of jellyfish swarms for achieving best results.

4.6. Algorithm comparison on solving benchmark functions

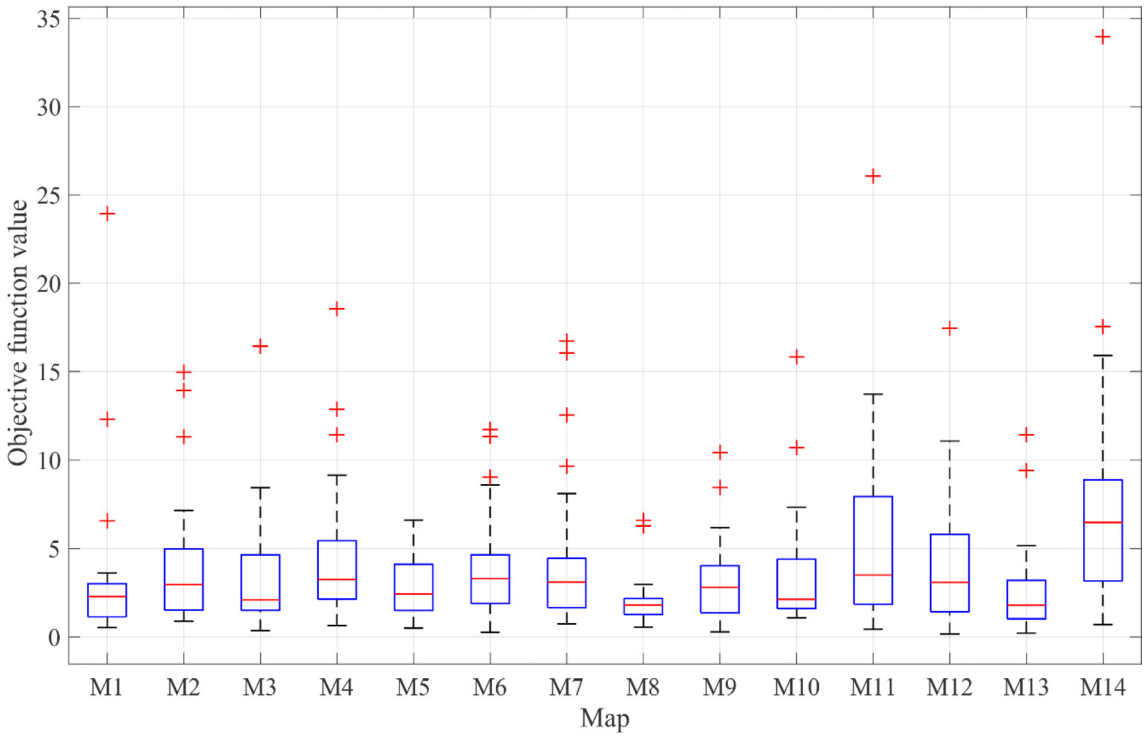
In all small/average-scale benchmark functions, following Karaboga and Bahriye [61], the population size and number of iterations were set to 50 and 10,000, respectively. Hence, the number of evaluations was 500,000. In the large-scale benchmark functions (CEC2005), a population size is set to 50; each test function was solved in 30 s. All the other internal parameters in these examples are presented in Table 3.

Statistic results after 30 runs on the JS and ten metaheuristic algorithms with 50 small/average-scale functions are presented in Table 4. For large-scale benchmark functions (CEC2005), Table 5 presents statistical results obtained after 30 runs, and Table 6 displays p-value of Wilcoxon rank sum test between the JS and compared algorithms.

All the results obtained from the mathematical tests (small/average/large-scale functions) are summarized in Table 7. In

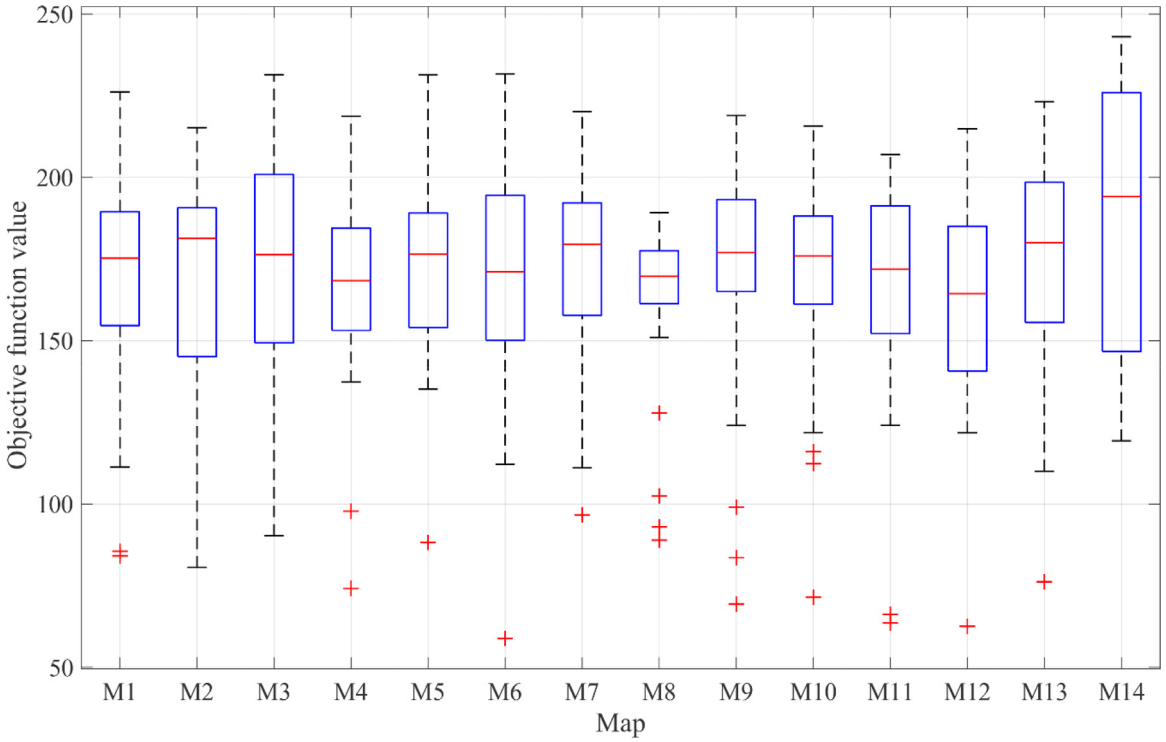


a) SumSquares function (F4).

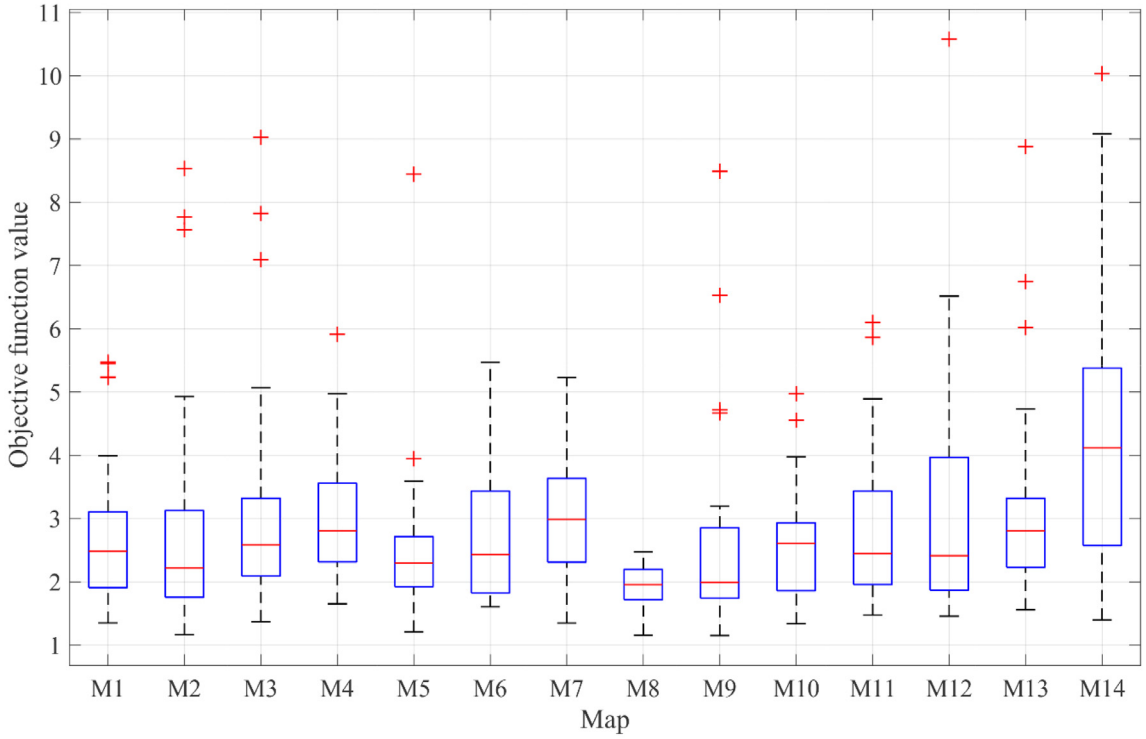


b) Powell function (F13).

Fig. 10. Box plots for effects of chaotic maps and random method on selected benchmark functions.
a) SumSquares function (F4).
b) Powell function (F13).
c) Rastrigin function (F22).
d) Griewank function (F41).

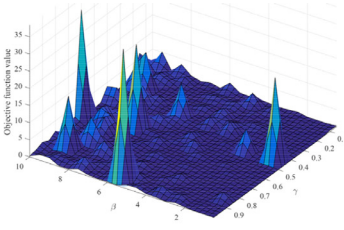


c) Rastrigin function (F22).

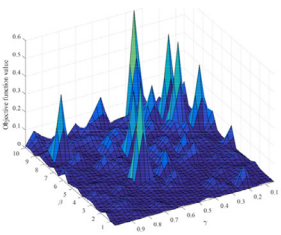


d) Griewank function (F41).

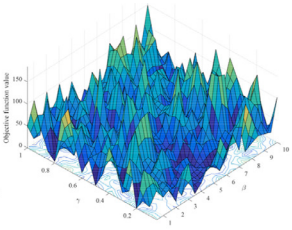
Fig. 10. Continued



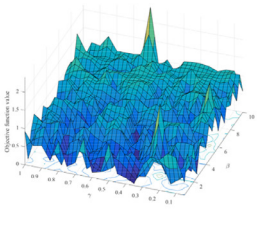
a) 3D view of the effect of β and γ on F4.



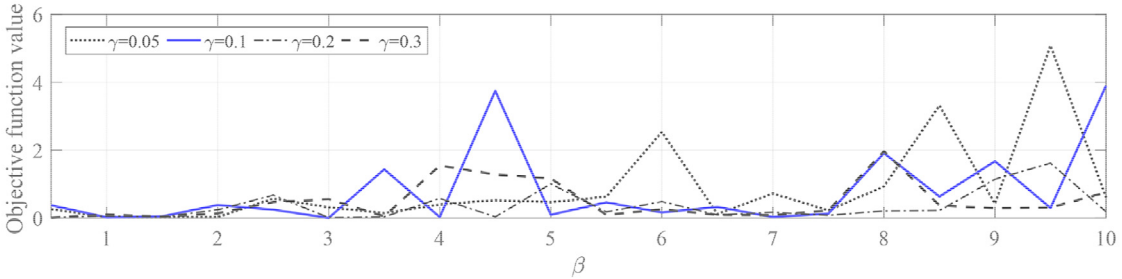
b) 3D view of the effect of β and γ on F13.



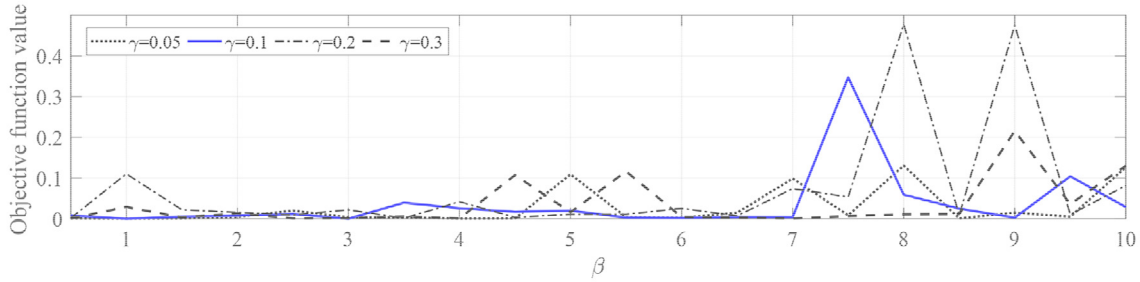
c) 3D view of the effect of β and γ on F22.



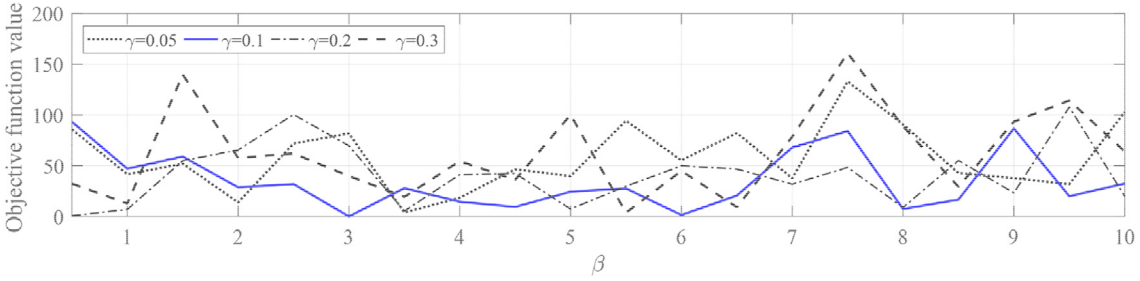
d) 3D view of the effect of β and γ on F41.



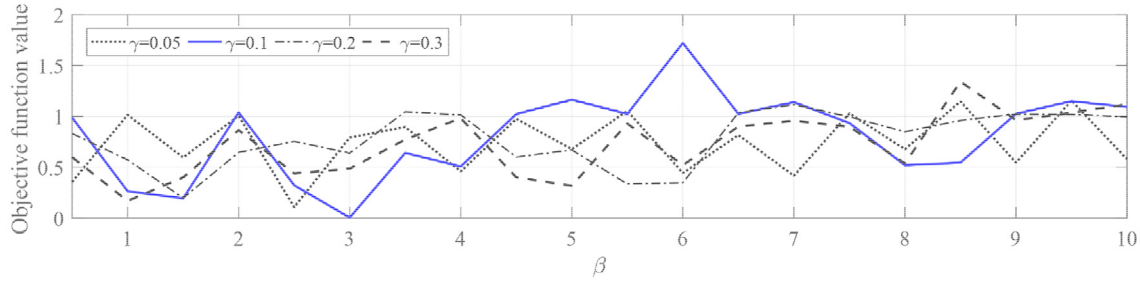
e) 2D view of the effect of β and γ on F4.



f) 2D view of the effect of β and γ on F13.



g) 2D view of the effect of β and γ on F22.



h) 2D view of the effect of β and γ on F41.

Fig. 11. Illustration of effects of β and γ on selected benchmark functions.

Table 4a
Optimization results of JS and WOA obtained in small/average-scale mathematical problems.

Function	JS			WOA		
	Mean	Std.	Time (Sec)	Mean	Std.	Time (Sec)
F1	**0**	**0**	0.66	**0**	**0**	1.93
F2	**0**	**0**	0.53	**0**	**0**	2.45
F3	**0**	**0**	1.13	**0**	**0**	2.87
F4	**0**	**0**	2.53	**0**	**0**	2.74
F5	**5.52E-05**	**3.89E-05**	1.02	1.31E-04	1.84E-04	4.66
F6	**0**	**0**	0.48	**0**	**0**	1.50
F7	**-1.00E+00**	**0**	0.48	**-1.00E+00**	**1.23E-12**	1.47
F8	**0**	**0**	0.77	**0**	**0**	1.46
F9	**0**	**0**	2.16	5.49E-02	2.63E-02	1.50
F10	**-5.00E+01**	**0**	0.46	**-5.00E+01**	**1.71E-09**	1.62
F11	**-2.10E+02**	**2.07E-11**	1.44	**-2.10E+02**	**2.73E-06**	1.78
F12	**0**	**0**	2.58	**0**	**0**	1.86
F13	**0**	**0**	4.08	1.10E-07	2.35E-07	3.14
F14	**0**	**0**	2.75	**0**	**0**	2.45
F15	**0**	**0**	1.62	**0**	**0**	8.63
F16	**3.69E-08**	**3.34E-08**	3.22	2.35E+01	2.86E-01	2.49
F17	**0**	**0**	2.56	6.67E-01	4.27E-07	2.44
F18	**9.98E-01**	**0**	1.31	**9.98E-01**	**0**	5.50
F19	**3.98E-01**	**0**	0.90	**3.98E-01**	**8.85E-12**	1.47
F20	**0**	**0**	0.45	**0**	**0**	1.45
F21	**0**	**0**	0.45	3.61E-09	5.68E-09	1.44
F22	**0**	**0**	2.50	**0**	**0**	2.60
F23	-1.09E+04	1.28E+03	2.28	-1.20E+04	3.97E+02	3.21
F24	**-1.80E+00**	**0**	0.53	**-1.80E+00**	**7.16E-12**	1.66
F25	**-4.69E+00**	**0**	1.27	-4.17E+00	4.00E-01	2.07
F26	**-9.66E+00**	**0**	0.82	-7.04E+00	8.29E-01	2.72
F27	**0**	**0**	0.43	**0**	**0**	1.47
F28	**-1.03E+00**	**0**	0.49	**-1.03E+00**	**0**	1.69
F29	**0**	**0**	0.46	**0**	**0**	1.45
F30	**0**	**0**	0.42	**0**	**0**	1.46
F31	**-1.87E+02**	**0**	0.57	**-1.87E+02**	**5.14E-09**	1.61
F32	**3.00E+00**	**0**	0.44	3.00E+00	7.87E-09	1.47
F33	**3.07E-04**	**0**	0.50	4.32E-04	3.16E-04	1.72
F34	**-1.02E+01**	**0**	1.13	-1.02E+01	8.69E-08	4.31
F35	**-1.04E+01**	**0**	1.16	**-1.04E+01**	**6.45E-08**	4.39
F36	**-1.05E+01**	**0**	1.21	**-1.05E+01**	**6.33E-08**	4.42
F37	**4.72E-05**	**5.84E-05**	4.23	4.62E-01	6.50E-01	2.92
F38	**4.94E-07**	**5.69E-07**	3.11	2.88E-01	3.22E-01	2.32
F39	**-3.86E+00**	**0**	1.62	-3.86E+00	2.22E-03	6.44
F40	**-3.32E+00**	**0**	2.09	-3.23E+00	1.65E-01	8.80
F41	**0**	**0**	0.65	5.21E-04	1.98E-03	3.05
F42	**0**	**0**	0.69	**0**	**0**	3.37
F43	**0**	**0**	6.37	1.44E-07	6.68E-08	9.95
F44	**0**	**0**	4.43	4.24E-07	2.69E-07	6.83
F45	**-1.08E+00**	**0**	0.54	**-1.08E+00**	**0**	1.64
F46	**-1.50E+00**	**0**	1.49	-9.41E-01	3.81E-01	1.98
F47	**-1.34E+00**	**3.02E-01**	2.33	-4.23E-01	1.95E-01	2.75
F48	**0**	**0**	0.53	1.42E-10	1.85E-10	1.77
F49	**0**	**0**	1.71	4.46E+02	3.24E+02	2.96
F50	**0**	**0**	1.98	3.61E+02	3.32E+02	3.11
Count of best hit	49			26		
Hit rate	98.00%			52.00%		
Total time Sec)			77.52			149.00

Std. = Standard deviation; Time (Sec) = computation time (unit: second); Bold numbers represent the best values.

particular, Table 7 shows that the three best algorithms with respect to hit rate were JS, TSA and ABC, with hit rates of 98%, 78% and 76%, respectively, for small/average-scale functions. JS took the least computation time (77.52 s) to solve 50 functions, and TSA and ABC required 106.69 s and 638.15 s, respectively. The convergence curves shown in Fig. 12 prove that the JS algorithm was definitely the fastest optimizer.

For the CEC2005 functions, Wilcoxon rank sum test (Table 7) indicates that JS outperforms the WOA (72/75), TSA (73/75), SOS (57/75), TLBO (61/75), FA (68/75), GSA (70/75), ABC (64/75), DE (63/75), PSO (65/75) and GA (65/75). The convergence curves (Fig. 13) confirm that the JS had a significantly higher convergence speed than the other ten algorithms.

Table 4b
Optimization results of TSA, SOS and TLBO obtained in small/average-scale mathematical problems.

Function	TSA			SOS			TLBO		
	Mean	Std.	Time (Sec)	Mean	Std.	Time (Sec)	Mean	Std.	Time (Sec)
F1	**0**	**0**	1.06	**0**	**0**	5.82	**0**	**0**	6.32
F2	**0**	**0**	1.66	**0**	**0**	5.85	**0**	**0**	6.05
F3	**0**	**0**	1.90	**0**	**0**	6.32	**0**	**0**	7.06
F4	**0**	**0**	1.65	**0**	**0**	7.16	**0**	**0**	6.86
F5	1.22E-02	0.003	3.79	9.13E-05	3.71E-05	8.91	8.93E-05	3.76E-05	8.43
F6	**0**	**0**	0.64	**0**	**0**	5.87	**0**	**0**	5.72
F7	**-1**	**0**	0.58	**-1.00E+00**	**0**	5.74	**-1.00E+00**	**0**	5.67
F8	**0**	**0**	0.57	**0**	**0**	5.71	**0**	**0**	5.84
F9	1.80E-05	3E-05	0.63	**0**	**0**	5.75	**0**	**0**	5.79
F10	**-50**	**0**	0.85	**-5.00E+01**	**0**	6.06	**-5.00E+01**	**0**	5.86
F11	**-210**	**2E-10**	1.01	-2.10E+02	0	6.44	**-2.10E+02**	**2.60E-12**	5.97
F12	**0**	**0**	0.98	**0**	**0**	6.61	**0**	**0**	6.38
F13	2.48E-02	0.009	2.43	**0**	**0**	8.17	7.44E-07	1.28E-06	7.10
F14	**0**	**0**	1.79	**0**	**0**	7.16	**0**	**0**	6.83
F15	**0**	**0**	3.52	**0**	**0**	14.25	**0**	**0**	14.75
F16	2.38E+01	0.541	1.88	1.04E-07	2.95E-07	7.04	1.46E+00	1.33E+00	7.05
F17	6.67E-01	4E-07	1.85	**0**	**0**	6.91	6.67E-01	0	6.18
F18	**9.98E-01**	**0**	4.90	**9.98E-01**	**0**	10.49	**9.98E-01**	**0**	9.95
F19	**3.98E-01**	**0**	0.65	3.98E-01	0	6.70	**3.98E-01**	**0**	5.68
F20	**0**	**0**	0.63	**0**	**0**	6.09	**0**	**0**	5.67
F21	**0**	**0**	0.59	3.38E-02	1.29E-01	5.70	**0**	**0**	5.65
F22	1.30E+02	37.91	2.28	**0**	**0**	6.65	9.99E+00	6.02E+00	6.52
F23	-7.08E+03	966.7	2.72	-1.21E+04	3.00E+01	7.70	-8.18E+03	6.14E+02	7.12
F24	**-1.80E+00**	**0**	0.76	-1.80E+00	0	7.02	**-1.80E+00**	**0**	5.90
F25	**-4.69E+00**	**0**	1.19	-4.69E+00	0	6.62	-4.62E+00	7.24E-02	6.27
F26	-9.65E+00	0.03	1.92	-9.66E+00	1.25E-03	8.28	-9.19E+00	2.59E-01	6.83
F27	**0**	**0**	0.64	**0**	**0**	5.53	**0**	**0**	5.75
F28	**-1.03E+00**	**0**	0.85	**-1.03E+00**	**0**	5.44	**-1.03E+00**	**0**	6.00
F29	**0**	**0**	0.61	**0**	**0**	5.19	**0**	**0**	5.64
F30	**0**	**0**	0.62	**0**	**0**	5.22	**0**	**0**	5.70
F31	**-1.87E+02**	**0**	0.82	-1.87E+02	0	5.50	**-1.87E+02**	**0**	5.91
F32	**3.00E+00**	**0**	0.63	**3.00E+00**	**0**	5.25	**3.00E+00**	**0**	5.68
F33	**3.07E-04**	**5E-10**	0.99	**3.07E-04**	**0**	5.58	**3.07E-04**	**0**	6.02
F34	**-1.02E+01**	**0**	3.69	-1.02E+01	0	8.50	**-1.02E+01**	**0**	9.44
F35	**-1.04E+01**	**0**	3.67	-1.00E+01	1.35E+00	8.51	**-1.04E+01**	**0**	9.55
F36	**-1.05E+01**	**0**	3.79	-1.04E+01	9.87E-01	8.65	**-1.05E+01**	**0**	9.59
F37	2.27E-03	0.003	2.20	5.80E-03	2.37E-02	6.95	2.75E-03	3.86E-03	7.28
F38	5.21E-04	9E-04	1.56	4.74E-05	7.53E-05	6.30	1.18E-04	1.47E-04	6.66
F39	**-3.86E+00**	**0**	5.67	**-3.86E+00**	**0**	10.84	**-3.86E+00**	**0**	12.01
F40	**-3.32E+00**	**0**	8.06	-3.29E+00	5.36E-02	13.13	-3.31E+00	3.02E-02	14.17
F41	**0**	**0**	2.44	**0**	**0**	6.70	**0**	**0**	6.76
F42	**0**	**0**	2.60	**0**	**0**	7.01	**0**	**0**	7.21
F43	**0**	**0**	9.79	**0**	**0**	14.25	6.91E-03	2.63E-02	14.56
F44	**0**	**0**	6.59	1.09E-02	3.37E-02	11.01	2.24E-02	4.83E-02	11.26
F45	**-1.08E+00**	**0**	0.74	-1.08E+00	0	5.49	**-1.08E+00**	**0**	5.91
F46	**-1.50E+00**	**0**	1.10	**-1.50E+00**	**0**	5.90	-1.39E+00	2.31E-01	6.28
F47	-1.14E+00	0.342	1.91	-7.93E-01	3.71E-01	6.67	-5.26E-01	2.53E-01	6.99
F48	**0**	**0**	0.90	**0**	**0**	5.60	**0**	**0**	6.09
F49	**0**	**0**	2.08	**0**	**0**	6.94	4.18E+01	1.22E+02	7.29
F50	**0**	**0**	2.28	5.38E-04	2.94E-03	7.13	3.27E+01	1.23E+02	7.35
Count of best hit	39			30			33		
Hit rate	78.00%			60.00%			66.00%		
Total time (Sec)			106.69			362.29			366.53

Std. = Standard deviation; Time (Sec) = computation time (unit: second); Bold numbers represent the best values.

4.7. Performance of JS algorithm

A robust optimization algorithm should be able to (1) explore the search space, (2) exploit the promising areas, and (3) converge to the best solution [10,70]. In this subsection, a comparative performance of JS, considering all these three aspects against ten state-of-the-art algorithms, is discussed.

4.7.1. Capacity of exploration

Multimodal benchmark functions have many local optima [10]. Consequently, they are usually used to evaluate the capability of exploration in the search space for an optimization algorithm. In this study, 33 small/average-scale (F18 to F50)

Table 4c
Optimization results of FA, GSA and ABC obtained in small/average-scale mathematical problems.

Function	FA			GSA			ABC		
	Mean	Std.	Time (Sec)	Mean	Std.	Time (Sec)	Mean	Std.	Time (Sec)
F1	**0**	**0**	19.87	**0**	**0**	17.266	**0**	**0**	14.21
F2	**0**	**0**	29.90	**0**	**0**	41.783	**0**	**0**	12.32
F3	5.20E-04	9.62E-05	37.32	**0**	**0**	41.956	**0**	**0**	13.35
F4	2.00E-04	9.85E-05	37.20	**0**	**0**	41.682	**0**	**0**	13.01
F5	3.32E-02	2.68E-02	39.45	1.02E-02	2.66E-03	44.012	3.00E-02	4.87E-03	15.00
F6	4.06E-11	5.10E-11	31.58	**0**	**0**	13.632	**0**	**0**	12.95
F7	-9.00E-01	3.05E-01	29.91	-9.92E-01	3.07E-02	13.634	**-1.00E+00**	**0**	12.56
F8	8.53E-12	8.39E-12	31.63	**0**	**0**	13.529	**0**	**0**	11.47
F9	8.88E-07	5.71E-07	32.01	**0**	**0**	15.846	9.30E-02	6.63E-02	10.64
F10	**-5.00E+01**	**3.30E-07**	32.18	**-5.00E+01**	**0**	17.842	**-5.00E+01**	**0**	10.86
F11	**-2.10E+02**	**1.14E-05**	33.12	**-2.10E+02**	**0**	21.935	**-2.10E+02**	**0**	10.96
F12	4.12E-07	1.31E-07	33.21	**0**	**0**	21.946	2.48E-04	1.83E-04	10.81
F13	2.18E-02	7.83E-03	37.12	2.48E-05	7.53E-06	36.701	3.13E-03	5.03E-04	11.90
F14	1.22E-02	3.25E-03	37.18	1.38E-08	1.97E-09	41.755	**0**	**0**	11.17
F15	1.29E-02	6.56E-03	38.78	**0**	**0**	43.557	**0**	**0**	12.75
F16	2.89E+01	1.37E+01	37.24	1.42E+01	3.39E-01	41.820	8.88E-02	7.74E-02	11.00
F17	6.99E-01	5.89E-02	37.22	6.67E-01	0	41.729	**0**	**0**	11.30
F18	**9.98E-01**	**0**	35.56	1.18E+00	3.62E-01	17.640	**9.98E-01**	**0**	15.03
F19	3.98E-01	6.74E-12	31.46	**3.98E-01**	**0**	13.613	**3.98E-01**	**0**	10.87
F20	1.78E-07	1.68E-07	31.57	**0**	**0**	13.611	**0**	**0**	10.94
F21	3.28E-10	2.00E-10	31.46	**0**	**0**	13.578	**0**	**0**	10.82
F22	3.40E+01	1.10E+01	37.56	1.27E+01	3.27E+00	42.087	**0**	**0**	11.50
F23	-7.43E+03	6.04E+02	37.88	-2.84E+03	4.51E+02	42.551	**-1.26E+04**	**0**	12.00
F24	-1.80E+00	5.01E-11	31.67	**-1.80E+00**	**0**	13.868	**-1.80E+00**	**0**	11.43
F25	-4.61E+00	7.43E-02	32.40	-4.58E+00	7.19E-02	17.331	**-4.69E+00**	**0**	11.60
F26	-8.77E+00	5.78E-01	33.95	-9.22E+00	2.34E-01	22.938	**-9.66E+00**	**0**	12.05
F27	6.58E-04	7.27E-04	31.54	4.96E-03	4.59E-03	13.646	**0**	**0**	11.01
F28	**-1.03E+00**	**1.11E-10**	31.72	**-1.03E+00**	**0**	13.829	**-1.03E+00**	**0**	11.35
F29	8.46E-09	9.30E-09	31.54	**0**	**0**	13.630	**0**	**0**	11.09
F30	7.14E-08	5.63E-08	31.53	**0**	**0**	13.615	**0**	**0**	11.05
F31	**-1.87E+02**	**1.42E-07**	31.67	-1.87E+02	2.00E-01	13.725	**-1.87E+02**	**0**	11.06
F32	**3.00E+00**	**9.06E-10**	31.47	**3.00E+00**	**0**	13.646	**3.00E+00**	**0**	11.06
F33	**3.07E-04**	**3.00E-10**	32.24	8.62E-04	1.65E-04	16.044	4.27E-04	6.04E-05	11.14
F34	**-1.02E+01**	**2.21E-07**	34.54	-8.07E+00	2.02E+00	18.474	**-1.02E+01**	**0**	15.79
F35	**-1.04E+01**	**2.25E-07**	34.64	**-1.04E+01**	**0**	18.501	**-1.04E+01**	**0**	16.05
F36	**-1.05E+01**	**2.73E-07**	34.66	**-1.05E+01**	**0**	18.556	**-1.05E+01**	**0**	15.80
F37	4.88E-02	1.23E-01	33.36	1.54E+00	1.06E+00	17.298	4.11E-02	2.31E-02	12.50
F38	1.53E-04	1.48E-04	32.75	7.02E-03	9.24E-03	16.709	2.95E-03	2.29E-03	11.73
F39	**-3.86E+00**	**7.86E-11**	36.44	**-3.86E+00**	**0**	19.442	**-3.86E+00**	**0**	18.45
F40	-3.28E+00	5.84E-02	39.90	**-3.32E+00**	**0**	24.633	**-3.32E+00**	**0**	20.87
F41	1.25E-03	2.98E-04	39.39	2.47E-04	1.35E-03	42.607	**0**	**0**	12.13
F42	5.17E-03	4.30E-04	38.23	2.23E-09	2.37E-10	42.776	**0**	**0**	12.55
F43	1.59E-06	2.69E-07	44.44	**0**	**0**	48.971	**0**	**0**	20.95
F44	1.65E-05	3.22E-06	41.50	**0**	**0**	45.944	**0**	**0**	17.69
F45	**-1.08E+00**	**3.20E-11**	31.58	**-1.08E+00**	**2.10E-03**	13.720	**-1.08E+00**	**0**	11.42
F46	-1.44E+00	1.75E-01	32.31	-8.57E-01	1.58E-01	17.262	-9.38E-01	2.08E-04	11.64
F47	-6.33E-01	3.09E-01	33.91	-3.10E-01	1.37E-01	22.850	-4.46E-01	1.34E-01	13.41
F48	4.68E-09	4.41E-09	31.71	8.99E-02	1.10E-01	13.849	**0**	**0**	11.50
F49	1.54E+01	2.02E+01	33.40	2.18E+02	2.64E+02	18.208	1.74E-01	6.82E-02	12.85
F50	3.35E+01	1.22E+02	34.40	1.75E+02	2.35E+02	23.198	8.23E+00	8.09E+00	12.58
Count of best hit	14			26			38		
Hit rate	28.00%			52.00%			76.00%		
Total time (Sec)			1707.30			1228.98			638.15

Std. = Standard deviation; Time (Sec) = computation time (unit: second); Bold numbers represent the best values.

and 21 high-scale (CF6 to CF12 with three levels of dimensions) multimodal benchmark functions were used to evaluate the optimization algorithms.

The analytical results in Table 7 indicate that JS algorithm solves 32/33 functions with global optima while WOA, TSA, SOS, TLBO, FA, GSA, ABC, DE, PSO and GA solve 14/33, 27/33, 16/33, 20/33, 10/33, 15/33, 26/33, 20/33, 11/33 and 11/33 functions, respectively, for the small/average-scale benchmark problmes. For the high-scale functions, results of Wilcoxon rank sum tests show that the JS algorithm outperforms the WOA (18/21), TSA (20/21), SOS (17/21), TLBO (17/21), FA (20/21), GSA (18/21), ABC (15/21), DE (15/21), PSO (18/21) and GA (18/21) in 30 s of running. Thus, the JS is a superior optimization algorithm in exploring the search space.

Table 4d
Optimization results of DE, PSO and GA obtained in small/average-scale mathematical problems.

Function	DE			PSO			GA		
	Mean	Std.	Time (Sec)	Mean	Std.	Time (Sec)	Mean	Std.	Time (Sec)
F1	**0**	**0**	12.81	**0**	**0**	9.81	**0**	**0**	11.86
F2	**0**	**0**	13.09	**0**	**0**	9.73	1.17E+03	7.66E+01	12.35
F3	**0**	**0**	15.24	**0**	**0**	10.60	1.11E+03	7.42E+01	12.71
F4	**0**	**0**	14.22	**0**	**0**	10.06	1.48E+02	1.24E+01	12.36
F5	1.36E-03	4.17E-04	15.59	1.16E-03	2.76E-04	12.04	1.81E-01	2.71E-02	14.80
F6	**0**	**0**	12.28	**0**	**0**	9.15	**0**	**0**	10.70
F7	**-1.00E+00**	**0**	12.19	**-1.00E+00**	**0**	9.09	**-1.00E+00**	**0**	10.62
F8	**0**	**0**	12.35	**0**	**0**	9.19	**0**	**0**	10.50
F9	4.09E-02	8.20E-02	12.37	**0**	**0**	9.33	1.49E-02	7.36E-03	11.17
F10	**-5.00E+01**	**0**	12.62	**-5.00E+01**	**0**	9.24	-5.00E+01	2.25E-05	11.30
F11	**-2.10E+02**	**0**	13.07	**-2.10E+02**	**0**	9.55	-2.09E+02	1.93E-01	11.42
F12	**0**	**0**	13.04	**0**	**0**	9.62	1.34E-02	4.53E-03	11.52
F13	2.17E-07	1.36E-07	14.21	1.10E-04	1.60E-04	10.76	9.70E+00	1.55E+00	13.41
F14	**0**	**0**	14.30	**0**	**0**	10.56	1.10E+01	1.39E+00	12.42
F15	**0**	**0**	15.96	**0**	**0**	11.83	7.40E+03	1.14E+03	14.16
F16	1.82E+01	5.04E+00	13.56	1.51E+01	2.42E+01	9.98	1.96E+05	3.85E+04	12.43
F17	6.67E-01	1.00E-09	13.41	6.67E-01	1.00E-08	9.83	1.22E+03	2.66E+02	12.41
F18	**9.98E-01**	**0**	16.60	**9.98E-01**	**0**	13.28	**9.98E-01**	**0**	15.00
F19	**3.98E-01**	**0**	12.27	**3.98E-01**	**0**	9.11	**3.98E-01**	**0**	10.54
F20	**0**	**0**	12.21	**0**	**0**	9.07	**0**	**0**	10.61
F21	**0**	**0**	12.87	**0**	**0**	9.09	**0**	**0**	10.56
F22	11.71673	2.538172	13.40	4.40E+01	1.17E+01	10.14	5.29E+01	4.56E+00	12.82
F23	-1.03E+04	5.22E+02	14.14	-6.91E+03	4.58E+02	10.63	-1.16E+04	9.33E+01	13.35
F24	**-1.80E+00**	**0**	12.35	-1.57E+00	1.20E-01	9.32	**-1.80E+00**	**0**	10.86
F25	-4.68E+00	1.25E-02	12.88	-2.49E+00	2.57E-01	9.66	-4.64E+00	9.79E-02	11.88
F26	-9.59E+00	6.42E-02	13.53	-4.01E+00	5.03E-01	10.28	-9.50E+00	1.41E-01	12.36
F27	**0**	**0**	12.20	**0**	**0**	9.15	4.24E-03	4.76E-03	10.74
F28	**-1.03E+00**	**0**	12.47	**-1.03E+00**	**0**	9.37	**-1.03E+00**	**0**	11.01
F29	**0**	**0**	12.01	**0**	**0**	9.07	6.83E-02	0.078216	10.56
F30	**0**	**0**	12.06	**0**	**0**	9.09	**0**	**0**	10.67
F31	**-1.87E+02**	**0**	12.24	**-1.87E+02**	**0**	9.35	**-1.87E+02**	**0**	10.94
F32	**3.00E+00**	**0**	12.09	**3.00E+00**	**0**	9.11	5.25E+00	5.870093	10.81
F33	4.27E-04	0.000273	12.54	4.91E-04	3.66E-04	9.45	5.62E-03	8.17E-03	11.80
F34	**-1.02E+01**	**0**	16.90	-2.09E+00	1.18E+00	13.27	-5.66E+00	3.87E+00	16.26
F35	**-1.04E+01**	**0**	17.00	-1.99E+00	1.42E+00	13.28	-5.34E+00	3.52E+00	16.64
F36	**-1.05E+01**	**0**	17.05	-1.88E+00	4.32E-01	13.39	-3.83E+00	2.45E+00	16.45
F37	2.40E-02	4.60E-02	13.87	3.61E-02	4.89E-02	10.72	3.03E-01	1.93E-01	13.25
F38	1.43E-04	1.45E-04	13.24	1.14E+01	7.36E+00	10.10	1.04E-02	9.08E-03	12.55
F39	**-3.86E+00**	**0**	20.06	-3.63E+00	0.116937	15.94	**-3.86E+00**	**0**	18.83
F40	-3.23E+00	0.047557	22.46	-1.86E+00	4.40E-01	18.08	-3.30E+00	5.01E-02	21.50
F41	0.001479	0.002958	13.96	1.74E-02	2.08E-02	10.71	1.06E+01	1.16E+00	13.77
F42	**0**	**0**	14.76	1.65E-01	4.94E-01	11.15	1.47E+01	1.78E-01	14.14
F43	**0**	**0**	22.45	2.07E-02	4.15E-02	18.40	1.34E+01	1.45E+00	21.91
F44	0.002198	0.004395	19.02	7.68E-03	1.63E-02	15.17	1.25E+02	1.20E+01	18.62
F45	**-1.08E+00**	**0**	12.43	-6.79E-01	2.75E-01	9.41	**-1.08E+00**	**0**	11.27
F46	**-1.50E+00**	**0**	12.86	-5.05E-01	2.14E-01	9.74	-9.68E-01	2.88E-01	12.16
F47	-1.05E+00	3.02E-01	13.66	-2.57E-03	3.52E-03	10.45	-6.36E-01	3.75E-01	12.85
F48	**0**	**0**	12.78	**0**	**0**	9.56	**0**	**0**	11.44
F49	5.99E+00	7.33E+00	14.38	1.46E+03	1.27E+03	10.76	4.30E-03	9.47E-03	13.23
F50	7.82E+02	1.05E+03	14.55	1.36E+03	1.33E+03	10.77	2.96E+01	1.60E+01	13.27
Count of best hit	32			24			15		
Hit rate	64.00%			48.00%			30.00%		
Total time (Sec)			707.60			536.45			648.76

Std. = Standard deviation; Time (Sec) = computation time (unit: second); Bold numbers represent the best values.

4.7.2. Capacity of exploitation

The exploitation capability of an optimization algorithm can be assessed by solving unimodal functions [10]. The capability of JS algorithm is evaluated by testing 17 small/average-scale (F1 to F17) and 15 high-scale (CF1 to CF5 with three levels of dimensions) unimodal benchmark functions, and then compared with ten well-known optimization algorithms (Table 7).

From the numerical results in small/average-scale functions, the JS achieves optimal solutions in 17/17 functions, while WOA, TSA, SOS, TLBO, FA, GSA, ABC, DE, PSO, and GA reach at optimal values by 12/17, 12/17, 14/17, 13/17, 4/17, 11/17, 12/17, 12/17, 13/17 and 4/17 functions, respectively. In the high-scale mathematical functions, results of Wilcoxon rank sum tests show that the JS algorithm is superior or equivalent to the WOA (15/15), TSA (15/15), SOS (8/15), TLBO (10/15), FA (13/15), GSA (15/15), ABC (14/15), DE (12/15), PSO (11/15) and GA (14/15) functions in 30 s of running. The analytical results indicate

Table 5a
Optimization results of JS and WOA for CEC2005 functions (CF1-CF14).

Function		JS		WOA	
		Mean	Std.	Mean	Std.
CF1	D=10	-4.5000E+02	5.3823E-14	-4.4949E+02	7.7299E-01
	D=30	-4.5000E+02	3.9535E-09	-2.1261E+02	1.1796E+02
	D=50	-4.5000E+02	3.2592E-07	1.4792E+03	8.8575E+02
CF2	D=10	-4.5000E+02	1.5973E-10	6.5412E+03	2.5867E+03
	D=30	-4.5000E+02	9.5321E-05	8.2709E+04	2.2740E+04
	D=50	-4.5000E+02	1.8824E-03	3.0821E+05	6.9954E+04
CF3	D=10	-4.4997E+02	1.1709E-02	1.8050E+06	2.6572E+06
	D=30	-4.4958E+02	1.2762E-01	8.9994E+07	3.1279E+07
	D=50	-4.4913E+02	1.8480E-01	2.4164E+08	8.8204E+07
CF4	D=10	-4.5000E+02	4.0974E-09	1.6616E+04	6.5592E+03
	D=30	-4.5000E+02	1.6167E-04	1.7178E+05	4.8224E+04
	D=50	-4.5000E+02	3.3532E-03	4.7871E+05	1.5757E+05
CF5	D=10	-3.0985E+02	2.7804E-01	1.8347E+03	2.9512E+03
	D=30	5.4987E+03	1.2328E+03	1.9077E+04	4.4953E+03
	D=50	1.5393E+04	1.8937E+03	3.3820E+04	5.5478E+03
CF6	D=10	3.9000E+02	1.8325E-04	2.1564E+04	3.3535E+04
	D=30	3.9000E+02	7.1423E-05	6.7711E+06	5.9720E+06
	D=50	3.9000E+02	2.2387E-04	1.3024E+08	8.5978E+07
CF7	D=10	1.0871E+03	1.1193E-01	1.0875E+03	3.2055E-01
	D=30	4.5168E+03	1.1179E+00	4.5617E+03	4.9021E+01
	D=50	6.0328E+03	1.0748E+01	6.2451E+03	6.4754E+01
CF8	D=10	-1.1962E+02	7.1797E-02	-1.1962E+02	1.1276E-01
	D=30	-1.1899E+02	6.3559E-02	-1.1910E+02	9.3847E-02
	D=50	-1.1878E+02	2.6391E-02	-1.1887E+02	7.0976E-02
CF9	D=10	-3.3000E+02	2.7956E-07	-2.9343E+02	1.1237E+01
	D=30	-3.3000E+02	1.7510E-06	-9.4120E+01	5.4169E+01
	D=50	-3.3000E+02	2.8864E-06	1.4438E+02	6.1285E+01
CF10	D=10	-3.3000E+02	2.8939E-07	-2.7106E+02	1.9159E+01
	D=30	-3.3000E+02	2.6208E-06	8.1210E+01	8.2002E+01
	D=50	-3.3000E+02	4.6059E-06	5.7968E+02	1.0487E+02
CF11	D=10	9.0021E+01	5.1709E-03	9.8516E+01	1.5382E+00
	D=30	9.0082E+01	7.0591E-03	1.2781E+02	3.0771E+00
	D=50	9.0146E+01	1.5251E-02	1.6341E+02	3.6507E+00
CF12	D=10	-4.5993E+02	2.5557E-02	1.5576E+04	1.0252E+04
	D=30	-4.5945E+02	1.0271E-01	5.9203E+05	2.2247E+05
	D=50	-4.5767E+02	2.9909E-01	3.1469E+06	1.1286E+06
CF13	D=10	-1.3000E+02	0.0000E+00	-1.2721E+02	1.4107E+00
	D=30	-1.3000E+02	1.3964E-14	-1.0602E+02	5.5823E+00
	D=50	-1.3000E+02	2.0441E-14	-7.1462E+01	1.2168E+01
CF14	D=10	-3.0000E+02	5.0528E-06	-2.9618E+02	3.6443E-01
	D=30	-3.0000E+02	1.0828E-05	-2.8634E+02	2.4654E-01
	D=50	-3.0000E+02	1.3849E-05	-2.7659E+02	3.1487E-01

Std. = Standard deviation.

that the JS algorithm demonstrates excellent exploitation capability owing to its better solutions than most of the compared algorithms.

4.7.3. Convergence capability

Intensification and diversification (I&D) are two major issues when designing a global search method [71,72]. Diversification generally refers to the ability to visit many and different regions of the search space, whereas intensification refers to the ability to obtain high quality solutions within those regions. Most classical metaheuristic algorithms have several components for intensification and diversification [72]. I&D components are operators, actions, or strategies of metaheuristic algorithms [71]. An effective tradeoff between them helps to reduce computational cost and implement efficient optimization [57].

In the mathematical tests, the JS algorithm only needs 77.52 s to solve 50 small/average-scale functions (Table 7), whereas the compared algorithms need at least 106.69 s (TSA) under the same hardware specifications and software settings. Moreover, the JS also attains 98% of hit rate in contrast to those of the WOA (52%), TSA(78%), SOS (60%), TLBO (66%), FA (28%), GSA (52%), ABC (76%), DE (64%), PSO (48%), GA (30%) for solving these functions. In the high scale functions (CEC2005 functions), results of Wilcoxon rank sum tests show that JS is better than WOA (72/75), TSA (73/75), SOS (57/75), TLBO (61/75), FA (68/75), GSA (70/75), ABC (64/75), DE (63/75), PSO (65/75) and GA (65/75) functions in 30 s of running (Table 7). Notably, the JS efficiently solves unimodal and multimodal functions as well as separate, non-separable, expanded and hybrid composite functions.

Table 5b
Optimization results of JS and WOA for CEC2005 functions (CF15-CF25).

Function		JS		WOA	
		Mean	Std.	Mean	Std.
CF15	D=10	1.2008E+02	2.2650E-02	6.5323E+02	1.0265E+02
	D=30	1.2030E+02	4.3559E-02	9.7212E+02	1.8599E+02
	D=50	1.2075E+02	9.3637E-02	1.0092E+03	2.4080E+02
CF16	D=10	1.2002E+02	7.5145E-03	3.9463E+02	6.3761E+01
	D=30	1.2010E+02	1.6650E-02	6.6210E+02	8.8326E+01
	D=50	1.2029E+02	3.6579E-02	8.0463E+02	1.1758E+02
CF17	D=10	1.2003E+02	8.9469E-03	4.5262E+02	7.5778E+01
	D=30	1.2012E+02	2.7902E-02	8.1564E+02	1.2122E+02
	D=50	1.2038E+02	4.7804E-02	1.0093E+03	1.7388E+02
CF18	D=10	1.2363E+01	3.5116E-01	1.0428E+03	6.9689E+01
	D=30	1.5710E+01	5.0164E-01	1.1497E+03	1.0835E+02
	D=50	2.3552E+01	3.0653E+00	1.1344E+03	2.0698E+02
CF19	D=10	1.5764E+02	2.3006E+01	1.0335E+03	7.4746E+01
	D=30	2.8483E+02	1.8846E+01	1.1422E+03	9.7412E+01
	D=50	4.1156E+02	2.0677E+01	1.1656E+03	2.1450E+02
CF20	D=10	8.5552E+02	1.0662E+02	1.0441E+03	6.5683E+01
	D=30	9.2790E+02	4.5222E+01	1.1390E+03	1.0395E+02
	D=50	9.1001E+02	8.3177E-03	1.1486E+03	2.1338E+02
CF21	D=10	3.6082E+02	2.4619E-01	1.5621E+03	2.0367E+02
	D=30	3.6611E+02	1.4866E+00	1.6552E+03	6.0127E+01
	D=50	3.8514E+02	4.6349E+00	1.7627E+03	4.3167E+01
CF22	D=10	3.8792E+02	1.1331E+01	1.3329E+03	5.8449E+01
	D=30	6.7314E+02	8.3737E+01	1.6625E+03	1.1953E+02
	D=50	1.4446E+03	1.6218E+02	1.8061E+03	1.0398E+02
CF23	D=10	5.3915E+02	4.0474E+02	1.6032E+03	1.4461E+02
	D=30	3.7710E+02	2.0617E+00	1.6858E+03	5.5016E+01
	D=50	4.1334E+02	7.3211E+00	1.7697E+03	3.2748E+01
CF24	D=10	4.6855E+02	3.9284E+01	1.2466E+03	4.2806E+02
	D=30	7.7144E+02	3.0221E+01	1.6506E+03	9.4291E+01
	D=50	6.6414E+02	3.9545E+01	1.7192E+03	2.7399E+01
CF25	D=10	1.2552E+03	2.0279E+02	1.5028E+03	2.6030E+02
	D=30	1.6244E+03	1.3541E+01	1.6924E+03	3.6518E+01
	D=50	1.6529E+03	1.1415E+01	1.7223E+03	3.4950E+01

Std. = Standard deviation.

To obtain the global optimum, an algorithm should be able to escape local optima, which requires a balance between exploration and exploitation. A good optimization algorithm should not converge prematurely to some local optimum [10]. The convergence curves of the JS for exemplary unimodal and multimodal benchmark functions are compared with WOA, TSA, SOS, TLBO, FA, GSA, ABC, DE, PSO and GA in Figs. 12 and 13. These figures show that the JS algorithm is superior to the other algorithms in all cases. Particularly, the curves of unimodal functions (F2, F5 and F15 in Fig. 12 and CF4 in Fig. 13) show that the JS has a remarkable characteristic to quickly exploit the promising areas without any disruption.

Similarly, the experimental results obtained by JS for multimodal functions (F22, F35 and F42 in Fig. 12; CF10 and CF12 in Fig. 13) demonstrate the diverse ability of the algorithm to escape local optima very quickly, which was attained by time control mechanism of JS algorithm. Finally, for the expanded function (CF14) and hybrid composite functions (CF16 and CF21) as shown in Fig. 13, the JS focused on exploration at the early iterations and switched to the exploitation phase near the end of iterations. The convergence graphs obviously show that the JS algorithm increases solution quality at early phase, escapes the local optima, and rapidly moving toward convergence of global optimum.

Results of analysis and visualization indicate that the JS algorithm well balances exploitation and exploration for I&D. This can be attributed to the merits of the algorithmic structure of JS, which include following ocean current (exploration), motions inside swarms of jellyfish (exploitation), and switching these motions by a time control mechanism (balance of exploitation and exploration).

5. Discrete design optimization of tower structures

This section presents the optimization results that were obtained by using JS algorithm to solve the three engineering optimization problems of the 25-bar, the 52-bar and the 582-bar towers, which are “small”-scale, “average”-scale and “large”-scale problems, respectively. Structural analyses were performed using the finite element method. The following subsections detail each problem and present the sensitivity of the internal parameters to optimum solution.

Table 5c
Optimization results of TSA, SOS and TLBO for CEC2005 functions (CF1-CF14).

Function		TSA		SOS		TLBO	
		Mean	Std.	Mean	Std.	Mean	Std.
CF1	D=10	-4.5000E+02	6.7430E-09	-4.5000E+02	0.0000E+00	-4.5000E+02	6.1549E-14
	D=30	-3.7577E+02	2.8866E+01	-4.5000E+02	1.6147E-13	-4.4699E+02	1.6378E+01
	D=50	3.6913E+03	5.6057E+02	-4.5000E+02	5.2445E-05	-3.1810E+02	1.6963E+02
CF2	D=10	-4.4199E+02	3.6659E+00	-4.5000E+02	2.3603E-14	-4.5000E+02	2.8377E-11
	D=30	3.5874E+04	5.7561E+03	-3.0306E+02	1.0062E+02	9.0772E+01	5.0637E+02
	D=50	1.3482E+05	1.2492E+04	7.2816E+03	2.7538E+03	1.0912E+04	3.6412E+03
CF3	D=10	2.9568E+06	1.2216E+06	1.2774E+05	9.6104E+04	2.4354E+05	1.3165E+05
	D=30	2.3168E+08	3.6955E+07	4.9882E+06	1.8355E+06	7.9732E+06	3.8561E+06
	D=50	8.3912E+08	1.1037E+08	2.2909E+07	1.1078E+07	5.2127E+07	1.8484E+07
CF4	D=10	-4.2772E+02	1.3105E+01	-4.5000E+02	6.2448E-14	-4.5000E+02	3.0580E-07
	D=30	4.9364E+04	6.4569E+03	4.3953E+03	2.2990E+03	7.8401E+03	3.2076E+03
	D=50	1.6222E+05	1.4486E+04	3.0707E+04	7.8018E+03	3.7042E+04	9.5660E+03
CF5	D=10	-3.0975E+02	2.4691E-01	-3.1000E+02	1.6083E-12	-3.1000E+02	1.5423E-10
	D=30	7.3763E+03	7.7898E+02	4.2522E+03	1.0163E+03	4.5478E+03	9.4085E+02
	D=50	2.3068E+04	1.3384E+03	1.1324E+04	1.5709E+03	1.3526E+04	1.9919E+03
CF6	D=10	3.9990E+02	2.9435E+00	3.9894E+02	2.8375E+01	4.0161E+02	2.8114E+01
	D=30	2.7079E+06	1.0149E+06	4.9087E+02	6.7646E+01	3.0496E+03	4.2618E+03
	D=50	1.0683E+09	2.5852E+08	8.4783E+02	3.8989E+02	4.5472E+06	3.3660E+06
CF7	D=10	1.0870E+03	5.2638E-07	1.0870E+03	4.5081E-13	1.0870E+03	4.5670E-13
	D=30	4.5180E+03	2.9328E-01	4.5163E+03	2.0477E-12	4.5163E+03	9.0403E-06
	D=50	6.0930E+03	2.2636E+01	6.0155E+03	5.8952E-01	6.0232E+03	1.8305E+01
CF8	D=10	-1.1956E+02	9.1623E-02	-1.1957E+02	7.8130E-02	-1.1954E+02	1.0037E-01
	D=30	-1.1896E+02	4.3760E-02	-1.1896E+02	5.9248E-02	-1.1892E+02	3.6231E-02
	D=50	-1.1878E+02	4.1685E-02	-1.1878E+02	4.3135E-02	-1.1877E+02	4.0666E-02
CF9	D=10	-3.0590E+02	4.5054E+00	-3.2169E+02	2.9421E+00	-3.2248E+02	2.9237E+00
	D=30	-1.1066E+02	1.1488E+01	-2.1951E+02	2.0244E+01	-2.3887E+02	1.5775E+01
	D=50	1.3269E+02	1.7833E+01	-6.5510E+01	4.0950E+01	-1.0363E+02	4.1544E+01
CF10	D=10	-2.9154E+02	4.5209E+00	-2.9961E+02	7.4435E+00	-3.0715E+02	7.7817E+00
	D=30	-7.1047E+01	1.4265E+01	-9.5523E+01	2.5487E+01	-9.6098E+01	2.9086E+01
	D=50	2.0552E+02	3.0683E+01	1.8125E+02	5.0021E+01	2.1026E+02	6.1931E+01
CF11	D=10	9.9121E+01	6.6643E-01	9.4335E+01	1.6895E+00	9.7448E+01	1.4493E+00
	D=30	1.3206E+02	9.3789E-01	1.2617E+02	5.3066E+00	1.3089E+02	1.2168E+00
	D=50	1.6688E+02	1.4014E+00	1.6251E+02	3.5644E+00	1.6596E+02	1.9926E+00
CF12	D=10	2.9263E+04	9.2267E+03	6.5977E+03	2.6475E+03	3.4696E+04	1.2006E+04
	D=30	1.0986E+06	1.2850E+05	4.5760E+05	9.5046E+04	9.7923E+05	1.7117E+05
	D=50	5.6003E+06	4.5330E+05	2.4628E+06	2.7832E+05	5.3626E+06	4.8373E+05
CF13	D=10	-1.2718E+02	3.9938E-01	-1.2866E+02	3.6240E-01	-1.2870E+02	3.4576E-01
	D=30	-1.0527E+02	1.4695E+00	-1.1588E+02	1.5433E+00	-1.2157E+02	3.1000E+00
	D=50	1.4197E+01	3.2135E+01	-9.6430E+01	3.9010E+00	-1.0871E+02	6.5870E+00
CF14	D=10	-2.9612E+02	1.3966E-01	-2.9656E+02	2.1123E-01	-2.9645E+02	2.3972E-01
	D=30	-2.8619E+02	1.1477E-01	-2.8652E+02	2.1435E-01	-2.8643E+02	1.5448E-01
	D=50	-2.7638E+02	1.8083E-01	-2.7666E+02	1.8333E-01	-2.7662E+02	1.8641E-01

Std. = Standard deviation.

5.1. Formulation of optimization problem

The purpose of this problem is to minimize structural weight or volume under constraints of element stresses and nodal displacements [73]. The problem of a structure with nm elements (in ng groups) and nn nodes can be stated as follows.

Minimize

$$W(A) = \sum_{i=1}^{nm} \gamma_i A_i \sqrt{(x_{i1} - x_{i2})^2 + (y_{i1} - y_{i2})^2 + (z_{i1} - z_{i2})^2} \tag{22}$$

Subject to

$$\sigma_i^c \le \sigma_i \le \sigma_i^t, \ i = 1, 2, \ldots, nm \tag{23}$$

$$\delta_{min} \le \delta_j \le \delta_{max}, \ j = 1, 2 \ldots, nn \tag{24}$$

$$A_{min} \le A_k \le A_{min}, \ k = 1, 2, \ldots, ng \tag{25}$$

Table 5d
Optimization results of TSA, SOS and TLBO for CEC2005 functions (CF15-CF25).

Function		TSA		SOS		TLBO	
		Mean	Std.	Mean	Std.	Mean	Std.
CF15	D=10	6.6465E+02	5.2974E+01	5.2434E+02	1.0278E+02	5.1433E+02	1.0546E+02
	D=30	8.3077E+02	5.2667E+01	6.4337E+02	5.6401E+01	6.6584E+02	6.6519E+01
	D=50	9.8510E+02	5.7939E+01	6.9108E+02	8.3189E+01	7.3218E+02	7.9429E+01
CF16	D=10	3.4257E+02	1.8783E+01	3.0506E+02	2.1927E+01	3.0472E+02	3.2300E+01
	D=30	5.5907E+02	2.8058E+01	4.5862E+02	7.3346E+01	5.0818E+02	1.0331E+02
	D=50	7.0750E+02	4.5503E+01	5.2674E+02	3.5868E+01	5.8158E+02	5.8711E+01
CF17	D=10	3.7310E+02	2.6704E+01	3.3833E+02	2.2602E+01	3.2085E+02	2.8284E+01
	D=30	6.5915E+02	6.8668E+01	5.2198E+02	1.0487E+02	5.2747E+02	1.0149E+02
	D=50	8.2988E+02	5.6143E+01	5.8324E+02	6.3752E+01	6.4233E+02	7.0682E+01
CF18	D=10	9.2171E+02	5.8770E+01	8.0628E+02	1.9635E+02	7.8854E+02	1.9785E+02
	D=30	1.0323E+03	1.8388E+01	9.4155E+02	1.9758E+01	9.6278E+02	2.6653E+01
	D=50	1.2249E+03	2.8586E+01	9.9715E+02	5.2485E+01	1.0737E+03	7.5896E+01
CF19	D=10	9.3126E+02	6.3745E+01	7.8969E+02	2.3794E+02	7.6432E+02	1.8910E+02
	D=30	1.0181E+03	1.4093E+01	9.3606E+02	9.8172E+00	9.5705E+02	1.4931E+01
	D=50	1.2975E+03	2.4751E+01	1.0218E+03	4.0891E+01	1.0773E+03	7.8152E+01
CF20	D=10	9.2477E+02	5.5922E+01	8.0232E+02	2.2934E+02	7.6495E+02	1.6210E+02
	D=30	1.0369E+03	2.1567E+01	9.3645E+02	1.2040E+01	9.6052E+02	1.9619E+01
	D=50	1.2298E+03	2.7515E+01	1.0189E+03	5.4611E+01	1.0744E+03	8.1996E+01
CF21	D=10	1.4424E+03	8.6845E+01	1.0611E+03	3.2999E+02	1.1789E+03	3.1262E+02
	D=30	1.5696E+03	1.8608E+01	1.1607E+03	2.8360E+02	1.5204E+03	1.3562E+02
	D=50	1.6500E+03	2.7223E+01	1.4777E+03	1.3587E+02	1.6213E+03	6.3242E+01
CF22	D=10	1.2297E+03	3.1588E+01	1.1512E+03	7.6975E+01	1.1603E+03	3.1917E+01
	D=30	1.5975E+03	5.5864E+01	1.3841E+03	3.8244E+01	1.4294E+03	5.0928E+01
	D=50	1.6895E+03	3.2591E+01	1.4604E+03	3.9068E+01	1.5299E+03	4.5575E+01
CF23	D=10	1.5047E+03	1.3801E+02	1.2401E+03	2.2377E+02	1.2789E+03	2.7432E+02
	D=30	1.5651E+03	1.9266E+01	1.2326E+03	2.6296E+02	1.5054E+03	1.1273E+02
	D=50	1.6512E+03	2.3700E+01	1.4929E+03	1.2957E+02	1.6430E+03	3.2443E+01
CF24	D=10	8.7083E+02	1.3048E+02	5.2019E+02	1.2242E+02	5.2334E+02	1.2914E+02
	D=30	1.4671E+03	3.9492E+01	9.7746E+02	3.9786E+02	1.2633E+03	2.6514E+02
	D=50	1.6631E+03	3.2342E+01	1.5497E+03	2.8813E+01	1.5933E+03	2.5909E+01
CF25	D=10	1.1821E+03	2.7925E+01	1.1141E+03	9.6469E+01	1.1253E+03	1.4181E+02
	D=30	1.5717E+03	2.0888E+01	1.4745E+03	1.0650E+02	1.5542E+03	6.9405E+01
	D=50	1.6790E+03	2.1252E+01	1.6019E+03	3.4807E+01	1.6216E+03	1.7013E+01

Std. = Standard deviation.

where W(A) is the weight of the structure; A is a vector of the sizing variables (cross-sectional areas of bars), which are the coordinates $x_{i1,2}$, $y_{i1,2}$, $z_{i1,2}$ of the nodes that limit the i^{th} element of the structure; γ_i denotes the material density of member i; and A_i is the cross-sectional area of member i, which can range between A_{min} and A_{max}.

To solve the stress and displacement constraints, the penalty function F_p is applied:

$$F_p = W(A) \times (1+\phi)^{\varepsilon} \tag{26}$$

where ϕ is the sum of stress and displacement penalties, and is given by

$$\phi = \sum_{i=1}^{nm} \phi_\sigma^i + \sum_{i=1}^{nn} \phi_\delta^j \tag{27}$$

The stress constraint penalty ϕ_σ^i of the i^{th} member and the displacement constraint penalty ϕ_δ^j of the j^{th} node are respectively defined as

$$\begin{cases} \phi_\sigma^i = 0 if\ \sigma_i^c \le \sigma_i \le \sigma_i^t \\ \phi_\sigma^i = \dfrac{\left|\sigma_i - \sigma^{t,c}\right|}{\left|\sigma^{t,c}\right|}\ if\ \sigma_i \langle \sigma_i^c\ or\ \sigma_i \rangle \sigma_i^t \end{cases} \tag{28}$$

$$\begin{cases} \phi_\delta^j = 0\ if\ \delta_{min} \le \delta_j \le \delta_{max} \\ \phi_\delta^j = \dfrac{\left|\delta_j - \delta^{min,max}\right|}{\left|\delta^{min,max}\right|}\ if\ \delta_j \langle \delta_{min}\ or\ \delta_j \rangle \delta_{max} \end{cases} \tag{29}$$

The exponent ε in the penalty function F_p is a function of the number of iterations, and is defined by

$$\varepsilon = \varepsilon_o \left(1 + \frac{t}{Max_{iter}}\right) \tag{30}$$

Table 5e
Optimization results of FA, GSA and ABC for CEC2005 functions (CF1-CF14).

Function		FA		GSA		ABC	
		Mean	Std.	Mean	Std.	Mean	Std.
CF1	D=10	-4.5000E+02	5.2723E-04	-4.5000E+02	1.0818E-05	-4.5000E+02	7.5333E-08
	D=30	-4.4998E+02	5.1249E-03	9.1308E+02	5.7846E+02	-4.4873E+02	4.6806E-01
	D=50	-4.4990E+02	1.8786E-02	3.0864E+04	3.9225E+03	4.6785E+02	1.2452E+02
CF2	D=10	-4.5000E+02	2.2281E-03	-3.2344E+02	6.7130E+01	-4.2988E+02	1.0915E+01
	D=30	1.9947E+03	1.0654E+03	1.9502E+04	2.1232E+03	3.0079E+04	4.7070E+03
	D=50	1.8596E+04	4.2676E+03	5.1379E+04	8.0481E+03	1.2140E+05	1.1808E+04
CF3	D=10	2.1189E+05	1.7158E+05	2.5483E+05	1.7683E+05	3.0308E+06	1.0476E+06
	D=30	6.6469E+06	2.9226E+06	3.4070E+07	2.4400E+07	1.7448E+08	3.5868E+07
	D=50	2.0013E+07	5.5491E+06	1.0217E+09	6.2318E+08	7.2024E+08	1.1665E+08
CF4	D=10	-4.4998E+02	9.4292E-03	9.7027E+03	1.8388E+03	-3.8355E+02	2.9342E+01
	D=30	1.1694E+04	3.5036E+03	6.2103E+04	1.2153E+04	4.0083E+04	6.1662E+03
	D=50	4.5131E+04	1.1627E+04	4.4682E+05	1.7768E+05	1.4909E+05	2.0278E+04
CF5	D=10	-3.0553E+02	1.2052E+00	3.6341E+03	1.2657E+03	-3.0998E+02	1.5108E-02
	D=30	1.6934E+03	2.6642E+02	2.2635E+04	2.6474E+03	5.9504E+03	6.5875E+02
	D=50	4.7282E+03	7.1775E+02	3.0361E+04	1.0975E+03	2.0936E+04	1.2091E+03
CF6	D=10	1.4071E+03	2.2011E+03	6.0892E+02	4.1373E+02	4.0305E+02	1.0103E+01
	D=30	3.4717E+03	3.7560E+03	1.1796E+08	7.4427E+07	4.6398E+05	1.7171E+05
	D=50	4.0318E+03	4.1695E+03	3.3747E+09	9.3485E+08	3.8599E+08	9.4955E+07
CF7	D=10	1.0889E+03	1.3360E+00	2.4739E+03	1.7241E+02	1.0870E+03	7.0954E-11
	D=30	4.5531E+03	1.6644E+01	1.1764E+04	4.1409E+02	4.5163E+03	1.1582E-05
	D=50	6.1144E+03	2.5518E+01	1.6369E+04	7.0527E+02	6.0162E+03	1.8672E-01
CF8	D=10	-1.1956E+02	8.1611E-02	-1.1966E+02	1.2302E-01	-1.1960E+02	6.7328E-02
	D=30	-1.1895E+02	4.5285E-02	-1.1955E+02	1.1004E-01	-1.1897E+02	3.9747E-02
	D=50	-1.1878E+02	3.5252E-02	-1.1951E+02	6.9018E-02	-1.1880E+02	5.2563E-02
CF9	D=10	-3.2536E+02	2.0296E+00	-3.2611E+02	1.7362E+00	-3.0693E+02	3.5525E+00
	D=30	-3.0069E+02	1.1860E+01	-2.8805E+02	7.6718E+00	-1.3200E+02	1.6073E+01
	D=50	-2.7175E+02	1.2786E+01	-2.2341E+02	1.3498E+01	8.9870E+01	1.9166E+01
CF10	D=10	-3.2321E+02	3.5961E+00	-3.2589E+02	2.0860E+00	-2.9320E+02	5.2157E+00
	D=30	-2.9795E+02	7.6788E+00	-2.9415E+02	7.9788E+00	-8.2134E+01	1.5132E+01
	D=50	-2.4875E+02	1.6511E+01	-2.2054E+02	1.7901E+01	1.8071E+02	1.9520E+01
CF11	D=10	9.3830E+01	1.4001E+00	9.0841E+01	3.9574E-01	9.8859E+01	8.2486E-01
	D=30	1.1522E+02	3.2294E+00	9.5626E+01	1.6532E+00	1.3168E+02	1.1574E+00
	D=50	1.4313E+02	4.7390E+00	1.0800E+02	3.3159E+00	1.6664E+02	2.0000E+00
CF12	D=10	4.2285E+02	8.6866E+02	3.4809E+02	1.5272E+03	3.0255E+04	8.9725E+03
	D=30	4.2515E+04	1.5101E+04	1.2710E+04	1.3869E+04	1.0609E+06	1.0856E+05
	D=50	3.9237E+05	2.3133E+05	7.2830E+04	4.2222E+04	5.4516E+06	3.9182E+05
CF13	D=10	-1.2914E+02	2.5473E-01	-1.2850E+02	3.9226E-01	-1.2757E+02	4.0539E-01
	D=30	-1.2482E+02	1.5043E+00	-1.2442E+02	1.2407E+00	-1.0782E+02	1.5315E+00
	D=50	-1.1739E+02	2.4519E+00	-1.2093E+02	2.0312E+00	-3.7779E+01	1.1561E+01
CF14	D=10	-2.9649E+02	3.9606E-01	-2.9552E+02	1.4039E-01	-2.9617E+02	1.6100E-01
	D=30	-2.8681E+02	3.4392E-01	-2.8582E+02	2.1213E-01	-2.8623E+02	1.3339E-01
	D=50	-2.7696E+02	3.8975E-01	-2.7614E+02	2.2148E-01	-2.7641E+02	1.6314E-01

Std. = Standard deviation.

where t is the number of the current iteration and Max_{iter} is maximum number of iterations. The initial value ε_0 can be chosen between 1.001 and 10,000 [73]. A sensitivity analysis shows that $\varepsilon_0 = 2$ is effective to solve this problem herein (Ref. Section 5.3).

5.2. Determination of population size and number of iterations

Population size and number of iterations are the two controlling parameters in JS algorithm. The number of iterations provides the stop criterion and can be set to a very high value (e.g., 1000 iterations). Another stop condtion is to use ten consecutive rates of change in the objective function with values of under 10^{-12}. In this study, the population size was set from 10 to 500. Three structural problems were used to analyze the sensitivity of population size to the optimal value; they were the 25-bar tower (small-scale problem with 8 variables), the 52-bar tower (average-scale problem with 16 variables), and the 582-bar tower (large-scale with 32 variables)

Table 8 reports the results that were obtained after 30 independent runs, and Table 9 displays details of the structural weight and structural analysis number that were obtained from each independent run in solving the three structural problems. In fact, with a population size of 40, JS could have reached the best optimum value in the fewest structural analyses. However, when the population size was lower than 40, JS either did not reach the optimum value or reached the optimum value in more structural analyses than required at a population size of 40. When the population size was higher than 40, JS reached the optimum design, but it required at least 1.3 times as many structural analyses as with a population size of 40. The results indicated that to optimize the performance of JS, the population size should be set to 40, and the number of

Table 5f
Optimization results of FA, GSA and ABC for CEC2005 functions (CF15-CF25).

Function		FA		GSA		ABC	
		Mean	Std.	Mean	Std.	Mean	Std.
CF15	D=10	6.4220E+02	1.0775E+02	4.3698E+02	1.8272E+02	6.1164E+02	6.1118E+01
	D=30	7.6278E+02	7.4978E+01	6.0016E+02	2.9520E+01	7.9045E+02	4.4727E+01
	D=50	7.2858E+02	1.0227E+02	6.0029E+02	7.6492E+00	8.9783E+02	5.3642E+01
CF16	D=10	3.0612E+02	1.6358E+01	2.8078E+02	1.6344E+01	3.2912E+02	1.9936E+01
	D=30	4.6111E+02	1.1999E+02	4.9656E+02	1.7703E+02	4.9683E+02	3.0706E+01
	D=50	4.6809E+02	5.4836E+01	4.1472E+02	7.8937E+01	6.1162E+02	3.2250E+01
CF17	D=10	3.4748E+02	2.9368E+01	3.0701E+02	1.3786E+01	3.5802E+02	2.3220E+01
	D=30	5.3758E+02	1.2977E+02	5.2602E+02	1.7489E+02	5.6883E+02	3.9486E+01
	D=50	5.3468E+02	5.8233E+01	5.8821E+02	1.1142E+02	7.4521E+02	5.8630E+01
CF18	D=10	9.2813E+02	1.4600E+02	1.0141E+03	6.4754E+01	8.6544E+02	4.8827E+01
	D=30	9.5300E+02	6.0329E+00	1.0274E+03	3.4472E+01	9.6581E+02	9.1495E+00
	D=50	1.0178E+03	1.3914E+01	1.0571E+03	1.0030E+01	1.1450E+03	3.0000E+01
CF19	D=10	8.8809E+02	1.4956E+02	1.0246E+03	3.1998E+01	8.8280E+02	6.1081E+01
	D=30	9.5402E+02	4.9789E+00	1.0348E+03	2.2022E+01	9.6708E+02	8.8420E+00
	D=50	1.0148E+03	1.3114E+01	1.0548E+03	9.7849E+00	1.1446E+03	1.9447E+01
CF20	D=10	9.3612E+02	1.4496E+02	1.0109E+03	5.0163E+01	8.4328E+02	7.8685E+01
	D=30	9.5400E+02	6.4410E+00	1.0255E+03	1.9295E+01	9.6904E+02	7.0157E+00
	D=50	1.0203E+03	1.2999E+01	1.0518E+03	1.0804E+01	1.1394E+03	2.1064E+01
CF21	D=10	1.4707E+03	2.0903E+02	1.3625E+03	8.8039E+01	1.2517E+03	2.0955E+02
	D=30	1.2333E+03	3.3087E+01	1.3402E+03	1.9812E+02	1.4106E+03	3.4653E+01
	D=50	1.3749E+03	4.3129E+01	1.5812E+03	5.1397E+00	1.5015E+03	1.5264E+01
CF22	D=10	1.1837E+03	2.4863E+01	1.2260E+03	4.9746E+01	1.1786E+03	1.7274E+01
	D=30	1.4318E+03	3.4470E+01	1.4573E+03	3.8299E+01	1.5199E+03	4.2264E+01
	D=50	1.4543E+03	1.6835E+01	1.5357E+03	1.8216E+01	1.5482E+03	2.9915E+01
CF23	D=10	1.4768E+03	1.8450E+02	1.5059E+03	4.2998E+01	1.3507E+03	1.7941E+02
	D=30	1.2890E+03	3.5560E+01	1.4541E+03	1.5298E+02	1.4252E+03	4.0302E+01
	D=50	1.3817E+03	7.2897E+01	1.5855E+03	6.3095E+00	1.5187E+03	1.7135E+01
CF24	D=10	7.0651E+02	1.1459E+02	1.4966E+03	1.9747E+02	4.7491E+02	1.8727E+01
	D=30	1.0097E+03	4.1928E+01	9.9142E+02	3.1021E+02	1.3204E+03	3.5709E+01
	D=50	1.3363E+03	7.5034E+01	1.5127E+03	7.1451E+00	1.5739E+03	2.1966E+01
CF25	D=10	1.1520E+03	1.9765E+01	1.6485E+03	1.0340E+01	1.1008E+03	6.4278E+00
	D=30	1.4289E+03	3.0457E+01	1.6134E+03	8.7491E+00	1.4844E+03	2.1372E+01
	D=50	1.5753E+03	8.5066E+00	1.6150E+03	9.7741E+00	1.6155E+03	1.1827E+01

Std. = Standard deviation.

structural analyses are 600 (15 iterations), 3,720 (93 iterations) and 14,840 (371 iterations) for 25-bar, 52-bar and 582-bar tower problems, respectively

5.3. Sensitivity analysis of initial value ε_o

Initial value ε_o (Eq. (30)) effects the penalty function F_p in solving the stress and displacement constraints. In this investigation, ε_o was set to various cases (case 1 (ε_o=1.001), case 2 (ε_o=1.5), case 3 (ε_o=2), case 4 (ε_o=2.5), case 5 (ε_o=3), case 6 (ε_o= 4), case 7 (ε_o=5), case8 (ε_o=10), case 9 (ε_o=100), case 10 (ε_o=500), case 11 (ε_o=1,000), and case 12 (ε_o=10,000)). Based on the results of previous subsection, the population size was set to 40 for the structural design problems (25-bar tower, 52-bar tower and 582-bar tower). After running 30 times for each case, statistical box plots are presented in Fig.14. It shows that case 3 ($\varepsilon_o = 2$) has narrower interquartile range and lower median than the other cases in all three problems. Fig. 14 presents that the case 3 gives a much more stable result of global optimum. Therefore, $\varepsilon_o = 2$ was chosen to generate initial penalty function for solving these problems.

5.4. Structural tower designs

5.4.1. 25-Bar tower

The first structural optimization problem that is solved in this study involves the spatial 25-bar tower that is displayed in Fig. 15a. The material density, modulus of elasticity, and stress limitations of the members in this structure are 0.1 lb/in^3, 10,000 ksi, and ± 40,000 psi, respectively. All nodes of the structure are subject to a displacement limitation of ±0.35 (in) in the X, Y, and Z directions, and the loads at the nodes are set to $P_{1x} = 1$ kips, $P_{3x} = 0.5$ kips, $P_{6x} = 0.6$ kips, $P_{1y} = P_{1z} = P_{2y} = P_{2z} = -10$ kips . The members of the structure were divided into eight groups, which are (1) A1, (2) A2–A5, (3) A6–A9, (4) A10–A11, (5) A12–A13, (6) A14–A17, (7) A18–A21, and (8) A22–A25. The discrete cross-sectional areas, included in the example with eight optimization variables, are selected from the set D = {0.1, 0.2, 0.3, 0.4, 0.5,0.6, 0.7,0.8, 0.9,1.0,1.1,1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0,2.1,2.2, 2.3, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4} (in^2).

Table 5g
Optimization results of DE, PSO and GA for CEC2005 functions (CF1-CF14).

Function		DE		PSO		GA	
		Mean	Std.	Mean	Std.	Mean	Std.
CF1	D=10	-4.5000E+02	0.0000E+00	-4.3623E+02	3.5701E+01	-4.5000E+02	4.7385E-04
	D=30	-4.5000E+02	1.4986E-04	5.6271E+02	1.0895E+03	-4.4589E+02	3.3762E+00
	D=50	-4.4851E+02	2.2625E-01	3.5764E+03	2.4934E+03	-3.3525E+02	4.9380E+01
CF2	D=10	-4.3504E+02	8.5143E+00	-4.3854E+02	3.7425E+01	-3.6494E+02	5.2935E+01
	D=30	2.1649E+04	3.3476E+03	1.9462E+02	2.8863E+02	3.3097E+03	1.3953E+03
	D=50	1.0479E+05	1.3198E+04	8.1961E+03	4.9480E+03	2.8380E+04	6.4453E+03
CF3	D=10	1.7630E+06	7.4714E+05	1.9764E+05	1.2636E+05	1.5185E+06	1.9489E+06
	D=30	1.2924E+08	2.8223E+07	9.6368E+06	4.9671E+06	2.7419E+07	1.0001E+07
	D=50	5.7569E+08	1.0358E+08	4.4040E+07	2.5887E+07	1.2129E+08	3.9397E+07
CF4	D=10	-4.4758E+02	1.7134E+00	-4.3368E+02	4.4558E+01	-2.8603E+01	2.5537E+02
	D=30	3.4469E+04	7.2791E+03	2.5623E+03	1.1345E+03	1.5665E+04	5.4464E+03
	D=50	1.2996E+05	1.2678E+04	2.6824E+04	7.3831E+03	5.4614E+04	1.6563E+04
CF5	D=10	-3.1000E+02	5.1447E-10	-3.1000E+02	0.0000E+00	9.6310E+02	1.0240E+03
	D=30	5.0300E+03	7.0513E+02	6.4027E+03	2.5863E+03	6.9663E+03	1.7419E+03
	D=50	1.7387E+04	1.5735E+03	1.0818E+04	2.5432E+03	1.4425E+04	2.5760E+03
CF6	D=10	4.0543E+02	7.1944E+00	4.1122E+02	4.3173E+01	1.2472E+03	2.3685E+03
	D=30	7.7502E+02	1.0057E+02	2.9500E+07	3.9422E+07	1.1176E+04	6.0462E+03
	D=50	2.0813E+04	6.6803E+03	1.8393E+08	2.5130E+08	5.3019E+05	4.0909E+05
CF7	D=10	1.0871E+03	8.6319E-02	1.0873E+03	1.8984E-01	1.0870E+03	2.6472E-03
	D=30	4.5163E+03	1.2411E-12	4.7945E+03	3.4723E+02	4.5163E+03	1.1278E-03
	D=50	6.0153E+03	1.2331E-08	6.9867E+03	6.4710E+02	6.0155E+03	1.2804E-01
CF8	D=10	-1.1957E+02	7.2938E-02	-1.1965E+02	7.8593E-02	-1.1954E+02	1.0465E-01
	D=30	-1.1897E+02	5.7807E-02	-1.1900E+02	5.5659E-02	-1.1891E+02	6.3446E-02
	D=50	-1.1879E+02	3.4874E-02	-1.1881E+02	4.5915E-02	-1.1874E+02	4.8331E-02
CF9	D=10	-3.3000E+02	5.4833E-10	-3.1468E+02	6.7965E+00	-3.3000E+02	3.5989E-03
	D=30	-2.5572E+02	5.7372E+00	-2.1174E+02	2.9755E+01	-3.0112E+02	6.8463E+00
	D=50	-1.1970E+02	8.8349E+00	-6.2178E+01	4.3770E+01	-2.0838E+02	2.1170E+01
CF10	D=10	-3.0843E+02	3.4366E+00	-3.1061E+02	8.2504E+00	-3.0422E+02	1.4007E+01
	D=30	-1.0670E+02	1.1740E+01	-2.0879E+02	3.6659E+01	-1.9835E+02	2.6339E+01
	D=50	1.2174E+02	2.5062E+01	-3.4454E+01	8.2174E+01	1.8592E+01	5.5918E+01
CF11	D=10	9.7435E+01	6.5998E-01	9.3391E+01	1.3313E+00	9.6795E+01	1.3146E+00
	D=30	1.3054E+02	1.0933E+00	1.1065E+02	3.5864E+00	1.1760E+02	4.0162E+00
	D=50	1.6639E+02	1.6183E+00	1.3388E+02	5.2808E+00	1.4083E+02	5.7895E+00
CF12	D=10	1.6996E+03	8.0016E+02	5.3708E+02	1.1410E+03	1.1021E+03	1.0916E+03
	D=30	3.0103E+05	3.5193E+04	1.5509E+05	1.7712E+05	1.5660E+05	4.6801E+04
	D=50	1.8995E+06	1.7651E+05	5.5310E+05	3.5623E+05	9.0384E+05	2.5204E+05
CF13	D=10	-1.2934E+02	1.0779E-01	-1.2934E+02	2.1229E-01	-1.2955E+02	1.1089E-01
	D=30	-1.2005E+02	8.4407E-01	-1.2606E+02	9.4801E-01	-1.2532E+02	7.4348E-01
	D=50	-1.0290E+02	1.0969E+00	-1.1945E+02	4.1982E+00	-1.1721E+02	2.7098E+00
CF14	D=10	-2.9630E+02	1.9376E-01	-2.9674E+02	5.0343E-01	-2.9636E+02	3.4388E-01
	D=30	-2.8632E+02	1.7778E-01	-2.8720E+02	3.3330E-01	-2.8667E+02	3.2965E-01
	D=50	-2.7644E+02	1.7465E-01	-2.7746E+02	3.5605E-01	-2.7690E+02	2.7324E-01

Std. = Standard deviation.

Figs. 15b and 15c display stress and displacement of each frame and node of 25-bar tower problem, respectively. Table 10 shows the statistical results of optimization for this test problem. The table indicates that all compared algorithms reached the optimum value (484.85 lb) and that JS and DAJA [73] required the fewest structural analyses-600 for JS and 511 for DAJA [73]. These results demonstrate that JS was one of the most efficient algorithms for solving the 25-bar tower problem.

5.4.2. 52-Bar tower

The second optimized structural problem involved the 52-bar tower that is shown in Fig. 16a. Attempts have been made to solve this problem using several optimization algorithms, such as HPSO [74], DHPSACO [75], CSS [76], CBO [77], TLBO [78], AFA [79], WCA [80], IMBA [80], HHS [81] and DAJA [73], among others. The members of this structure are divided into 12 groups, (1) A1–A4, (2) A5–A10, (3) A11–A13, (4) A14–A17, (5) A18–A23, (6) A24–A26, (7) A27-A30, (8) A31-A36, (9) A37-A39, (10) A40-A43, (11) A44-A49 and (12) A50–A52. The material density is 7860.0 kg/m^3 and the modulus of elasticity is 2.07×10^5 MPa. Each member had a stress limit of $\pm$180 MPa. Both loads P_x= 100 kN and P_y= 200 kN were applied at nodes 17–20. The discrete variables are adopted from Li et al. [74] and Sadollah et al. [74,80] on the basis of AISC specifications.

Fig. 16b displays stress constraint violation with all the frames in the stress limit. In Table 11, the three best optimal structural masses were obtained by JS (1899.678 kg), CBO (1899.35 kg) [77] and CSS (1897.62 kg) [76]. JS required the fewest analyses (3,720) while CBO and CSS required 3,840 and 5,000 analyses, respectively. Therefore, JS is the most efficient algorithm for solving this problem.

Table 5h
Optimization results of DE, PSO and GA for CEC2005 functions (CF15-CF25).

Function		DE		PSO		GA	
		Mean	Std.	Mean	Std.	Mean	Std.
CF15	D=10	4.9913E+02	5.3388E+01	4.3768E+02	1.8575E+02	3.5159E+02	1.7623E+02
	D=30	8.0885E+02	4.7697E+01	6.2461E+02	7.7714E+01	4.9273E+02	8.4094E+01
	D=50	8.6108E+02	6.3999E+01	6.6421E+02	5.4318E+01	5.3068E+02	6.4125E+01
CF16	D=10	3.4227E+02	2.5304E+01	2.5566E+02	2.8106E+01	2.5930E+02	2.4382E+01
	D=30	5.3534E+02	4.8340E+01	4.5462E+02	1.5936E+02	4.1300E+02	1.3500E+02
	D=50	6.1841E+02	4.5226E+01	4.4006E+02	9.3679E+01	4.6915E+02	6.8629E+01
CF17	D=10	3.6380E+02	3.2995E+01	2.5736E+02	2.2427E+01	2.6178E+02	2.5020E+01
	D=30	6.0631E+02	4.9807E+01	5.3442E+02	1.5456E+02	4.6382E+02	1.4067E+02
	D=50	7.4849E+02	7.6239E+01	5.8545E+02	1.1267E+02	5.3622E+02	7.1047E+01
CF18	D=10	9.7118E+02	5.2679E+01	9.0685E+02	1.4708E+02	9.1230E+02	1.8443E+02
	D=30	9.6327E+02	1.0120E+01	9.4364E+02	2.1143E+01	9.4756E+02	8.3966E+00
	D=50	1.1324E+03	2.3732E+01	9.8127E+02	2.9751E+01	1.0099E+03	2.3720E+01
CF19	D=10	9.7198E+02	4.9066E+01	9.5989E+02	6.8328E+01	9.7107E+02	7.8234E+01
	D=30	9.6056E+02	1.0166E+01	9.4174E+02	1.7242E+01	9.4447E+02	2.5534E+01
	D=50	1.1394E+03	3.2477E+01	9.7824E+02	2.7976E+01	1.0189E+03	1.9036E+01
CF20	D=10	9.6455E+02	6.2575E+01	9.3488E+02	1.0603E+02	9.7519E+02	7.2608E+01
	D=30	9.6131E+02	9.4227E+00	9.4277E+02	2.7732E+01	9.4479E+02	2.8828E+01
	D=50	1.1355E+03	2.4729E+01	9.7735E+02	1.7529E+01	1.0191E+03	2.6692E+01
CF21	D=10	1.3066E+03	1.9695E+02	1.1467E+03	3.6554E+02	1.3229E+03	2.3653E+02
	D=30	1.3255E+03	4.6071E+01	1.3530E+03	1.8017E+02	9.9627E+02	2.4417E+02
	D=50	1.4559E+03	1.3065E+01	1.3713E+03	1.2801E+02	1.1856E+03	2.2119E+02
CF22	D=10	1.2251E+03	2.8941E+01	1.1767E+03	7.1417E+01	1.1711E+03	5.5890E+01
	D=30	1.5233E+03	4.0516E+01	1.3621E+03	7.5979E+01	1.3859E+03	3.5374E+01
	D=50	1.5613E+03	2.6705E+01	1.4127E+03	4.5447E+01	1.4209E+03	2.2685E+01
CF23	D=10	1.3480E+03	1.6724E+02	1.2957E+03	2.4350E+02	1.4519E+03	2.0382E+02
	D=30	1.3679E+03	4.6073E+01	1.3109E+03	2.0009E+02	1.0769E+03	2.5970E+02
	D=50	1.4593E+03	1.4107E+01	1.4077E+03	1.1911E+02	1.2286E+03	2.1299E+02
CF24	D=10	7.5521E+02	2.1635E+02	1.0524E+03	3.5062E+02	5.2699E+02	1.8854E+02
	D=30	1.2910E+03	2.0336E+01	1.2078E+03	2.6437E+02	5.0985E+02	5.1258E+01
	D=50	1.5163E+03	3.4227E+01	1.5003E+03	5.2575E+01	1.4761E+03	5.2606E+01
CF25	D=10	1.1080E+03	1.1841E+01	1.4477E+03	2.0510E+02	1.2475E+03	2.3458E+02
	D=30	1.4393E+03	2.4457E+01	1.5417E+03	7.6100E+01	1.4809E+03	7.0120E+01
	D=50	1.5959E+03	9.4524E+00	1.5914E+03	2.2460E+01	1.5645E+03	8.8503E+00

Std. = Standard deviation.

5.4.3. 582-Bar tower

The last test problem involved the 582-bar tower structure that is shown in Fig. 17. This structure was originally proposed by Hasançebi et al. [82] and subsequently studied in detail by Kaveh and Talatahari [83]. The tower has 582 elements in 32 independent groups. A single loading condition of 5 kN lateral loads in both X- and Y-directions and −30 kN vertical loads in the Z-direction, applied to all nodes, is used. The cross sections and radii of gyration of the elements are selected from the list of 140 W-shape steel profiles (Table 12). The modulus of elasticity (E) and yield stress (F_y) are 200 GPa and 248 MPa, respectively.

The problem has three constraints in this problem. The first is that the displacements of each node do not exceed 8 cm. The second is a limit on stress according to ASD-AISC specifications [84]. The stress limits are set as follows.

$$\begin{cases} \sigma_i^+ = 0.6\ F_y\ \sigma_i \geq 0 \\ \sigma_i^-\ \sigma_i < 0 \end{cases} \tag{31}$$

where σ_i^- depends on the slenderness ratio:

$$\sigma_i^- = \begin{cases} \left[\dfrac{\left(1-\frac{\lambda_i^2}{2C_c^2}\right)F_y}{\frac{5}{3}+\frac{3\lambda_i}{8C_c}-\frac{\lambda_i^3}{8C_c^3}} \right] & for\ \lambda_i < C_c \\ \dfrac{12\pi^2 E}{23\lambda_i^2} & for\ \lambda_i \geq C_c \end{cases} \tag{32}$$

C_c is the slenderness ratio (λ) that divides the elastic from the inelastic buckling regions, and is defined as follows.

$$C_c = \sqrt{\frac{2\pi^2 E}{F_y}} \tag{33}$$

Table 6a
p-value of Wilcoxon rank sum test between JS and WOA, TSA, SOS, TLBO and FA algorithms for solving CEC2005 functions.

JS vs.		WOA	Sig.	TSA	Sig.	SOS	Sig.	TLBO	Sig.	FA	Sig.
CF1	D=10	**8.98E-12**	+	**8.98E-12**	+	1.00E+00	≈	9.99E-01	≈	**8.98E-12**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	1.00E+00	≈	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.67E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF2	D=10	**1.51E-11**	+	**1.51E-11**	+	1.00E+00	≈	1.00E+00	≈	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF3	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF4	D=10	**1.51E-11**	+	**1.51E-11**	+	1.00E+00	≈	**4.28E-04**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF5	D=10	**1.51E-11**	+	**3.81E-03**	+	1.00E+00	≈	1.00E+00	≈	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.01E-07**	+	1.00E+00	≈	9.98E-01	≈	1.00E+00	≈
	D=50	**1.51E-11**	+	**1.51E-11**	+	1.00E+00	≈	1.00E+00	≈	1.00E+00	≈
CF6	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF7	D=10	**5.00E-05**	+	**4.25E-02**	+	1.00E+00	≈	1.00E+00	≈	**1.51E-11**	+
	D=30	**1.46E-11**	+	**4.24E-09**	+	1.00E+00	≈	1.00E+00	≈	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	1.00E+00	≈	1.00E+00	≈	**1.51E-11**	+
CF8	D=10	7.30E-01	≈	**2.32E-03**	+	**1.08E-02**	+	**1.68E-04**	+	**2.66E-03**	+
	D=30	1.00E+00	≈	**1.82E-02**	+	**3.51E-02**	+	**2.73E-06**	+	**7.82E-03**	+
	D=50	1.00E+00	≈	3.48E-01	≈	3.59E-01	≈	5.94E-02	≈	5.09E-01	≈
CF9	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF10	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF11	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF12	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF13	D=10	**6.06E-13**	+	**6.06E-13**	+	**6.06E-13**	+	**6.06E-13**	+	**6.06E-13**	+
	D=30	**3.79E-12**	+	**3.79E-12**	+	**3.79E-12**	+	**3.79E-12**	+	**3.79E-12**	+
	D=50	**7.32E-12**	+	**7.32E-12**	+	**7.32E-12**	+	**7.32E-12**	+	**7.32E-12**	+
CF14	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF15	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF16	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF17	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF18	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF19	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF20	D=10	**1.02E-09**	+	**1.52E-03**	+	8.61E-01	≈	9.88E-01	≈	**1.50E-04**	+
	D=30	**1.74E-10**	+	**3.56E-09**	+	**4.15E-06**	+	**1.34E-06**	+	**2.06E-06**	+
	D=50	**1.82E-02**	+	**1.51E-11**	+	**3.06E-10**	+	**8.65E-07**	+	**1.51E-11**	+
CF21	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF22	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**7.32E-11**	+	**1.46E-09**	+	5.85E-01	≈	**3.07E-02**	+	6.52E-01	≈
CF23	D=10	**1.19E-10**	+	**6.56E-09**	+	**2.59E-07**	+	**1.19E-07**	+	**9.25E-09**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF24	D=10	**7.32E-11**	+	**1.51E-11**	+	1.00E+00	≈	9.95E-01	≈	**2.25E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	8.81E-02	≈	**4.45E-10**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF25	D=10	**2.78E-04**	+	**3.07E-02**	+	1.00E+00	≈	1.00E+00	≈	2.70E-01	≈
	D=30	**1.60E-09**	+	1.00E+00	≈	1.00E+00	≈	1.00E+00	≈	1.00E+00	≈
	D=50	**3.69E-11**	+	**3.80E-07**	+	1.00E+00	≈	1.00E+00	≈	1.00E+00	≈
	+	**72**		**73**		**57**		**61**		**68**	
	≈	**3**		**2**		**18**		**14**		**7**	

"+" indicates poorer performance than that of JS, and "≈" indicates no significant difference of performance between the compared algorithm and JS.

Table 6b
p-value of Wilcoxon rank sum test between JS and GSA, ABC, DE, PSO and GA algorithms for solving CEC2005 functions.

JS vs.		GSA	Sig.	ABC	Sig.	DE	Sig.	PSO	Sig.	GA	Sig.
CF1	D=10	**8.98E-12**	+	**8.98E-12**	+	**1.00E+00**	≈	1.00E+00	≈	**1.75E-07**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.67E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF2	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF3	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF4	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF5	D=10	**1.51E-11**	+	1.00E+00	≈	1.00E+00	≈	1.00E+00	≈	**1.51E-11**	+
	D=30	**1.51E-11**	+	**3.62E-02**	+	9.33E-01	≈	7.27E-02	≈	**5.57E-04**	+
	D=50	**1.51E-11**	+	**1.67E-11**	+	**5.53E-05**	+	1.00E+00	≈	9.69E-01	≈
CF6	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF7	D=10	**1.51E-11**	+	1.00E+00	≈	1.00E+00	≈	**1.09E-02**	+	9.98E-01	≈
	D=30	**1.51E-11**	+	1.00E+00	≈	1.00E+00	≈	**5.48E-07**	+	1.00E+00	≈
	D=50	**1.51E-11**	+	1.00E+00	≈	1.00E+00	≈	**2.79E-10**	+	1.00E+00	≈
CF8	D=10	8.30E-01	≈	1.11E-01	≈	**3.64E-03**	+	9.80E-01	≈	**8.40E-04**	+
	D=30	1.00E+00	≈	1.09E-01	≈	1.06E-01	≈	7.77E-01	≈	**1.39E-05**	+
	D=50	1.00E+00	≈	9.68E-01	≈	8.91E-01	≈	9.93E-01	≈	**1.00E-04**	+
CF9	D=10	**1.51E-11**	+	**1.51E-11**	+	1.00E+00	≈	**1.50E-11**	+	**3.06E-10**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF10	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.50E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF11	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF12	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF13	D=10	**6.06E-13**	+	**6.06E-13**	+	**6.06E-13**	+	**6.06E-13**	+	**6.06E-13**	+
	D=30	**3.79E-12**	+	**3.79E-12**	+	**3.79E-12**	+	**3.79E-12**	+	**3.79E-12**	+
	D=50	**7.32E-12**	+	**7.32E-12**	+	**7.32E-12**	+	**7.32E-12**	+	**7.32E-12**	+
CF14	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF15	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF16	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF17	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF18	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF19	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF20	D=10	**1.29E-07**	+	6.19E-01	≈	**4.33E-05**	+	**2.87E-02**	+	**1.78E-04**	+
	D=30	**7.79E-09**	+	**5.97E-07**	+	**9.30E-07**	+	**2.37E-06**	+	**1.58E-05**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF21	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF22	D=10	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.96E-02**	+	**5.16E-03**	+	**1.03E-03**	+	8.83E-01	≈	7.90E-01	≈
CF23	D=10	**2.93E-08**	+	**1.90E-07**	+	**1.51E-07**	+	**9.35E-08**	+	**2.75E-08**	+
	D=30	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF24	D=10	**2.04E-11**	+	**9.37E-08**	+	**1.44E-10**	+	**3.37E-06**	+	**1.00E+00**	**≈**
	D=30	**1.35E-02**	+	**1.51E-11**	+	**1.51E-11**	+	**5.33E-08**	+	**1.00E+00**	**≈**
	D=50	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+	**1.51E-11**	+
CF25	D=10	**1.51E-11**	+	1.00E+00	≈	1.00E+00	≈	**7.98E-04**	+	9.99E-01	≈
	D=30	1.00E+00	≈	1.00E+00	≈	1.00E+00	≈	1.00E+00	≈	1.00E+00	≈
	D=50	1.00E+00	≈	1.00E+00	≈	1.00E+00	≈	1.00E+00	≈	1.00E+00	≈
	+	**70**		**64**		**63**		**65**		**65**	
	≈	**5**		**11**		**12**		**10**		**10**	

"+" indicates poorer performance than that of JS, and "≈" indicates no significant difference of performance between the compared algorithm and JS.

Table 7a
Number of best hits in solving small/average-scale mathematical optimization problems.

Fun.	C	JS	WOA	TSA	SOS	TLBO	FA	GSA	ABC	DE	PSO	GA
F1-F5	US	5	4	4	4	4	2	4	4	4	4	1
F6-F17	UN	12	8	8	10	9	2	7	8	8	9	3
F18-F26	MS	8	5	6	3	5	1	4	9	5	4	5
F27-F50	MN	24	9	21	13	15	9	11	17	15	7	6
Total		49	26	39	30	33	14	26	38	32	24	15
Hit rate (%)		98%	52%	78%	60%	66%	28%	52%	76%	64%	48%	30%
Total computational time (Sec)		77.52	149.00	106.69	362.29	366.53	1707.30	1228.98	638.15	707.60	536.45	648.76

Nomenclature: characteristics (C), separable (S), non-separable (N), multimodal (M), unimodal (U), expanded (E) and hybrid composite (H) functions; "+" indicates poorer performance than that of JS, and "≈" indicates no significant difference of performance between compared algorithm and JS.

Table 7b
Results of Wilcoxon rank sum tests in solving high-scale mathematical optimization problems (CEC2005 functions).

Fun.	C	JS vs.	WOA	TSA	SOS	TLBO	FA	GSA	ABC	DE	PSO	GA
CF1-CF5	U	+	15	15	8	10	13	15	14	12	11	14
		≈	0	0	7	5	2	0	1	3	4	1
CF6-CF12	M	+	18	20	17	17	20	18	15	15	18	18
		≈	3	1	4	4	1	3	6	6	3	3
CF13-CF14	E	+	6	6	6	6	6	6	6	6	6	6
		≈	0	0	0	0	0	0	0	0	0	0
CF15-CF25	H	+	33	32	26	28	29	31	29	30	30	27
		≈	0	1	7	5	4	2	4	3	3	6
Total		+	72	73	57	61	68	70	64	63	65	65
		≈	3	2	18	14	7	5	11	12	10	10

Nomenclature: characteristics (C), separable (S), non-separable (N), multimodal (M), unimodal (U), expanded (E) and hybrid composite (H) functions; "+" indicates poorer performance than that of JS, and "≈" indicates no significant difference of performance between compared algorithm and JS.

Table 8
Sensitivity analysis of population size on optimal value using JS algorithm.

Population size	25-bar tower Weight (lb)	NSA	52-bar tower Weight (kg)	NSA	582-bar tower Volume (m3)	NSA
10	490.793	1,250	1910.187	1,600	20.322	3,870
20	485.380	3,240	1902.910	3,900	20.354	7,420
30	484.616	3,540	1901.687	6,000	20.210	11,190
40	**484.854**	**600**	**1899.678**	**3,720**	**20.153**	**14,840**
50	484.930	5,500	1899.659	5,000	20.162	20,000
60	484.854	9,420	1899.708	8,640	20.188	22,440
70	484.646	10,500	1899.574	5,670	20.172	28,000
80	484.643	11,680	1899.574	6,480	20.216	32,000
90	484.645	13,410	1899.719	13,860	20.204	36,000
100	484.979	12,700	1899.767	19,700	20.154	39,000
250	484.854	28,500	1899.719	38,500	21.219	12,750
500	484.639	71,000	1899.739	86,000	21.576	38,500

NSA: Number of structural analyses.

Consistent with the ASD-AISC design code, the third constraint is the maximum allowable slenderness ratio, which is 200 or 300 for compression or tension elements, respectively.

$$\lambda_i = \frac{k_i l_i}{r_i} \le \begin{cases} 300 \ \textit{for members under tension} \\ 200 \ \textit{for members under compression} \end{cases} \tag{34}$$

where λ_i is the slenderness ratio of the i^{th} member, and l_i and r_i are the length and radius of gyration of the i^{th} member. Should the constraint on the slenderness ratio of the compression elements not be satisfied, then the allowable stress must not exceed $\left(\frac{12\pi^2 E}{23\lambda_i^2}\right)$ [84].

Table 9
Results of 30 independent optimization runs for solving structural design problems.

25-bar tower Weight (lb)	NSA	52-bar tower Weight (kg)	NSA	582-bar tower Volume (m^3)	NSA
485.793	5,960	1902.706	8,000	21.317	15,520
485.249	4,480	1904.979	8,000	20.209	15,120
484.854	**600**	1904.149	6,480	20.645	15,600
485.249	5,600	1899.763	7,760	20.162	14,840
485.250	7,120	1899.706	5,680	20.557	16,000
485.049	4,800	1900.397	5,520	20.281	14,960
487.243	5,080	1902.533	8,000	20.417	14,840
485.769	6,320	1904.984	5,840	20.469	16,000
486.496	3440	1903.092	8,000	21.115	15,320
484.854	1,200	1900.986	7,040	20.920	16,000
486.091	3,720	1904.339	8,000	20.598	16,000
486.100	5,080	1901.050	7,600	20.388	15,720
485.049	6,000	1904.991	6,960	21.751	14,840
485.769	5,040	1899.685	4,920	21.416	14,840
487.243	4,480	1900.564	3,880	20.881	15,400
485.049	5,640	1902.961	8,000	21.682	15,600
484.854	**600**	1901.185	7,160	20.509	16,000
486.259	4,360	1904.127	8,000	20.388	14,920
487.299	7,920	**1899.678**	**3,720**	21.136	14,840
485.025	5,720	1902.533	7,040	20.589	14,840
484.854	880	1901.390	5,280	20.715	14,840
485.049	5,680	1899.756	7,480	21.302	14,840
486.873	4,960	1899.746	7,120	20.408	16,000
484.854	720	1901.878	5,000	20.664	16,000
486.324	4,320	1903.860	8,000	21.073	16,000
485.574	6,360	1899.765	7,800	**20.153**	**14,840**
484.854	720	1903.092	8,000	21.448	14,840
485.761	7,680	1902.923	8,000	21.630	14,840
487.127	6,080	1902.923	5,640	20.220	14,840
487.299	5,680	1904.514	6,600	20.682	15,080

NSA: Number of structural analyses.

Table 10a
Comparison of optimal solutions to 25-bar tower problem (HPSO, ABC, MBA, CBO, ECBO, and TLBO).

Design variables (in^2)	HPSO [74]	ABC [86]	MBA [87]	CBO [88]	ECBO [88]	TLBO [78]
A_1	0.1	0.1	0.1	0.1	0.1	0.1
A_2~A_5	0.3	0.3	0.3	0.3	0.3	0.3
A_6~A_9	3.4	3.4	3.4	3.4	3.4	3.4
A_{10}~A_{11}	0.1	0.1	0.1	0.1	0.1	0.1
A_{12}~A_{13}	2.1	2.1	2.1	2.1	2.1	2.1
A_{14}~A_{17}	1	1	1	1	1	1
A_{18}~A_{21}	0.5	0.5	0.5	0.5	0.5	0.5
A_{22}~A_{25}	3.4	3.4	3.4	3.4	3.4	3.4
Weight (lb)	484.85	484.85	484.85	484.85	484.85	484.85
Worst weight (lb)	N/A	485.05	485.048	N/A	N/A	N/A
Mean weight (lb)	N/A	484.94	484.885	486.87	485.89	N/A
Std. (lb)	N/A	N/A	0.072	N/A	N/A	N/A
NSA	25,000	24,250	2,150	2,040	7,050	4,000

NSA: Number of structural analyses.

Fig. 18 shows stress of each element, displacement of each node and slenderness constraint violation of each member in 582-bar tower. Table 13 presents the optimum designs that were obtained by Hasançebi et al. [82], Kaveh et al. [75,83], and Mortazavi [85]. The best design was obtained by JS, with the smallest average volume and low standard deviation of optimization result over 30 independent runs. PSO [82], and ECBO [83] required 50,000 and 19,700 structural analyses, respectively, while JS needed 14,840 analyses to converge to the optimum. Although DHPSACO [75] and iPSO [85] required only 8,500 and 2,360 analyses, respectively, they yielded mean optima that were higher than that obtained by JS with an optimized steel volume of 20.153 m^3. Therefore, JS was a competitive optimizer for solving the 582-bar tower design problem.

Table 10b
Comparison of optimal solutions to 25-bar tower problem (AFA, HHS, aeDE, DBB-BC, DAJA, and JS).

Design variables (in^2)	AFA [73,79]	HHS [81]	aeDE [89]	DBB-BC [90]	DAJA [73]	JS (This study)
A_1	0.1	0.1	0.1	0.1	0.1	**0.1**
A_2~A_5	0.3	0.3	0.3	0.3	0.3	**0.3**
A_6~A_9	3.4	3.4	3.4	3.4	3.4	**3.4**
A_{10}~A_{11}	0.1	0.1	0.1	0.1	0.1	**0.1**
A_{12}~A_{13}	2.1	2.1	2.1	2.1	2.1	**2.1**
A_{14}~A_{17}	1	1	1	1	1	**1**
A_{18}~A_{21}	0.5	0.5	0.5	0.5	0.5	**0.5**
A_{22}~A_{25}	3.4	3.4	3.4	3.4	3.4	**3.4**
Weight (lb)	484.85	484.85	484.85	484.85	484.85	**484.85**
Worst weight (lb)	N/A	N/A	486.1	N/A	484.85	**487.30**
Mean weight (lb)	N/A	484.95	485.01	N/A	484.85	**485.80**
Std. (lb)	N/A	0.365	0.273	N/A	0	**0.869**
NSA	7,100	1,739	1,440	20,000	511	**600**

NSA: Number of structural analyses.

Table 11a
Comparison of optimal solutions to 52-bar tower problem (HPSO, DHPSACO, CSS, CBO, TLBO, and AFA).

Design variables (mm^2)	HPSO [74]	DHPSACO [75]	CSS [73,76]	CBO [77]	TLBO [78]	AFA [73,79]
A_1~A_4	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055
A_5~A_{10}	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288
A_{11}~A_{13}	363.225	494.193	388.386	388.386	494.193	363.225
A_{14}~A_{17}	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219
A_{18}~A_{23}	940	1008.385	940	939.998	940	939.988
A_{24}~A_{26}	494.193	285.161	494.193	506.451	494.193	494.193
A_{27}~A_{30}	2238.705	2290.318	2238.705	2238.705	2238.705	2238.705
A_{31}~A_{36}	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385
A_{37}~A_{39}	388.386	388.386	494.193	506.451	494.193	641.289
A_{40}~A_{43}	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868
A_{44}~A_{49}	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288
A_{50}~A_{52}	792.256	506.451	494.193	506.451	494.193	494.193
Weight (kg)	1905.5	1904.83	1897.62	1899.35	1902.605	1903.37
Worst weight (kg)	N/A	N/A	N/A	2262.8	N/A	N/A
Mean weight (kg)	N/A	N/A	N/A	1963.12	N/A	N/A
Std. (kg)	N/A	N/A	N/A	106.01	N/A	N/A
NSA	150,000	5,300	5,000	3,840	6,000	52,600

NSA: Number of structural analyses.

Table 11b
Comparison of optimal solutions to 52-bar tower problem (WCA, IMBA, HHS, aeDE, DAJA, and JS).

Design variables (mm^2)	WCA [80]	IMBA [80]	HHS [81]	aeDE [89]	DAJA [73]	JS (This study)
A_1~A_4	4658.055	4658.055	4658.055	4658.055	4658.055	**4658.055**
A_5~A_{10}	1161.288	1161.288	1161.288	1161.288	1161.288	**1161.288**
A_{11}~A_{13}	494.193	494.193	494.193	494.193	494.193	**494.193**
A_{14}~A_{17}	3303.219	3303.219	3303.219	3303.219	3303.219	**3303.219**
A_{18}~A_{23}	940	940	940	940	940	**939.998**
A_{24}~A_{26}	494.193	494.193	494.193	494.193	494.193	**506.451**
A_{27}~A_{30}	2238.705	2238.705	2238.705	2238.705	2238.705	**2238.705**
A_{31}~A_{36}	1008.385	1008.385	1008.385	1008.385	1008.385	**1008.385**
A_{37}~A_{39}	494.193	494.193	494.193	494.193	494.193	**388.386**
A_{40}~A_{43}	1283.868	1283.868	1283.868	1283.868	1283.868	**1283.868**
A_{44}~A_{49}	1161.288	1161.288	1161.288	1161.288	1161.288	**1161.288**
A_{50}~A_{52}	494.193	494.193	494.193	494.193	494.193	**506.451**
Weight (kg)	1902.605	1902.605	1902.605	1902.605	1902.605	**1899.678**
Worst weight (kg)	1912.646	1904.83	N/A	1925.714	1903.944	**1904.991**
Mean weight (kg)	1909.856	1903.076	1904.587	1906.735	1902.74	**1902.142**
Std. (kg)	7.09	1.13	1.309	6.679	0.446	**1.860**
NSA	7,100	4,750	4,523	3,720	3,321	**3,720**

NSA: Number of structural analyses.

Table 12
Profile list for 582-bar tower [85].

W-shape profile list						
W27 × 178	W21 × 122	W18 × 50	W14 × 455	W14 × 74	W12 × 136	W10 × 77
W27 × 161	W21 × 111	W18 × 46	W14 × 426	W14 × 68	W12 × 120	W10 × 68
W27 × 146	W21 × 101	W18 × 40	W14 × 398	W14 × 61	W12 × 106	W10 × 60
W27 × 114	W21 × 93	W18 × 35	W14 × 370	W14 × 53	W12 × 96	W10 × 54
W27 × 102	W21 × 83	W16 × 100	W14 × 342	W14 × 48	W12 × 87	W10 × 49
W27 × 94	W21 × 73	W16 × 89	W14 × 311	W14 × 43	W12 × 79	W10 × 45
W27 × 84	W21 × 68	W16 × 77	W14 × 283	W14 × 38	W12 × 72	W10 × 39
W24 × 162	W21 × 62	W16 × 67	W14 × 257	W14 × 34	W12 × 65	W10 × 33
W24 × 146	W21 × 57	W16 × 57	W14 × 233	W14 × 30	W12 × 58	W10 × 30
W24 × 131	W21 × 50	W16 × 50	W14 × 211	W14 × 26	W12 × 53	W10 × 26
W24 × 117	W21 × 44	W16 × 45	W14 × 193	W14 × 22	W12 × 50	W10 × 22
W24 × 104	W18 × 119	W16 × 40	W14 × 176	W12 × 336	W12 × 45	W8 × 67
W24 × 94	W18 × 106	W16 × 36	W14 × 159	W12 × 305	W12 × 40	W8 × 58
W24 × 84	W18 × 97	W16 × 31	W14 × 145	W12 × 279	W12 × 35	W8 × 48
W24 × 76	W18 × 86	W16 × 26	W14 × I32	W12 × 252	W12 × 30	W8 × 40
W24 × 68	W18 × 76	W14 × 730	W14 × 120	W12 × 230	W12 × 26	W8 × 35
W24 × 62	W18 × 71	W14 × 665	W14 × 109	W12 × 210	W12 × 22	W8 × 31
W24 × 55	W18 × 65	W14 × 605	W14 × 99	W12 × 190	W10 × 112	W8 × 28
W21 × 147	W18 × 60	W14 × 550	W14 × 90	W12 × 170	W10 × 100	W8 × 24
W21 × 132	W18 × 55	W14 × 500	W14 × 82	W12 × 152	W10 × 88	W8 × 21

Table 13
Comparison of optimal solutions to 582-bar tower problem.

Design variables	PSO [82]	DHPSACO [75]	ECBO [83]	iPSO [85]	JS (This study)
1	W8 × 21	W8 × 24	W8 × 21	W8 × 21	**W8 × 21 (39.74)**
2	W12 × 79	W12 × 72	W27 × 94	W8 × 21	**W24 × 84 (159.35)**
3	W8 × 24	W8 × 28	W8 × 24	W8 × 21	**W8 × 21 (39.74)**
4	W10 × 60	W12 × 58	W12 × 72	W21 × 73	**W12 × 65 (123.23)**
5	W8 × 24	W8 × 24	W8 × 24	W12 × 53	**W8 × 21 (39.74)**
6	W8 × 21	W8 × 24	W8 × 21	W8 × 21	**W8 × 21 (39.74)**
7	W8 × 48	W10 × 49	W10 × 54	W8 × 21	**W10 × 45 (85.81)**
8	W8 × 24	W8 × 24	W8 × 24	W8 × 21	**W8 × 21 (39.74)**
9	W8 × 21	W8 × 24	W8 × 21	W8 × 21	**W8 × 21 (39.74)**
10	W10 × 45	W12 × 40	W14 × 48	W8 × 21	**W10 × 54 (101.94)**
11	W8 × 24	W12 × 30	W8 × 24	W18 × 76	**W8 × 21 (39.74)**
12	W10 × 68	W12 × 72	W12 × 53	W24 × 62	**W10 × 68 (128.39)**
13	W14 × 74	W18 × 76	W12 × 72	W10 × 49	**W24 × 84 (159.35)**
14	W8 × 48	W10 × 49	W10 × 49	W10 × 49	**W14 × 53 (100.64)**
15	W18 × 76	W14 × 82	W14 × 74	W12 × 79	**W12 × 72 (136.13)**
16	W8 × 31	W8 × 31	W8 × 31	W21 × 62	**W8 × 21 (39.74)**
17	W8 × 21	W14 × 61	W16 × 67	W14 × 43	**W12 × 58 (109.68)**
18	W16 × 67	W8 × 24	W8 × 24	W16 × 26	**W8 × 21 (39.74)**
19	W8 × 24	W8 × 21	W8 × 21	W8 × 21	**W8 × 21 (39.74)**
20	W8 × 21	W12 × 40	W14 × 34	W8 × 21	**W12 × 40 (75.48)**
21	W8 × 40	W8 × 24	W8 × 24	W8 × 21	**W8 × 21 (39.74)**
22	W8 × 24	W14 × 22	W8 × 21	W8 × 24	**W8 × 21 (39.74)**
23	W8 × 21	W8 × 31	W8 × 21	W8 × 24	**W16 × 36 (68.39)**
24	W10 × 22	W8 × 28	W8 × 24	W8 × 21	**W8 × 21 (39.74)**
25	W8 × 24	W8 × 21	W8 × 21	W16 × 67	**W8 × 21 (39.74)**
26	W8 × 21	W8 × 21	W8 × 21	W8 × 31	**W8 × 21 (39.74)**
27	W8 × 21	W8 × 24	W8 × 24	W8 × 24	**W8 × 21 (39.74)**
28	W8 × 24	W8 × 28	W8 × 21	W8 × 21	**W8 × 21 (39.74)**
29	W8 × 21	W16 × 36	W8 × 21	W8 × 21	**W8 × 21 (39.74)**
30	W8 × 21	W8 × 24	W8 × 24	W8 × 21	**W8 × 21 (39.74)**
31	W8 × 24	W8 × 21	W8 × 21	W8 × 21	**W8 × 21 (39.74)**
32	W8 × 24	W8 × 24	W8 × 24	W8 × 28	**W8 × 21 (39.74)**
Volume (m^3)	22.3958	22.0607	21.340	20.9464	**20.153**
Worst volume (m^3)	N/A	N/A	N/A	N/A	**21.7510**
Mean volume (m^3)	N/A	N/A	21.491	21.3297	**20.7909**
Std. (m^3)	N/A	N/A	0.134	0.9553	**0.4848**
NSA	50,000	8,500	19,700	2,360	**14,840**

NSA: Number of structural analyses.

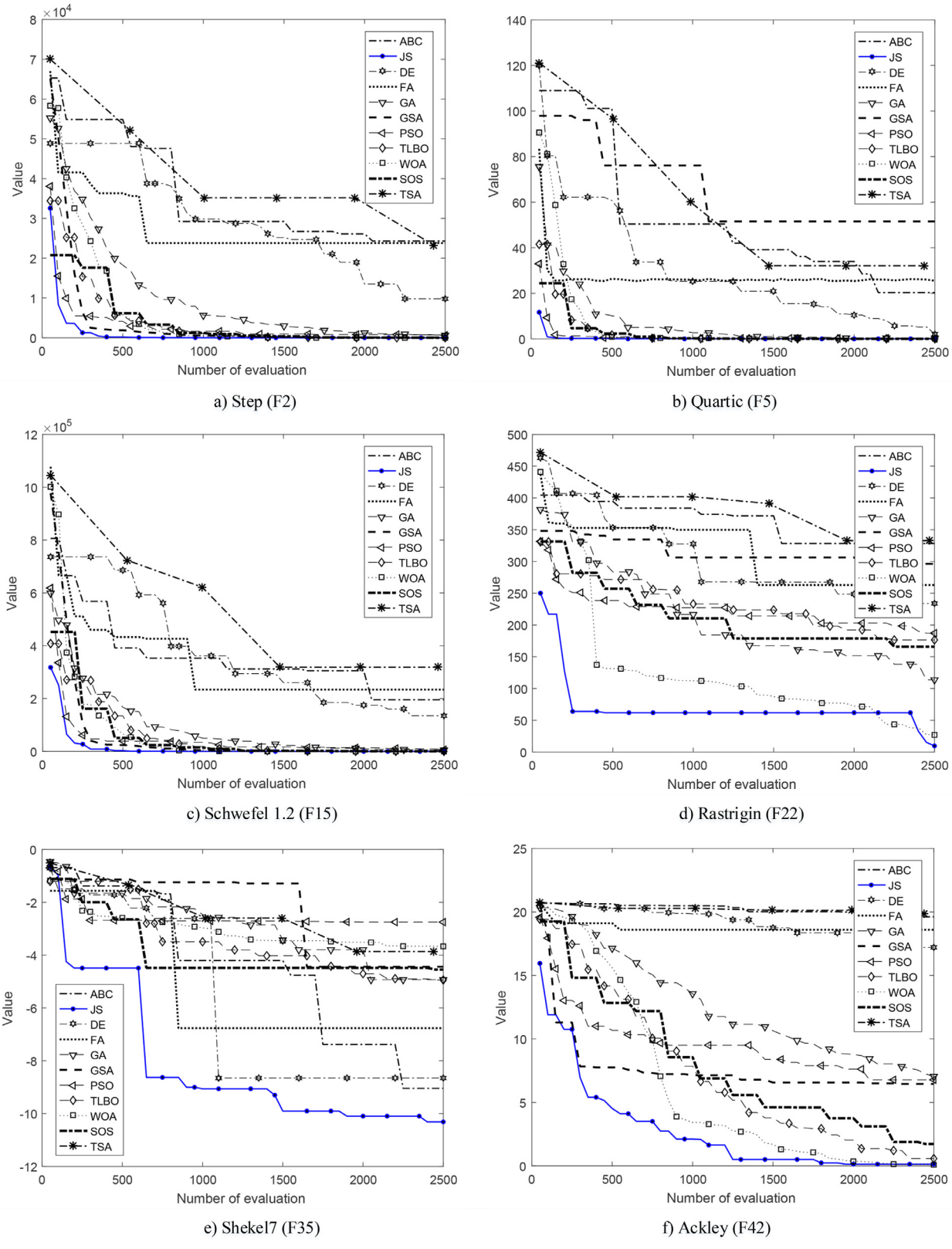
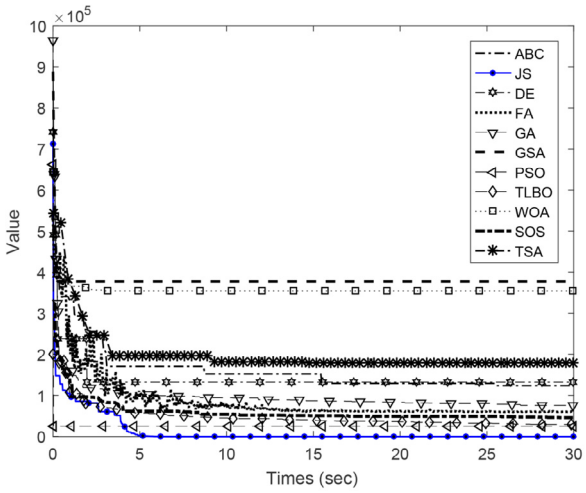
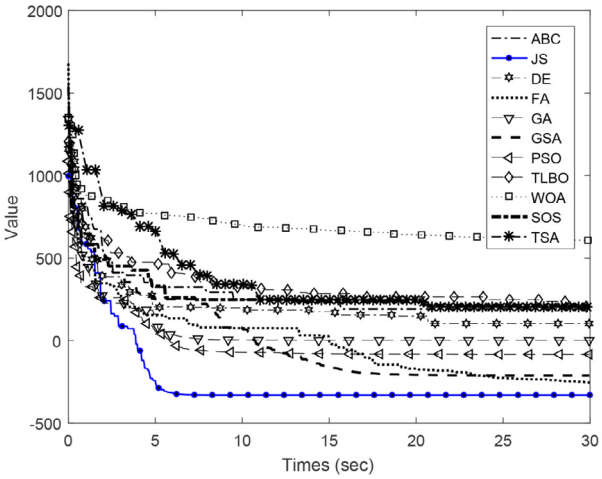


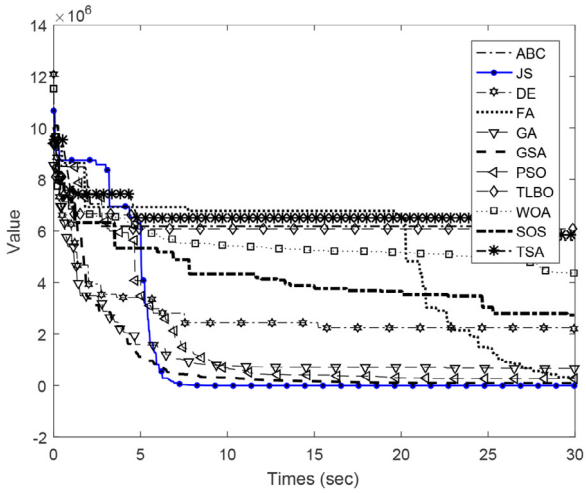
Fig. 12. Convergence curves for selected small/average-scale mathematical problems.



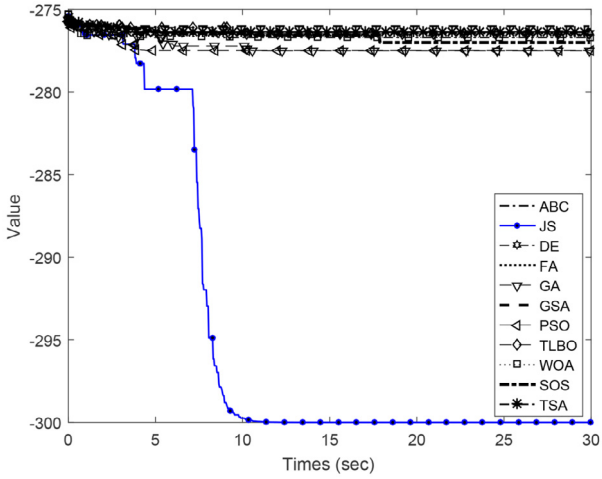
a) Shifted Schwefel's Problem 1.2 with Noise in Fitness (CF4)



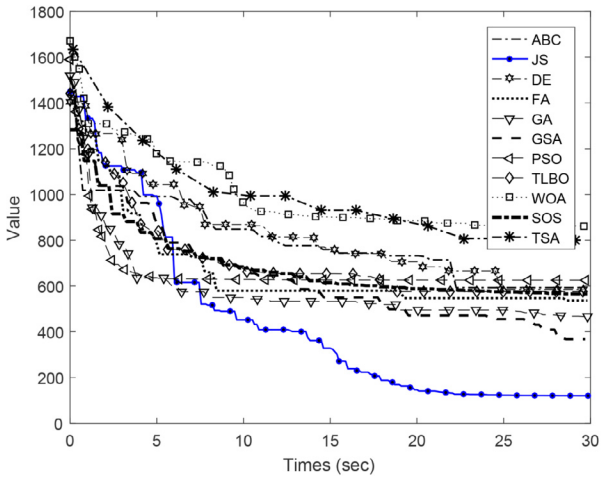
b) Shifted Rotated Rastrigin's Function (CF10)



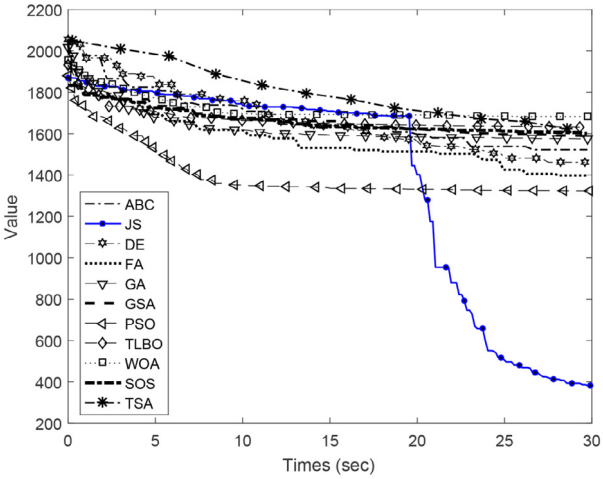
c) Schwefel's Problem 2.13 (CF12)



d) Shifted Rotated Expanded Scaffer's CF6 (CF14)

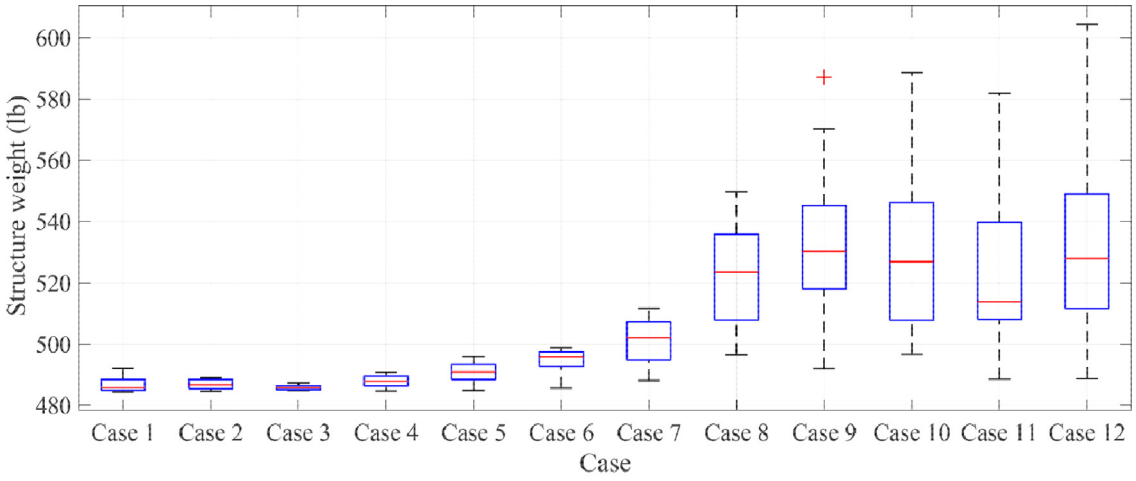


e) Rotated Hybrid Composition Function (CF16)

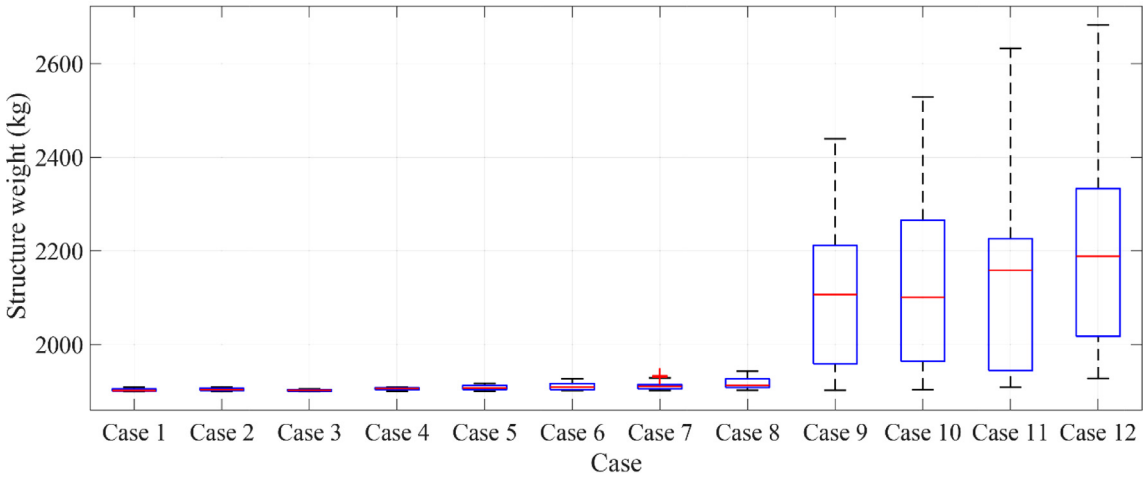


f) Rotated Hybrid Composition Function (CF21)

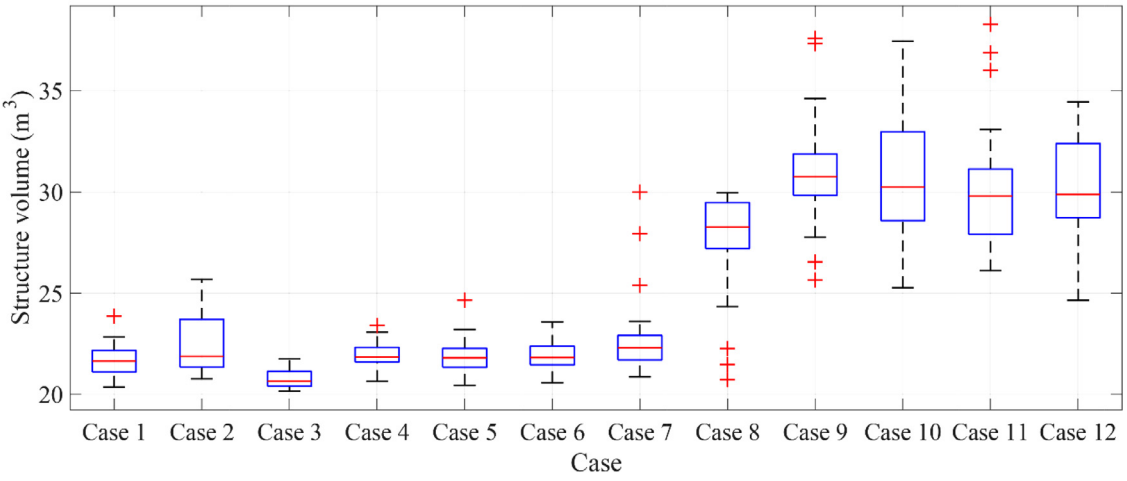
Fig. 13. Convergence curves for selected large scale CEC2005 functions ($D = 50$).



a) 25-bar tower.

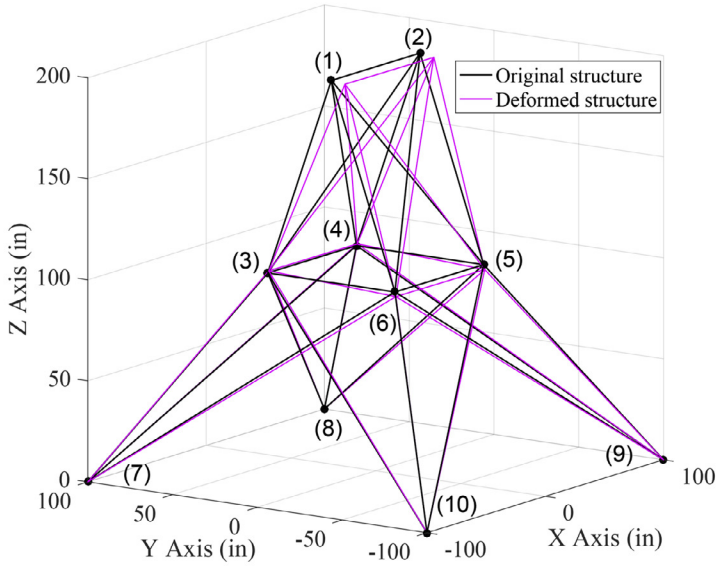


b) 52-bar tower.

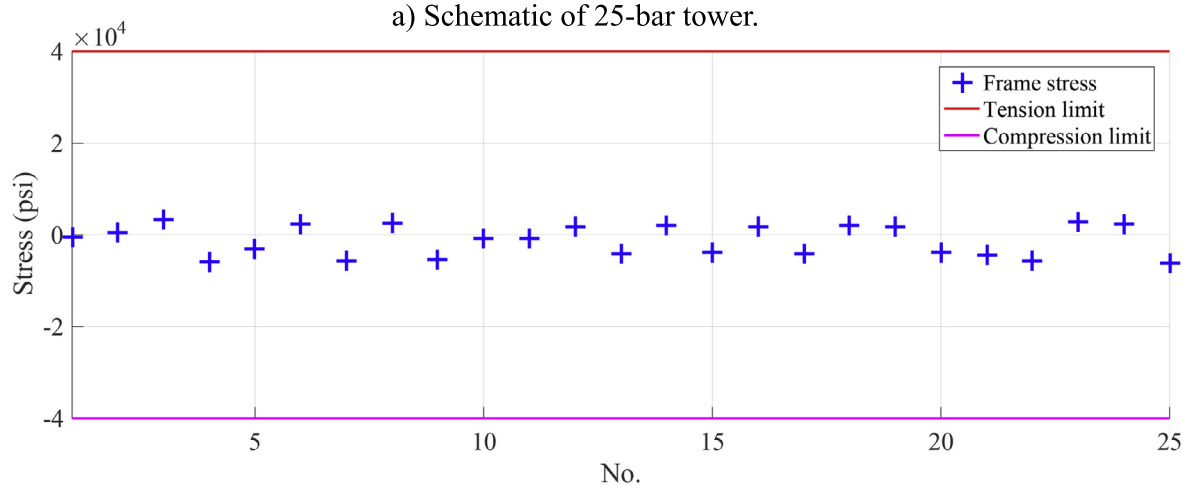


c) 582-bar tower.

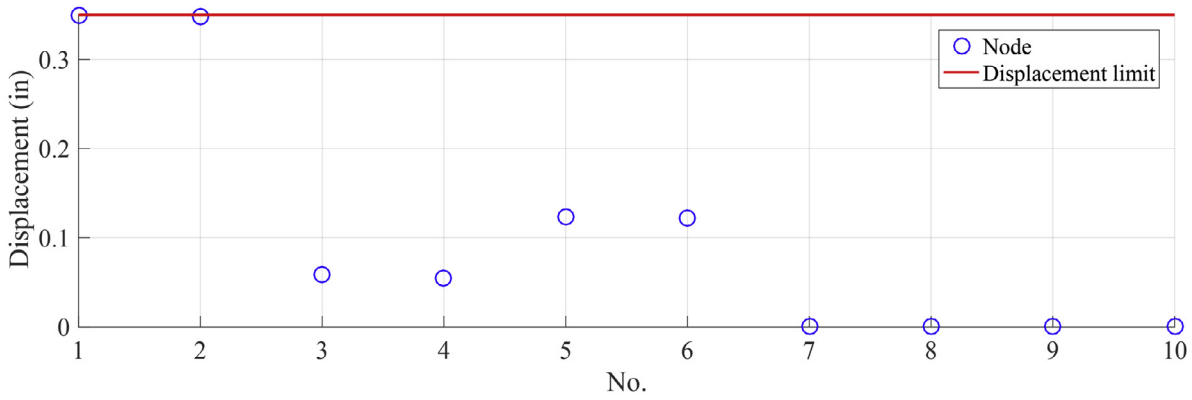
Fig. 14. Box plots for effects of initial value ε_0 on optimum solutions.



a) Schematic of 25-bar tower.

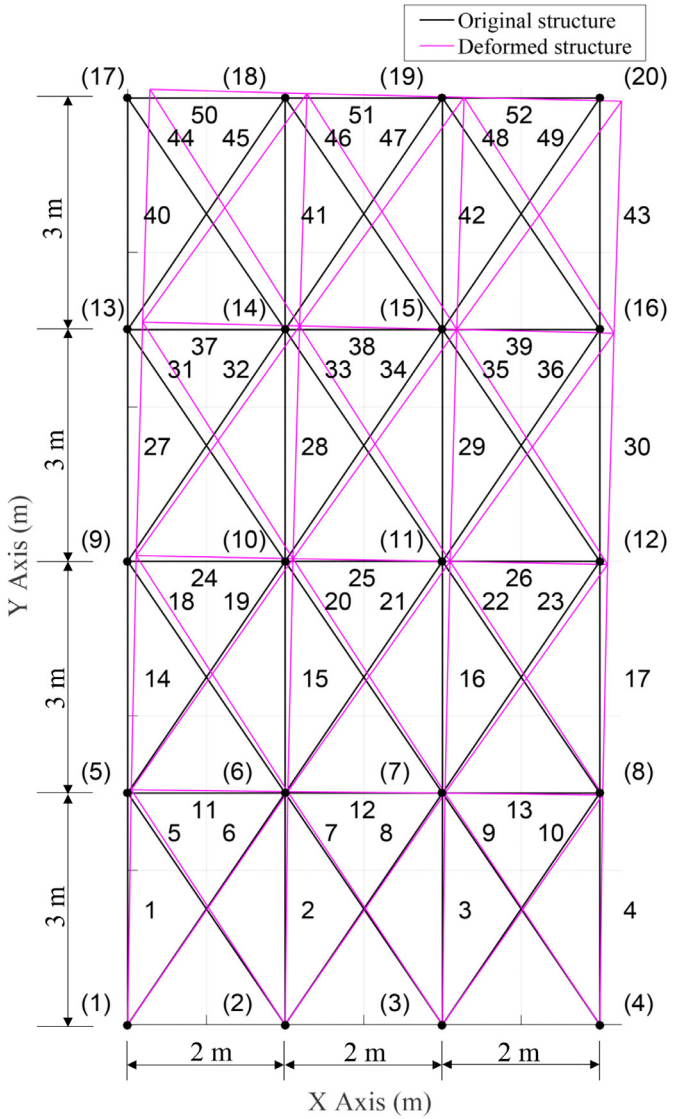


b) Stress constraint violation.

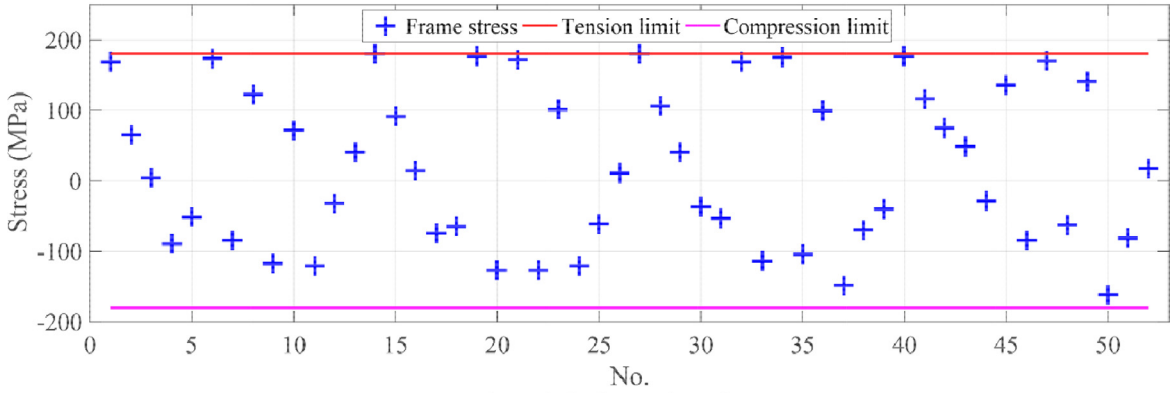


c) Displacement constraint violation.

Fig. 15. 25-bar tower problem.



a)Schematic of 52-bar tower.



b) Stress constraint violation of 52-bar tower.

Fig. 16. 52-bar tower problem.

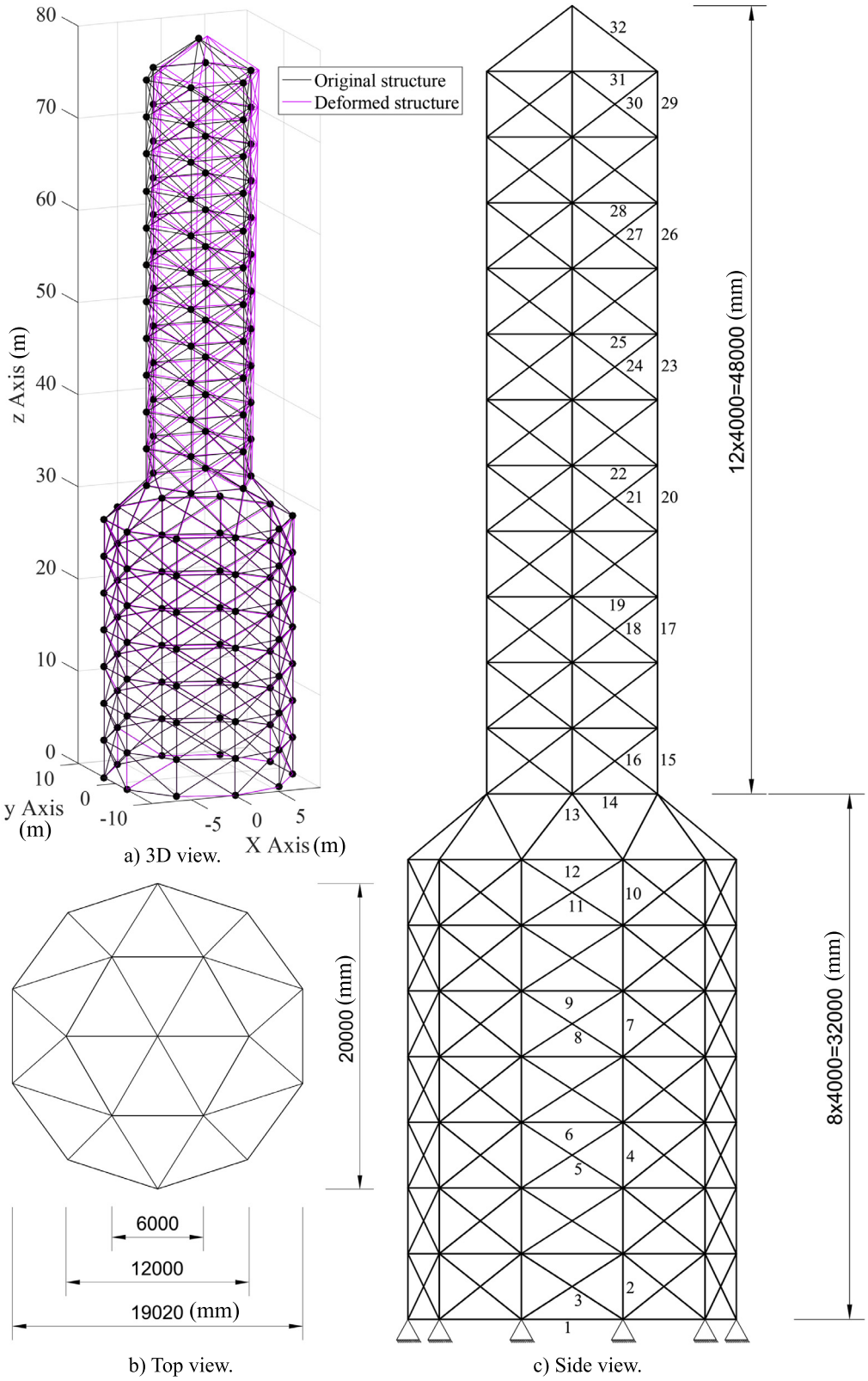
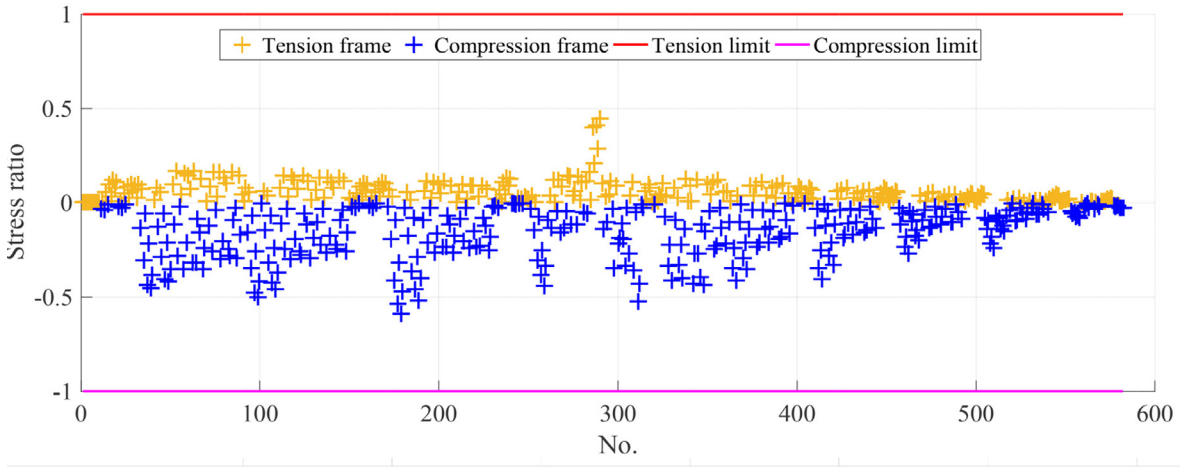
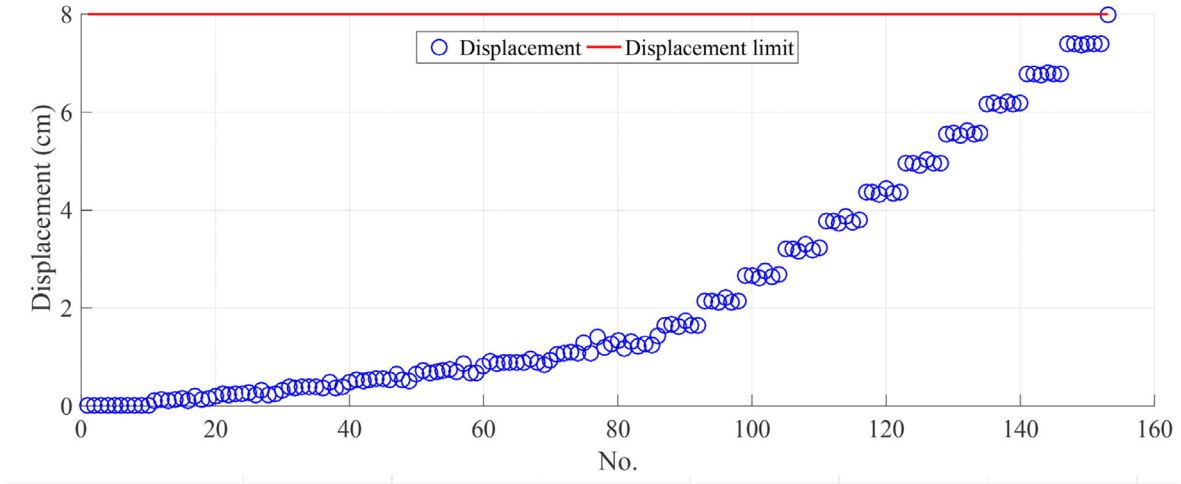


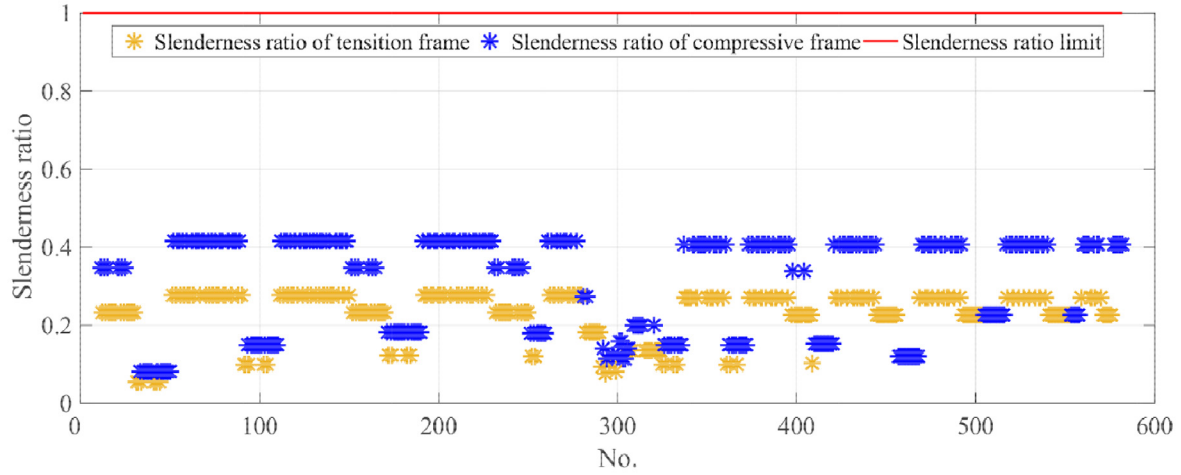
Fig. 17. Schematic of 582-bar tower.



a) Stress constraint violation.



b) Displacement constraint violation.



c) Slenderness ratio constraint violation.

Fig. 18. Constraint violation of 582-bar tower.

6. Conclusion

This study developed artificial Jellyfish Search, a new metaheuristic optmization algorithm that is inspired by the behavior of jellyfish for seeking food in the ocean, which involves following ocean current at the beginning, movements inside jellyfish swarms as time goes by, and a time control mechanism for switching among these motions. Initially, the exploration stage begins when jellyfish follow ocean currents to find the best locations (exploration stage). As time passes, a jellyfish swarm is established, and each jellyfish continues to move inside the swarm to find a better location, including active and passive motions, making the exploitation in this stage (exploitation stage). Meanwhile, a time control mechanism switches between these motions. After repeating the loop, jellyfish bloom occurs, which is the optimum phase. The numerical experiments have proved that JS algorithm well balances exploration and exploitation to find the optimum value.

The three powerful features of JS are as follows: (1) it has only two internal parameters; (2) it is easy to code; and (3) it is easy to apply. 50 small/average- and 25 large-scale mathematical functions (CEC2005) with various dimensions, are used to validate the JS algorithm. Successful hit-rate result on small/average functions indicates that JS can search the optimal locations better than the other algorithms. Notably, JS requires less time and faster convergence than the other algorithms. On the large-scale mathematical functions, the significance of Wilcoxon rank sum test further confirms that JS has ability to find optimal values in large-scale functions. The JS obviously outperforms the WOA, TSA, SOS, TLBO, FA, GSA, ABC, DE, PSO and GA algorithms in the mathematical benchmark tests.

Three steel structural optimization problems (involving the design of a 25-bar tower, a 52-bar tower and a 582-bar tower) were solved using the proposed JS optimizer. The best weight or volume of 25-bar tower, 52-bar tower and 582-bar tower design are 484.854 lb, 1899.678 kg and 20.153 m^3, respectively, which are the best solutions of steel structures with fewer numbers of evaluation than previous studies. The JS algorithm achieves better results owing to its appropriate balance of exploration and exploitation by switching the movements with a time control mechanism, and a chaotic map used to improve the diversity of initial population. Thus, JS is potentially an excellent metaheuristic algorithm for solving various engineering optimization problems.

Source codes and supplementary data

The fundamental codes and data that support the findings of this study are available at ResearchGate with the identifier https://www.researchgate.net/profile/Jui-Sheng_Chou.

Declaration of Competing Interest

We wish to declare that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgement

The authors would like to thank the Ministry of Science and Technology, Taiwan, for financially supporting this research under contract MOST 107-2221-E-011-035-MY3.

References

[1] S. Gao, C.W. de Silva, Estimation distribution algorithms on constrained optimization problems, Appl. Math. Comput. 339 (2018) 323–345.
[2] M.S. Kıran, M. Gündüz, Ö.K. Baykan, A novel hybrid algorithm based on particle swarm and ant colony optimization for finding the global minimum, Appl. Math. Comput. 219 (2012) 1515–1521.
[3] B.A. Hassan, T.A. Rashid, Operational framework for recent advances in backtracking search optimisation algorithm: a systematic review and performance evaluation, Appl. Math. Comput. 370 (2020) 124919.
[4] M. Birattari, L. Paquete, T. Stützle, K. Varrentrapp, Classification of metaheuristics and design of experiments for the analysis of components, Teknik Rapor (2001) AIDA-01-05.
[5] J. Liu, C. Wu, G. Wu, X. Wang, A novel differential search algorithm and applications for structure design, Appl. Math. Comput. 268 (2015) 246–269.
[6] W. Xiang, S. Ma, M. An, hABCDE: a hybrid evolutionary algorithm based on artificial bee colony algorithm and differential evolution, Appl. Math. Comput. 238 (2014) 370–386.
[7] M.S. Kiran, TSA: tree-seed algorithm for continuous optimization, Expert Syst. Appl. 42 (2015) 6686–6698.
[8] K. Hussain, M.N. Mohd Salleh, S. Cheng, Y. Shi, Metaheuristic research: a comprehensive survey, Artif. Intell. Rev. 52 (2019) 2191–2233.
[9] W. Zhao, L. Wang, Z. Zhang, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, Knowl. Based Syst. 163 (2019) 283–304.
[10] Q. Askari, I. Younas, M. Saeed, Political optimizer: a novel socio-inspired meta-heuristic for global optimization, Knowl. Based Syst. 195 (2020) 105709.
[11] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1992) 66–73.
[12] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997) 341–359.
[13] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evolut. Comput. 3 (1999) 82–102.
[14] D. Simon, Biogeography-based optimization, IEEE Trans. Evolut. Comput. 12 (2008) 702–713.
[15] S. Kirkpatrick, C.D. Gelatt Jr, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[16] V. Černý, Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm, J. Optim. Theory Appl. 45 (1985) 41–51.
[17] O.K. Erol, I. Eksin, A new optimization method: big Bang–Big Crunch, Adv. Eng. Softw. 37 (2006) 106–111.
[18] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a Gravitational Search Algorithm, Inf. Sci. 179 (2009) 2232–2248.

[19] B. Webster, P.J. Bernhard, A local search optimization algorithm based on natural principles of gravitation, in: Proceedings of the International Conference on Information and Knowledge Engineering, 2003, pp. 255–261.
[20] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta Mech 213 (2010) 267–289.
[21] R. Formato, Central force optimization: a new metaheuristic with applications in applied electromagnetics. Progress Electromag. Res.. PIER 77, 425–491, in, Progress In Electromagnetics Research, 2007, pp. 425-491
[22] H. Du, X. Wu, J. Zhuang, in: Small-World Optimization Algorithm For Function Optimization, Springer Verlag, 2006, pp. 264–273.
[23] H. Shah-Hosseini, Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation, Int. J. Comput. Sci. Eng. 6 (2011) 132–140.
[24] J. Kennedy, R. Eberhart, in: Particle Swarm Optimization, IEEE, Piscataway, NJ, United States, 1995, pp. 1942–1948.
[25] M. Dorigo, M. Birattari, T. Stützle, Ant colony optimization artificial ants as a computational intelligence technique, IEEE Comput. Intell. Mag. 1 (2006) 28–39.
[26] H.A. Abbass, MBO: marriage in honey bees optimization a haplometrosis polygynous swarming approach, in: Congress On Evolutionary Computation 2001, 2001, pp. 207–214. Soul.
[27] M. Roth, S. Wicker, Termite: A Swarm Intelligent Routing Algorithm for Mobilewireless Ad-Hoc Networks, 2006, pp. 155–184.
[28] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. 39 (2007) 459–471.
[29] H. Sun, H. Luş, R. Betti, Identification of structural models using a modified Artificial Bee Colony algorithm, Comput. Struct. 116 (2013) 59–74.
[30] P.C. Pinto, T.A. Runkler, J.M.C. Sousa, WASP swarm algorithm for dynamic MAX-SAT problems, in: 8th International Conference on Adaptive and Natural Computing Algorithms, ICANNGA 2007, Warsaw, 2007, pp. 350–357.
[31] M. Farshchin, C.V. Camp, M. Maniat, Multi-class teaching–learning-based optimization for truss design with frequency constraints, Eng. Struct. 106 (2016) 355–369.
[32] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2001) 60–68.
[33] M.R. Maheri, M.M. Narimani, An enhanced harmony search algorithm for optimum design of side sway steel frames, Comput. Struct. 136 (2014) 78–89.
[34] J.-S. Chou, N.-M. Nguyen, FBI inspired meta-optimization, Appl. Soft Comput. 93 (2020) 106339.
[35] F. Glover, Tabu search, Part I, ORSA J. Comp. 1 (1989) 190–206.
[36] S. He, Q.H. Wu, J.R. Saunders, A novel group search optimizer inspired by animal behavioural ecology, in: 2006 IEEE Congress on Evolutionary Computation, CEC 2006, Vancouver, BC, 2006, pp. 1272–1278.
[37] A.H. Kashan, League Championship Algorithm: a new algorithm for numerical function optimization, in: International Conference on Soft Computing and Pattern Recognition, SoCPaR 2009, Malacca, 2009, pp. 43–48.
[38] T. Bastian, M.K.S. Lilley, S.E. Beggs, G.C. Hays, T.K. Doyle, Ecosystem relevance of variable jellyfish biomass in the Irish Sea between years, regions and water types, Estuarine, Coast. Shelf Sci. 149 (2014) 302–312.
[39] K. Landow, Best treatment of jellyfish stings? Postgrad. Med. 107 (2000) 27–28.
[40] P.J. Fenner, J.A. Williamson, J.W. Burnett, J. Rifkin, First aid treatment of jellyfish stings in Australia response to a newly differentiated species, Med. J. Aust. 158 (1993) 498–501.
[41] G.L. Mariottini, L. Pane, Mediterranean jellyfish venoms: a review on scyphomedusae, Mar. Drugs 8 (2010).
[42] S. Fossette, N.F. Putman, K.J. Lohmann, R. Marsh, G.C. Hays, A biologist's guide to assessing ocean currents: a review, Mar. Ecol. Prog. Ser. 457 (2012) 285–301.
[43] S. Fossette, A.C. Gleiss, J. Chalumeau, T. Bastian, C.D. Armstrong, S. Vandenabeele, M. Karpytchev, G.C. Hays, Current-oriented swimming by jellyfish and its role in bloom maintenance, Curr. Biol. 25 (2015) 342–347.
[44] G.L. Mariottini, L. Pane, Mediterranean jellyfish venoms: a review on scyphomedusae, Mar. Drugs 8 (2010) 1122–1152.
[45] L. Brotz, W.W.L. Cheung, K. Kleisner, E. Pakhomov, D. Pauly, Increasing jellyfish populations: trends in Large Marine Ecosystems, Hydrobiologia 690 (2012) 3–20.
[46] Z. Dong, D. Liu, J.K. Keesing, Jellyfish blooms in China: dominant species, causes and consequences, Mar. Pollut. Bull. 60 (2010) 954–963.
[47] A.J. Richardson, A. Bakun, G.C. Hays, M.J. Gibbons, The jellyfish joyride: causes, consequences and management responses to a more gelatinous future, Trends Ecol. Evol. (Amst.) 24 (2009) 312–322.
[48] R.H. Condon, C.M. Duarte, K.A. Pitt, K.L. Robinson, C.H. Lucas, K.R. Sutherland, H.W. Mianzan, M. Bogeberg, J.E. Purcell, M.B. Decker, Recurrent jellyfish blooms are a consequence of global oscillations, Proc. Natl. Acad. Sci. 110 (2013) 1000–1005.
[49] G.C. Hays, Ocean currents and marine life, Curr. Biol. 27 (2017) R470–R473.
[50] D. Zavodnik, Spatial aggregations of the swarming jellyfish Pelagia noctiluca (Scyphozoa), Mar. Biol. 94 (1987) 265–269.
[51] M.S. Kıran, O. Fındık, A directed artificial bee colony algorithm, Appl. Soft Comput. 26 (2015) 454–462.
[52] T. Xiang, X. Liao, K.-w. Wong, An improved particle swarm optimization algorithm combined with piecewise linear chaotic map, Appl. Math. Comput. 190 (2007) 1637–1645.
[53] J.-S. Chou, N.-T. Ngo, Modified firefly algorithm for multidimensional optimization in structural design problems, Struct. Multidiscip. Optim. 55 (2017) 2013–2028.
[54] A. Kaveh, R. Mahdipour Moghanni, S.M. Javadi, Optimum Design of Large Steel Skeletal Structures Using Chaotic Firefly Optimization Algorithm Based On the Gaussian map, Structural and Multidisciplinary Optimization, 2019.
[55] R.M. May, Simple mathematical models with very complicated dynamics, Nature 261 (1976) 459.
[56] W.-C. Hong, Y. Dong, L.-Y. Chen, S.-Y. Wei, SVR with hybrid chaotic genetic algorithms for tourism demand forecasting, Appl. Soft Comput. 11 (2011) 1881–1890.
[57] J. Xu, J. Zhang, Exploration-exploitation tradeoffs in metaheuristics: survey and analysis, in: Proceedings of the 33rd Chinese Control Conference, 2014, pp. 8633–8638.
[58] X.-S. Yang, Firefly algorithms for multimodal optimization, in: O. Watanabe, T. Zeugmann (Eds.), Stochastic Algorithms: Foundations and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 169–178.
[59] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[60] M.Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112.
[61] D. Karaboga, B. Akay, A comparative study of Artificial Bee Colony algorithm, Appl. Math. Comput. 214 (2009) 108–132.
[62] P.J. Ballester, J. Stephenson, J.N. Carter, K. Gallagher, Real-parameter optimization performance study on the CEC-2005 benchmark with SPC-PNX, in: 2005 IEEE Congress on Evolutionary Computation, 491, 2005, pp. 498–505.
[63] R.J. Bolton, D.J. Hand, N.M. Adams, Determining hit rate in pattern search, in: D.J. Hand, N.M. Adams, R.J. Bolton (Eds.), Pattern Detection and Discovery, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 36–48.
[64] M.F. Verde, N.A. Macmillan, C.M. Rotello, Measures of sensitivity based on a single hit rate and false alarm rate: the accuracy, precision, and robustness of′, Az, and A', Percept Psychophys. 68 (2006) 643–654.
[65] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18.
[66] A.H. Gandomi, X.S. Yang, S. Talatahari, A.H. Alavi, Firefly algorithm with chaos, Commun. Nonlinear Sci. Numer. Simul. 18 (2013) 89–98.
[67] I. Fister, X.S. Yang, J. Brest, in: On the Randomized Firefly Algorithm, Studies in Computational Intelligence, 2014, pp. 27–48.
[68] A. Saxena, R. Kumar, S. Das, β-Chaotic map enabled Grey Wolf Optimizer, Appl. Soft Comput. 75 (2019) 84–105.
[69] S.K. Dinkar, K. Deep, Opposition based Laplacian Ant Lion Optimizer, J. Comput. Sci. 23 (2017) 71–90.

[70] S.N. Chegini, A. Bagheri, F. Najafi, PSOSCALF: a new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems, Appl. Soft Comput. 73 (2018) 697–726.
[71] C. Blum, A. Roli, Metaheuristics in combinatorial optimization: overview and conceptual comparison, ACM Comput. Surv. 35 (2003) 268–308.
[72] M. Lozano, C. García-Martínez, Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: overview and progress report, Comput. Oper. Res. 37 (2010) 481–497.
[73] S.O. Degertekin, L. Lamberti, I.B. Ugur, Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm, Appl. Soft Comput. 79 (2019) 363–390.
[74] L.J. Li, Z.B. Huang, F. Liu, A heuristic particle swarm optimization method for truss structures with discrete variables, Comput. Struct. 87 (2009) 435–443.
[75] A. Kaveh, S. Talatahari, A particle swarm ant colony optimization for truss structures with discrete variables, J. Construct. Steel Res. 65 (2009) 1558–1568.
[76] A. Kaveh, S. Talatahari, A charged system search with a fly to boundary method for discrete optimum design of truss structures, Asian J. Civil Eng. 11 (2010) 277–293.
[77] A. Kaveh, V.R. Mahdavi, Colliding bodies optimization method for optimum discrete design of truss structures, Comput. Struct. 139 (2014) 43–53.
[78] T. Dede, Application of teaching-learning-based-optimization algorithm for the discrete optimization of truss structures, KSCE J. Civil Eng. 18 (2014) 1759–1767.
[79] A. Baghlani, M. Makiabadi, M. Sarcheshmehpour, Discrete optimum design of truss structures by an improved firefly algorithm, Adv. Struct. Eng. 17 (2014) 1517–1530.
[80] A. Sadollah, H. Eskandar, A. Bahreininejad, J.H. Kim, Water cycle, mine blast and improved mine blast algorithms for discrete sizing optimization of truss structures, Comput. Struct. 149 (2015) 1–16.
[81] M.Y. Cheng, D. Prayogo, Y.W. Wu, M.M. Lukito, A hybrid harmony search algorithm for discrete sizing optimization of truss structure, Autom. Construct. 69 (2016) 21–33.
[82] O. Hasançebi, S. Çarbaş, E. Doğan, F. Erdal, M.P. Saka, Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures, Comput. Struct. 87 (2009) 284–302.
[83] A. Kaveh, M. Ilchi Ghazaan, Enhanced colliding bodies optimization for design problems with continuous and discrete variables, Adv. Eng. Softw. 77 (2014) 66–75.
[84] A.I.o.S. Construction, Manual of Steel Construction: Allowable Stress Design, American Institute of Steel Construction, 1989.
[85] A. Mortazavi, V. Toğan, Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer, Struct. Multidiscip. Optim. 54 (2016) 715–736.
[86] M. Sonmez, Discrete optimum design of truss structures using artificial bee colony algorithm, Struct. Multidiscip. Optim. 43 (2011) 85–97.
[87] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm for optimization of truss structures with discrete variables, Comput. Struct. 102-103 (2012) 49–63.
[88] A. Kaveh, M. Ilchi Ghazaan, A comparative study of CBO and ECBO for optimal design of skeletal structures, Comput. Struct. 153 (2015) 137–147.
[89] V. Ho-Huu, T. Nguyen-Thoi, T. Vo-Duy, T. Nguyen-Trang, An adaptive elitist differential evolution for optimization of truss structures with discrete design variables, Comput. Struct. 165 (2016) 59–75.
[90] D. Prayogo, M.Y. Cheng, Y.W. Wu, A.A. Herdany, H. Prayogo, Differential Big Bang - Big Crunch algorithm for construction-engineering design optimization, Autom. Construct. 85 (2018) 290–304.