Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 8548639, 32 pages
https://doi.org/10.1155/2021/8548639

Research Article

Social Network Search for Solving Engineering Optimization Problems

Hadi Bayzidi,[1] **Siamak Talatahari**,[1,2] **Meysam Saraee**,[1]
and Charles-Philippe Lamarche[3]

[1] *Department of Civil Engineering, University of Tabriz, Tabriz, Iran*
[2] *Engineering Faculty, Near East University, North Cyprus, Mersin 10, Turkey*
[3] *Department of Civil and Building Engineering, Université de Sherbrooke, Sherbrooke, Canada*

Correspondence should be addressed to Siamak Talatahari; siamak.talat@gmail.com

Received 12 June 2021; Revised 16 August 2021; Accepted 3 September 2021; Published 30 September 2021

Academic Editor: Radu-Emil Precup

In this paper, a new metaheuristic optimization algorithm, called social network search (SNS), is employed for solving mixed continuous/discrete engineering optimization problems. The SNS algorithm mimics the social network user's efforts to gain more popularity by modeling the decision moods in expressing their opinions. Four decision moods, including imitation, conversation, disputation, and innovation, are real-world behaviors of users in social networks. These moods are used as optimization operators that model how users are affected and motivated to share their new views. The SNS algorithm was verified with 14 benchmark engineering optimization problems and one real application in the field of remote sensing. The performance of the proposed method is compared with various algorithms to show its effectiveness over other well-known optimizers in terms of computational cost and accuracy. In most cases, the optimal solutions achieved by the SNS are better than the best solution obtained by the existing methods.

1. Introduction

Optimization is a part of the nature of human works, in which almost all of the human decisions go through an optimal process [1]. Optimization is embedded in the essence of the many branches of science, for example, a system with minimal energy in physics, the maximum profit in business, survival of the best organism in biology, and designing an engineering system that satisfies a set of constraints [2, 3]. Almost all of the engineering problems contain several nonlinear and complex constraints depending on the design criteria and safety rules.

Over the last decades, various types of methods have been developed to solve constrained engineering problems. Two well-known groups of these methods are mathematical and metaheuristic methods. The idea of mathematical methods can be attributed to the development of the calculus of variations [4]. These methods employ the gradient of the objective function and constraints of the problem to find the optimal solution. The results of these methods are exact. However, these approaches search in a space near the starting point, which makes them sensitive to the initial starting point. In other words, just a correct starting point can lead to the global optima. In dealing with complex optimization problems, these methods are not suitable and frequently reach local solutions, and in some real applications, the gradient of the objective function and constraints is impossible to be calculated [5]. These drawbacks encourage researchers towards metaheuristic methods. Metaheuristic methods try to combine basic heuristic methods with randomization and rule-based theories, which are usually taken from natural phenomena such as evolution, swarm intelligence, and governing laws in different physics theories. Metaheuristic algorithms are approximate, but their results have high accuracy and are very close to the global optimum solution [6]. These methods are problem-independent, and the starting point does not determine the quality of the final solutions. Besides, these methods employ different operators to perform a global search in the space of the problem at an appropriate speed. These features have made them popular in recent decades. Also, these types of algorithms are among the most popular techniques that are employed for solving optimization problems in different fields, such as computer and electrical engineering [7], water, geotechnical and transport engineering [8], structure and infrastructures engineering [9], robotic [10], project and construction management [11], feature selection and data mining [12, 13], industrial and manufacturing [14], and medicine and biology [15].

Glover [16] introduced the term metaheuristic firstly. The word metaheuristic is a combination of two old Greek words: meta and heuristic. The word heuristic has its origin in the old Greek work heuriskein, which means the art of discovering new strategies (rules) to solve problems. The suffix meta also is a Greek word that means "upper-level methodology" [17]. Almost every metaheuristic algorithm follows the general process shown in Figure 1. Algorithm steps cause fundamental differences in the performance of algorithms when faced with different problems. In the other words, algorithm steps represent the unique operators of each algorithm in which new solutions are generated. The operators of each algorithm refer to the optimal process of a particular phenomenon that those algorithms have imitated.

According to the type of basic phenomena of each method, metaheuristic algorithms can be classified into four main categories: (1) evolutionary, (2) swarm intelligence, (3) physics-based, and (4) human-based algorithms. Evolutionary algorithms are motivated by natural evolution. Swarm intelligence algorithms model the natural behavior of animals in teamwork such as foraging and hunting. Physical phenomena and laws of science inspire physics-based algorithms, and finally, human-based algorithms mimic various optimal behaviors of humans in different conditions. Some of the most popular and novel algorithms are presented in Table 1.

Each of these algorithms can behave differently when dealing with different problems, so that one particular algorithm may not solve some problems. Therefore, it is necessary to create a new high-performance optimization algorithm that is able to solve more types of problems. Novel metaheuristic methods are developed to find the optimal solution for complex and large-scale problems in less time than previous ones, with higher accuracy. These aims are satisfied by developing more robust algorithms that have a better ability to search the space of problems to find a better solution. In addition, this property arises from the right balance between exploration and exploitation of the proposed algorithm. Exploitation means searching around the current best solutions, while exploration tries to explore the search space more efficiently, often by randomization [42].

In addition to inventing novel algorithms based on natural phenomena, developing new algorithms using hybridizing the operators of the current methods or modifying them is a hot topic in the field of metaheuristic algorithms. Firefly algorithm with chaos [43], hybrid particle swarm optimizer, ant colony strategy and harmony search scheme (HPSACO) [44], island-based cuckoo search with highly disruptive polynomial mutation (iCSPM) [45], quantum-behaved developed swarm optimizer (QDSO) [46], hybrid self-assembly with particle swarm optimization (SAPSO) [47], upgraded whale optimization algorithm (UWOA) [48], fuzzy controllers with slime mould algorithm (SMAF) [49], and hybrid invasive weed optimization-shuffled frog-leaping (SFLA-IWO) [50, 51] are some the newly developed hybrid or modified optimization algorithms.

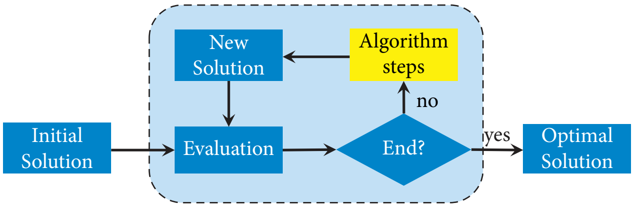


Figure 1: The general form of optimization algorithms.

Table 1: List of some popular and new metaheuristic algorithms.

Algorithm	References
Evolution strategy (ES)	[18]
Genetic algorithms (GA)	[19]
Ant colony optimization (ACO)	[20]
Particle swarm optimization (PSO)	[21]
Differential evolution (DE)	[22]
Cuckoo search (CS)	[23]
Bat algorithm (BA)	[24]
Charged system search (CSS)	[25]
Firefly algorithm (FA)	[26]
Eagle strategy (ES)	[27]
Krill herd algorithm (KH)	[28]
Flower pollination algorithm (FPA)	[29]
Grey wolf optimizer (GWO)	[30]
Optimization based on phylogram analysis (OPA)	[31]
Whale optimization algorithm (WOA)	[32]
Developed swarm optimizer (DSO)	[33]
Stochastic paint optimizer (SPO)	[34]
Chaos game optimization (CGO)	[35, 36]
Atomic orbital search (AOS)	[37, 38]
Material generation algorithm (MGA)	[39]
Crystal structure algorithm (CryStAl)	[40]
Social network search (SNS)	[41]

Social network search (SNS) algorithm is a robust metaheuristic algorithm that was innovated as a novel method for solving optimization problems, and its results showed that it is capable of outperforming various methods in dealing with different optimization problems [42]. The SNS algorithm simulates human behavior as users of a social network. Social network users can influence the opinions of other users on the network by sharing their views, opinions, and thoughts. Each of the users can also share their thoughts on the network and affect other people's opinions. In other words, the SNS simulates particular moods where the views and opinions of users are influenced under their communications. This paper investigates the performance of the SNS algorithm using 14 constrained engineering optimization problems and a real application in the field of satellite image segmentation. The obtained results are compared with other optimizers in terms of best function value and number of function evaluations, and in most cases, the solutions of the SNS are better than the other methods.

The rest of this paper is organized as follows. Section 2 describes the SNS algorithm and constraint-handling technique. The performance of the SNS algorithm in solving optimization problems is evaluated against other methods in Section 3. Finally, conclusions are given in Section 4.

2. Materials and Methods

This section presents the general framework of the SNS algorithm and the utilized constraint-handling technique for solving engineering optimization problems.

2.1. Social Network Search (SNS). Human beings are social species, which always try to communicate with each other. Social networks are virtual tools that were created for this goal with the advent of technology. The proposed SNS algorithm simulates the interactive behavior among users in social networks to achieve more popularity. Social networks are platforms where users can interact virtually with other users. Interacting with other users of the network may affect their opinions. The process of interacting with and influencing other users of the network goes through an optimal process so that users are always trying to increase their level of popularity on the network.

The main property of social networks is that users can follow other persons, as shown in Figure 2. If a user shares a new post, that person's followers may be informed about the shared topic. This feature (fast propagation of views) has turned networks into a powerful tool for promoting information and ideas, which is due to having high connectivity of users in the social networks, as demonstrated in Figure 2.

In social networks, users' viewpoint can be affected by other views in different moods containing imitation, conversation, disputation, and innovation. One of these moods that look like real-world social behavior creates the new solution in the SNS algorithm. Description and mathematical modeling of these moods are as follows [42].

2.1.1. Mood 1: Imitation. Imitation means that the views of other users are attractive, and usually, users try to imitate each other in expressing their opinions as follows:

$$\begin{aligned} X_{i\text{new}} &= X_j + \text{rand}(-1,1)\times R,\\ R &= \text{rand}\,(0,1)\times r, \\ r &= X_j - X_i, \end{aligned} \tag{1}$$

where X_j represents the vector of the jth user's view, which is selected randomly ($i \neq j$), X_i is the view vector of the ith user, and rand$(-1,1)$ and rand $(0,1)$ are two random vectors in intervals $[-1, 1]$ and $[0, 1]$, respectively. In this mood, the new solution will be generated according to imitation space (Figure 3(a)), and this space is created using the radii of shock and popularity. The shock radius (R) reflects the amount of influence of the jth user, and its magnitude is considered as a multiple of r. The value of r shows the popularity radius of the jth user, which is calculated based on the difference in the opinions of the ith and jth users. Also, the final effect of the shock radius is reflected by multiplying its value to a random vector in the interval of $[-1, 1]$, in which if the components of the random vector are positive, the shared view will be agreed with the jth opinion and vice versa. The process of the imitation mood is illustrated in Figure 3(a). As can be seen, using equation (1), the space of imitation will be formed, and then a point as a new view will be shared on the network.

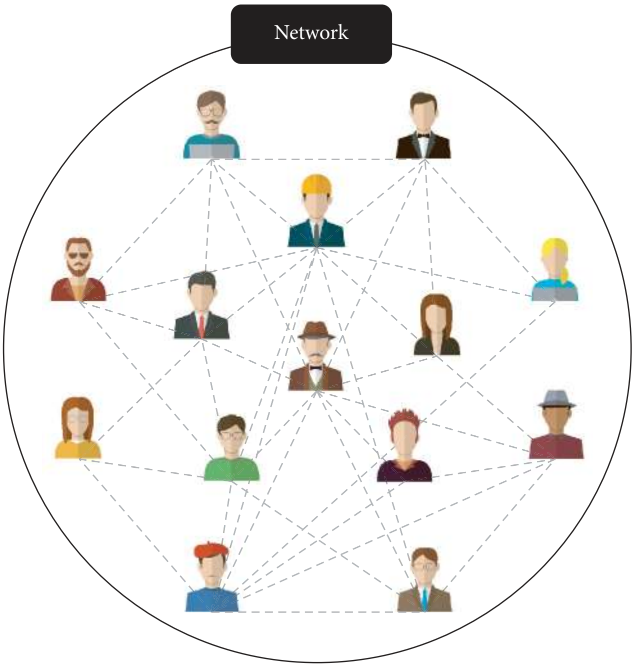
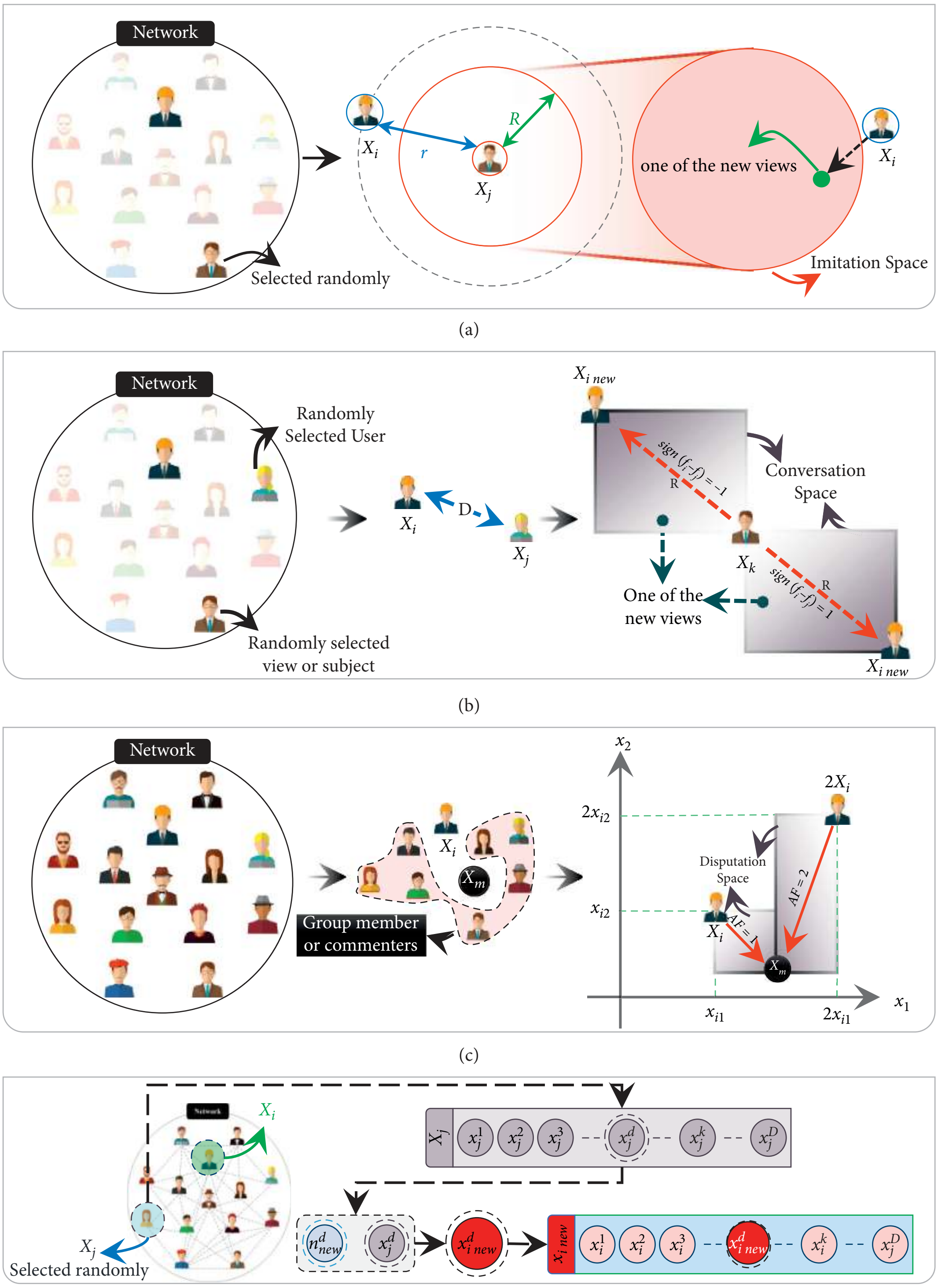


Figure 2: A general model for the social network.

2.1.2. Mood 2: Conversation. In social networks, users can communicate with each other and benefit from their conversation about different issues according to

$$\begin{aligned} X_{i\text{new}} &= X_k + R,\\ R &= \text{rand}\,(0,1)\times D, \\ D &= \text{sign}\big(f_i - f_j\big)\times\big(X_j - X_i\big), \end{aligned} \tag{2}$$

where rand $(0,1)$ is a random vector in the interval $[0, 1]$, X_j and X_k are the vectors of two randomly selected positions somehow $i \neq j \neq k$, and f_i and f_j are the objective functions of X_i and X_j, respectively. This mood models a state in which users learn from each other and increase their information about events. In conversation, users find a sight about a specific issue through other views, and finally, due to the differences in opinions, they can draw a new vision of the issue under discussion. X_k demonstrates the vector of the issue, which is randomly chosen to speak about it; also, R is the effect of chat, which is based on the differences of opinion and represents the change in their beliefs about the



(a)

(b)

(c)

(d)

Figure 3: The process of different moods in the SNS algorithm. (a) Imitation. (b) Conversation. (c) Disputation. (d) Innovation.

issue. D is the difference between the views of users. In addition, sign(.) is the sign function and $\mathrm{sign}(f_i - f_j)$ determines the moving direction of X_k by comparing f_i and f_j. The process of this decision mood is shown in Figure 3(b). As can be seen, the user's view about the issue changes as a result of conversations with the jth user. The changed opinion is considered as a new view to share with others. Changing the user's view about the events is considered as the relocation of the events.

2.1.3. Mood 3: Disputation. The disputation mood imagines a state where users explain their views about events to others and defend their opinion. In this situation, users see different views from other persons and may be influenced by the expressed reasons. The new affected view in disputation is as follows:

$$X_{i\text{new}} = X_i + \text{rand}(0,1) \times (M - AF \times X_i),$$
$$M = \frac{\sum_t^{N_r} X_t}{N_r}, \tag{3}$$
$$\text{AF} = 1 + \text{round}(\text{rand}),$$

where rand $(0,1)$ is a random vector in the interval [0, 1] and N_r is a random integer between 1 and N_{user} (N_{user} is the population size or network size). N_r determines the number of users who participate in disputation, and participants are selected randomly. AF is the admission factor, which indicates the insistence from users on their opinion in discussions with other persons and is a random integer that can be either 1 or 2. round(.) is a function that rounds its input to the nearest integer number. The process of disputation mood is shown in Figure 3(c).

2.1.4. Mood 4: Innovation. Sometimes a topic that users share on the networks comes from their new experiences and thoughts. In this mood, the new solution is developed by changing a randomly selected variable of X_i as follows:

$$X_{i\text{new}} = [x_1, x_2, x_3, \ldots, x^d_{inew}, \ldots, x_D],$$
$$x^d_{i\text{new}} = t \times x^d_j + (1-t) \times n^d_{\text{new}},$$
$$n^d_{\text{new}} = lb_d + \text{rand}_1 \times (ub_d - lb_d), \tag{4}$$
$$t = \text{rand}_2,$$

where d is the dth variable that is selected randomly in the interval [1, D], D is the number of problem variables, and rand$_1$ and rand$_2$ are two random numbers in the interval [0, 1]. Also, ub_d and lb_d are maximum and minimum values for the dth variable. n^d_{new} represents the new idea about the selected dimension. x^d_j is the current idea about the dth variable presented by another user (jth user which selected randomly and $i \neq j$) that ith user wants to change it because of the new idea (n^d_{new}). Innovation models a state in which a person thinks about a specific issue, perhaps looks at that issue in a novel way, and is able to understand the nature of that problem more accurately or can find a completely different view about it. A particular subject may have distinct features, and each of them affects the understanding of the problem. As a result, by changing the idea about one of them (x^d_i), the general concept of the subject will change, and a novel view will be achieved. $x^d_{i\text{new}}$ is a new insight into the issue under consideration from the dth viewpoint and is replaced with the current view (x^d_i). The outline of the construction of the new view is shown in Figure 3(d).

In the SNS algorithm, only one of the predefined four models, so-called decision moods, will be selected and executed randomly for each user in each iteration of the algorithm. In other words, all of the moods described here are real-world behaviors of users in social networks, and it seems that the correct assumption is that only one of these moods occurs at a specific time (iteration) for each users. As a result, the chance of occurrence of these moods is considered to be small by using a random procedure with a uniform distribution.

An important point is that the SNS algorithm has no specific parameter to be fine-tuned, and this feature is one of its superiority. In the third mood of the SNS algorithm, AF is defined as a random integer, and it can be considered a deterministic parameter whose value is generated randomly. To utilize the SNS algorithm, it just needs to determine the number of users (population size) and the maximum number of evaluations or iterations. The flowchart of the SNS algorithm is illustrated in Figure 4. Besides, the MATLAB code of the SNS algorithm for solving engineering optimization problems is available in [52].

2.2. Constraint-Handling Technique. Most engineering optimization problems aim to find optimal solutions under special conditions, which are usually based on resource limitations, design principles, and safety requirements. These special conditions are called constraints, and the main aim of constraint optimization is to find a feasible solution. A constrained optimization problems can be formulated as

$$\begin{aligned}
&\text{minimize:}\\
&\quad f(X), \quad X = (x_1, x_2, \ldots, x_d),\\
&\quad \text{subject to:}\\
&\quad g_i(X) \leq 0, \quad i = 1, 2, \ldots, n_g,\\
&\quad h_j(X) = 0, \quad j = 1, 2, \ldots, n_h,\\
&\quad lb_k \leq (x_k) \leq ub_k, \quad k = 1, 2, \ldots, d,
\end{aligned} \tag{5}$$

where the function $f(X)$ is an objective function, X is a vector of solution variables (and they can be continuous, discrete, or mixed), $g_i(X)$ and $h_j(X)$ are inequality and equality constraints, respectively, in which n_g is the number of inequality constraints and n_h is the number of equality constraints, d represents the number of variables, and lb_k and ub_k are the minimum and the maximum permissible values for the kth variable, respectively. The point worth mentioning is that a feasible solution satisfies all constraints. In contrast, infeasible solutions do not satisfy at least one constraint [53]. Also, in dealing with equality constraints, $h_j(X) = 0$ is replaced with an inequality $|h_j(X)| - \delta \leq 0$, where δ is a positive tolerance value. Another approach for handling the equality constraints is to replace $h_j(X) = 0$ with two inequality constraints of the type $h_j(X) \leq 0$ and $h_j(X) \geq 0$. This strategy facilitates convergence to the optimum design [54]. Therefore, all of the constraints can be transformed into inequality constraints.

Metaheuristic algorithms cannot solve constraint optimization problems, directly. Therefore, it needs to equip them with an additional tool for handling constraints. A group of methods was developed for this proposal which is called constraint-handling techniques (CHTs). The CHTs enable the optimization methods to handle the objective function and constraints, simultaneously. CHTs are grouped

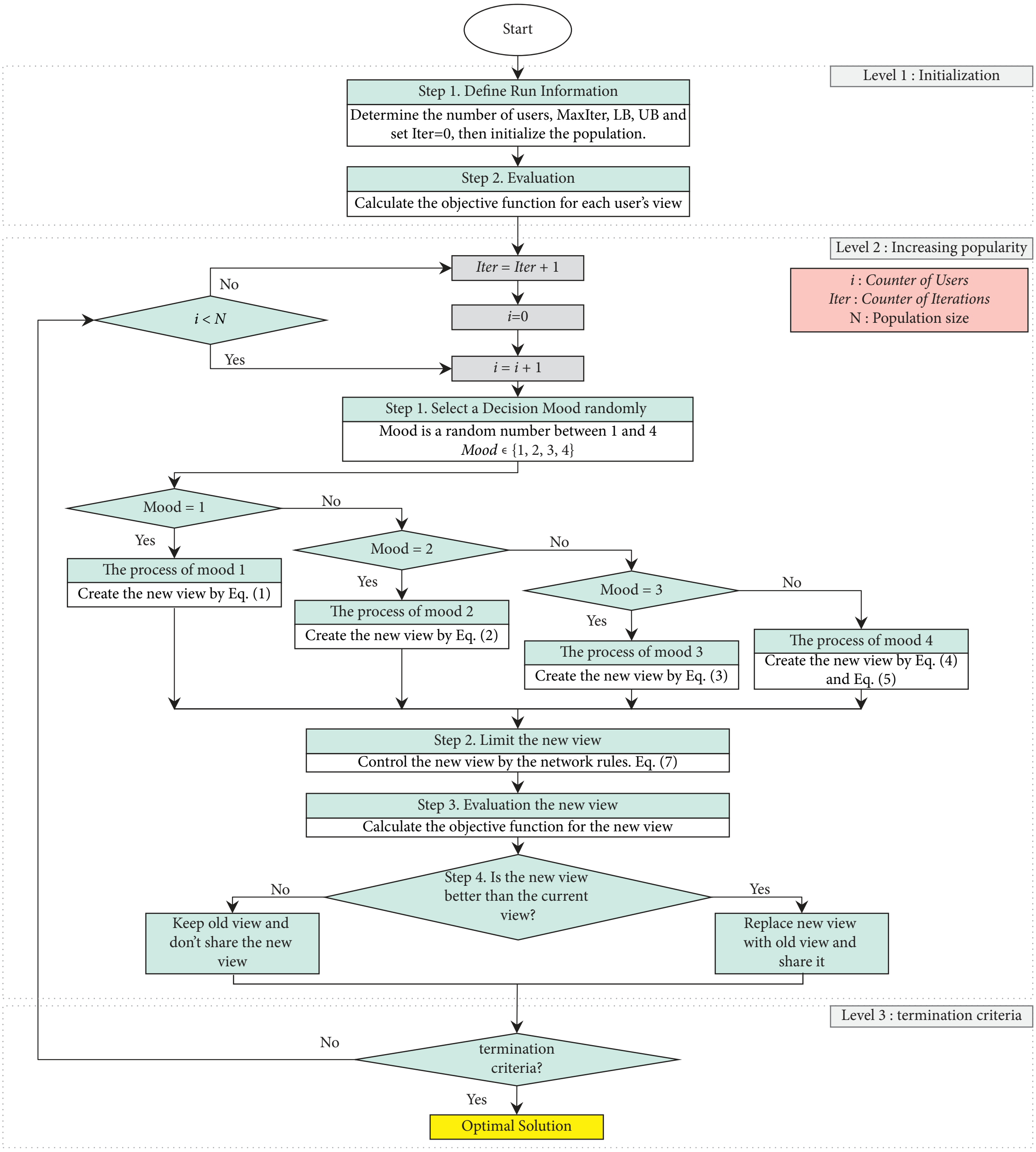

Start
Level 1 : Initialization
Step 1. Define Run Information
Determine the number of users, MaxIter, LB, UB and set Iter=0, then initialize the population.
Step 2. Evaluation
Calculate the objective function for each user's view
Level 2 : Increasing popularity
i : Counter of Users
Iter : Counter of Iterations
N : Population size
Iter = Iter + 1
No
i < N
i=0
Yes
i = i + 1
Step 1. Select a Decision Mood randomly
Mood is a random number between 1 and 4
Mood ϵ {1, 2, 3, 4}
Mood = 1
No
Yes
The process of mood 1
Create the new view by Eq. (1)
Mood = 2
No
Yes
The process of mood 2
Create the new view by Eq. (2)
Mood = 3
No
Yes
The process of mood 3
Create the new view by Eq. (3)
The process of mood 4
Create the new view by Eq. (4) and Eq. (5)
Step 2. Limit the new view
Control the new view by the network rules. Eq. (7)
Step 3. Evaluation the new view
Calculate the objective function for the new view
Step 4. Is the new view better than the current view?
No
Yes
Keep old view and don't share the new view
Replace new view with old view and share it
Level 3 : termination criteria
No
termination criteria?
Yes
Optimal Solution

Figure 4: The flowchart of the SNS algorithm.

into five categories: (1) penalty functions, (2) special representations and operators, (3) repair algorithms, (4) separation of objectives and constraints, and (5) hybrid methods [53]. The first method, penalty functions, is a simple and standard procedure for handling constraints. In the penalty function approach, a penalty term is added to the objective function, and then a constrained optimization problem is transformed into an unconstrained one. A penalty function can be formulated as follows:

$$F(X) = f(X) + P(X), \tag{6}$$

where $F(X)$ is the fitness function, which expresses the unconstrained state of the constrained problem, $f(X)$ is the objective function, and $P(X)$ is the penalty term that denotes the violation of constraints and is calculated as follows:

$$\begin{aligned} P(X) = {} & \sum_{i=1}^{n_g} \alpha_i \times \max\left(0,\ g_i(X)\right) \\ & + \sum_{j=1}^{n_h} \beta_j \times \max\left(0, \left|h_j(X)\right| - \delta\right), \end{aligned} \tag{7}$$

where $\max(0, g_i(X))$ and $\max(0, |h_j(X)| - \delta)$ represent the value of the violations of the solutions according to the ith inequality and jth equality constraints, respectively. Also, α_i and β_j are penalty factors for these constraints, respectively. The magnitude of penalty factors affects the quality of answers, and the suitable penalty factors are problem-dependent.

To solving constrained optimization problems, metaheuristic methods and CHT should be liked for recognizing feasible search space. Then, the optimizer should try to find the optimum or a near-optimum solution in the feasible region. Therefore, in each iteration of an algorithm, the fitness of the population is evaluated according to objective and constraint(s), and based on the calculated fitness function, the next generation of the population will be generated. In other words, the algorithm will identify the problem's search space using the fitness of the current population.

3. Results and Discussion

This section evaluates the performance of the SNS algorithm using 15 benchmark problems in various fields of engineering. One of these problems deals with the segmentation of satellite images in the field of remote sensing as a real application of metaheuristic algorithms in engineering. Each of these examples was run 30 times independently using the SNS algorithm, and the results are compared with different counterpart algorithms from the literature. In selecting the counterpart algorithm, an attempt has been made to use the results of newly developed methods.

3.1. Cantilever Beam. This problem is a structural engineering design example that is related to the weight optimization of a cantilever beam with a square cross section [55]. The beam is rigidly supported at one end, and a vertical force acts on the free node of the cantilever, as shown in Figure 5. The beam consists of five hollow square blocks with constant thickness, whose heights (or widths) are the decision variables, and the thickness is held fixed (here 2/3). This problem can be expressed analytically as follows:

$$\begin{aligned}
&\text{minimize:}\\
&\quad f(X) = 0.0624\left(x_1 + x_2 + x_3 + x_4 + x_5\right),\\
&\quad \text{subject to:}\\
&\quad g(X) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0,\\
&\quad \text{variable range:}\\
&\quad 0.01 \le x_i \le 100, \qquad i = 1, \ldots, 5.
\end{aligned} \tag{8}$$

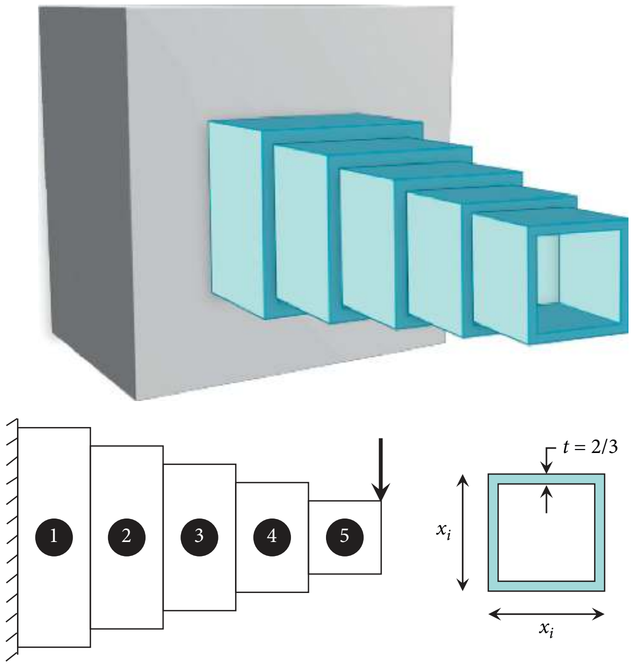


Figure 5: Schematic representation of cantilever beam.

The best solutions for solving this problem obtained by the SNS and various methods are listed in Table 2. It can be seen that the solution obtained by the SNS is better than that of the other methods. In addition, the SNS terminated after 12,000 evaluations. The statistical results of the SNS and other methods are listed in Table 3, and based on them, it can be seen that the SNS algorithm has obtained a more accurate answer in a smaller number of function evaluations (NFEs).

3.2. Optimal Design of I-Shaped Beam. The other typical engineering optimization problem is the I-beam design problem, which aims to minimize the vertical deflection of the beam shown in Figure 6. It simultaneously satisfies the cross-sectional area and stress constraints under given loads. The width of flange $b (= x_1)$, the height of section $h (= x_2)$, the thickness of the web $t_w (= x_3)$, and the thickness of the flange $t_f (= x_4)$ are variables of this problem. The maximum vertical deflection of the beam is $f(x) = \mathrm{PL}^3/48\mathrm{EI}$ when the length of the beam (L) and modulus of elasticity (E) are 5200 cm and 523.104 kN/cm^2, respectively. The objective function and constraints of this problem are formulated as follows:

Table 2: Best results of the cantilever beam design example.

Algorithm	Variables					Constraint	
	x_1	x_2	x_3	x_4	x_5	$g_1(X)$	$f(X)$
CS [18]	6.0089	5.3049	4.5023	3.5077	2.1504	$-6.448E-05$	1.33999
MFO [56]	5.98487	5.31672	4.49733	3.51361	2.16162	$4.182E-09$	1.33998
ALO [57]	6.01812	5.31142	4.48836	3.49751	2.15832	$-2.995E-06$	1.33995
SOS [58]	6.01878	5.30344	4.49587	3.49896	2.15564	$1.393E-04$	1.33996
SSA [59]	6.01513	5.30930	4.49500	3.50142	2.15278	$4.0578E-09$	1.33995
SNS (present study)	6.01545	5.31066	4.48800	3.50528	2.15428	$-3.2718E-07$	1.33995

MFO: moth-flame optimization algorithm. ALO: ant lion optimizer. SSA: salp swarm algorithm.

Table 3: Comparative results of SNS with other methods for the cantilever beam design problem.

Algorithm	Worst	Mean	Best	SD	NFEs
SOS [58]	NA	1.33997	1.33996	$1.1E-5$	15,000
CGO [37]	1.340602	1.340052	1.339970	$1.2245E-04$	100,000
AOS [39]	1.491711	1.351954	1.339957	0.02499743	100,000
MGA [40]	1.3402011	1.3400526	1.3399756	$6.99E-05$	100,000
SNS (present study)	1.3399576	1.3399576	1.3399576	$1.1102E-15$	12,000

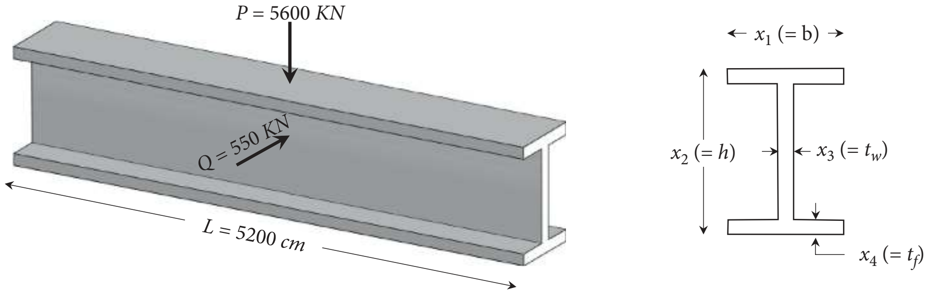


Figure 6: A 3D view of beam design problem.

minimize:

$$f(X) = \frac{5000}{x_3\left(x_2 - 2x_4\right)^3/12 + \left(x_1 x_4^3/6\right) + 2bx_4\left(x_2 - x_4/2\right)^2}$$

subject to:

$$g_1(X) = 2x_1x_3 + x_3\left(x_2 - 2x_4\right) \leq 300,$$

$$g_2(X) = \frac{18x_2 \times 10^4}{x_3\left(x_2 - 2x_4\right)^3 + 2x_1x_3\left(4x_4^2 + 3x_2\left(x_2 - 2x_4\right)\right)} + \frac{15x_1 \times 10^3}{\left(x_2 - 2x_4\right)x_3^2 + 2x_3x_1^3} \leq 56, \tag{9}$$

variable range:

$$10 \leq x_1 \leq 50,$$

$$10 \leq x_2 \leq 80,$$

$$0.9 \leq x_3 \leq 5,$$

$$0.9 \leq x_4 \leq 5.$$

Many optimizers have solved this nonlinearly constrained problem, and Table 4 presents the best results of these methods. In addition, the statistical results for comparing the performance of the SNS methods are provided in Table 5. For this case study, the SNS needs 3600 function evaluations to reach these results, and it can be seen that the SNS performs superior compared to other methods.

3.3. Three-Bar Truss Design Problem. This case considers a 3-bar planar truss structure shown in Figure 7. The volume of a statically loaded 3-bar truss is to be minimized subject to stress (σ) constraints on each of the truss members. The objective is to evaluate the optimal cross-sectional areas, $A_1\,(=x_1)$ and $A_2\,(=x_2)$. The mathematical formulation is given below:

$$
\begin{aligned}
&\text{minimize:}\\
&f(X) = \left(2\sqrt{2}\,x_1 + x_2\right) \times l,\\
&\text{subject to:}\\
&g_1(X) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0,\\
&g_2(X) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0,\\
&g_3(X) = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \le 0,\\
&l = 100\,\text{cm},\\
&P = 2\text{kN/cm}^3,\\
&\sigma = 2\text{kN/cm}^3,\\
&\text{variable range:}\\
&0 \le x_1,\\
&x_2 \le 1.
\end{aligned}
\tag{10}
$$

The best results of different methods are presented in Table 6. Also, Table 7 provides the statistical results of these algorithms. It can be seen that the best objective value of the SNS is equal or better than that of other methods. The required number of function evaluations (NFEs) for the SNS algorithm is 4800, which is much lower than that of other algorithms.

3.4. Tubular Column Design. This problem is an example of designing a uniform column of the tubular section to carry a compressive load at minimum cost. This problem has two design variables, the mean diameter of the column $d\,(=x_1)$ and the thickness of tube $t\,(=x_2)$, which are shown in Figure 8. This column is made of a material with a yield stress of $\sigma_y = 500\,\text{kgf/cm}^2$ and a modulus of elasticity of $E = 0.85 \times 10^6 6\ \text{kgf/cm}^2$. The optimization model of this problem is given as follows:

$$
\begin{aligned}
&\text{minimize:}\\
&f(X) = 9.8x_1x_2 + 2x_1,\\
&\text{subject to:}\\
&g_1(X) = \frac{P}{\pi x_1 x_2 \sigma_y} - 1 \le 0,\\
&g_2(X) = \frac{8PL^2}{\pi^3 E x_1 x_2\left(x_1^2 + x_2^2\right)} - 1 \le 0,\\
&g_3(X) = \frac{2.0}{x_1} - 1 \le 0,\\
&g_4(X) = \frac{x_1}{14} - 1 \le 0,\\
&g_5(X) = \frac{0.2}{x_2} - 1 \le 0,\\
&g_6(X) = \frac{x_2}{8} - 1 \le 0,\\
&\text{variable range:}\\
&2 \le x_1 \le 14,\\
&0.2 \le x_2 \le 0.8.
\end{aligned}
\tag{11}
$$

According to the constraints g_1 and g_2, the included stress in the column should be less than the buckling and yield stresses, respectively. Also, other constraints (g_3, g_4, g_5, and g_6) clamp the variables of the problem to the ranges of the variables. This problem was previously solved using various methods, and the best results of these methods and SNS are presented in Table 8. The SNS uses 1250 evaluations to solve this problem. In addition, the statistical results of some methods are reported in Table 9. According to these results, the SNS has found better results than other algorithms.

3.5. Speed Reducer Design. In mechanical systems, one of the essential parts of the gearbox is the speed reducer, and it can be employed for several applications [65]. In this optimization problem (see Figure 9), the weight of the speed reducer is to be minimized with subject to 11 constraints. This problem has seven variables, face width $b\,(=x_1)$, module of teeth $m\,(=x_2)$, the number of teeth in the pinion $z\,(=x_3)$, length of the first shaft between bearings $l_1\,(=x_4)$, length of the second shaft between bearings $l_2\,(=x_5)$, the diameter of first shafts $d_1\,(=x_6)$, and the diameter of second shafts $d_2\,(=x_7)$. The mathematical formulation of this problem is formulated as follows:

Table 4: Best results for the optimal design of I-shaped beam problem.

Algorithm	Variables				Constraints		
	x_1	x_2	x_3	x_4	$g_1(X)$	$g_2(X)$	$f(X)$
IARSM [60]	79.99	48.42	0.9	2.4	0.0869999	−1.524540	0.0131
CS [18]	80	50	0.9	2.3216	−0.012005	−1.570020	0.01307
GWO [61]	80	50	0.9	2.3217	−0.009059	−1.570071	0.0131
EMGO-FCR [61]	80	50	0.9	2.3200	−0.176000	−1.567179	0.0131
SOS [58]	80	50	0.9	2.3217	−0.000222	−1.570224	0.01307
AEFA-C [61]	79.9671	49.99	0.9	2.3164	−0.560371	−1.559518	0.0131
SNS (present study)	80	50	0.9	2.3217	0	−1.570228	0.0130741

IARSM: improved adaptive response surface method. EMGO-FCR: ensemble meta model-based global optimization using fuzzy clustering. AEFA-C: artificial electric field algorithm.

Table 5: Comparison of the results of SNS with the other methods for the I-shaped beam problem.

Algorithm	Worst	Mean	Best	SD	NFEs
CS [18]	0.01353646	0.0132165	0.0130747	0.0001345	5000
SOS [58]	NA	0.0130884	0.0130741	$4.0E-5$	5000
AOS [39]	0.0138140	0.0131788	0.0130741	$1.555E-04$	100,000
SNS (present study)	0.0130764	0.0130743	0.0130741	$4.313E-07$	3600

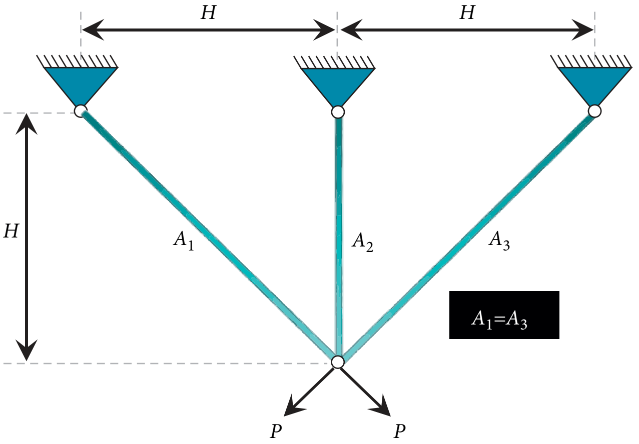


Figure 7: A schematic model of three-bar truss.

Table 6: Best results of the three-bar truss design problem.

Algorithm	Variables		Constraints			
	x_1	x_2	$g_1(X)$	$g_2(X)$	$g_3(X)$	$f(X)$
GA [62]	0.788915	0.407569	$9.6430E-07$	−1.464873605	−0.53512542	263.8958857
PSO [62]	0.788669	0.408265	$4.8650E-07$	−1.464082376	−0.535917137	263.8958434
ICA [62]	0.788625	0.408389	$8.4180E-07$	−1.463941244	−0.536057913	263.8958452
CS [18]	0.78867	0.40902	$-2.9000E-04$	−0.26853	−0.73176	263.9716
WCA [5]	0.788651	0.408316	$0.0000E+00$	−1.464024	−0.535975	263.895843
GWO [62]	0.788648	0.408325	$3.3400E-08$	−1.464014397	−0.535985569	263.8960063
ALO [57]	0.78866281	0.4082831	$-5.3170E-12$	−1.464062005	−0.53593799	263.8958434
MFO [56]	0.78824477	0.4094669	$7.7090E-12$	−1.462717072	−0.537282927	263.8959796
SSA [59]	0.78866541	0.4082757	$3.0000E-10$	−1.464070360	−0.53592963	263.8958434
WSA [62]	0.788683	0.408227	$-7.2400E-07$	−1.464126180	−0.53587454	263.8958434
SNS (present study)	0.78868473	0.4082211	$3.2978E-08$	−1.4641325	−0.5358675	263.8958434

WCA: water cycle algorithm. WSA: water strider algorithm.

Table 7: Comparative results of SNS with other methods for the three-bar truss design problem.

Algorithm	Worst	Mean	Best	SD	NFEs
GA [62]	264.82080546	263.96803663	263.89588573	$1.66862E-01$	50,000
PSO [62]	264.58490296	263.95741428	263.89584341	$1.36897E-01$	50,000
ICA [62]	263.91413326	263.89932689	263.89584519	$4.11693E-03$	50,000
CS [18]	NA	264.0669	263.97156	$9.00000E-05$	15,000
WCA [5]	263.896201	263.895903	263.895843	$8.71000E-05$	5250
GWO [62]	263.90421778	263.89795501	263.89600631	$1.61422E-03$	50,000
WSA [62]	263.89743217	263.89606687	263.8958434	$3.11960E-04$	50,000
CGO [37]	263.8960068	263.8958511	263.895843	$2.51E-05$	100,000
AOS [39]	263.895845	263.895843	263.895843	$8.26E-09$	100,000
SNS (present study)	263.8958561	263.8958462	263.8958434	$3.31056E-06$	4800

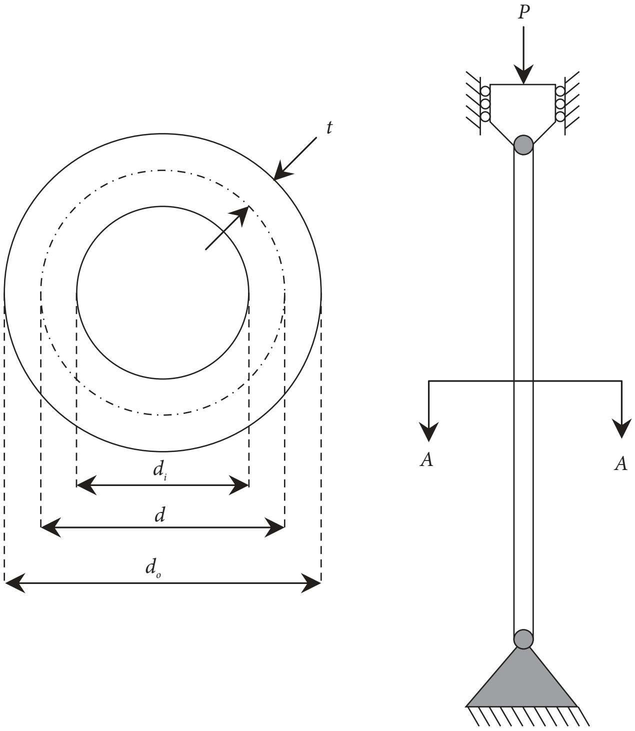


Figure 8: The 3D model of tubular column.

Table 8: Best results of the tubular column example.

Algorithm	Variables		Constraints						
	x_1	x_2	$g_1(X)$	$g_2(X)$	$g_3(X)$	$g_4(X)$	$g_5(X)$	$g_6(X)$	$f(X)$
CS [18]	5.45139	0.29196	−0.0241	−0.1095	−0.633	−0.610	−0.315	−0.635	26.53217
ISA [63]	5.45115623	0.29196547	$-2.5E-10$	$-1.8E-10$	−0.6331	−0.6106	−0.3149	−0.635	26.5313
SNS	5.45115623	0.29196547	$-2.6E-10$	$-1.8E-10$	−0.6331	−0.6106	−0.3149	−0.635	26.4994969

Table 9: Comparative results of SNS with other methods for the tubular column example.

Algorithm	Worst	Mean	Best	SD	NFEs
ISA [63]	26.532	26.531	26.531	$1.70E-04$	3000
CS [18]	26.53972	26.53504	26.53217	$1.93E-03$	15,000
FA [64]	NA	28.74	26.52	2.08	3000
AOS [39]	26.60831361	26.53161399	26.53137828	$1.0300E-03$	100,000
SNS (present study)	26.48637095	26.48636249	26.48636147	$2.2160E-06$	1250

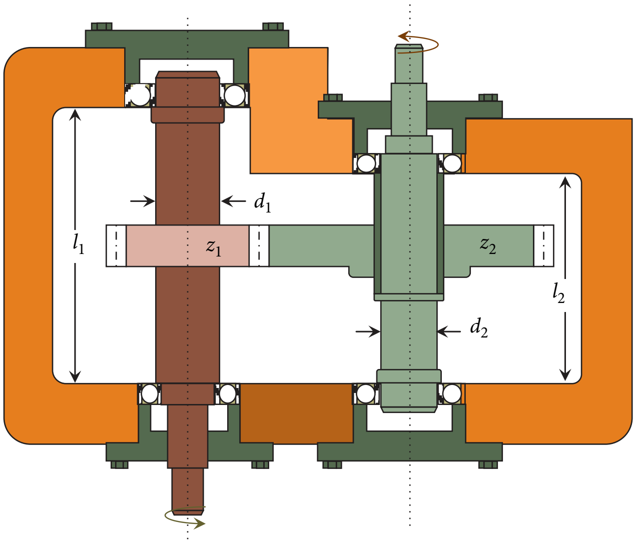


FIGURE 9: A schematic representation of speed reducer.

minimize:

$$f(X) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right),$$

subject to:

$$g_1(X) = \frac{27}{x_1x_2^2x_3} - 1 \le 0,$$

$$g_2(X) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0,$$

$$g_3(X) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \le 0,$$

$$g_4(X) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \le 0,$$

$$g_5(X) = \frac{\sqrt{\left(745x_4/x_2x_3\right)^2 + 16.9 \times 10^6}}{110x_6^3} - 1 \le 0,$$

$$g_6(X) = \frac{\sqrt{\left(745x_5/x_2x_3\right)^2 + 157.5 \times 10^6}}{85x_7^3} - 1 \le 0,$$

$$g_7(X) = \frac{x_2x_3}{40} - 1 \le 0,$$

$$g_8(X) = \frac{5x_2}{x_1} - 1 \le 0,$$

$$g_9(X) = \frac{x_1}{12x_2} - 1 \le 0,$$

$$
\begin{aligned}
& g_{10}(X) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0 \\
& g_{11}(X) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0, \\
& \text{variable range:} \\
& 2.6 \leq x_1 \leq 3.6, \\
& 0.7 \leq x_2 \leq 0.8, \\
& x_3 \in \{17,\ 18,\ 19,\ \ldots,\ 28\}, \\
& 7.3 \leq x_4, \\
& x_5 \leq 8.3, \\
& 2.9 \leq x_6 \leq 3.9, \\
& 5 \leq x_7 \leq 5.5.
\end{aligned}
\tag{12}
$$

This engineering problem has 11 constraints, seven nonlinear constraints and four linear inequality constraints, which are considered based on (1) bending stress of the gear teeth, (2) surface stress, (3) transverse deflections of the shafts, and (4) stresses in the shafts. The comparison of the best optimal solution with various optimization methods is given in Table 10. The SNS method requires 3750 evaluations to find its solution. The statistical results of SNS and ten optimization methods are compared in Table 11. Among the compared optimization algorithms, the SNS has the lowest number of function evaluations while its results are better than those of the other methods.

3.6. Piston Lever. The main objective of this problem is to locate the piston components, $H(= x_1)$, $B(= x_2)$, $D(= x_3)$, and $X(= x_4)$, by minimizing the oil volume when the lever of the piston is lifted up from ${}^{0^{\circ}}$ to 45°, as shown in Figure 10. The formulation of this problem is given as follows:

$$
\begin{aligned}
& \text{minimize:} \\
& f(X) = \frac{1}{4}\pi x_3^2 (L_2 - L_1), \\
& \text{subject to:} \\
& g_1(X) = QL\cos\theta - R \times F \leq 0, \\
& g_2(X) = Q(L - x_4) - M_{\max} \leq 0, \\
& g_3(X) = 1.2(L_2 - L_1) - L_1 \leq 0, \\
& g_4(X) = \frac{x_3}{2} - x_2 \leq 0, \\
& \text{where} \\
& R = \frac{\left|-x_4(x_4\sin\theta + x_1) + x_1(x_2 - x_4\cos\theta)\right|}{\sqrt{(x_4 - x_2)^2 + x_1^2}}, \\
& F = \frac{\pi P x_3^2}{4}, \\
& L_1 = \sqrt{(x_4 - x_2)^2 + x_1^2}, \\
& L_2 = \sqrt{(x_4\sin\theta + x_1)^2 + (x_2 - x_4\cos\theta)^2}, \\
& \theta = {}^{45^{\circ}}, \\
& Q = 10,000 \text{ lbs}, \\
& L = 240 \text{ in}, \\
& M_{\max} = 1.8 \times 10^6 \text{ lbs in}, \\
& P = 1500 \text{ psi}, \\
& \text{variable range:} \\
& 0.05 \leq x_1,\ x_2,\ x_4 \leq 500, \\
& 0.05 \leq x_3 \leq 120.
\end{aligned}
\tag{13}
$$

These inequality constraints consider the force equilibrium, the maximum bending moment of the lever, minimum piston stroke, and geometrical conditions. The best solutions obtained by SNS and some of the other algorithms are presented in Table 12. In addition, the performance of the PSO [71], DE [71], GA [71], hybrid particle swarm optimization (HPSO) [71], HPSO with Q-learning [71], CS [18], ISA [63], CGO [37], MGA [40], AOS [39], and SNS is summarized in Table 13. The SNS algorithm obtains its results after 5000 evaluations, and its results are far better than those of other methods.

Table 10: Best results of the speed reducer design example.

	CS [18]	WCA [5]	BA [66]	ABC [67]	APSO [68]	SHO [69]	SNS
x_1	3.50150	3.50000	3.50000	3.50000	3.50131	3.50159	3.50000
x_2	0.70000	0.70000	0.70000	0.70000	0.70000	0.70000	0.70000
x_3	17.00000	17.00000	17.00000	17.00000	18.00000	17.00000	17.00000
x_4	7.60500	7.30000	7.30001	7.30000	8.12781	7.30000	7.30000
x_5	7.81810	7.71532	7.71532	7.80000	8.04212	7.80000	7.71532
x_6	3.35200	3.35021	3.35021	3.35022	3.35245	3.35127	3.35021
x_7	5.28750	5.28665	5.28665	5.28668	5.28708	5.28874	5.28665
$g_1(X)$	−0.07430	−0.07392	−0.07400	−0.07392	−0.12569	−0.07434	−0.07392
$g_2(X)$	−0.19830	−0.19800	−0.19800	−0.19800	−0.28490	−0.19836	−0.19800
$g_3(X)$	−0.43490	−0.49917	−0.49900	−0.49917	−0.34888	−0.49980	−0.49917
$g_4(X)$	−0.90080	−0.90464	−0.90500	−0.90156	−0.89804	−0.90162	−0.90464
$g_5(X)$	−0.00110	−0.66685	0.00000	0.00000	−0.66559	−0.66717	0.00000
$g_6(X)$	−0.00040	0.00000	0.00000	−0.00063	−0.00026	−0.00117	0.00000
$g_7(X)$	−0.70250	−0.70250	−0.70300	−0.70250	−0.68500	−0.70250	−0.70250
$g_8(X)$	−0.00040	0.00000	0.00000	0.00000	−0.00038	−0.00045	0.00000
$g_9(X)$	−0.58320	−0.58333	−0.58300	−0.58333	−0.58318	−0.58314	−0.58333
$g_{10}(X)$	−0.08900	−0.05133	−0.05100	−0.05133	−0.14754	−0.05111	−0.05133
$g_{11}(X)$	−0.01300	0.00000	0.00000	−0.01070	−0.04058	−0.01056	0.00000
$f(X)$	3000.98100	2994.47107	2994.46710	2997.05841	3187.63049	2998.55070	2994.47107

APSO: accelerated particle swarm optimization. SHO: spotted hyena optimizer.

Table 11: Comparative results of SNS with other methods for the speed reducer design example.

Algorithm	Worst	Mean	Best	SD	NFEs
CS [18]	3009.00	3007.1997	3000.981	$4.96E+00$	250,000
ABC [67]	NA	2997.058412	2997.058412	$0.00E+00$	30,000
WCA [5]	2994.505578	2994.474392	2994.471066	$7.40E-03$	15,150
APSO [68]	4443.01763900	3822.64062400	3187.63048600	$3.66E+02$	30,000
SHO [69]	3003.889	2999.64	2998.5507	$1.93E+00$	NA
SSA	3015.662612	3005.574377	2996.021720	$4.63E+00$	NA
WOA	3233.598124	3042.915023	2996.604340	$4.08E+01$	NA
CSS	3106.216451	3005.658912	2996.492478	$4.86E+00$	NA
CGO [37]	2995.504933	2994.465397	2994.443649	0.110282	100,000
FACSS [70]	3006.419746	2999.413798	2996.3752376	$4.82E+00$	NA
SNS (present study)	2994.4710992	2994.4710696	2994.4710662	$7.00E-06$	3750

FACSS: fuzzy adaptive charged system search.

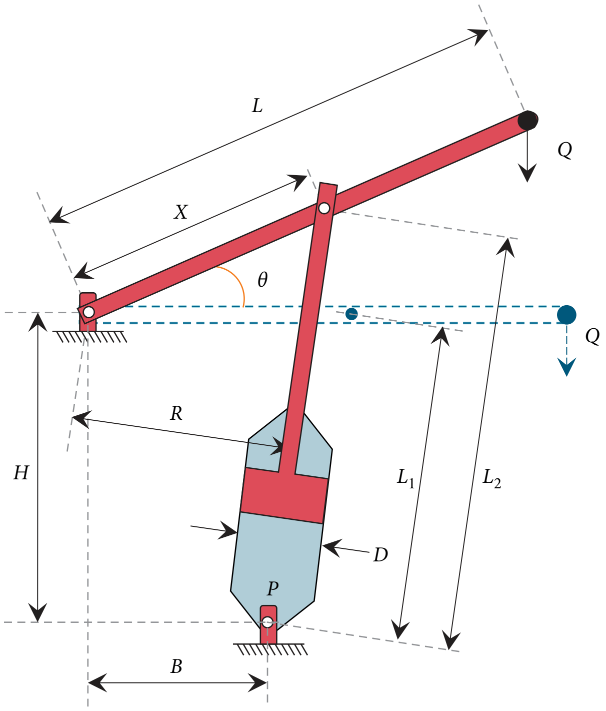


Figure 10: A model of piston lever problem.

Table 12: Best results of the piston lever example.

	PSO [71]	DE [71]	GA [71]	HPSO [71]	CS [18]	SNS
x_1	133.3	129.4	250.0	135.5	0.050	0.050
x_2	2.44	2.43	3.96	2.48	2.043	2.042
x_3	117.14	119.80	60.03	116.62	120.000	120.000
x_4	4.75	4.75	5.91	4.75	4.085	4.083
$g_1(X)$	NA	NA	NA	NA	−1744.912	0.000
$g_2(X)$	NA	NA	NA	NA	−600,000	−600,000
$g_3(X)$	NA	NA	NA	NA	−117.185	−117.187
$g_4(X)$	NA	NA	NA	NA	$-4.50E-04$	$-9.74E-12$
$f(X)$	122	159	161	162	8.427	8.412698349

Table 13: Comparative results of SNS with other methods for the piston lever example.

Algorithm	Worst	Mean	Best	SD	NFEs
PSO [71]	294	166	122	51.7	50,000
DE [71]	199	187	159	14.2	50,000
GA [71]	216	185	161	18.2	50,000
HPSO [71]	197	187	162	13.4	50,000
HPSO with Q-learning [71]	168	151	129	13.4	50,000
CS [18]	168.5920	40.2319	8.4271	59.0552	50,000
ISA [63]	610.6	226.5	8.4	111.2	12,500
CGO [37]	167.4728087	45.0486599	8.412813813	67.24763	100,000
AOS [39]	167.6649862	33.7412759	8.419142742	93.46674724	100,000
MGA [40]	167.4732134	32.4688925	8.413406652	29.96370439	100,000
SNS (present study)	167.4727747	24.3189743	8.412698349	47.71792646	5000

3.7. *Corrugated Bulkhead Design.*

This problem aims to minimize the weight of a corrugated bulkhead in a chemical tanker [72], in which the design variables are the width (x_1), depth (x_2), length (x_3), and plate thickness (x_4). The mathematical model of this optimization problem is given as follows:

minimize:

$$f(X) = \frac{5.885x_4(x_1+x_3)}{x_1+\sqrt{|x_3^2-x_2^2|}},$$

subject to:

$$g_1(X) = -x_4x_2\left(0.4x_1+\frac{x_3}{6}\right)+8.94\left(x_1+\sqrt{|x_3^2-x_2^2|}\right)\le 0,$$

$$g_2(X) = -x_4x_2^2\left(0.2x_1+\frac{x_3}{12}\right)+2.2\left(8.94\left(x_1+\sqrt{|x_3^2-x_2^2|}\right)\right)^{4/3}\le 0,$$

$$g_3(X) = -x_4+0.0156x_1+0.15\le 0,$$

$$g_4(X) = -x_4+0.0156x_3+0.15\le 0,$$

$$g_5(X) = -x_4+1.05\le 0,$$

$$g_6(X) = -x_3+x_2\le 0,$$

variable range:

$$0\le x_1,\ x_2,\ x_3\le 100, \qquad (14)$$

$$0\le x_4\le 5.$$

Tables 14 and 15 compare the best and statistical results of SNS and other optimizers, respectively. According to these results, the SNS significantly improves the solution quality of other algorithms. In addition, the SNS method solves this problem after 3125 evaluations that is the lowest value among other methods.

3.8. *Design of Pressure Vessel.*

A cylindrical vessel is capped at both ends by hemispherical heads, as shown in Figure 11. The objective is to minimize the total cost, including the cost of material, forming, and welding. This problem has four variables including the thickness of the shell $T_s(=x_1)$, the thickness of the head $T_h(=x_2)$, the inner radius $R=(x_3)$, and the length of the cylindrical section of the vessel, not including the head $L(=x_4)$. In addition, x_1 and x_2 are integer multiples of 0.0625 in, while the other variables are continuous. The optimization problem can be expressed as follows:

Table 14: Best results of the corrugated bulkhead design.

	CS [18]	VIGMM3 [61]	AEFA-C [61]	SNS
x_1	37.1179498	57.69231	57.69277	57.69230732
x_2	33.0350210	34.14762	34.13296	34.14762029
x_3	37.1939476	57.69231	57.55294	57.69230729
x_4	0.7306255	1.05000	1.05007	1.05
$g_1(X)$	−23.3537699	−0.25839	−240.89634	−240.6946226
$g_2(X)$	−15.9738532	$-2.220E-16$	−11.59051	$-1.46827E-05$
$g_3(X)$	−0.00158548	$-9.769E-15$	$-6.864E-05$	$-5.80038E-09$
$g_4(X)$	−0.00039992	$-5.551E-16$	$-2.250E-03$	$-6.20877E-09$
$g_5(X)$	0.3193745	0.00000	$-7.581E-05$	$-6.50591E-13$
$g_6(X)$	−4.1589266	0.68949	−23.41997	−23.544687
$f(X)$	5.894331	6.84296	−6.84584	6.842958019

VIGMM3: vibration-based ideal gas molecular movement. AEFA-C: artificial electric field algorithm.

Table 15: Comparative results of SNS with other methods for the corrugated bulkhead design.

Algorithm	Worst	Mean	Best	SD	NFEs
FA [64]	NA	10.23	7.21	1.95	12,000
LF-FA [64]	NA	8.83	6.95	1.26	12,000
LS-LF-FA [64]	NA	7.44	6.86	0.67	12,000
AD-IFA [64]	NA	7.21	6.84	0.58	12,000
AOS [39]	7.066936186	7.060808377	6.84295801	$6.4911E-04$	100,000
SNS (present study)	6.843074399	6.842979802	6.842960515	$2.0942E-05$	3125

LF-FA: Levy flight firefly algorithm. LS-LF-FA: logarithmic spiral path flight firefly. AD-IFA: spiral-Levy flight firefly algorithm.

minimize:

$$f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3,$$

subject to:

$$g_1(X) = -x_1 + 0.0193x_3 \leq 0,$$

$$g_2(X) = -x_2 + 0.00954x_3 \leq 0,$$

$$g_3(X) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0,$$

$$g_4(X) = x_4 - 240 \leq 0,$$

variable range:

$$x_1,\ x_2 \in \{1 \times 0.0625,\ 2 \times 0.0625,\ 3 \times 0.0625,\ \ldots,\ 1600 \times 0.0625\},$$

$$10 \leq x_3,$$

$$x_4 \leq 200. \tag{15}$$

This problem has been used to evaluate the performance of many algorithms. Tables 16 and 17 compare the best and statistical results of SNS and other algorithms, respectively. The SNS needs 6000 NFEs for solving this problem that is much lower than that of other algorithms.

3.9. Design of Tension/Compression Spring. The tension/compression spring design problem is described in [81] for which the objective is to minimize the weight of a tension/compression spring, as shown in Figure 12. This problem is subject to constraints on minimum deflection, shear stress, surge frequency, limits on the outside diameter, and design variables. The design variables are the mean coil diameter $D\,(= x_1)$, the wire diameter $d\,(= x_2)$, and the number of active coils $N\,(= x_3)$. The problem can be stated as

minimize:

$$f(X) = (x_3 + 2)x_2x_1^2,$$

subject to:

$$g_1(X) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \leq 0,$$

$$g_2(X) = \frac{4x_2^2 - x_1x_2}{12566\left(x_2x_1^3 - x_1^4\right)} + \frac{1}{5108x_1^2} - 1 \leq 0,$$

$$g_3(X) = 1 - \frac{140.45x_1}{x_2^2 x_3} \leq 0, \tag{16}$$

$$g_4(X) = \frac{x_1 + x_2}{1.5} - \leq 0,$$

variable range:

$$0.05 \leq x_1 \leq 2,$$

$$0.25 \leq x_2 \leq 1.3,$$

$$2 \leq x_3 \leq 15.$$

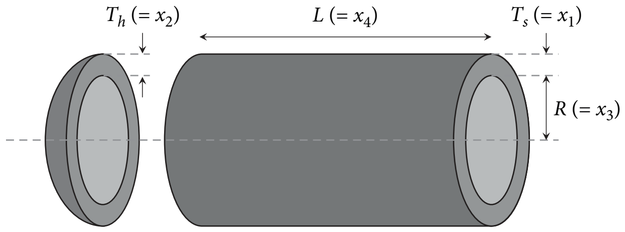


Figure 11: Schematic view of pressure vessel design.

Table 16: Best results of the pressure vessel design.

	G-QPSO [73]	HPSO [74]	CPSO [75]	CDE [76]	SAP [77]	ABC [67]	CS [18]	EO [78]
x_1	0.81250	0.81250	0.81250	0.81250	0.81250	0.81250	0.81250	0.81250
x_2	0.43750	0.43750	0.43750	0.43750	0.43750	0.43750	0.43750	0.43750
x_3	42.09840	42.09840	42.09130	42.09840	40.32390	42.09845	42.09845	42.09845
x_4	176.63720	176.63660	176.74650	176.63760	200.00000	176.63660	176.63660	176.63660
$g_1(X)$	0.00000	0.00000	0.00000	0.00000	−0.03432	0.00000	0.00000	0.00000
$g_2(X)$	−0.03580	−0.03580	−0.00036	−0.03580	−0.05285	−0.03588	−0.03588	−0.03588
$g_3(X)$	−0.21790	3.12260	−118.76870	−3.70512	−27.10585	−0.00023	−0.00005	−0.00005
$g_4(X)$	−63.36280	−63.36340	−63.25350	−63.36230	−40.00000	−63.36340	−63.36340	−63.36340
$f(X)$	6059.7208	6059.7143	6061.0777	6059.7340	6288.7445	6059.7143	6059.7143	6059.71430
	MFO [56]	GWO [31]	WOA [33]	APSO [68]	IAPSO [68]	NDE [79]	MCEO [80]	SNS
x_1	0.8125	0.812500	0.812500	0.8125	0.8125	0.8125	0.8125	0.81250
x_2	0.4375	0.434500	0.437500	0.4375	0.4375	0.4375	0.4375	0.43750
x_3	42.098445	42.089181	42.0982699	42.0984	42.0984	42.0984455	42.0984455	42.09845
x_4	176.636596	176.758731	176.638998	176.6374	176.6366	176.636595	176.636596	176.63660
$g_1(X)$	$-1.15E-08$	−0.00017880	$-3.39E-06$	$-8.80E-07$	$-4.09E-13$	$-1.4E-15$	$-1.13E-\ 10$	0.00000
$g_2(X)$	−0.03588083	−0.03296921	−0.03588250	−0.03588126	$-3.58E-2$	−0.00035880	−0.037564	−0.03588
$g_3(X)$	0.04023295	−40.6168247	−1.25270175	−1.33153860	$-1.39E-07$	−0.00000001	$-4.73E-\ 04$	0.00000
$g_4(X)$	−63.363404	−63.241269	−63.361002	−63.3626	−63.3634	−0.63363404	−63.3634	−63.36340
$f(X)$	6059.7143	6051.5639	6059.7410	6059.72418	6059.71433	6059.71433	6059.7143	6059.71434

G-QPSO: Gaussian quantum-behaved particle swarm optimization. CPSO: co-evolutionary particle swarm optimization. CDE: co-evolutionary differential evolution. SAP: self-adaptive penalty approach. EO: equilibrium optimizer. MCEO: multilevel cross entropy optimizer.

Table 17: Comparative results of SNS with other methods for the pressure vessel design.

Algorithm	Worst	Mean	Best	SD	NFEs
SAP [77]	6308.15	6293.843	6288.745	$7.41E+00$	3000
HPSO [74]	6288.677	6099.9323	6059.7143	$8.62E+01$	81,000
CDE [76]	6059.734	6085.23	6371.046	$4.30E+01$	204,800
CPSO [75]	6363.8041	6147.1332	6061.0777	$8.65E+01$	200,000
PSO [73]	14076.324	8756.6803	6693.7212	$1.49E+03$	8000
QPSO [73]	8017.2816	6839.9326	6059.7209	$4.79E+02$	8000
G-QPSO [73]	7544.4925	6440.3786	6059.7208	$4.48E+02$	8000
ABC [67]	NA	6245.308144	6059.714736	$2.05E+02$	30,000
CS [18]	6495.347	6447.736	6059.714335	$5.03E+02$	15,000
WOA [33]	NA	6068.05	6059.741	$6.57E+01$	6300
APSO [68]	7544.49272	6470.71568	6059.7242	$3.27E+02$	200,000
EO [78]	7544.4925	6668.114	6059.7143	$5.66E+02$	15,000
CGO [37]	6330.958685	6250.957354	6247.672819	$1.07E+01$	100,000
SNS (present study)	6410.086886	6097.100294	6059.714335	$9.28E+01$	6000

Table 18 compares the SNS with many optimization algorithms in terms of best optimization results, and Table 19 presents the statistical results of these algorithms. The SNS algorithm solves this problem in 9000 evaluations, and among the compared methods, just WOA [33] and MCEO [80] used a fewer number of evaluations, while their results are not as good as SNS.

3.10. Design of Welded Beam. This benchmark problem was introduced by Coello [77] and has been tackled by many researchers. As illustrated in Figure 13, the beam is under a vertical force. The goal of this problem is to find the minimum manufacturing cost of the welded beam. The problem is subject to seven constraints of stress, deflection, welding, and geometry. The variables are weld thickness

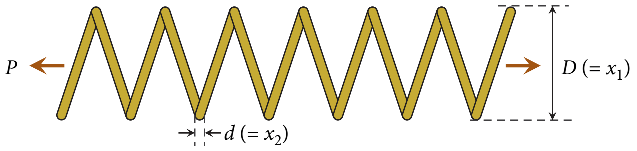


Figure 12: Tension/compression spring.

Table 18: Best results of tension/compression spring design.

Algorithm	Variables			Constraints				
	x_1	x_2	x_3	$g_1(X)$	$g_2(X)$	$g_3(X)$	$g_4(X)$	$f(X)$
CPSO [75]	0.051728	0.357644	11.244543	$-8.25E-04$	$-2.52E-05$	−4.051306	−0.727085	0.012674
WCA [5]	0.051680	0.356500	11.300400	$-1.60E-13$	$-7.90E-14$	−4.053300	−0.727800	0.012665
ABC [67]	0.051749	0.358179	11.203763	$0.00E+00$	$0.00E+00$	−4.056663	−0.726713	0.012665
SFOA [65]	0.051800	0.359000	11.279000	$-3.24E-06$	$-3.58E-07$	−4.060000	−0.726000	0.012700
APSO [68]	0.052588	0.378343	10.138862	$-1.55E-04$	$-8.33E-04$	−4.089171	−1.069069	0.012700
IAPSO [68]	0.051685	0.356629	11.294175	$-1.97E-10$	$-4.64E-10$	−4.053610	−1.091686	0.012665
MFO [56]	0.051994	0.364109	10.868421	$-4.10E-06$	$3.04E-06$	−4.068140	−0.722600	0.012667
GWO [31]	0.051690	0.356737	11.288850	$-7.91E-05$	$-7.51E-06$	−4.053380	−0.727720	0.012666
WOA [33]	0.051200	0.345200	12.004000	$-5.60E-04$	$-3.00E-05$	−4.027400	−0.735700	0.012676
SHO [69]	0.051144	0.343751	12.095500	$-3.30E-04$	$1.16E-05$	−4.025790	−0.736740	0.012674
NDE [79]	0.051689	0.356718	11.288968	$0.00E+00$	$0.00E+00$	−4.053785	−0.727728	0.012665
SSA [59]	0.051207	0.345215	12.004032	$-5.60E-04$	$-3.70E-05$	−4.027410	−0.735720	0.012676
SNS	0.051587	0.354268	11.434058	$-1.37E-08$	$-3.18E-10$	−4.048919	−0.729430	0.012665

Table 19: Comparative results of tension/compression spring design.

Algorithm	Worst	Mean	Best	SD	NFEs
CPSO [75]	0.012924	0.01273	0.0126747	$5.20E-05$	200,000
HPSO [74]	0.0127191	0.0127072	0.0126652	$1.58E-05$	81,000
CDE [76]	0.01279	0.012703	0.0126702	$2.70E-05$	204,800
PSO [73]	0.071802	0.019555	0.012857	$1.17E-02$	20,000
QPSO [73]	0.018127	0.013854	0.012669	$1.34E-03$	20,000
G-QPSO [73]	0.015869	0.012996	0.012666	$6.28E-04$	20,000
WCA [5]	0.012952	0.012746	0.012665	$8.06E-05$	11,750
ABC [67]	NA	0.012709	0.012665	$1.28E-02$	30,000
APSO [68]	0.014937	0.013297	0.0127	$6.85E-04$	120,000
IAPSO [68]	0.01782864	0.01367653	0.01266523	$1.57E-03$	20,000
WOA [33]	NA	0.0127	0.0126763	$3.00E-04$	4410
MCEO [80]	0.01350901	0.0127196	0.01266051	$3.79E-05$	2000
EO [78]	0.013997	0.013017	0.012666	$3.91E-04$	15,000
SNS (present study)	0.012765873	0.012684717	0.012665246	$2.38549E-05$	9000

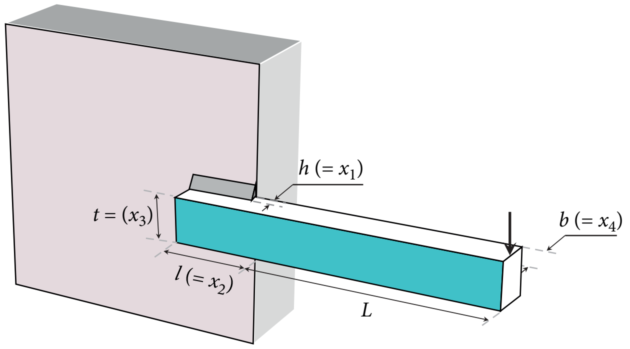


Figure 13: Schematic of the welded beam structure with indication of design variables.

$h(=x_1)$, height $l(=x_2)$, length $t(=x_3)$, and bar thickness $b(=x_4)$, as shown in Figure 13. The objective function can be mathematically be stated as

$$
\begin{aligned}
&\text{minimize:}\\
&f(X) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2),\\
&\text{subject to:}\\
&g_1(X) = \tau(X) - \tau_{\max} \le 0,\\
&g_2(X) = \sigma(X) - \sigma_{\max} \le 0,\\
&g_3(X) = \delta(X) - \delta_{\max} \le 0,\\
&g_4(X) = x_1 - x_4 \le 0,\\
&g_5(X) = P - P_c(X) \le 0,\\
&g_6(X) = 0.125 - x_1 \le 0,\\
&g_7(X) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0,\\
&\tau(X) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},\\
&\tau' = \frac{P}{\sqrt{2}x_1x_2},\\
&\tau'' = \frac{MR}{J},\\
&M = P\left(L + \frac{x_2}{2}\right),\\
&R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},\\
&J = 2\left\{\sqrt{2}\,x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\},\\
&\sigma(\overrightarrow{X}) = \frac{6PL}{x_4x_3^2},\\
&\delta(\overrightarrow{X}) = \frac{6PL^3}{Ex_3^2x_4},\\
&P_c(\overrightarrow{X}) = \frac{4.013E\sqrt{x_3^2x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),\\
&P = 6000\,\text{lb},\\
&L = 14\,\text{in},\\
&\delta_{\max} = 0.25\,\text{in},\\
&E = 30 \times 10^6\,\text{psi},\\
&G = 12 \times 10^6\,\text{psi},\\
&\tau_{\max} = 13,600\,\text{psi},\\
&\sigma_{\max} = 30,000\,\text{psi}.\\
&\text{variable range:}\\
&0.1 \le x_1,\\
&x_4 \le 2,\\
&0.1 \le x_2,\\
&x_3 \le 10.
\end{aligned}
\tag{17}
$$

Tables 20 and 21 compare the best and statistical results of various optimizer in dealing with welded beam design problem. The SNS algorithm needs 9000 evaluations, which is the lowest NFE among other algorithms, while its results are better. In addition, the SNS algorithm has the lowest standard deviation that shows its robustness in solving this problem.

3.11. Design of Gear Train. The gear train design problem is an unconstrained discrete design problem in mechanical engineering and was introduced by Sandgren [85]. The purpose of this benchmark task is to minimize the gear ratio defined as the ratio of the angular velocity of the output shaft to the angular velocity of the input shaft. The number of teeth of gears $n_A(=x_1)$, $n_B(=x_2)$, $n_C(=x_3)$, and $n_D(=x_4)$ are considered as the design variables, and Figure 14 illustrates the 3D model of this problem. The mathematical formulation is provided as follows:

$$
\begin{aligned}
&\text{minimize:}\\
&f(X) = \left(\frac{1}{6.931} - \frac{x_3x_2}{x_1x_4}\right)^2,\\
&\text{variable range:}\\
&x_1, x_2, x_3, x_4 \in \{12, 13, 14, \ldots, 60\}.
\end{aligned}
\tag{18}
$$

The best results of 19 algorithms include the SNS are presented in Table 22. It can be seen that all the algorithms find the optimum solution, except the PSO [62] and BBO [62]. In addition, the statistical results of 14 algorithms are compared in Table 23. The proposed method outperformed most of the other algorithms in terms of the mean, worst, SD, and NFEs.

3.12. A Reinforced Concrete Beam Design. Amir and Hasegawa [91] presented a simplified optimization problem of designing a reinforced concrete beam, as shown in Figure 15. The beam is assumed to be simply supported with a span of 30 ft and subjected to a live load of 2000 lbf and a dead load

Table 20: Best results of welded beam design.

	BBO [62]	PSO [62]	ICA [62]	WCA [5]	ABC [67]	EO [78]	TEO [82]	SSA [59]
x_1	0.1854860	0.219292	0.205799	0.205728	0.205730	0.2057000	0.20568100	0.2057000
x_2	4.3129000	3.430416	3.469634	3.470522	3.470489	3.4705000	3.47230500	3.4714000
x_3	8.4399030	8.433559	9.034950	9.036620	9.036624	9.0366000	9.03513300	9.0366000
x_4	0.2359020	0.236204	0.205806	0.205729	0.205730	0.2057000	0.20579600	0.2057000
$g_1(X)$	−114.190503	−0.193642	0.020688	0.010832	0.000000	0.0094704	−0.60927688	−0.8256803
$g_2(X)$	−6.7171616	0.017873	−0.017763	0.119259	−0.000002	0.0014958	0.22468860	−0.7106346
$g_3(X)$	−0.0504000	−0.016900	−0.000007	−0.000001	0.000000	−0.0000001	−0.00011500	0.0000000
$g_4(X)$	−3.2078633	−3.276396	−3.390414	−3.390662	−3.432984	−3.3906592	−3.39027259	−3.3908203
$g_5(X)$	−0.0604860	−0.094292	−0.080799	−0.080728	−0.080730	−0.0807295	−0.08068100	−0.0807000
$g_6(X)$	−0.2350000	−0.235000	−0.236000	−0.235540	−0.235540	−0.2355403	−0.23553783	−0.2355381
$g_7(X)$	−2639.74280	−2668.48570	−5.951908	0.057684	0.000000	0.0008607	−5.15644919	−2.6033472
$f(X)$	1.9180550	1.852720	1.725135	1.724856	1.724852	1.7249000	1.72528400	1.7249100
	WSA [62]	MCEO [80]	CBO [62]	GWO [62]	SHO [69]	WOA [33]	CCSA [83]	SNS
x_1	0.20573000	0.20572964	0.2057220	0.2056770	0.2055630	0.2053960	0.2057000	0.2057296
x_2	3.47048900	3.47048866	3.4704100	3.4708940	3.4748460	3.4842930	3.4702000	3.4704887
x_3	9.03662400	9.03662391	9.0372760	9.0385580	9.0357990	9.0374260	9.0362000	9.0366239
x_4	0.20573000	0.20572964	0.2057350	0.2057390	0.2058110	0.2062760	0.2057000	0.2057296
$g_1(X)$	−0.02539959	0.00000363	−0.0246881	−0.0519206	−1.4028657	−21.5450190	−0.0020610	0.0000000
$g_2(X)$	−0.05312238	−0.00002880	−5.1106833	−14.2018357	−6.3837568	−84.7713395	−0.6569560	0.0000000
$g_3(X)$	0.00000000	0.00000000	−0.0000130	−0.0000620	−0.0002480	−0.0008800	−0.0000045	0.0000000
$g_4(X)$	−3.39065616	−3.39000000	−3.3905161	−3.3902412	−3.3898697	−3.3852837	−3.3906250	−3.3906591
$g_5(X)$	−0.08073000	−0.08072964	−0.0807220	−0.0806770	−0.0805630	−0.0803960	−0.0807296	−0.0807296
$g_6(X)$	−0.23554035	−0.23554032	−0.2360000	−0.2355503	−0.2355421	−0.2355825	−0.2355406	−0.2355403
$g_7(X)$	−0.03155555	−0.00001860	−0.7536625	−1.6632990	−6.7607624	−48.2829292	−0.3906801	0.0000000
$f(X)$	1.72485200	1.72485230	1.7246630	1.7252320	1.7256610	1.7304990	1.7249000	1.7248523

CCSA: conscious neighborhood-based crow search algorithm.

Table 21: Comparative results of welded beam design.

Algorithm	Worst	Mean	Best	SD	NFEs
PSO [62]	3.841845	2.613785	1.85272	$4.71E-01$	50,000
CPSO [75]	1.782143	1.748831	1.728024	$1.29E-02$	200,000
HPSO [74]	1.814295	1.749040	1.724852	$4.00E-02$	81,000
CDE [76]	1.824105	1.768158	1.733461	$2.22E-02$	204,800
ICA [62]	2.237755	1.79433	1.725135	$1.10E-01$	50,000
BBO [62]	3.606933	2.630412	1.918055	$4.11E-01$	50,000
FA [54]	2.345579	1.878656	1.731206	$2.68E-01$	50,000
CSS [26]	1.759479	1.739654	1.724866	$8.06E-03$	NA
WCA [5]	1.744697	1.726427	1.724856	$4.29E-03$	46,500
ABC [67]	NA	1.741913	1.724852	$3.10E-02$	30,000
MFO [62]	1.724852	1.732109	1.950241	$3.40E-02$	50,000
SCA [62]	1.786863	1.849364	1.925162	$3.47E-02$	50,000
APSO [68]	1.993999	1.877851	1.736193	$7.61E-02$	50,000
SHO [69]	1.726064	1.725828	1.725661	$2.87E-04$	NA
SSA [62]	1.725886	1.823426	2.246638	$1.28E-01$	50,000
GWO [62]	1.725232	1.72631	1.728487	$7.71E-04$	50,000
WOA [33]	NA	1.7320	1.730499	0.0226	9900
TEO [82]	1.931161	1.768040	1.725284	$5.82E-02$	NA
EO [78]	1.736725	1.726482	1.724853	$3.26E-03$	15,000
WSA [62]	1.725068	1.724908	1.724852	$4.15E-05$	50,000
FACSS [70]	1.736565	1.730223	1.724853	$6.363E-7$	100,000
T-CSS [84]	1.735656	1.730212	1.724860	$2.00E-06$	100,000
SNS (present study)	1.725051	1.724880	1.724852	$5.18E-05$	9000

T-CSS: tribe-charged system search.

of 1000 lbf, including the weight of the beam. The concrete compressive strength (σ_c) is 5 ksi, and the yield stress of the reinforcing steel (σ_y) is 50 ksi. The cost of concrete is $0.02/in2/linear ft, and the cost of steel is $1.0/in2/linear ft. To minimize the total cost of the structure, the area of the reinforcement $A_s (= x_1)$, the width of the beam $b (= x_2)$, and

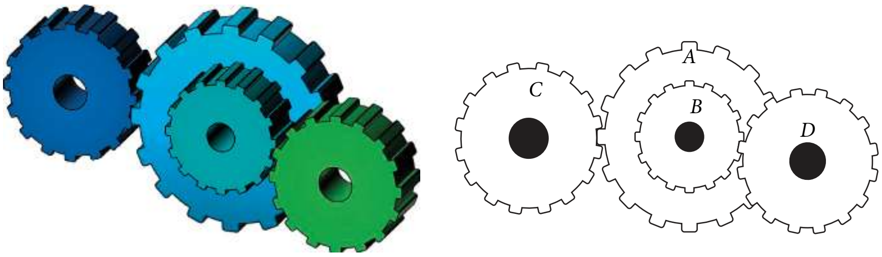


Figure 14: The 3D model of gear train.

Table 22: Best results of the gear train design.

Algorithm	Variables				$f(X)$
	x_1	x_2	x_3	x_4	
GA [62]	49	19	16	43	$2.70E-12$
PSO [62]	34	13	20	53	$2.31E-11$
ICA [62]	43	16	19	49	$2.70E-12$
CS [18]	43	16	19	49	$2.70E-12$
ABC [67]	49	16	19	43	$2.70E-12$
MSFWA [86]	49	19	16	43	$2.70E-12$
MBA [87]	43	16	19	49	$2.70E-12$
BBO [62]	53	26	15	51	$2.31E-11$
NNA [62]	49	16	19	43	$2.70E-12$
GWO [62]	49	19	16	43	$2.70E-12$
ISA [63]	43	19	16	49	$2.70E-12$
APSO [68]	43	16	19	49	$2.70E-12$
IAPSO [68]	43	16	19	49	$2.70E-12$
MVO [88]	43	16	19	49	$2.70E-12$
MFO [56]	43	19	16	49	$2.70E-12$
ALO [57]	49	19	16	43	$2.70E-12$
PSOSCALF [89]	49	19	16	43	$2.70E-12$
WSA [62]	43	16	19	49	$2.70E-12$
SNS (present study)	43	19	16	49	$2.70085714E-12$

MBA: mine blast algorithm.

Table 23: Comparative results of SNS with other methods for the gear train design.

Algorithm	Worst	Mean	Best	SD	NFEs
GA [62]	$1.5247E-08$	$1.6212E-09$	$2.7009E-12$	$3.2174E-09$	50,000
PSO [62]	$1.0222E-06$	$7.9383E-08$	$2.3078E-11$	$1.8147E-07$	50,000
ICA [62]	$2.3576E-09$	$8.0417E-10$	$2.7009E-12$	$7.7862E-10$	50,000
UPSO [90]	N.A	$3.80562E-08$	$2.700857E-12$	$1.09E-07$	100,000
MFO [62]	$2.7265E-08$	$7.5337E-09$	$2.3078E-11$	$9.3539E-09$	50,000
SCA [62]	$2.3576E-09$	$8.8113E-10$	$2.7009E-12$	$6.4529E-10$	50,000
SSA [62]	$2.7265E-08$	$1.9822E-09$	$2.7009E-12$	$4.5748E-09$	50,000
BBO [62]	$4.2018E-07$	$4.5418E-08$	$2.3078E-11$	$7.2953E-08$	50,000
CBO [62]	$1.1173E-08$	$2.1032E-09$	$2.3078E-11$	$2.4025E-09$	50,000
NNA [62]	$2.3576E-09$	$6.1128E-10$	$2.7009E-12$	$6.1167E-10$	50,000
GWO [62]	$9.9216E-10$	$3.3777E-10$	$2.7009E-12$	$4.0956E-10$	50,000
WOA [62]	$6.5123E-09$	$9.6633E-10$	$2.7009E-12$	$1.1296E-09$	50,000
WSA [62]	$1.3616E-09$	$1.6800E-10$	$2.7009E-12$	$3.8265E-10$	50,000
SNS (present study)	$1.36165E-09$	$1.68012E-10$	$2.700857E-12$	$3.74894E-10$	25,000

the depth of the beam $h(=x_3)$ have to be determined. The structure should be proportioned to have a required strength based upon the ACI building code 318-77 as follows:

$$M_u = 0.9A_s\sigma_y(0.8h)\left(1.0 - 0.59\frac{A_s\sigma_y}{0.8bh\sigma_c}\right) \geq 1.4M_d + 1.7M_l, \quad (19)$$

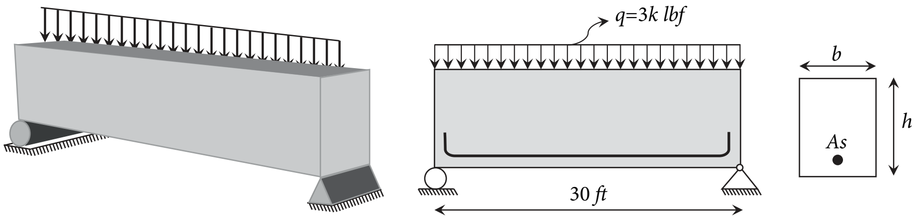


Figure 15: A schematic view of reinforced concrete beam.

where M_u, M_d, and M_l are the flexural strength, dead load, and live load moments of the beam, respectively. In this case, $M_d = 1350$ in kip and $M_l = 2700$ in kip. The depth to width ratio of the beam is restricted to be less than or equal to 4. The optimization problem can be expressed as

$$\begin{aligned}
&\text{minimize:}\\
&f(X) = 2.9x_1 + 0.6x_2x_3,\\
&\text{subject to:}\\
&g_1(X) = \frac{x_2}{x_3} - 4 \le 0,\\
&g_2(X) = 180 + 7.375\frac{x_1^2}{x_3} - x_1x_2 \le 0,\\
&\text{variable range:}\\
&x_1 \in \{6, 6.16, 6.32, 6.6, 7, 7.11, 7.2, 7.8, 7.9, 8, 8.4\},\\
&x_2 \in \{28, 29, 30, \ldots, 40\},\\
&5 \le x_3 \le 10.
\end{aligned} \tag{20}$$

It is clear that the variables x_1 and x_2 are discrete, while x_3 is continuous. The SNS method requires1000 evaluations to reach the optimum solution. Table 24 presents the results of optimum designs obtained by the SNS and other methods for this problem. In addition, the statistical results of FA [54], CS [18], AOS [39], and SNS are compared in Table 25. Obviously, the performance of the SNS method is better than other algorithms.

3.13. Car Side Impact Design. On the foundation of the European Enhanced Vehicle-Safety Committee (EEVC) procedures, a car is exposed to a side impact, and the aim of this benchmark problem is minimizing the weight of the door using nine influence parameters including thicknesses of B-pillar inner ($= x_1$), B-pillar reinforcement ($= x_2$), floor side inner ($= x_3$), cross members ($= x_4$), door beam ($= x_5$), door beltline reinforcement ($= x_6$), roof rail ($= x_7$), materials of B-pillar inner ($= x_8$), floor side inner ($= x_9$), barrier height ($= x_{10}$), and hitting position ($= x_{11}$). Youn et al. [93] simplified this optimization problem's analytical formulation and sped up computations using the global response surface methodology to approximate the structural weight and response to the impact. Figure 16 shows a model for the car side impact problem. According to the simplified models, the optimization problem is formulated as follows:

$$\begin{aligned}
&\text{minimize:}\\
&f(X) = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_7,\\
&\text{subject to:}\\
&g_1(X) = 1.16 - 0.3717x_2x_4 - 0.00931x_2x_{10} - 0.484x_3x_9 + 0.01343x_6x_{10} - 1 \le 0,\\
&g_2(X) = 46.36 - 9.9x_2 - 12.9x_1x_2 + 0.1107x_3x_{10} - 32 \le 0,\\
&g_3(X) = 33.86 + 2.95x_3 + 0.1792x_3 - 5.057x_1x_2 - 11.0x_2x_8 - 0.0215x_5x_{10} - 9.98x_7x_8 + 22.0x_8x_9 - 32 \le 0,\\
&g_4(X) = 28.98 + 3.818x_3 - 4.2x_1x_2 + 0.0207x_5x_{10} + 6.63x_6x_9 - 7.7x_7x_8 + 0.32x_9x_{10} - 32 \le 0,\\
&g_5(X) = 0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.0008757x_5x_{10} + 0.08045x_6x_9\\
&\qquad + 0.00139x_8x_{11} + 0.00001575x_{10}x_{11} - 0.32 \le 0,\\
&g_6(X) = 0.214 + 0.00817x_5 - 0.131x_1x_8 - 0.0704x_1x_9 + 0.03099x_2x_6 - 0.018x_2x_7 + 0.0208x_3x_8 + 0.121x_3x_9 - 0.00364x_5x_6\\
&\qquad + 0.0007715x_5x_{10} - 0.0005354x_6x_{10} + 0.00121x_8x_{11} + 0.00184x_9x_{10} - 0.02x_2^2 - 0.32 \le 0,\\
&g_7(X) = 0.74 - 0.61x_2 - 0.163x_3x_8 + 0.001232x_3x_{10} - 0.166x_7x_9 + 0.227x_2^2 - 0.32 \le 0,
\end{aligned}$$

Table 24: Best results of a reinforced concrete beam design.

	GHN-EP [71]	GA [92]	FA [54]	CS [18]	ISA [63]	AOS [39]	SNS
x_1	6.32	7.20	6.32	6.32	6.32	6.32	6.32
x_2	34	32	34	34	34	34	34
x_3	8.637180	8.0451	8.5000	8.5000	8.5000	8.5	8.5
$g_1(X)$	−0.7745	−2.8779	−0.2241	−0.2241	−0.2241	−0.22409498	−0.22409411
$g_2(X)$	−0.0635	−0.0224	0	0	0	$-1.00E-07$	$5.27667E-12$
$f(X)$	362.00648	366.1459	359.2080	359.2080	359.2080	359.2080	359.2080

GHN-EP: generalized Hopfield network-based extended penalty approach.

Table 25: Comparative results of SNS with other methods for a reinforced concrete beam design.

Algorithm	Worst	Mean	Best	SD	NFEs
FA [54]	669.150	460.706	359.2080	80.73870	25,000
CS [18]	NA	NA	359.2080	NA	5000
AOS [39]	362.2535	359.3306872	359.2080	0.59614901	100,000
SNS (present study)	362.634	359.3222001	359.2080	0.61498581	1000

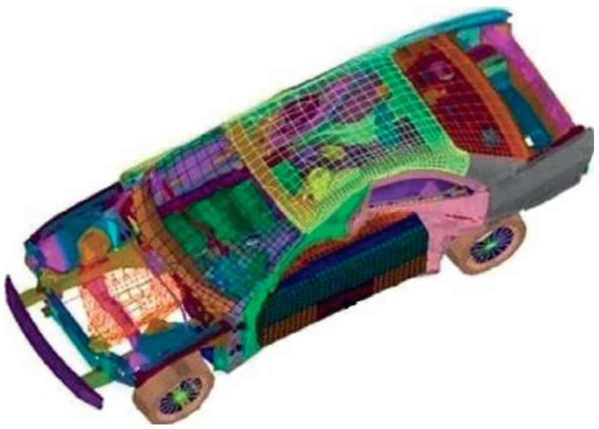


Figure 16: A model of car side impact problem.

$$
\begin{aligned}
&g_8(X) = 4.72 - 0.5x_4 - 0.19x_2x_3 - 0.0122x_4x_{10} + 0.009325x_6x_{10} + 0.000191x_{11}^2 - 4 \le 0,\\
&g_9(X) = 10.58 - 0.674x_1x_2 - 1.95x_2x_8 + 0.02054x_3x_{10} - 0.0198x_4x_{10} + 0.028x_6x_{10} - 9.9 \le 0,\\
&g_{10}(X) = 16.45 - 0.489x_3x_7 - 0.843x_5x_6 + 0.0432x_9x_{10} - 0.0556x_9x_{11} - 0.000786x_{11}^2 - 15.7 \le 0,\\
&\text{variable range:}\\
&0.5 \le x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7 \le 1.5,\\
&x_8, x_9 \in \{0.192,\ 0.345\},,\\
&-30 \le x_{10},\\
&x_{11} \le +30.
\end{aligned}
\tag{21}
$$

The design of car side impact is also used as a benchmark problem to evaluate the performance of various methods. The best results of the SNS and these algorithms are presented in Table 26. It should be noted that the results of other algorithms evaluated by Gandomi et al. [18, 54, 66] have different variables ranges, but in this paper, the variable ranges of [93] are utilized. Table 27 summarizes the statistical results obtained by the different optimization algorithms for the car side impact design problem. In this case, the SNS achieves its results with 20,000 NFEs. The CS

Table 26: Best results of the car side impact design example.

	PSO [54]	DE [54]	GA [54]	CS [18]	BA [66]	SNS
x_1	0.50000	0.50000	0.50005	0.50000	0.50000	0.5
x_2	1.11670	1.11670	1.28017	1.11643	1.11670	1.115933208
x_3	0.50000	0.50000	0.50001	0.50000	0.50000	0.5
x_4	1.30208	1.30208	1.03302	1.30208	1.30208	1.302918991
x_5	0.50000	0.50000	0.50001	0.50000	0.50000	0.5
x_6	1.50000	1.50000	0.50000	1.50000	1.50000	1.5
x_7	0.50000	0.50000	0.50000	0.50000	0.50000	0.5
x_8	0.34500	0.34500	0.34994	0.34500	0.34500	0.345
x_9	0.19200	0.19200	0.19200	0.19200	0.19200	0.192
x_{10}	−19.54935	−19.54935	10.3119	−19.54935	−19.54935	−19.6388662
x_{11}	−0.00431	−0.00431	0.00167	−0.00431	−0.00431	1.49192$E-06$
$g_1(X)$	−0.617505	−0.617505	−0.43167	−0.617423	−0.61750525	−0.61849499
$g_2(X)$	−0.092707	−0.092707	−0.09839	−0.092703	−0.09270723	−0.09273231
$g_3(X)$	−0.100627	−0.100627	−0.13685	−0.100625	−0.10062705	−0.10061401
$g_4(X)$	−0.034209	−0.034209	−0.02699	−0.034182	−0.03420961	−0.03418562
$g_5(X)$	−4.278328	−4.278328	−3.77008	−4.277761	−4.27832783	−4.28314394
$g_6(X)$	−7.2838105	−7.2838105	−3.36103	−7.282103	−7.28381045	−7.29404065
$g_7(X)$	−0.0026365	−0.0026365	−0.00025	3.647$E-05$	−0.00263652	0
$g_8(X)$	−2.425$E-05$	−2.425$E-05$	−8.6156$E-06$	1.395$E-06$	−2.43$E-05$	0
$g_9(X)$	−0.9654269	−0.9654269	−0.58567	−0.965154	−0.96542696	−0.96669762
$g_{10}(X)$	−0.1666041	−0.1666041	0.502506	−0.166604	−0.16660413	−0.16739262
$f(X)$	22.84474	22.84298	22.85653	22.84294	22.84474	22.84297965

Table 27: Comparative results of SNS with other methods for the car side impact design example.

Algorithm	Worst	Mean	Best	SD	NFEs
PSO [54]	23.21354	22.89429	22.84474	0.15017	20,000
DE [54]	24.12606	23.22828	22.84298	0.34451	20,000
GA [54]	26.240578	23.51585	22.85653	0.66555	20,000
BA [66]	23.21354	22.89273	22.84474	0.17383	20,000
FA [54]	24.06623	22.89376	22.84298	0.16667	20,000
CS [18]	23.25998	22.85858	22.84294	0.07612	20,000
WCA [94]	23.37093376	22.97516442	22.84303648	1.9772$E-01$	NA
MBA [94]	23.48894217	22.93642104	22.84359640	1.5258$E-01$	NA
ER-WCA [94]	24.45531280	23.06992534	22.84326461	3.502$E-01$	NA
WOA [94]	27.36081368	24.81448617	23.04216220	9.657$E-01$	NA
FACSS [70]	23.05362562	22.91212354	22.84907401	4.726$E-02$	NA
CSS [84]	24.863563	23.523265	23.007336	0.562345	100,000
T-CSS [84]	23.800904	22.903653	22.847848	0.078565	100,000
SNS (present study)	23.18454939	22.88145736	22.84296954	0.101801211	20,000

ER-WCA: evaporation rate-based water cycle algorithm.

method has better performance than all of the methods, according to the results presented in [18]. In addition, the best result of the SNS is better than those of other algorithms.

3.14. Cantilever Stepped Beam. This problem is a good benchmark to verify the capability of the optimization methods for solving continuous, discrete, and mixed variable structural design problems. This problem aims to minimize the volume of the beam. The width of segments $(x_1, x_2, x_3, x_4, x_5)$ and height of them $(x_6, x_7, x_8, x_9, x_{10})$ are chosen to be the design variables. These ten variables are illustrated in Figure 17. Except for bending stress constraints, a specified aspect ratio is imposed such that the ratio of height to width in the segments of the beam is limited to be less than 20. The problem is formulated as follows:

minimize:

$$f(X) = D\left(x_1x_6l_1 + x_2x_7l_2 + x_3x_8l_3 + x_4x_9l_4 + x_5x_{10}l_5\right),$$

subject to:

$$g_1(X) = \frac{6Pl_5}{x_5x_{10}^2} - \sigma_d \leq 0,$$

$$g_2(X) = \frac{6P\left(l_5 + l_4\right)}{x_4x_9^2} - \sigma_d \leq 0,$$

$$g_3(X) = \frac{6P\left(l_5 + l_4 + l_3\right)}{x_3x_8^2} - \sigma_d \leq 0,$$

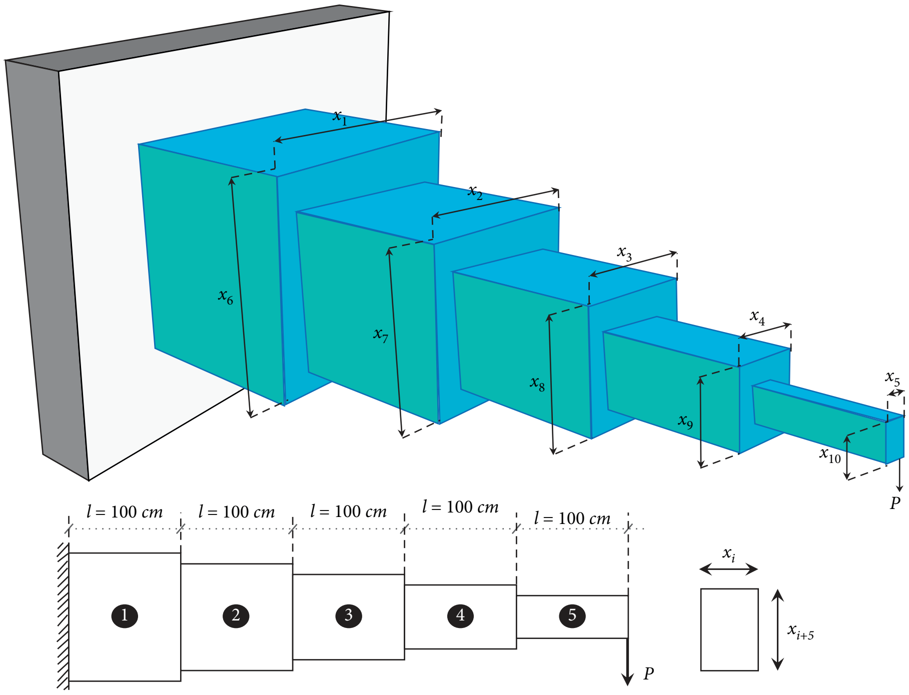


Figure 17: A 3D model of stepped cantilever beam.

$$g_4(X) = \frac{6P(l_5 + l_4 + l_3 + l_2)}{x_2 x_7^2} - \sigma_d \le 0,$$

$$g_5(X) = \frac{6P(l_5 + l_4 + l_3 + l_2 + l_1)}{x_1 x_6^2} - \sigma_d \le 0,$$

$$g_6(X) = \frac{Pl^3}{3E}\left(\frac{1}{I_5} + \frac{1}{I_4} + \frac{1}{I_3} + \frac{1}{I_2} + \frac{1}{I_1}\right) - \Delta_{max} \le 0,$$

$$g_7(X) = \frac{x_{10}}{x_5} - 20 \le 0,$$

$$g_8(X) = \frac{x_9}{x_4} - 20 \le 0,$$

$$g_9(X) = \frac{x_8}{x_3} - 20 \le 0,$$

$$g_{10}(X) = \frac{x_7}{x_2} - 20 \le 0,$$

$$g_{11}(X) = \frac{x_6}{x_1} - 20 \le 0,$$

$$P = 50000N,$$

$$\sigma_d = 14,000N/\text{cm}^2,$$

$$E = 2 \times 10^7 N/\text{cm}^2,$$

$$\Delta_{\max} = 2.7\ \text{cm},$$

$$D = 1.0,$$

variable range:

$$x_1 \in \{1, 2, 3, 4, 5\},$$

$$x_2, x_3 \in \{2.4, 2.6, 2.8, 3.1\},$$

$$1 \le x_4,$$

$$x_5 \le 5,$$

$$x_6 \in \{30, 31, 32, \ldots, 65\}, \qquad (22)$$

$$x_7, x_8 \in \{45, 50, 55, 60, 65\},$$

$$30 \le x_9,$$

$$x_{10} \le 65.$$

The first five constraints are related to the bending stresses in each beam segment that must be lower than the allowable limit (σ_d). Also, the deflection of the cantilever beam tip must be smaller than the limit deflection ($\Delta_{\max}$). The aspect ratio between the height and width of the cross sections must be less than 20 and is applied by the last five constraints. Six of the variables ($x_1, x_2, x_3, x_6, x_7, x_8$) are discrete, and the rest of them (x_4, x_5, x_9, x_{10}) are continuous.

The best and statistical results of the FA [54], thermal exchange optimization (TEO) [82], PSO [82], and SNS are presented in Tables 28 and 29, respectively. To solve this problem, the SNS needs 20,000 evaluations, which is lower than the NFEs of other compared methods, and at the same time, it outperforms all of them in terms of best, mean, worst, and SD.

3.15. Real Application of SNS in Remote Sensing (Segmentation of Satellite Images). Image segmentation is an important topic in the field of remote sensing due to the increasing volume of collected images from satellites, airplanes, and

Table 28: Best results of the cantilever stepped beam.

	FA [54]	TEO [82]	PSO [82]	SNS (present study)
x_1	3	3	5	3
x_2	3.1	3.1	3.1	3.1
x_3	2.6	2.6	2.6	2.6
x_4	2.205	2.21629046531169	2.204556	2.204555692
x_5	1.750	1.76910085340763	1.749757	1.749757012
x_6	60	60	47	60
x_7	55	55	55	55
x_8	50	50	50	50
x_9	44.091	44.014420969625	44.09111	44.09111383
x_{10}	34.995	34.8445777966116	34.995140	34.99514024
$g_1(X)$	−1.83171281	−33.157060	$-1.49E-06$	$-1.81899E-12$
$g_2(X)$	−2.74873225	−25.554426	$-2.38E-06$	$-1.81899E-12$
$g_3(X)$	−153.846153	−153.846154	−153.8461538	−153.8461538
$g_4(X)$	−1203.41242	−1203.412423	−1203.412423	−1203.412423
$g_5(X)$	−111.111111	−111.111111	−419.1942055	−111.1111111
$g_6(X)$	−2.44816450	−2.447883	−2.44430588	−2.448137336
$g_7(X)$	−0.00285714	−0.303792	0	$-3.19744E-14$
$g_8(X)$	−0.00408163	−0.140500	$-4.54E-09$	$-3.55271E-15$
$g_9(X)$	−0.76923076	−0.769231	−0.769230769	−0.769230769
$g_{10}(X)$	−2.25806451	−2.258065	−2.258064516	−2.258064516
$g_{11}(X)$	0	0	−10.6	0
$f(X)$	63893.52	63994.018919	69393.430796	63893.4307958715

Table 29: Comparative results of SNS with other methods for the cantilever stepped beam.

Algorithm	Worst	Mean	Best	SD	NFEs
FA [54]	64262.99420	64144.75312	63893.52578	175.91879	50,000
PSO [82]	NA	NA	69393.430796	NA	300,000
TEO [82]	NA	NA	63994.018919	NA	300,000
SNS (present study)	63905.69523997	63893.88749760	63893.43079588	$2.21E+00$	20,000

other platforms [95]. Image segmentation aims to partition an image into several homogenous sections such that the combination of no two adjacent sections is homogenous. Segmentation is a difficult task due to poor resolution, unfavorable environmental conditions, ambiguous regions, and the presence of pixels with a weak local correlation in satellite images [96, 97]. Metaheuristic methods are proper tools that can deal with the difficulty of discovering the homogeneity measure in the images [98]. In this section, the SNS algorithm is employed for segmenting color satellite images, and then its results have been compared with different optimization algorithms.

Thresholding techniques are the most common methods that are used as the objective function in image segmentation. In this study, Kapur's entropy method is employed for this purpose, and its mathematical formulation is as follows:

$$\begin{aligned}
H_0 &= -\sum_{i=0}^{t_1-1}\left(\frac{p_i}{\omega_0}\right)\log_2\left(\frac{p_i}{\omega_0}\right);\\
H_1 &= -\sum_{i=t_1}^{t_2-1}\left(\frac{p_i}{\omega_1}\right)\log_2\left(\frac{p_i}{\omega_1}\right);\\
H_j &= -\sum_{i=t_j}^{t_{j+1}-1}\left(\frac{p_i}{\omega_j}\right)\log_2\left(\frac{p_i}{\omega_j}\right);\\
H_m &= -\sum_{i=t_m}^{N-1}\left(\frac{p_i}{\omega_m}\right)\log_2\left(\frac{p_i}{\omega_m}\right),
\end{aligned} \tag{23}$$

where

$$\omega_0 = \sum_{i=0}^{t_1-1} p_i;\ \omega_1 = \sum_{i=t_1}^{t_2-1} p_i;\ \omega_j = \sum_{i=t_j}^{t_{j+1}-1} p_i;\ \omega_m = \sum_{i=t_m}^{N-1} p_i, \tag{24}$$

where $H_0, H_1, \ldots, H_m$ are the entropy values of $m+1$ various sections or classes, p_i is the probability of the pixel intensity, and N is the total number of distinct intensity levels.

The utilized image and its histogram patterns are shown in Figure 18. This satellite image is taken from *Pléiades Satellite Imagery* to carry out the experimental study for segmentation. It can be seen that the histogram of the satellite image has a multimodal pattern, and it is very difficult to segment such an image that possesses immense information content.

The experiment was carried out 10 times to choose the best of each algorithm. Figure 19 gives the segmented images for four different levels of thresholding (n) and compares the

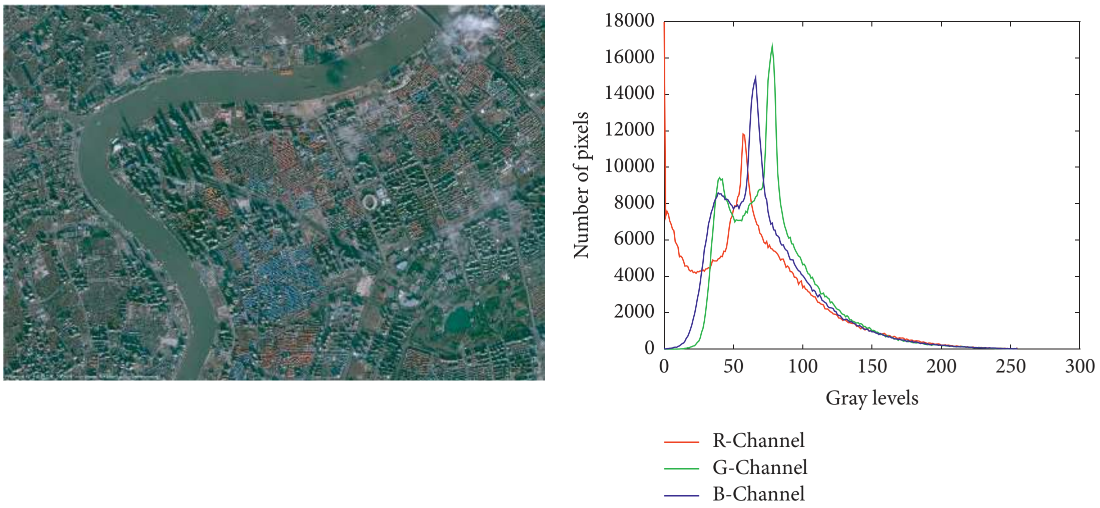


Figure 18: The utilized satellite image and its histograms (https://www.satpalda.com/gallery).

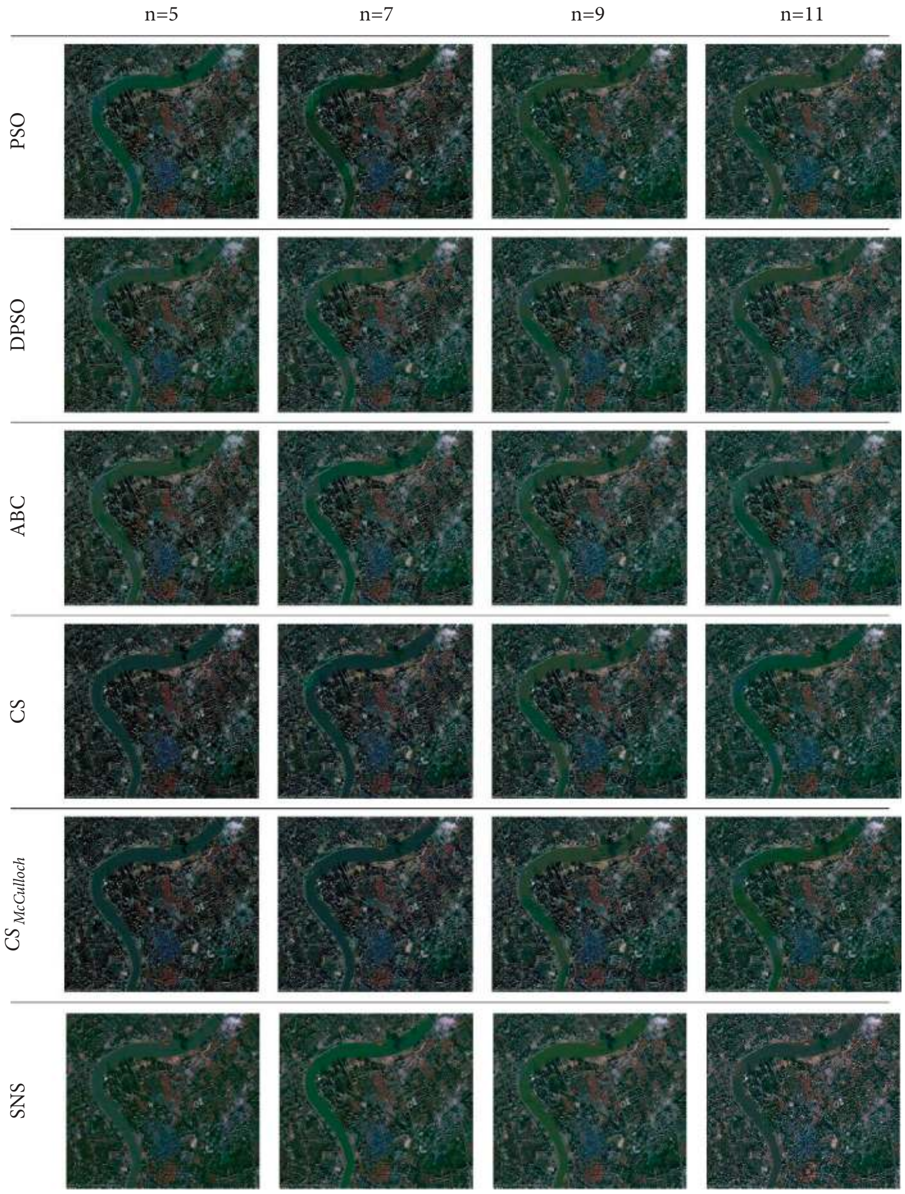


Figure 19: Segmentation results of test images using metaheuristic algorithms for four different threshold levels.

Table 30: Comparison of threshold values between algorithms.

n	Methods		
	PSO [96]	DPSO [96]	ABC [96]
5	40 82 123 164 207 62 100 138 176 217 54 93 131 171 210	43 86 128 171 211 65 104 142 180 217 51 95 99 147 166	40 82 121 162 205 63 100 139 177 217 56 95 134 173 214
7	30 60 90 119 151 184 220 52 80 105 133 161 189 219 44 74 104 132 161 190 219	33 66 97 129 160 191 221 54 82 109 137 166 195 225 43 73 103 132 161 190 219	33 65 95 126 157 188 220 54 81 109 137 166 195 225 46 76 104 132 161 190 217
9	22 44 68 94 120 148 175 203 231 47 68 89 110 132 155 180 203 229 21 48 75 101 128 155 181 206 230	22 45 70 96 121 146 171 197 224 44 63 82 103 125 147 171 197 226 21 47 72 95 119 145 172 199 225	22 48 73 99 125 152 179 204 231 29 51 71 94 121 147 171 197 228 21 48 75 101 126 151 177 201 226
11	17 38 59 78 98 119 140 162 183 207 231 28 46 66 84 101 118 135 153 175 198 224 18 36 56 77 99 121 141 161 181 202 226	16 35 54 74 94 116 139 163 186 209 231 44 63 82 101 119 138 157 176 194 214 234 18 39 59 80 101 122 143 164 186 208 230	16 32 50 70 91 113 136 159 182 206 231 28 46 66 86 106 126 144 164 184 204 225 18 37 56 76 96 118 140 162 185 208 230
n	CS [96]	$\text{CS}_{\text{McCulloch}}$ [96]	SNS (present study)
5	48 89 132 181 210 49 91 133 179 215 51 93 140 182 219	45 88 130 172 208 47 89 132 178 212 48 88 133 173 209	45 88 131 174 215 43 57 71 97 173 56 95 134 174 214
7	42 81 110 145 182 221 245 42 83 115 143 179 208 239 43 77 109 143 181 217 242	39 76 107 139 173 204 232 40 77 108 142 174 203 233 41 76 106 140 175 205 234	35 66 96 126 157 189 220 40 50 51 66 116 182 206 46 75 103 132 161 191 223
9	34 61 79 110 133 159 189 210 233 25 56 77 109 137 161 191 210 240 30 62 79 110 132 164 198 214 245	25 51 75 103 130 156 182 209 235 20 53 70 108 135 159 189 211 238 29 61 75 109 130 163 194 213 240	24 47 72 97 122 148 174 200 225 12 12 14 48 62 115 157 186 191 18 47 74 99 123 148 173 198 226
11	28 47 65 87 105 132 145 167 191 220 239 25 48 69 91 110 132 147 165 191 214 240 28 50 66 85 108 135 148 167 189 211 235	23 45 64 84 105 129 145 164 186 211 235 24 46 67 89 107 131 146 162 188 213 238 22 48 65 85 106 130 147 163 185 212 235	21 41 63 83 103 123 143 165 187 209 232 11 12 40 61 83 100 110 180 217 219 251 16 34 54 74 95 117 139 161 183 206 230

Table 31: Comparison of PSNR between algorithms.

n	PSO [96]	DPSO [96]	ABC [96]	CS [96]	$\text{CS}_{\text{McCulloch}}$ [96]	SNS
5	20.3884	20.4267	20.4707	20.4781	20.4816	**26.0094**
7	21.1449	21.1886	21.2327	21.155	21.316	**26.1077**
9	22.2473	22.3048	22.3342	22.5418	22.7496	**26.7082**
11	23.139	23.2316	23.3399	23.5449	23.6527	**27.6536**

Table 32: Comparison of FSIM between algorithms.

n	PSO [96]	DPSO [96]	ABC [96]	CS [96]	$\text{CS}_{\text{McCulloch}}$ [96]	SNS
5	0.9425	0.9427	0.9433	0.9402	0.9445	0.96655
7	0.9097	0.9125	0.9141	0.9071	0.9174	0.97626
9	0.947	0.9479	0.9486	0.9441	0.9549	0.98178
11	0.9175	0.9179	0.9214	0.9156	0.9276	0.98441

results of the SNS algorithm with PSO, Darwinian PSO (DPSO), ABS, CS, and CS algorithm with McCulloch's method ($\text{CS}_{\text{McCulloch}}$). Also, Table 30 presents the comparison of threshold values between the SNS and other methods.

Various criteria can be used for comparing the performance of metaheuristic algorithms in satellite image segmentation. Peak signal to noise ratio (PSNR) and feature similarity index (FSIM) are two quantitative performance metrics, which are utilized in this study [99]. PSNR measures the accuracy of the reconstructed image and is formulated as follows:

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{(1/mn)\sum_{j=1}^{m}\sum_{k=1}^{n}\left(X_{j,k} - X'_{j,k}\right)^2}\right), \tag{25}$$

where mn is the size of image and X and X' are the main and the processed images, respectively. In addition, FSIM is a criterion that calculates the similarity of the thresholded and original images as follows:

$$\text{FISM} = \frac{\sum_{x\in X} S_L(x) PC_m(x)}{\sum_{x\in X} PC_m(x)}, \tag{26}$$

where $S_L(x)$ shows the similarity of images and $\mathrm{PC}_m(x)$ is the phase congruency map. Tables 31 and 32 present the values of PSNR and FSIM metrics, respectively. According to these results, the SNS algorithm achieved better PSNR and FSIM values for all thresholding levels.

This comprehensive study demonstrates that the developed SNS has competence among the other metaheuristic algorithms. Based on the results in solving classical engineering problems, it can be concluded that the SNS algorithm can perform superior to other algorithms in dealing with semi-real constrained problems. In addition, the image segmentation problem results show the SNS algorithm's ability to solve real-world problems.

4. Conclusion

The social network search (SNS) is a newly developed metaheuristic algorithm that mimics the behavior of social network users in expressing their opinions. In the present study, the SNS algorithm was employed for solving 14 semi-real constrained optimization problems and one real-world application in the field of remote sensing at a relatively low computational cost. From the comparative study, the SNS has shown its potential to handle various constrained optimization problems, and its performance is much better than other state-of-the-art algorithms in terms of the selected performance metrics. This is partly because there are no parameters to be fine-tuned in the SNS. In addition, it is worth mentioning that a simple penalty function method is used for constraint handling, while other compared methods maybe used advanced methods for this task.

This algorithm uses four moods of users in the social networks, namely, imitation, conversation, disputation, and innovation. Users are influenced to express their new views using these four moods simulated from real-world behaviors of users in social networks that randomly accrue for each of them. As further studies, different modifications can be employed to improve the performance of the SNS. Some of these editions are listed below:

(i) In the course of iterations, each user is affected during a randomly selected mood. Developing this random selection to an adaptive selection may affect the performance.

(ii) In the imitation mood, the new view is created inside the imitation space. A new model for this space can have a high impact.

(iii) The shock radius (R) and popularity radius (r) are two important key parameters for improving the imitation mood output.

(iv) In the imitation, conversation, and innovation moods, a random user (X_j) is selected. The selection of this user affects extremely the performance of the SNS. Another selection mechanism can be useful.

(v) The subject (X_k) in conversation mood has an effective impact on the quality of the newly generated solutions.

(vi) In conversation mood, the direction and size of movements are affected by $\mathrm{sign}(f_i - f_j)$. The change of this factor in an adaptive way that affects the size of movements is desirable.

(vii) In the disputation mood, a random number of users are considered. Different strategies can be integrated with this mood. For example, different neighborhood topologies can be used. In addition, dynamic regrouping schema can be useful to improve the performance of disputation mood.

(viii) A new mood can be designed to improve the ability of the SNS by modeling another specific situation in social networks.

Hybridization of the proposed algorithm with other popular algorithms is a common way to benefit from the idea of different metaheuristics to develop a more robust optimization algorithm. In addition, the ability of this algorithm should be examined in dealing with other complex real-world optimization problems in different branches of science.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors gratefully acknowledge the partial financial support from the Natural Science and Engineering Research Council of Canada (NSERC) Discovery Grants Program.

References

[1] A. Kaveh and T. Bakhshpoori, *Metaheuristics: Outlines, MATLAB Codes and Examples*, Springer, Berlin, Germany, 2019.

[2] M. J. Kochenderfer and T. A. Wheeler, *A Algorithms for Optimization*, MIT Press, Cambridge, MA, USA, 2019.

[3] R.-E. Precup, E.-L. Hedrea, R.-C. Roman, E. M. Petriu, A.-I. Szedlak-Stinean, and C.-A. Bojan-Dragos, "Experiment-based approach to teach optimization techniques," *IEEE Transactions on Education*, vol. 64, no. 2, pp. 88–94, 2021.

[4] S. S. Rao, *Engineering Optimization*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2009.

[5] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm-a novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers & Structures*, vol. 110-111, no. 111, pp. 151–166, 2012.

[6] X. S. Yang, "Optimization and metaheuristic algorithms in engineering," *Metaheuristics in Water, Geotechnical and Transport Engineering*, vol. 1–23, 2013.

[7] N. Razmjooy, M. Ashourian, and Z. Foroozandeh, "Metaheuristics and optimization in computer and electrical engineering," vol. 696, 2021.

[8] X. S. Yang, A. H. Gandomi, S. Talatahari, and A. H. Alavi, "Metaheuristics in water, geotechnical and transport engineering," *Metaheuristics in Water, Geotechnical and Transport Engineering*, Elsevier Science, Amsterdam, Netherlands, 2013.

[9] A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, "Metaheuristic applications in structures and infrastructures," *Metaheuristic Applications in Structures and Infrastructures*, Elsevier Science, Amsterdam, Netherlands, 2013.

[10] S. Fong, S. Deb, and A. Chaudhary, "A review of metaheuristics in robotics," *Computers and Electrical Engineering*, vol. 43, pp. 278–291, 2015.

[11] T. W. Liao, P. J. Egbelu, B. R. Sarker, and S. S. Leu, "Metaheuristics for project and construction management - a state-of-the-art review," *Automation in Construction*, vol. 20, no. 5, pp. 491–505, 2011.

[12] C. Dhaenens and L. Jourdan, "Metaheuristics for data mining," *4OR*, vol. 17, no. 2, pp. 115–139, 2019.

[13] S. C. Yusta, "Different metaheuristic strategies to solve the feature selection problem," *Pattern Recognition Letters*, vol. 30, no. 5, pp. 525–534, 2009.

[14] F. Xhafa and A. Abraham, "Metaheuristics for scheduling in industrial and manufacturing applications studies in computational intelligence," *Computational Intelligence*, vol. 128, p. 345, 2008.

[15] A. Nakib, "Design of static metaheuristics for medical image analysis," *Metaheuristics for Medicine and Biology*, vol. 704, pp. 1–22, 2017.

[16] F. Glover and G. A. Kochenberger, *Handbook of Metaheuristics*, Springer, New York, NY, USA, 2003.

[17] A. Kaveh, *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, Springer, Cham, Switzerland, 2nd edition, 2016.

[18] I. Rechenberg, *Evolutionsstrategien, Medizinische Informatik und Statistik*, Springer, Berlin, Heidelberg, Germany, 1978.

[19] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.

[20] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[21] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the ICNN'95-International Conference on Neural Networks*, pp. 1942–1948, IEEE, Perth, Australia, 1995.

[22] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[23] X. S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, no. 4, pp. 330–343, 2010.

[24] X.-S. Yang, "A new metaheuristic Bat-inspired Algorithm," *Nature Inspired Cooperative Strategies for Optimization*, vol. 284, pp. 65–74, 2010.

[25] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search," *Acta Mechanica*, vol. 213, no. 3-4, pp. 267–289, 2010.

[26] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.

[27] X.-S. Yang and S. Deb, "Eagle strategy using Lévy Walk and firefly algorithms for stochastic optimization," *Nature Inspired Cooperative Strategies for Optimization*, vol. 284, pp. 101–111, 2010.

[28] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.

[29] X.-S. Yang, "Flower pollination algorithm for global optimization, unconventional computation and natural computation," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 240–249, 2012.

[30] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[31] A. Soares, R. Râbelo, and A. Delbem, "Optimization based on phylogram analysis," *Expert Systems with Applications*, vol. 78, pp. 32–50, 2017.

[32] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[33] R. Sheikholeslami and S. Talatahari, "Developed swarm optimizer: a new method for sizing optimization of water distribution systems," *Journal of Computing in Civil Engineering*, vol. 30, no. 5, Article ID 04016005, 2016.

[34] A. Kaveh, S. Talatahari, and N. Khodadadi, "Stochastic paint optimizer: theory and application in civil engineering," *Engineering with Computers*, vol. 2020, pp. 1–32, 2020.

[35] S. Talatahari and M. Azizi, "Chaos Game Optimization: a novel metaheuristic algorithm," *Artificial Intelligence Review*, vol. 54, no. 2, pp. 917–1004, 2020.

[36] S. Talatahari and M. Azizi, "Optimization of constrained mathematical and engineering design problems using chaos game optimization," *Computers and Industrial Engineering*, vol. 145, Article ID 106560, 2020.

[37] M. Azizi, "Atomic orbital search: a novel metaheuristic algorithm," *Applied Mathematical Modelling*, vol. 93, pp. 657–683, 2021.

[38] M. Azizi, S. Talatahari, and A. Giaralis, "Optimization of engineering design problems using atomic orbital search algorithm," *IEEE Access*, vol. 9, pp. 102497–102519, 2021.

[39] S. Talatahari, M. Azizi, and A. H. Gandomi, "Material generation algorithm: a novel metaheuristic algorithm for optimization of engineering problems," *Processes*, vol. 9, no. 5, p. 859, 2021.

[40] S. Talatahari, M. Azizi, M. Tolouei, B. Talatahari, and P. Sareh, "Crystal structure algorithm (CryStAl): a metaheuristic optimization method," *IEEE Access*, vol. 9, pp. 71244–71261, 2021.

[41] S. Talatahari, H. Bayzidi, and M. Saraee, "Social network search for global optimization," *IEEE Access*, vol. 9, pp. 92815–92863, 2021.

[42] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.

[43] A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, "Firefly algorithm with chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 1, pp. 89–98, 2013.

[44] A. Kaveh and S. Talatahari, "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures," *Computers and Structures*, vol. 87, no. 5-6, pp. 267–283, 2009.

[45] B. H. Abed-alguni, "Island-based cuckoo search with highly disruptive polynomial mutation," *International Journal of Artificial Intelligence*, vol. 17, pp. 57–82, 2019.

[46] S. Talatahari and M. Azizi, "Optimal design of real-size building structures using quantum-behaved developed swarm optimizer," *The Structural Design of Tall and Special Buildings*, vol. 29, p. e1747, 2020.

[47] H. Zapata, N. Perozo, W. Angulo, and J. Contreras, "A hybrid swarm algorithm for collective construction of 3D structures," *International Journal of Artificial Intelligence*, vol. 18, pp. 1–18, 2020.

[48] M. Azizi, R. G. Ejlali, S. A. Mousavi Ghasemi, and S. Talatahari, "Upgraded Whale Optimization Algorithm for fuzzy logic based vibration control of nonlinear steel structure," *Engineering Structures*, vol. 192, pp. 53–70, 2019.

[49] R.-E. Precup, R.-C. David, R.-C. Roman, A.-I. Szedlak-Stinean, and E. M. Petriu, "Optimal tuning of interval type-2 fuzzy controllers for nonlinear servo systems using slime mould algorithm," , 2021.

[50] A. Kaveh, S. Talatahari, and N. Khodadadi, "Hybrid invasive weed optimization-shuffled frog-leaping algorithm for optimal design of truss structures," *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, vol. 44, no. 2, pp. 405–420, 2020.

[51] A. Kaveh, S. Talatahari, and N. Khodadadi, "The hybrid invasive weed optimization-shuffled frog-leaping algorithm applied to optimal design of frame structures," *Periodica Polytechnica: Civil Engineering*, vol. 63, pp. 882–897, 2019.

[52] B. Hadi, *Social Network Search for Solving Engineering Problems*, https://www.mathworks.com/matlabcentral/fileexchange/97577-social-network-search-for-solving-engineering-problems, 2021.

[53] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Applied Soft Computing*, vol. 10, no. 2, pp. 629–640, 2010.

[54] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using Firefly Algorithm," *Computers & Structures*, vol. 89, no. 23-24, pp. 2325–2336, 2011.

[55] H. Chickermane and H. C. Gea, "Structural optimization using a new local approximation method," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 5, pp. 829–846, 1996.

[56] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

[57] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.

[58] M.-Y. Cheng and D. Prayogo, "Symbiotic Organisms Search: a new metaheuristic optimization algorithm," *Computers & Structures*, vol. 139, pp. 98–112, 2014.

[59] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.

[60] G. G. Wang, "Adaptive response surface method using inherited Latin hypercube design points," *Journal of Mechanical Design*, vol. 125, no. 2, pp. 210–220, 2003.

[61] Y. A. Anita, A. Yadav, and N. Kumar, "Artificial electric field algorithm for engineering optimization problems," *Expert Systems with Applications*, vol. 149, Article ID 113308, 2020.

[62] A. Kaveh and A. Dadras Eslamlou, "Water strider algorithm: a new metaheuristic and applications," *Structures*, vol. 25, pp. 520–541, 2020.

[63] A. H. Gandomi and D. A. Roke, "Engineering optimization using interior search algorithm," in *Proceedings of the 2014 IEEE Symposium Series on Computational Intelligence-SIS 2014: 2014 IEEE Symposium on Swarm Intelligence*, pp. 20–26, Orlando, FL, USA, December 2015.

[64] J. Wu, Y.-G. Wang, K. Burrage, Y.-C. Tian, B. Lawson, and Z. Ding, "An improved firefly algorithm for global continuous optimization problems," *Expert Systems with Applications*, vol. 149, Article ID 113340, 2020.

[65] D. Sattar and R. Salim, "A smart metaheuristic algorithm for solving engineering problems," *Engineering with Computers*, vol. 37, pp. 2389–2417, 2020.

[66] X. S. Yang and A. Hossein Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464–483, 2012.

[67] B. Akay and D. Karaboga, "Artificial bee colony algorithm for large-scale problems and engineering design optimization," *Journal of Intelligent Manufacturing*, vol. 23, no. 4, pp. 1001–1014, 2012.

[68] N. Ben Guedria, "Improved accelerated PSO algorithm for mechanical engineering optimization problems," *Applied Soft Computing*, vol. 40, pp. 455–467, 2016.

[69] G. Dhiman and V. Kumar, "Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications," *Advances in Engineering Software*, vol. 114, pp. 48–70, 2017.

[70] S. Talatahari, M. Azizi, and M. Toloo, "Fuzzy adaptive charged system search for global optimization," *Applied Soft Computing*, vol. 109, Article ID 107518, 2021.

[71] P. Kim and J. Lee, "An integrated method of particle swarm optimization and differential evolution," *Journal of Mechanical Science and Technology*, vol. 23, no. 2, pp. 426–434, 2009.

[72] A. Ravindran, K. M. Ragsdell, and G. V. Reklaitis, *Engineering Optimization*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2006.

[73] L. d. S. Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1676–1683, 2010.

[74] Q. He and L. Wang, "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization," *Applied Mathematics and Computation*, vol. 186, no. 2, pp. 1407–1422, 2007.

[75] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, pp. 89–99, 2007.

[76] H. F. zhuo, L. Wang, and Q. He, "An effective co-evolutionary differential evolution for constrained optimization," *Applied Mathematics and Computation*, vol. 186, pp. 340–356, 2007.

[77] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.

[78] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Knowledge-based systems equilibrium optimizer: a novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, Article ID 105190, 2019.

[79] A. W. Mohamed, "A novel differential evolution algorithm for solving constrained engineering optimization problems," *Journal of Intelligent Manufacturing*, vol. 29, no. 3, pp. 659–692, 2018.

[80] F. MiarNaeimi, G. Azizyan, and M. Rashki, "Multi-level cross entropy optimizer (MCEO): an evolutionary optimization algorithm for engineering problems," *Engineering with Computers*, vol. 34, no. 4, pp. 719–739, 2018.

[81] J. Arora, *Introduction to Optimum Design*, Elsevier, Amsterdam, Netherlands, 2012.

[82] A. Kaveh and A. Dadras, "A novel meta-heuristic optimization algorithm: thermal exchange optimization," *Advances in Engineering Software*, vol. 110, pp. 69–84, 2017.

[83] H. Zamani, M. H. Nadimi-Shahraki, and A. H. Gandomi, "CCSA: conscious neighborhood-based crow search algorithm for solving global optimization problems," *Applied Soft Computing Journal*, vol. 85, 2019.

[84] S. Talatahari and M. Azizi, "Tribe-charged system search for global optimization," *Applied Mathematical Modelling*, vol. 93, pp. 115–133, 2021.

[85] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design optimization," *Journal of Mechanical Design*, vol. 112, no. 2, pp. 223–229, 1990.

[86] X. Han, L. Yue, Y. Dong, Q. Xu, G. Xie, and X. Xu, "Efficient hybrid algorithm based on moth search and fireworks algorithm for solving numerical and constrained engineering optimization problems," *The Journal of Supercomputing*, vol. 76, pp. 9404–9429, 2020.

[87] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems," *Applied Soft Computing*, vol. 13, no. 5, pp. 2592–2612, 2013.

[88] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-Verse Optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, 2016.

[89] S. N. Chegini, A. Bagheri, and F. Najafi, "PSOSCALF: a new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems," *Applied Soft Computing*, vol. 73, pp. 697–726, 2018.

[90] K. E. Parsopoulos and M. N. Vrahatis, "Unified particle swarm optimization for solving constrained engineering optimization problems," *Lecture Notes in Computer Science*, vol. 3612, pp. 582–591, 2005.

[91] B. Hossain and T. Hasegawa, "Nonlinear mixed-discrete structural optimization," *Journal of Structural Engineering*, vol. 115, pp. 626–646, 1989.

[92] Y. Yun, *Study on Adaptive Hybrid Genetic Algorithm and its Applications to Engineering Design Problems*, MSc thesis, Waseda University, Tokyo, Japan, 2005.

[93] B. D. Youn and K. K. Choi, "A new response surface methodology for reliability-based design optimization," *Computers and Structures*, vol. 82, no. 2-3, pp. 241–256, 2004.

[94] A. R. Yildiz, H. Abderazek, and S. Mirjalili, "A comparative study of recent non-traditional methods for mechanical design optimization," *Archives of Computational Methods in Engineering*, vol. 27, no. 4, pp. 1031–1048, 2020.

[95] L. Wan, Y. Li, H. Cen et al., "Combining UAV-based vegetation indices and image classification to estimate flower number in oilseed rape," *Remote Sensing*, vol. 10, no. 9, p. 1484, 2018.

[96] S. Suresh and S. Lal, "An efficient cuckoo search algorithm based multilevel thresholding for segmentation of satellite images using different objective functions," *Expert Systems with Applications*, vol. 58, pp. 184–209, 2016.

[97] B. A. Johnson and L. Ma, "Image segmentation and object-based image analysis for environmental monitoring: recent areas of interest, researchers' views on the future priorities," *Remote Sensing*, vol. 12, no. 11, p. 1772, 2020.

[98] R. Al-Ruzouq, A. Shanableh, M. B. A. Gibril, and S. Al-Mansoori, "Image segmentation parameter selection and ant colony optimization for date palm tree detection and mapping from very-high-spatial-resolution aerial imagery," *Remote Sensing*, vol. 10, 2018.

[99] H. Jia, C. Lang, D. Oliva, W. Song, and X. Peng, "Hybrid grasshopper optimization algorithm and differential evolution for multilevel satellite image segmentation," *Remote Sensing*, vol. 11, no. 9, p. 1134, 2019.