



# Search and Rescue Optimization Algorithm for Size Optimization of Truss Structures with Discrete Variables

Amir Shabani\*, Behrouz Asgarian\*\* and Miguel Salido\*\*\*

#### ARTICLE INFO

Article history: Received: September 2018. Revised: January 2019. Accepted: February 2019. Keywords: Truss Optimization Metaheuristic Algorithm Discrete variables Constrained optimization **Abstract:** In this paper, a new metaheuristic algorithm is developed to sizing optimization of truss structures with discrete variables. The proposed algorithms namely search and rescue optimization algorithm (SAR), imitates the exploration behavior of humans during search and rescue operations. The performance of the proposed algorithm is evaluated using several discrete truss design problems and the obtained results compared with the results of other optimization algorithms. The comparisons demonstrated that the best averages and standard deviations of results were obtained by SAR for all the studied problems and the proposed algorithm outperforms the other compared optimization algorithms in terms of finding the optimized weight of the truss (accuracy). According to the numerical results, it can be concluded that SAR is a very efficient and robust algorithm for designing truss structures with discrete variables.

# 1. Introduction

Nowadays, limitations of resources and development of computer science has led to optimization techniques which are widely used in engineering and industry. In design optimization, different goals are considered; such as minimization of cost, material usage, and production time performance or maximization of performance, lifetime service, and quality [1], [2]. Structural optimization problems have large number of design variables, constraints, and many local minima and they are generally highly nonlinear and constrained. The truss optimum design is a kind of structural optimization design problem and the minimization of weight or volume under specific criteria is normally the aim of this problem. Optimization techniques can be classified into two categories: classical and metaheuristic methods [3].

Classical methods such as mathematical programming often need gradient information to find an optimum solution and they are not suitable for constrained, non-differentiable, and multimodal problems such as truss optimization. Unlike them, metaheuristic algorithms do not require any gradient information of the problem [4]. Most of metaheuristic algorithms are inspired by physical or natural phenomena. For instance, particle swarm optimization (PSO) [5], [6] has been developed based on movement of birds or fish to find food sources. Simplified dolphin echolocation (SDE) [7], [8] is based on the hunting technique of dolphins. They send echo in different directions and listen to the reflections and then move towards their prey. The capability to escape from local optima and simple implementation of metaheuristic algorithms have led to the wide use of them in various engineering optimization problems. During the last decade, many metaheuristic algorithms have been utilized for structural optimization; including harmony search (HS) [9], charged system search (CSS) [10], modified dolphin monitoring (MDM) [11] and cuckoo search algorithm (CS) [12]. Optimal design of truss structures is an important field structural optimization and many metaheuristic in algorithms have been developed for truss optimization; such

<sup>\*</sup> PhD Candidate, Faculty of Civil Engineering, K.N.Toosi University of Technology, Tehran, Iran.

<sup>\*\*</sup> Corresponding Author: Professor, Faculty of Civil Engineering, K.N.Toosi University of Technology, Tehran, Iran. Email: Asgarian@kntu.ac.ir

<sup>\*\*\*</sup> Professor, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Valencia, Spain

as particle swarm optimization (PSO) [13], simulated annealing (SA) [14], ant colony optimization (ACO) [15], teaching learning based optimization algorithm (TLBO) [16] and enhanced colliding bodies optimization (ECBO) [17].

Design variables can be categorized into continuous or discrete. Weight minimization of truss with discrete design variables are very difficult and they can be considered as NP-hard problems [18]. Most of metaheuristic algorithms were proposed for continuous optimizations and then developed for discrete ones. Search and rescue optimization algorithm (SAR) [19] has been recently proposed by authors for continuous optimization. This algorithm was inspired by the explorations which were carried out by humans during search and rescue operations. The preliminary studies show that it is extremely efficient and outperforms the existing algorithms in solving continuous engineering optimization problems. In this study, SAR is applied to size optimization of truss structures with discrete variables and the performance of the proposed algorithm is compared with those of the other methods.

The remainder of the paper is organized as follows: Section 2 presents the description of SAR. Next, the structural optimization is briefly described in section 3. Afterward, in order to compare the proposed algorithm with the other optimization methods, four truss design problems are presented in section 4. Finally, section 5 presents the concluding remarks.

## 2. Search and Rescue Optimization Algorithm

Like other live creatures, human beings search for different purposes as groups. Search can be done for variety of goals such as hunting, finding food sources or lost people. One type of the group search is search and rescue operations. A new metaheuristic algorithm called search and rescue optimization algorithm (SAR) has been proposed by the authors based on search and rescue operations. These operations are sometimes carried out to find specific people who are lost. In the following, the procedure of finding lost people is described considering the main concepts of this operation. The members of the search group can recognize the clues and traces of lost people based on the training received. Several guidelines are developed for search and rescue operations by some organisations such as American Society for Testing and Materials (ASTM) and the National Fire Protection Association (NFPA) [20]. The discovered clues reveal different information about lost people. For example, some clues indicate the probability of the presence of lost people in a specific position. The training help the humans (the member of search group) to evaluate the clues. Also, they can use compass and global positioning system (GPS) to find their locations. The obtained information is shared between the humans by communication equipment. Information of clues is gathered during the search. Humans set aside some clues whenever they found better clues in other positions but information of the abandoned clues are used to improve searching operations. Hence, clues can be categorized as two types:

1. Hold clue: there is a human in the clue position and searches around it.

2. Abandoned clue: The human who found the clue has left it to find better clues, but the information of the clue is available for other humans.

In SAR algorithm, the positions of humans are equal to the solutions of the optimization problem and the amount of clues in these positions denote the objective function of them. In the following, the mathematical modelling of SAR for solving a "maximization problem" is described.

The flowchart of SAR is presented in Fig. 1. The positions of hold and abandoned clues are stored in matrices **X** (human position matrix) and **M** (memory matrix), respectively. They are  $N \times D$  matrices where *D* is the dimension of the problem and *N* is the number of humans. This information creates clues matrix (matrix C) by Eq.1.

$$C = \begin{bmatrix} X \\ M \end{bmatrix} = \begin{bmatrix} X_{11} & \dots & X_{1D} \\ \vdots & \ddots & \vdots \\ X_{N1} & \dots & X_{ND} \\ M_{11} & \dots & M_{1D} \\ \vdots & \ddots & \vdots \\ M_{N1} & \dots & M_{ND} \end{bmatrix}$$
(1)

Where  $X_{NI}$  is the position of the 1<sup>st</sup> dimension for the  $N^{\text{th}}$  human and  $M_{ID}$  is the position of the  $D^{\text{th}}$  dimension for the 1<sup>st</sup> memory. According to the information, humans search around the clues or seek in directions that are created by connecting the clues together [21]. So, the procedure of searching in search and rescue operations can be classified into social and individual phases.



#### 2.1 Social Phase

In this phase, humans search based on the gathered information. They connect the found clues together and search in those directions [21]. To model this phase, firstly, a clue from the clues matrix (matrix C) is randomly selected for each human. Then, the search direction is created by Eq.2.

$$SD_i = (X_i - C_k), \qquad k \neq i$$
 (2)

Where  $X_i$ ,  $C_k$ , and  $SD_i$  are the position of the *i*<sup>th</sup> human, the position of the *k*<sup>th</sup> clue, and the search direction for the *i*<sup>th</sup> human, respectively. *k* is a random integer number ranging between 1 and 2*N*. For *i*=*k*,  $C_i$  will be equal to  $X_i$ . So, *k* is chosen in such a way that  $k \neq i$ . Also, searching around better clues increases the probability of finding the lost person. Therefore, the search is done around the position that has better clues. In other words, if there are better clues in the position of the *i*<sup>th</sup> human compared to the position of the *k*<sup>th</sup> clue, (for maximization problems: the value of objective function for solution  $X_i$  is greater than that of  $C_k$ ),  $X_i$  is selected to search and vice versa. Furthermore, the humans try to explore a certain location only one time in search and rescue operations. Hence, movements of the humans toward

each other should be limited. For this purpose, only some dimensions of  $X_i$  will be changed by moving in the direction of Eq. 2. This limitation is applied by the binomial crossover operator. Finally, new position of the *i*<sup>th</sup> human in all dimensions is calculated by Eq.3.

$$X_{i,j}' = \begin{cases} \begin{cases} C_{k,j} + r1 \times SD_{i,j} & \text{if } f(C_k) > f(X_i) \\ X_{i,j} + r1 \times SD_{i,j} & \text{otherwise} \end{cases} & \text{if } r2 < SE \text{ or } j = j_{rand} \\ X_{i,j} & \text{otherwise} \end{cases}$$
(3)

Where  $X'_{i,j}$  is the new position of the *j*<sup>th</sup> dimension for the *i*<sup>th</sup> human,  $C_{k,j}$  is the position of the *j*<sup>th</sup> dimension for the stored clue  $k^{th}$ ,  $f(C_k)$  and  $f(X_i)$  are the objective function values for the solution  $C_k$  and  $X_i$ , respectively. r1 is a random number with a uniform distribution in the range [-1, 1]. r2 is a uniformly distributed random number in the range [0, 1]. r2 is different for each dimension, but r1 is fixed for all dimensions.  $j_{rand}$  is a random integer number ranged between 1 and D which ensures that at least one dimension of  $X'_{i,j}$  is different from  $X_{i,j}$ . *SE* (Social effect) is an algorithm parameter ranged between 0 and 1. SE is used to control the effect of group members on each other in the social phase

## 2.2 Individual Phase

In individual phase, humans search regardless of the position and amount of clues found by others. They search around their current positions. The idea of connecting different clues is used for this phase. The new position of the *i*<sup>th</sup> human is calculated as following equation:

$$X'_{i} = X_{i} + r3 \times (C_{k} - C_{m}), \qquad i \neq k \neq m$$
(4)

Where *k* and *m* are random integer numbers ranging between 1 and 2*N* that  $i \neq k \neq m$ . r3 is a random number with a uniform distribution ranging between 0 and 1. The matrix C is updated in each human search phase.

#### 2.3 Boundary Control

The solutions obtained by social and individual phases should be located in the solution space, and if they are out of allowable solution space, they should be modified. The new position of the  $i^{th}$  human is modified by Eq.5.

$$X_{i,j}' = \begin{cases} (X_{i,j} + X_j^{max})/2 & \text{if } X_{i,j}' > X_j^{max} \\ (X_{i,j} + X_j^{min})/2 & \text{if } X_{i,j}' < X_j^{min} \\ = 1, \dots, D \end{cases}$$
(5)

Where  $X^{max_j}$  and  $X^{min_j}$  indicate the maximum and minimum of the *j*<sup>th</sup> dimension, respectively.

#### 2.4 Update Information and Positions

If the new solution  $X'_i$  generated in social or individual phases, is better than the previous one (for maximization problems: the value of objective function for solution  $X'_i$  is greater than that of  $X_i$ ), the previous position ( $X_i$ ) will be stored in a random position of the memory matrix (M) and this position will be accepted as a new position. Otherwise, this position is discarded and the memory is not updated. This step can be defined as follows:

$$M_{n} = \begin{cases} X_{i} & if \ f(X_{i}') > f(X_{i}) \\ M_{n} & otherwise \end{cases}$$
(6)

$$X_{i} = \begin{cases} X'_{i} & \text{if } f(X'_{i}) > f(X_{i}) \\ X_{i} & \text{otherwise} \end{cases}$$
(7)

Where  $\mathbf{M}_{n}$  is the position of the  $n^{\text{th}}$  stored clue in the memory matrix. n is a random integer number in the range [1, N].

#### 2.5 Abandon Clues

The lost people may be injured. So, the searching space must be explored in the shortest possible time and humans must stop unsuccessful searching around clues after some effort. To model this behavior, the number of unsuccessful searches of each human is stored. At first, Unsuccessful Search Number (*USN*) is set to 0 for each human. Also, whenever a human finds better clues it is set to 0 for that human; otherwise, it will increase by 1 point.

$$USN_i = \begin{cases} USN_i + 1 & if f(X'_i) < f(X_i) \\ 0 & otherwise \end{cases}$$
(8)

A solution is left when it cannot be improved after a specific number of searches (Maximum Unsuccessful Search Number (*MU*)). Then, a new solution replaces it using Eq.9.  $X_{i,j} = X_j^{min} + r4 \times (X_j^{max} - X_j^{min}), \quad j = 1, ..., D$  (9) Where  $X_{i,j}$  is position of the *j*<sup>th</sup> dimension for the *i*<sup>th</sup> human, *r4* is a random number for the *j*<sup>th</sup> dimension generated with a uniform distribution ranged between 0 and 1. The *MU* 

parameter indicates the maximum number of unsuccessful searches before leaving a clue. MU directly relates to the dimension of the problem. As the search space increases, the maximum number of unsuccessful searches also increases.

# 2.6 Pseudo Code of Continuous Search and Rescue Optimization Algorithm (SAR)

The modelling of human searches in search and rescue operations is done in two phases as introduced in the previous section. The pseudo code of this algorithm has been presented in Algorithm 1 for solving a maximization problem. Position sorting is performed only once before the iterations begin.

Algorithm 1	Pseudo	code of	continuous	SAR.
-------------	--------	---------	------------	------

#### 1. Begin:

- 2. Randomly initialize a population of 2N solutions uniformly distributed in the range  $[X_i^{min}, X_i^{max}]$ , j=1, ..., D
- 3. Sort the solutions in the decreasing order and find the current best position (Xbest)
- 4. Use the first half of the sorted solutions for human positions (X) and the others for memory matrix (M)
- 5. Define the algorithm parameters (SE, MU) and set USN\_i=0 where i=1,...,N
- 6. While stop criterion is not satisfied do
- 7. **For** i=1 to N **do**
- 8. Update the clues matrix (C) by Eq.1
- 9. Generate search direction (SD<sub>i</sub>) using Eq.2
- $10. \quad \text{For } j{=}1 \text{ to } D \text{ do} \\$
- 11. Calculate the new j<sup>th</sup> dimension of the i<sup>th</sup> human using Eq.3
- 12. Boundary control of the new j<sup>th</sup> dimension of the i<sup>th</sup> human using Eq.5
- 13. End For
- 14. Update the n<sup>th</sup> memory using Eq.6
- 15. Update the position of the i<sup>th</sup> human using Eq.7
- 16. Update USN<sub>i</sub> using Eq.8
- 17. Update the clues matrix (C) by Eq.1
- 18. Calculate the new position of the i<sup>th</sup> human using Eq.4
- 19. Boundary control of the new position of the i<sup>th</sup> human using Eq.5
- 20. Update the n<sup>th</sup> memory using Eq.6
- 21. Update the position of the i<sup>th</sup> human using Eq.
- 22. Update USN<sub>i</sub> using Eq.8
- 23. If  $USN_i > MU$  do
- 24. For j=1 to D do
- 25. Calculate the j<sup>th</sup> dimension of the i<sup>th</sup> human using Eq.8
- 26. End for
- $27. \qquad USN_i = 0$
- 28. End If
- 29. End for
- 30. Find the current best position and update Xbest
- 31. End while
- 32. Return Xbest
- 33. End

## 2.7 Discrete SAR algorithm

SAR is developed for continuous optimization problems. In order to handle discrete problems, the continuous solutions of SAR generated by Eq.5 will be rounded to the nearest allowable discrete values using Eq.10.

$$X_{discrete}^{new} = Fix(X_{continuous}^{new})$$
(10)

Where Fix(X) is a function which rounds each element of the solution to the nearest allowable discrete value. Also, sometimes the discrete solution generated by Eq.10 and the social or individual phases is not different from the current one (previous position of human). Therefore, these steps will be repeated until different solutions are achieved.

#### 3. Statement of the truss optimization problems

For size optimization of truss structures, the objective is to minimize weight of the structure subjected to stress and deflection constraints. In this problems, cross sectional areas are considered as design variables and they are selected from an allowable list of standard sections. The problem formulation is given as follows:

Minimize: 
$$W(\{A\}) = \sum_{i=1}^{n} \gamma_i . A_i . L_i$$
 (11)

Subjected to:

$$\delta_{\min} \le \delta_j \le \delta_{\max}, \qquad j = 1, 2, \dots, m \tag{12}$$

$$\sigma_{min} \le \sigma_p \le \sigma_{max}, \qquad p = 1, 2, \dots, n \tag{13}$$

$$\sigma_p^b \le \sigma_p \le 0, \qquad p = 1, 2, \dots, nc \tag{14}$$

$$A_p \in allowable \ section \ list = [S_1, S_2, S_3, \dots, S_k]$$
(15)

Where A denotes a vector containing the cross sectional areas of members (design variables), W({A}) is the weight of the truss,  $\gamma_i$ , A<sub>i</sub>, and L<sub>i</sub> are the material density, the cross sectional area and the length of members, respectively. m is the number of nodes, n is the number of member and nc is the number of compression members.  $\delta_j$  is the displacement of node j and  $\delta_{min}$  and  $\delta_{max}$  are corresponding lower and upper limits. Also,  $\sigma_p$  is the stress of member p and  $\sigma_{min}$  and  $\sigma_{max}$  are corresponding lower and upper limits.  $\sigma_p^{b}$  are stress and buckling stress of the member p, respectively. A<sub>p</sub> is the cross-sectional area of the member p that is chosen from allowable section list that includes k elements. The elements are sorted in ascending orders in the allowable section list.

SAR is proposed for unconstrained optimization problems. To handle constraints, a penalty function approach is utilized. In this method, a value is added to the objective function based on the total constraint violation and the constrained optimization problem is transformed into an unconstrained one using the following formulation:

$$F = \left(1 + \varepsilon_1 \cdot \left(\sum_{j=1}^{K} \Phi^j\right)\right)^{\varepsilon_2} \times W(\{A\})$$
(16)

Where *F* is unconstrained objective function,  $\varepsilon_1$  and  $\varepsilon_2$  are the coefficients of the penalty function. In this study,  $\varepsilon_1$  increases from 1 in the first iteration to 15 in the last iteration and  $\varepsilon_2$  is set to 2. K is the number of constraints and  $\Phi^j$  related to the *j*<sup>th</sup> constraint is calculated for each constraint as follows:

$$\Phi^{j} = \begin{cases} \left| \frac{c_{max}^{j} - c^{j}}{c_{max}^{j}} \right| & \text{if } c^{j} > c_{max}^{j} \\ \left| \frac{c_{min}^{j} - c^{j}}{c_{min}^{j}} \right| & \text{if } c^{j} < c_{min}^{j} \\ 0 & \text{otherwise} \end{cases}$$
(17)

Where  $c_{min}^{j}$  and  $c_{max}^{j}$  are lower and upper bound of the  $j^{th}$  constraint and  $c^{j}$  is the value of the  $j^{th}$  constraint.

#### 4. Numerical examples and results

In this section, the performance of SAR was investigated using four truss design examples. The objective is to minimize the weight of the truss using discrete variables. The design variables are cross sectional areas of all members. The obtained results of SAR were compared with results of other researchers. For each design example, 50 independent runs were performed by SAR and the best, average and standard deviation of the results were presented for each problem. The proposed algorithm and direct stiffness method for analysis of truss structures were coded in MATLAB and all runs were performed on a 64-bit computer with an Intel i7 (3.4 GHz) processor and 32GB of RAM. For solving the truss examples, the population size of SAR were considered as follows: 25, 10, 20, 25, and 15, respectively. Also, SE and MU (two control parameters of SAR) were respectively set to 0.3 and 30×D (D is the number of variables) for all the examples. In the following tables, "NA", "CV", "STD", and "MNSA" mean not available, constraint violation, standard deviation and maximum number of structural analyses, respectively. The best results are highlighted in bold in the following tables.

#### 4.1 Planar 10-bar truss design

The 10-bar truss design problem, shown in Fig. 2, is a standard truss optimization which was frequently solved by many researchers. This cantilever truss including ten bars is subjected to a single loading condition P = 100 kips at nodes 2 and 4. The elastic modulus is  $10^4$  ksi and the material density is 0.1 lb/in<sup>3</sup> for all members. Cross sectional areas of all the members are selected from the discrete list L=[ 1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38,

3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.50, 30.00, 33.50] (in<sup>2</sup>). The stress limitations of all members are  $\pm 25$  ksi and the displacements of the free nodes in all directions had to be less than  $\pm 2$  in. The results obtained by SAR for solving 10-bar truss are presented in Table 1.



Fig. 2: The 10-bar truss design problem

	1					0	
 Design Variables	HPSO	BB-BC	ABC	SCA	MSCA	HHS	SAR
A1	30	33.5	33.5	26.5	33.5	33.5	33.5
A2	1.62	1.62	1.62	2.62	1.62	1.62	1.62
A3	22.9	22.9	22.9	26.5	22.9	22.9	22.9
A4	13.5	14.2	14.2	18.8	14.2	14.2	14.2
A5	1.62	1.62	1.62	1.62	1.62	1.62	1.62
A6	1.62	1.62	1.62	2.38	1.62	1.62	1.62
A7	7.97	7.97	7.97	11.5	7.97	7.97	7.97
A8	26.5	22.9	22.9	22	22.9	22.9	22.9
A9	22	22.00	22	19.9	22	22	22
A10	1.8	1.62	1.62	1.8	1.62	1.62	1.62
Best	5531.98	5490.74	5490.74	5633.44	5490.74	5490.74	5490.74
CV	None						
Average	NA	5494.17	5510.35	5838.26	5492.64	5493.49	5490.757
STD	NA	12.42	NA	220.39	2.42	10.46	0.138
MNSA	50000	8694	25800	10000	10000	5000	10000

Table 1. Comparison of the statistical results for the 10-bar truss design

Also, this truss was designed by other optimization methods, including a heuristic particle swarm optimization (HPSO) [22], big bang-big crunch optimization (BB-BC) [15], artificial bee colony algorithm (ABC) [23], sine-cosine algorithm (SCA), a modified sine-cosine algorithm (MSCA) [24], and a hybrid harmony search algorithm (HHS) [25] the results of which are reported in this table. According to the table, most of the algorithms are able to find the optimum weight of the truss (5490.74 lb). SAR outperforms the other algorithms in terms of average and standard deviation and these values are significantly better than the others. SAR found the best design in most of 50 independent runs. These results indicate the high ability of SAR for avoiding the local minima and finding the global minimum. The statistical results of HHS and BB-BC achieved after 5000 and 8694 structural analyses are lower than that of SAR. The maximum number of structure analysis of SAR is set to 10000. The average of designs obtained by BB-BC, ABC, SCA, MSCA and HHS were 3.4, 347.5, 19.6, 1.9 and 2.7 lbs heavier than that of SAR, respectively.

Fig. 3 shows the convergence curves of the best run and the average of 50 independent runs of SAR for the 10-bar truss. SAR found the best design after only 3554 structural analyses and the weight of worst design obtained by SAR is

5491.717 lb. The worst design of SAR is lighter than the weights of average designs of the compared algorithms.



Fig. 3: The convergence curve of SAR for 10-bar truss

#### 4.2 Spatial 25-bar truss design

Fig. 4 shows the spatial 25-bar truss design. Many optimization methods were applied to solve this well-known optimization problem. The elastic modulus and the material density of all members are  $10^4$  ksi and 0.1 lb/in<sup>3</sup>, respectively. This truss is subjected to the two loading conditions presented in Table 2. Due to the symmetry of the structure, the 25 members of the truss are divided into 8

groups, as follows: (1) A1, (2) A2-A5, (3) A6-A9, (4) A10-A11, (5) A12-A13, (6) A14-A17, (7) A18-A21, (8) A22-A25.



Fig. 4: The 25-bar truss design problem

The set of allowable section areas used for this problem is L=[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.6, 2.8, 3.0,

3.2, 3.4] (in<sup>2</sup>). The displacements of the free nodes in both directions are limited to  $\pm 0.35$  in and the allowable stress of each group is  $\pm 40$  ksi.

Table 2. Nodal loading (ksi) for the spatial 25-bar truss

Node		Case 1			Case 2	
	$P_x$	$P_y$	$P_z$	$P_x$	$P_y$	$P_z$
1	0	20	-5	1	10	-5
2	0	-20	-5	0	10	-5
3	0	0	0	0.5	0	0
6	0	0	0	0.5	0	0

SAR is compared with eight different methods including a heuristic particle swarm optimization (HPSO) [22], colliding bodies optimization (CBO) [17], enhanced colliding bodies optimization (ECBO) [17], sine-cosine algorithm (SCA) [24], a modified sine-cosine algorithm (MSCA) [24], artificial bee colony algorithm (ABC) [23], big bang–big crunch optimization (BB-BC) [15] and a hybrid harmony search algorithm (HHS) [25]. The optimization results of these algorithms are summarize in Table 3.

Table 3. Comparison of the statistica	al results for the 25-bar truss d	esign problem
---------------------------------------	-----------------------------------	---------------

	-						<b>U</b>		
Element groups	HPSO	CBO	ECBO	SCA	MSCA	ABC	BB-BC	HHS	SAR
1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1
2	0.3	0.3	0.3	0.1	0.3	0.3	0.3	0.3	0.3
3	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4
4	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
5	2.1	2.1	2.1	1.4	2.1	2.1	2.1	2.1	2.1
6	1	1	1	1.2	1	1	1	1	1
7	0.5	0.5	0.5	0.7	0.5	0.5	0.5	0.5	0.5
8	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4
Best	484.85	484.85	484.85	486.29	484.85	484.85	484.85	484.85	484.85
CV	None	None	None						
Average	NA	486.87	485.89	491.17	484.94	484.94	485.1	484.946	484.88
STD	NA	NA	NA	2.55	0.22	NA	0.44	0.365	0.068
MNSA	25000	20000	20000	5000	5000	24250	9090	5000	5000

All the algorithms except SCA converged to the best weight of 484.85 lb. Although the number of structural analyses of the compared algorithms are equal or more than that of SAR, the average of results achieved by SAR is lighter than those of others. So, it can be concluded that convergence rate of SAR is much faster than the others. Also, standard deviation of SAR is remarkably lower compared to the others and it is more stable than them.

According to these results, SAR is highly efficient in terms of finding optimized weight and convergence rate. The convergence curves of the best run and the average of 50 independent runs of SAR for the 25-bar truss are shown in Fig. 5. SAR converges to optimum design after only 1369 structural analyses and the weight of worst design obtained by SAR is 485.049 lb.



Fig. 5: The convergence curve of SAR for 25-bar truss

## 2.5 Planar 52-bar truss design

The 52-bar truss design problem is shown in Fig. 6. This truss including 52 members is classified into 12 groups: (1) A1–A4, (2) A5–A10, (3) A11–A13, (4) A14–A17, (5) A18–A23, (6) A24–A26, (7) A27–A30, (8) A31–A36, (9) A37–A39, (10) A40–A43, (11) A44–A49, and (12) A50–A52. The modulus of elasticity and the material density of all members are 207 GPa and 7860 kg/m<sup>3</sup>, respectively. The stress limitations of all members are  $\pm 180$  MPa and there is no displacement limitation. The set of cross sectional areas used in this design problem is shown in Table 4. The truss is subjected to 100 KN in the positive x-direction and 200 KN the positive y-direction at nodes 17, 18, 19 and 20.



This design problem has been previously investigated for discrete design variables using some optimization algorithms including a heuristic particle swarm optimization (HPSO) [22], a particle swarm ant colony optimization (DHPSACO) [13], an adaptive elitist differential evolution (AEDE) [26], sine-cosine algorithm (SCA) [24], a modified sine-cosine algorithm (MSCA) [24] and colliding bodies optimization (CBO) [27].

Table 4. The list of allowable cross sectional areas										
No.	in <sup>2</sup>	mm <sup>2</sup>	No.	in <sup>2</sup>	mm <sup>2</sup>					
1	0.111	71.613	33	3.84	2477.414					
2	0.141	90.968	34	3.87	2496.769					
3	0.196	126.451	35	3.88	2503.221					
4	0.25	161.29	36	4.18	2696.769					
5	0.307	198.064	37	4.22	2722.575					
6	0.391	252.258	38	4.49	2896.768					
7	0.442	285.161	39	4.59	2961.284					
8	0.563	363.225	40	4.8	3096.768					
9	0.602	388.386	41	4.97	3206.445					
10	0.766	494.193	42	5.12	3303.219					
11	0.785	506.451	43	5.74	3703.218					
12	0.994	641.289	44	7.22	4658.055					
13	1	645.16	45	7.97	5141.925					
14	1.228	792.256	46	8.53	5503.215					
15	1.266	816.773	47	9.3	5999.988					
16	1.457	939.998	48	10.85	6999.986					
17	1.563	1008.385	49	11.5	7419.34					
18	1.62	1045.159	50	13.5	8709.66					
19	1.8	1161.288	51	13.9	8967.724					
20	1.99	1283.868	52	14.2	9161.272					
21	2.13	1374.191	53	15.5	9999.98					
22	2.38	1535.481	54	16	10322.56					
23	2.62	1690.319	55	16.9	10903.2					
24	2.63	1696.771	56	18.8	12129.01					
25	2.88	1858.061	57	19.9	12838.68					
26	2.93	1890.319	58	22	14193.52					
27	3.09	1993.544	59	22.9	14774.16					
28	3.13	2019.351	60	24.5	15806.42					
29	3.38	2180.641	61	26.5	17096.74					
30	3.47	2238.705	62	28	18064.48					
31	3.55	2290.318	63	30	19354.8					
32	3.63	2341.931	64	33.5	21612.86					

The convergence curves of the best run and the average of 50 independent runs of SAR for the 52-bar truss are shown in Fig. 7. SAR converges to optimum design after only 5913 structural analyses and the weight of worst design obtained by SAR is 1903.94 kg, which is lower than the average results of the others.



Fig. 7: The convergence curve of SAR for 52-bar truss

The results of these optimization techniques and SAR are summarized in Table 5. According to this table, the best design found by CBO and DHPSACO did not satisfy the constraints. AEDE, MSCA and SAR found the lightest weight of the 52-bar truss that meet the constraints. To achieve these results, they required 3402, 10000 and 8000 structural analyses, respectively. AEDE has the fastest

convergence rate among the compared algorithms and SAR is the second fast algorithm for these problems. SAR outperforms the others in terms of average and standard deviation of the results.

Table 5. Comparison of the statistical results for the 52-bar truss design problem

Element groups	HPSO	DHPSACO	AEDE	SCA	MSCA	CBO	SAR
1	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055
2	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288
3	363.225	494.193	494.193	363.225	494.193	388.386	494.193
4	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219
5	939.998	1008.385	939.998	1045.159	939.998	939.998	939.998
6	494.193	285.161	494.193	506.451	494.193	506.451	494.193
7	2238.705	2290.318	2238.705	2238.705	2238.705	2238.705	2238.705
8	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385
9	388.386	388.386	494.193	641.289	494.193	506.451	494.193
10	1283.868	1283.868	1283.868	1690.319	1283.868	1283.868	1283.868
11	1161.288	1161.288	1161.288	1045.159	1161.288	1161.288	1161.288
12	792.256	506.451	494.193	645.16	494.193	506.451	494.193
Best	1905.49	1904.83	1902.605	1947.535	1902.605	1899.35	1902.605
CV	None	2.70E-02	None	None	None	4.85E-04	None
Average	NA	N/A	1906.735	1958.564	1904.129	1963.12	1902.834
STD	NA	N/A	6.679	9.37	2.67	106.01	0.353
MNSA	100000	5300	3402	10000	10000	3840	8000

#### 4.4 Spatial 72-bar truss design

The spatial 72-bar truss shown in Fig. 8, is a four level tower. The material density and the elastic modulus are 0.1 lb/in<sup>3</sup> and  $10^4$  ksi, respectively. The members are divided into 16 groups using symmetry of the truss, as follows: (1)A1-A4, (2) A5-A12, (3) A13-A16, (4) A17-A18, (5) A19-A22, (6) A20-A30, (7) A31-A34, (8) A35-A36, (9) A37-A40, (10) A41-A48, (11) A49-A52, (12) A53-A54, (13) A55-A58, (14) A59-A62, (15) A63-A70, (16) A71-A72. The displacements of the free nodes in both directions are limited to  $\pm 0.25$  in.

The two variants of this truss design problem are as follows:

Case1: The allowable cross sectional areas are selected from the following list:

L = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2] (in<sup>2</sup>).

Case 2: The allowable cross sectional areas are selected from the list presented in Table 4. The members are subjected to the stress limits of  $\pm 25$  ksi. The two loading conditions shown in Table 6, are considered for designing this truss.



Fig. 8: The 72-bar truss design problem

Table 6. Nodal loading (ksi) for the spatial 72-bar truss

Node	C	Case 1		Case 2			
	$P_x$	$P_y$	$P_x$	$P_y$	$P_x$	$P_y$	
17	5	5	-5	0	0	-5	
18	0	0	0	0	0	-5	
19	0	0	0	0	0	-5	
20	0	0	0	0	0	-5	

The results obtained by SAR and several optimization algorithms for 72-bar truss design variant 1 are presented in Table 7.

 Table 7. Comparison of the statistical results for the 72-bar truss design problem (case 1)

Element	инс	HDSO	WCA	MBA	IMBA	SAP
groups	groups		WCA	MDA	IMDA	SAK
1	1.9	2.1	1.9	1.9	1.9	1.9
2	0.5	0.6	0.5	0.5	0.5	0.5
3	0.1	0.1	0.1	0.1	0.1	0.1
4	0.1	0.1	0.1	0.1	0.1	0.1
5	1.4	1.4	1.4	1.4	1.4	1.4
6	0.5	0.5	0.5	0.5	0.5	0.5
7	0.1	0.1	0.1	0.1	0.1	0.1
8	0.1	0.1	0.1	0.1	0.1	0.1
9	0.5	0.5	0.5	0.5	0.5	0.5
10	0.5	0.5	0.5	0.5	0.5	0.5
11	0.1	0.1	0.1	0.1	0.1	0.1
12	0.1	0.1	0.1	0.1	0.1	0.1
13	0.2	0.2	0.2	0.2	0.2	0.2
14	0.6	0.5	0.6	0.6	0.6	0.6
15	0.4	0.3	0.4	0.4	0.4	0.4
16	0.6	0.7	0.6	0.6	0.6	0.6
Best	385.54	388.94	385.54	385.54	385.54	385.54
CV	None	None	None	None	None	None
Average	386.04	NA	387.665	385.842	385.765	385.596
STD	1.155	NA	1.62	0.55	0.41	0.295
MNSA	5000	12500	50000	50000	50000	15000

These algorithms include a hybrid harmony search algorithm (HHS) [25], a heuristic particle swarm optimization (HPSO) [22], water cycle algorithm (WCA) [28], mine blast algorithm (MBA) [28] and an improved version of MBA (IMBA) [28]. From Table 7, it can be seen that all the algorithms except HPSO converged to the best design of the truss. HHS requires significantly less number of structural analyses than all the compared algorithms to find the optimum design and the fastest algorithm. The average design weight of SAR is 385.616 lb with a standard deviation of 0.296 lb. These values are lower than those of the other algorithms. WCA, MBA and IMBA required 50000 structural analyses, but SAR only needed 15000 structural analyses to achieve these results. Hence, SAR significantly outperforms them in this problem.



Fig. 9: The convergence curve of SAR for 72-bar truss (case 1)

Fig. 9 illustrates the convergence curves of the best run and the average of 50 independent runs of SAR for the 72-bar truss (case 1). SAR converges to optimum design after only 7371 structural analyses and the weight of worst design obtained by SAR is 386.948 lb.

Table 8 Presents the results of SAR and other methods including colliding bodies optimization (CBO) [17], enhanced colliding bodies optimization (ECBO) [17], an improved ray optimization (IRO) [29], sine-cosine algorithm (SCA) [24], a modified sine-cosine algorithm (MSCA) [24], water cycle algorithm (WCA) [28], mine blast algorithm (MBA) [28] and an improved version of MBA (IMBA) [28] for 72-bar truss design variant 2. According to this table, ECBO, IRO, MSCA, WCA, IMBA and SAR successfully found the best design of the truss. The average and standard deviation of the results obtained by SAR are better than those of the compared algorithms, while the number of structural analyses of SAR (10000) is equal or lower than the others. Therefore, SAR has the fastest convergence rate among the compared algorithms and outperforms them in terms of average and standard deviation. After SAR, the results of IMBA are better than the others, but IMBA requires five times more structural analyses than the proposed algorithm (50000 structural analyses for IMBA and 10000 structural analyses for SAR).

The convergence curves of the best run and the average of 50 independent runs of SAR for the 72-bar truss (case 2) are displayed in Fig. 10. SAR converges to optimum design after only 7125 structural analyses and the weight of worst design obtained by it is 392.39 lb.



Fig. 10: The convergence curve of SAR for 72-bar truss (case 2)

								· /	
Element groups	CBO	ECBO	IRO	SCA	MSCA	WCA	MBA	IMBA	SAR
1	2.13	1.99	1.99	3.13	1.99	1.99	1.8	1.99	1.99
2	0.563	0.563	0.442	0.563	0.563	0.442	0.602	0.442	0.442
3	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111
4	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111
5	1.228	1.228	1.228	1.266	1.228	1.228	1.266	1.228	1.228
6	0.442	0.442	0.563	0.442	0.442	0.563	0.563	0.563	0.563
7	0.141	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111
8	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111
9	0.442	0.563	0.563	0.563	0.563	0.563	0.442	0.563	0.563
10	0.563	0.563	0.563	0.442	0.563	0.563	0.442	0.563	0.563
11	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111
12	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111
13	0.196	0.196	0.196	0.196	0.196	0.196	0.196	0.196	0.196
14	0.563	0.563	0.563	0.563	0.563	0.563	0.563	0.563	0.563
15	0.391	0.391	0.391	0.442	0.391	0.391	0.442	0.391	0.391
16	0.563	0.563	0.563	0.442	0.563	0.563	0.602	0.563	0.563
Best	391.23	389.334	389.334	402.96	389.33	389.334	390.739	389.33	389.334
CV	None	None	None	None	None	None	None	None	None
Average	456.69	391.59	408.17	412.62	390.67	389.941	395.432	389.82	389.649
STD	NA	NA	NA	11.73	1.063	1.43	3.04	0.84	0.604
MNSA	20000	20000	20000	10000	10000	50000	50000	50000	10000

Table 8. Comparison of the statistical results for the 72-bar truss design problem (case 2)

# 5. Conclusion

In this paper, a novel metaheuristic algorithm namely search and rescue optimization algorithm (SAR) is presented to sizing optimization of truss structures with discrete variables. SAR is developed based on mimicking the explorations behavior of humans during search and rescue operations.

The proposed algorithm consists of two phases including social phase and individual phase and the implementation of it is relatively simple. The results of the tests done in this paper have shown that combining these two phases along with the use of memory lead to a balance between exploration and exploitation processes in SAR.

Four truss design problems are considered to evaluate the performance and efficiency of SAR. The comparisons demonstrate that the best averages and standard deviations of results were obtained by SAR for all the studied problems and the proposed algorithm outperforms the other compared optimization algorithms in terms of finding the optimized weight of the truss (accuracy). Also, SAR requires lower number of structural analyses in comparison with most of the compared algorithms. According to the numerical results, it can be concluded that SAR is a very efficient and robust algorithm for designing truss structures with discrete variables.

Future works may focus on hybrid versions of SAR with other well-known optimization algorithms such as PSO, GA and ACO.

#### References

[1] Deb, K. "Optimization for engineering design: Algorithms and examples", PHI Learning Pvt. Ltd. 2012.

[2] Yang, X. S. "A new metaheuristic bat-inspired algorithm". Nature inspired cooperative strategies for optimization (NICSO 2010). Springer 2010.

[3] Kaveh, A., Azar, B. F. & Talatahari, S. "Ant colony optimization for design of space trusses". Int. J. Space. Struct., vol. 23, 2008, p. 167-181.

[4] Cheng, M. Y. & Prayogo, D. "A novel fuzzy adaptive teaching–learning-based optimization (FATLBO) for solving structural optimization problems". Eng. Comput., vol. 33, 2017, p. 55-69.

[5] Kennedy, J., Eberhart, R. "Particle swarm optimization". In: NETWORKS, I. I. C. O. N. (ed.). Perth, WA, Australia: IEEE 1995.

**[6]** Du, K. L. & Swamy, M. "Particle swarm optimization". Search and optimization by metaheuristics. Springer 2016.

[7] Kaveh, A. & Hosseini, P. "A simplified dolphin echolocation optimization method for optimum design of trusses". Int. J. Optim. Civil. Eng., vol. 4, 2014, p. 381-397.
[8] Kaveh, A., Hosseini Vaez, S. & Hosseini, P. "Simplified dolphin echolocation algorithm for optimum design of frame". Smart Struct Syst, vol. 21, 2018, p. 321-333.

[9] Lee, K. S., Geem, Z. W., Lee, S. H. & Bae, K. W. "The harmony search heuristic algorithm for discrete structural optimization". Eng. Optimiz., vol. 37, 2005, p. 663-684.

[10] Kaveh, A. & Talatahari, S. "Charged system search for optimal design of frame structures". Appl. Soft. Comput., vol. 12, 2012, p. 382-393.

[11] Kaveh, A., Vaez, S. H. & Hosseini, P. "Performance of the Modified Dolphin Monitoring Operator for Weight Optimization of Skeletal Structures". Period. Polytech-CIV., vol., 2017, p.

**[12]** Kaveh, A., Bakhshpoori, T. & Azimi, M. "Seismic optimal design of 3D steel frames using cuckoo search algorithm". The Structural Design of Tall and Special Buildings, vol. 24, 2015, p. 210-227.

[13] Kaveh, A. & Talatahari, S. "A particle swarm ant colony optimization for truss structures with discrete variables". J. Constr. Steel. Res., vol. 65, 2009, p. 1558-1568.

[14] Lamberti, L. "An efficient simulated annealing algorithm for design optimization of truss structures". Comput. Struct., vol. 86, 2008, p. 1936-1953.

[15] Camp, C. V. & Bichon, B. J. "Design of space trusses using ant colony optimization". J. Struct. Eng., vol. 130, 2004, p. 741-751.

[16] Degertekin, S. & Hayalioglu, M. "Sizing truss structures using teaching-learning-based optimization". Comput. Struct., vol. 119, 2013, p. 177-188.

[17] Kaveh, A. & Ghazaan, M. I. "A comparative study of CBO and ECBO for optimal design of skeletal structures". Comput. Struct., vol. 153, 2015, p. 137-147.

**[18]** Stolpe, M. "Truss optimization with discrete design variables: a critical review". Struct. Multidiscipl. Optim., vol. 53, 2016, p. 349-374.

[19] Shabani, A., Asgarian, B., Gharebaghi, S. A., Salido, M. A. & Giret, A. " A New Optimization Algorithm Based on Search and Rescue Operations ". Math. Probl. Eng., vol. 2019, 2019.

**[20]** Astm. "F1847: Standard Guide for Demonstrating Minimum Skills of Search and Rescue Dogs". West Conshohocken, PA, USA: ASTM International 2012.

[21] Management, K. E. "SAR Field Search Methods: Search Tecniques Used by Trained Teams in the Field" [Online]. Kentucky Emergency Management. Available: https://kyem.ky.gov/Who%20We%20Are/Documents/SAR %20Field%20Search%20Methods.pdf [Accessed May 07 2019].

[22] Li, L., Huang, Z., Liu, F. & Wu, Q. "A heuristic particle swarm optimizer for optimization of pin connected structures". Comput. Struct., vol. 85, 2007, p. 340-349.

[23] Sonmez, M. "Discrete optimum design of truss structures using artificial bee colony algorithm". Struct. Multidiscipl. Optim., vol. 43, 2011, p. 85-97.

**[24]** Gholizadeh, S. & Sojoudizadeh, R. "Modified Sine-Cosine Algorithm for Sizing Optimization of Truss Structures with Discrete Design Variables". Iran University of Science & Technology, vol. 9, 2019, p. 195-212.

[25] Cheng, M.Y., Prayogo, D., Wu, Y. W. & Lukito, M. M. "A Hybrid Harmony Search algorithm for discrete sizing optimization of truss structure". Automat. Constr., vol. 69, 2016, p. 21-33.

[26] Ho Huu, V., Nguyen Thoi, T., Vo Duy, T. & Nguyen Trang, T. "An adaptive elitist differential evolution for optimization of truss structures with discrete design variables". Comput. Struct., vol. 165, 2016, p. 59-75.

[27] Kaveh, A. & Mahdavi, V. "Colliding bodies optimization method for optimum discrete design of truss structures". Comput. Struct., vol. 139, 2014, p. 43-53.

[28] Sadollah, A., Eskandar, H., Bahreininejad, A. & Kim, J. H. "Water cycle, mine blast and improved mine blast

algorithms for discrete sizing optimization of truss structures". Comput. Struct., vol. 149, 2015, p. 1-16.

**[29]** Kaveh, A., Ghazaan, M. I. & Bakhshpoori, T. "An improved ray optimization algorithm for design of truss structures". Period. Polytech-CIV., vol. 57, 2013, p. 97–112-97–112.