Jellyfish Search (JS) (JS Chou and DN Truong, 2021)

Meta-heuristic Optimization Algorithms:

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;

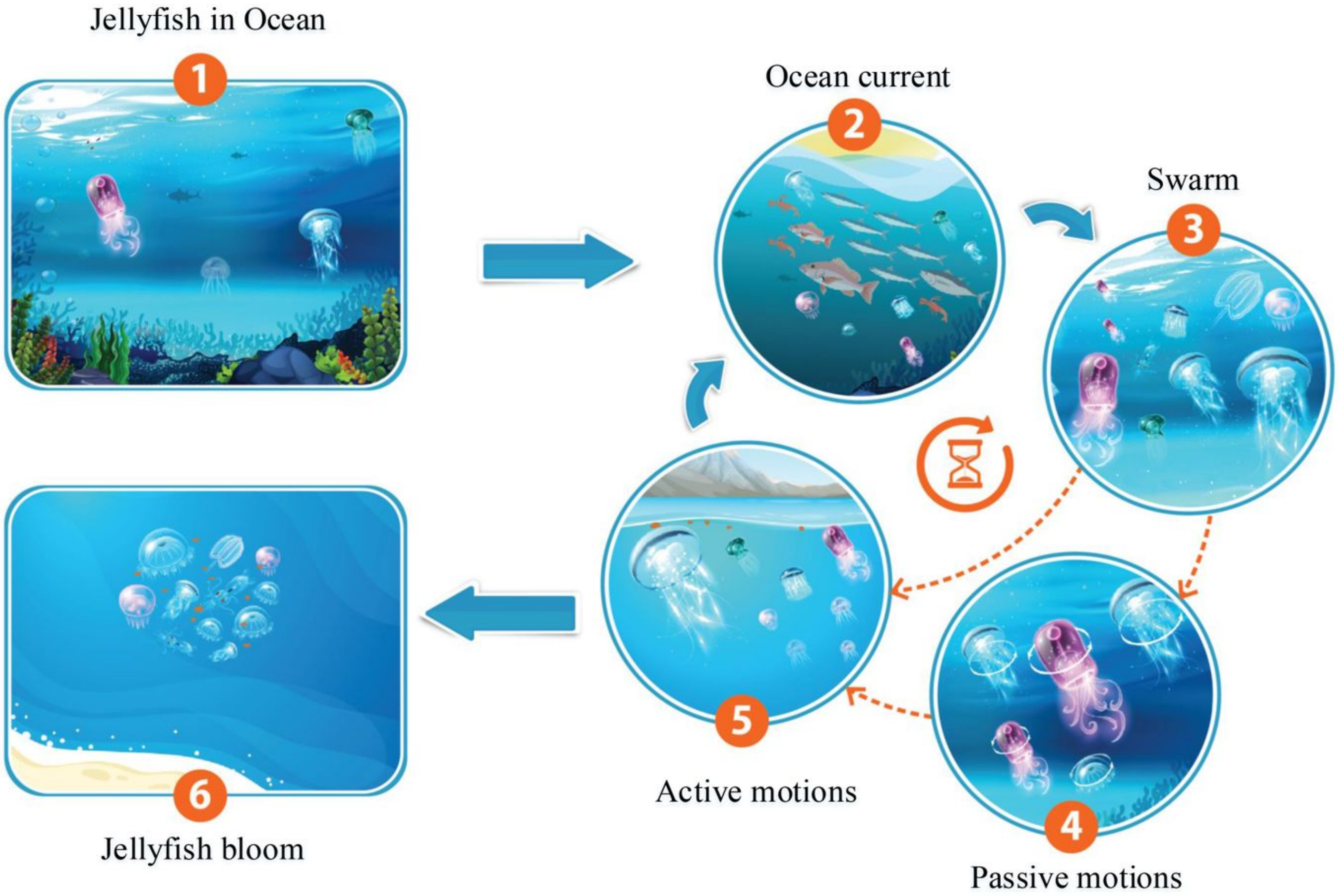
(iii) they can bypass local minima; and

(iv) they can be utilized to solve a wide range of problems in various fields

A novel metaheuristic optimizer inspired by behavior of Jellyfish in ocean

- Jellyfish have features that enable them to control their movements. Despite this ability, they mostly drift in the water, depending on currents and tides .
- When conditions are favorable, jellyfish can form a swarm, and a large mass of jellyfish is called a jellyfish bloom .
- 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..
- 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 optimizer.

Behaviour of Jellyfish in the Ocean:



Algorithm:

The proposed optimization algorithm is based on three idealized rules:

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

Equations:

The direction of the ocean current (trend) is determined by averaging all the vectors from each jellyfish in the ocean to jellyfish that is currently in the best location.

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

Now, the new location of each jellyfish is given by

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

where, $\mu = \frac{\sum X_i}{n_{Pop}}$

nPop = number of jellyfish

$X*$ = jellyfish currently with the best location in the swarm

Xi = Logistic chaotic value of location of the i th jellyfish.

Xi(t) = Location of i th jellyfish at time t

μ = mean location of all jellyfish

β = distribution coefficient

Equations:

In swarm, jellyfish are passive (type A) and active (type B) motions, respectively. 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.

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

where γ is the Motion coefficient

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

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

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

where f is an objective function of location X

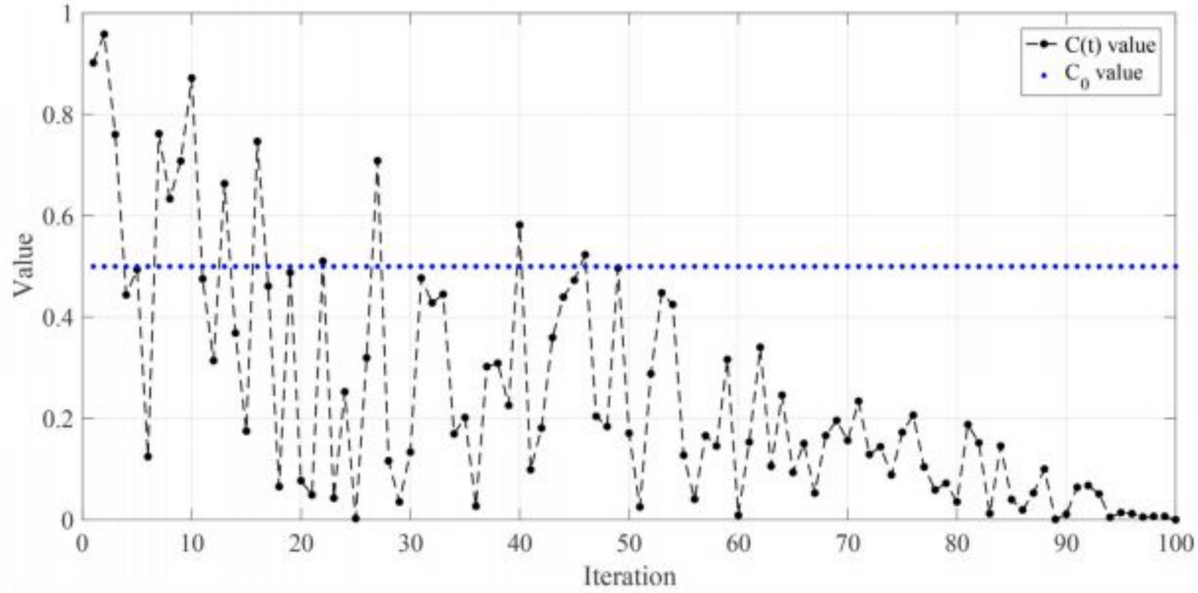
Type B motion is the motion of jellyfish is given by:

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

Equations:

Time control mechanism: 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 Co. The time control function is a random value that fluctuates from 0 to 1 over time.

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



a) Time control function.

Equations:

Population initialization: The population of jellyfish is normally initialized using logistic map, which is one of the simplest chaotic maps

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

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.

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.

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

$X_{i,d}$ 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.

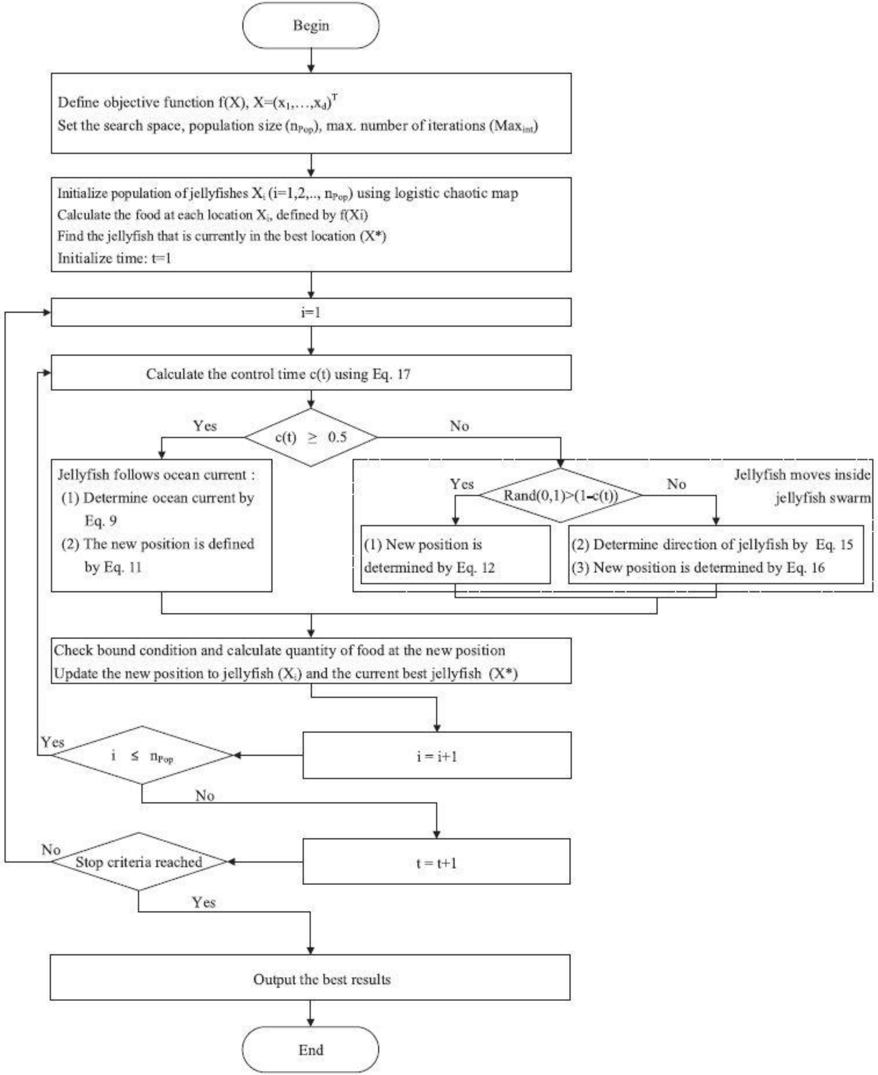
Objective Functions Used:

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.

FlowChart:



Begin
Define objective function f(X), X=(x1,…,xd)T
Set the search space, population size (nPop), max. number of iterations (Maxint)
Initialize population of jellyfishes Xi (i=1,2,.., nPop) using logistic chaotic map
Calculate the food at each location Xi, defined by f(Xi)
Find the jellyfish that is currently in the best location (X*)
Initialize time: t=1
i=1
Calculate the control time c(t) using Eq. 17
c(t) ≥ 0.5
Yes
No
Jellyfish follows ocean current :
(1) Determine ocean current by Eq. 9
(2) The new position is defined by Eq. 11
Jellyfish moves inside jellyfish swarm
Rand(0,1)>(1-c(t))
Yes
No
(1) New position is determined by Eq. 12
(2) Determine direction of jellyfish by Eq. 15
(3) New position is determined by Eq. 16
Check bound condition and calculate quantity of food at the new position
Update the new position to jellyfish (Xi) and the current best jellyfish (X*)
i = i+1
i ≤ nPop
Yes
No
t = t+1
Stop criteria reached
No
Yes
Output the best results
End

Pseudocode:

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 ($i=1,2,\ldots, n_{Pop}$) *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 $c(t) \geq 0.5$*: 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

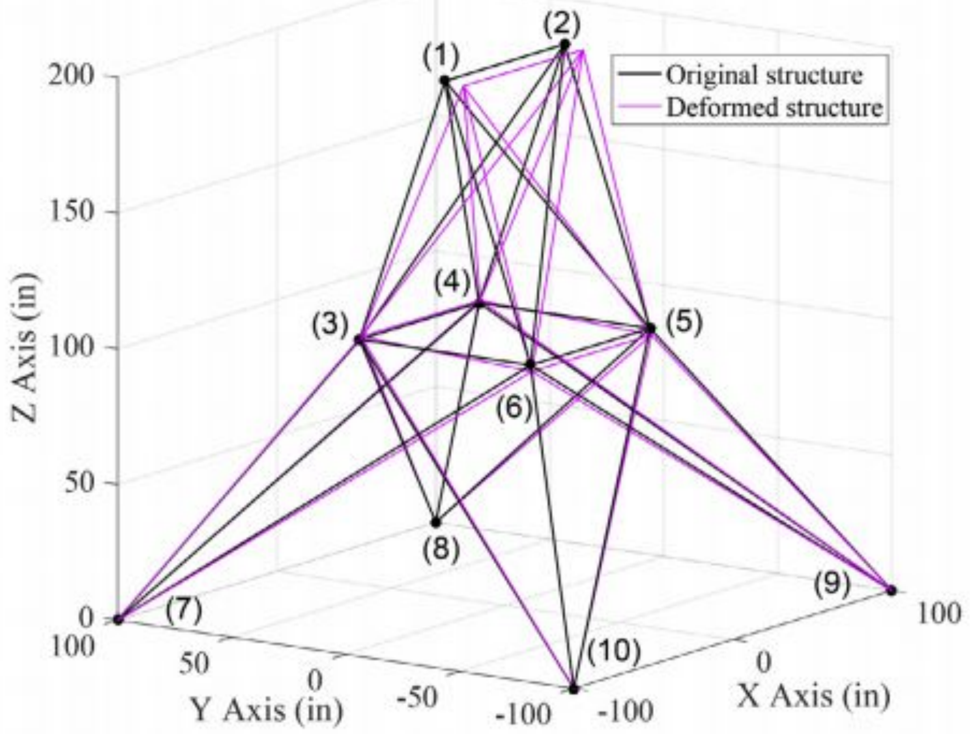
Problems which can be solved using Jellyfish Search

Structural tower designs:

- 25-Bar tower
- 52-Bar tower
- 582-Bar tower

25 - Bar Tower:

The first structural optimization problem that is solved in this study involves the spatial 25-bar tower that is displayed.



a) Schematic of 25-bar tower.

25 - Bar Tower:

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

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

Subject to

$$\sigma_i^c \leq \sigma_i \leq \sigma_i^t, \ i = 1, 2, \ldots, nm$$

$$\delta_{min} \leq \delta_j \leq \delta_{max}, \ j = 1, 2 \ldots, nn$$

$$A_{min} \leq A_k \leq A_{min}, \ k = 1, 2, \ldots, ng$$

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

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

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

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

$$\varepsilon = \varepsilon_0 \left(1 + \frac{t}{Max_{iter}}\right) \tag{30}$$

The table indicates that all compared algorithms reached the optimum value (484.85 lb) and that JS and DAJA required the fewest structural analyses-600 for JS and 511 for DAJA. These results demonstrate that **JS was one of the most efficient algorithms for solving the 25-bar tower problem.**

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

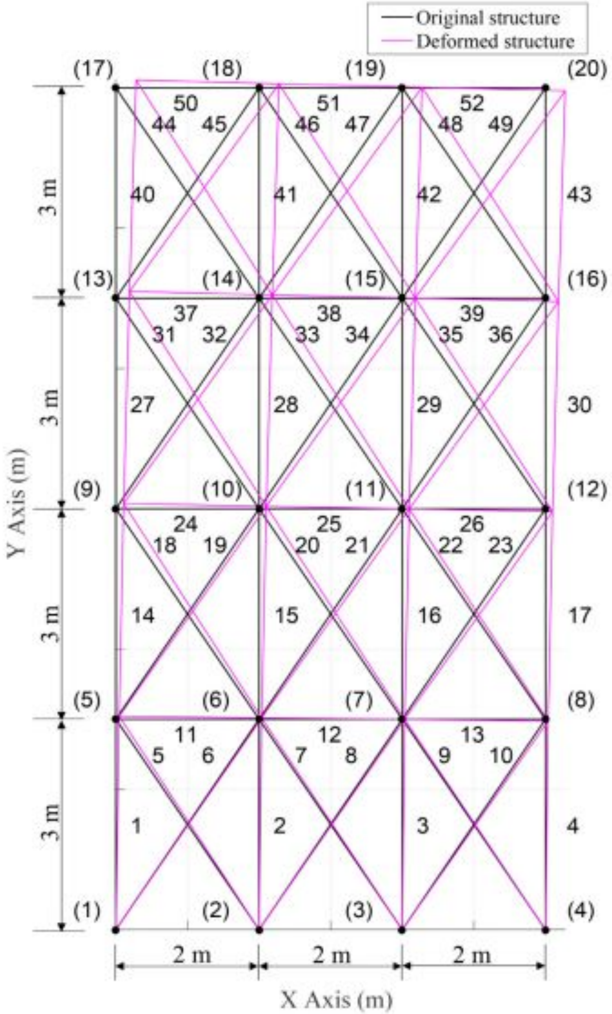
Design variables (in^2)	AFA [73,79]	HHS [81]	aeDE [89]	DBB-BC [90]	DAJA [73]	JS (This study)
A_1	0.1	0.1	0.1	0.1	0.1	**0.1**
$A_2 \sim A_5$	0.3	0.3	0.3	0.3	0.3	**0.3**
$A_6 \sim A_9$	3.4	3.4	3.4	3.4	3.4	**3.4**
$A_{10} \sim A_{11}$	0.1	0.1	0.1	0.1	0.1	**0.1**
$A_{12} \sim A_{13}$	2.1	2.1	2.1	2.1	2.1	**2.1**
$A_{14} \sim A_{17}$	1	1	1	1	1	**1**
$A_{18} \sim A_{21}$	0.5	0.5	0.5	0.5	0.5	**0.5**
$A_{22} \sim A_{25}$	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.

52 - Bar Tower:

The second optimized structural problem involved the 52-bar tower that is shown in Figure.

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



a)Schematic of 52-bar tower.

582 - Bar Tower:

The last test problem involved the 582-bar tower structure that is shown in Figure. This structure was originally proposed by Hasançebi et al. and subsequently studied in detail by Kaveh and Talatahari . 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 140W-shape steel profiles.. The modulus of elasticity (E) and yield stress (Fy) are 200 GPa and 248 MPa, respectively.

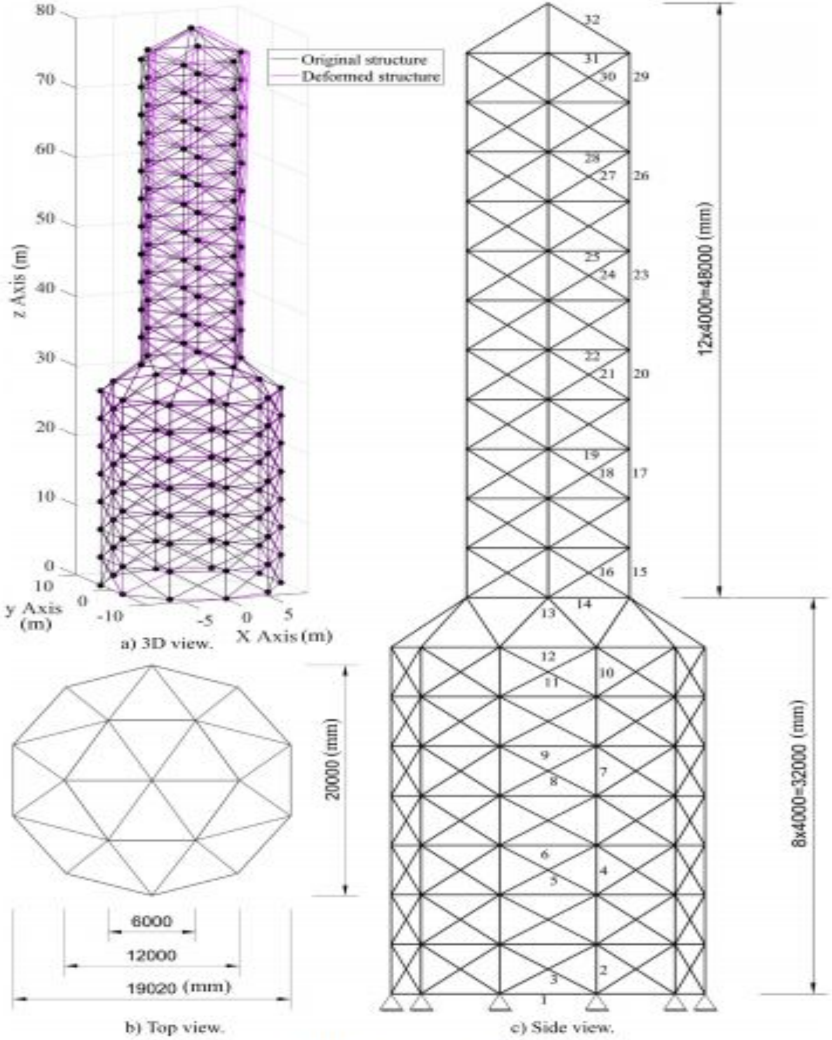


Fig. 17 Schematic of 582-bar tower.

Conclusion:

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.Notably, JS requires less time and faster convergence than the other algorithms. On the large-scale mathematical functions.

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 m3, 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

Thanks!