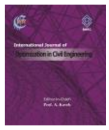
INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2021; 11(2): 177-194



SYMBIOTIC ORGANISMS SEARCH AND HARMONY SEARCH ALGORITHMS FOR DISCRETE OPTIMIZATION OF STRUCTURES

S. Talatahari[1*, †], V. Goodarzimehr[2] and S. Shojaee[2]
[1]*Department of Civil Engineering, The university of Tabriz, Tabriz, Iran*
[2]*Engineering Faculty, Shahid Bahonar University of Kerman, Kerman, Iran*

ABSTRACT

In this work, a new hybrid Symbiotic Organisms Search (SOS) algorithm introduced to design and optimize spatial and planar structures under structural constraints. The SOS algorithm is inspired by the interactive behavior between organisms to propagate in nature. But one of the disadvantages of the SOS algorithm is that due to its vast search space and a large number of organisms, it may trap in a local optimum. To fix this problem Harmony search (HS) algorithm, which has a high exploration and high exploitation, is applied as a complement to the SOS algorithm. The weight of the structures' elements is the objective function which minimized under displacement and stress constraints using finite element analysis. To prove the high capabilities of the new algorithm several spatial and planar benchmark truss structures, designed and optimized and the results have been compared with those of other researchers. The results show that the new algorithm has performed better in both exploitation and exploration than other meta-heuristic and mathematics methods.

Keywords: discrete variables, symbiotic organisms search; harmony search; size optimization; structural optimization; truss structures; meta-heuristic algorithm.

Received:10 January 2021; Accepted: 20 May 2021

1. INTRODUCTION

So far, a lot of optimization methods, are proposed by researchers. According to new human needs and engineering problems become more complicated, engineers need more powerful tools to optimize engineering problems. Classical optimization methods, despite good performance, face obstacles and issues. For example, classical optimization methods, in

*Corresponding author: Department of Civil Engineering, University of Tabriz, Tabriz, Iran
†E-mail address: siamak.talat@gmail.com (S. Talatahari)

addition to the derivability and continuity of the objective function and the constraints of the problem, may not reach the global optimization, especially when the starting point is close to local optimization. Also, the solution time increases dramatically with an increasing number of problem variables.

In recent decades, researchers developed new optimization methods to solve the mentioned problems. These methods are inspiring by nature and the law of physics, which are so-called metaheuristic methods, such as: Genetic Algorithm (GAs) [1], which model the process of natural evolution.The Harmony Search algorithm (HS) [2] is derived from modeling and simulating the process that a composer goes through to harmonize a piece of music. Ant Colony Optimization (ACO) [3] is inspired by the collective behavior of ants. Big Bang-Big Crunch (BB-BC) [4] Inspired by the theory of the evolution of the universe, it is called the Big Bang-Big Density Theory. Particle Swarm Optimization [5] In the PSO algorithm, the members of the population of the answers are directly related to each other and solve the problem by exchanging information with each other and recalling good memories of the past. Teaching-Learning-Based Optimization (TLBO) [6] Inspired by the learning and teaching process. Charge System Search (CSS) [7] is one of the newest meta-exploration algorithms that searches the problem space using the laws governing electrical physics. Symbiotic Organisms Search (SOS) [8] Simulates the interactive behavior between creatures in nature.

The metaheuristic methods have a relatively similar process to achieve the optimal solution. In most of these methods, the algorithm starts by generating several random answers in the feasible space. Then they move to the optimal point by performing a series of processes. Since these algorithms are based on population and in the search process moving towards the optimal answer, they also contribute to random search. Therefore, they are more likely to arrive at the overall optimal results than classical methods.

Recently, to improve the performance of metaheuristic algorithms, some researchers have tried to provide a more effective method by hybridizing these algorithms together, such as: A Particle Swarm Ant Colony Optimization (PSACO) [9]. Hybrid Algorithm of Harmony Search, Particle Swarm and Ant Colony [10]. A Hybrid Harmony Search algorithm [11]. PSO and Convex Approximation [12]. Particle Swarm Optimization and Genetic Algorithm [13]. Teaching-Learning-Based Optimization and Harmony Search [14]. One of the good advantages of hybrid algorithms is that they cover each other's shortcomings. Because many of these algorithms have some defects, so by hybridizing them, we can reach a better algorithm. According to previous researches, the performance of hybrid algorithms has been better than the initial algorithms.

In this paper, by hybridizing SOS and HS algorithms, a new hybrid algorithm has been developed, which is called (SOSHS). The SOS algorithm can easily explore the entire search space. But, due to the possibility of being trapped in the local optimal, the HS algorithm is used, which has a high power of global search. To demonstrate the capabilities of this new algorithm, we have designed and optimized several truss structures. The results show that this algorithm performs better than other meta-exploration methods.

2. DISCRETE OPTIMUM DESIGN PROBLEM OF TRUSS STRUCTURES

A structural optimization problem with discrete design variables is known as a nonlinear problem with nonlinear constraints. To optimizing the size of a truss structure, the cross-sectional area of the truss members is the design variable of the problem. The objective function of the problem is the weight of the truss structure. In discrete size optimization problems, the main task is to select the optimal members' section from a list of standards sections. So that the weight of the structure minimized while meeting the design constraints. The problem of optimal design for discrete variables expresses as follow:

$$
\begin{aligned}
& && A=[A_1, A_2, ..., A_n] \\
& \text{Subject to} && A \in D_i, D_i=\{d_{i,1}, d_{i,2}, ..., d_{i,r(i)}\} \\
& \text{To minimize} && W=\sum_{e=1}^{N_m} \gamma_e . l_e . A_e \\
& && \sigma^l < \sigma_e < \sigma^u \\
& && \delta^l < \delta_e < \delta^u
\end{aligned} \tag{1}
$$

where A is the vector containing the design variables; D_i is an allowable set of discrete values for the design variable A_i ; n is the number of design variables or the number of member groups; $r(i)$ is the number of available discrete values for the i^{th} design variable; W is the weight of truss; γ_e is the unit weight; l_e is the length of each member; N_m is the number of structure members; This minimum design also has to satisfy the constraints on each member stress σ_e and deflection δ_c at each connection c. To control these constraints, a penalty method can be used as:

$$\text{if} \quad \sigma^l < \sigma_e < \sigma^u \quad \text{then} \quad \varphi_\sigma^e = 0 \tag{2}$$

$$\text{if} \quad \sigma_e < \sigma^l \quad \text{or} \quad \sigma_e > \sigma^u \quad \text{then} \quad \varphi_\sigma^e = \left| \frac{\sigma_e - \sigma^{l,u}}{\sigma^{l,u}} \right| \tag{3}$$

$$\varphi_\sigma^k = \sum_{e=1}^{N_m} \varphi_\sigma^e \tag{4}$$

$$\text{if} \quad \delta^l \le \delta_{c(x,y.z)} \le \delta^u \quad \text{then} \quad \varphi_{\delta(x,y,z)}^c = 0 \tag{5}$$

$$\text{if} \quad \delta_{c(x,y.z)} < \delta^l \quad \text{or} \quad \delta_{c(x,y.z)} > \delta^u \quad \varphi_{\delta(x,y,z)}^c = \left| \frac{\delta_{c(x,y.z)} - \delta^{l,u}}{\delta^{l,u}} \right| \tag{6}$$

$$\varphi_\delta^k = \sum_{c=1}^{N_m} [\varphi_{\delta x}^c + \varphi_{\delta y}^c + \varphi_{\delta z}^c] \tag{7}$$

The final penalty function ψ^k for a truss structure is as:

$$\psi^k = (1 + \varphi_\sigma^k + \varphi_\delta^k)^\varepsilon \tag{8}$$

where ε is a positive penalty coefficient. The value of penalized weight can be defined as:

$$F^k = \psi^k . w^k \tag{9}$$

3. HEURISTIC SYMBIOTIC ORGANISMS SEARCH AND HARMONY SEARCH FOR TRUSS STRUCTURES

3.1 Review of continuous symbiotic organisms search algorithm

The Symbiotic Organisms Search algorithm, at first introduced by Cheng and Prayogo [8], simulates the interaction behavior between organisms in nature. Organisms rarely live alone because they need each other to provide food and even survival. This relationship based on dependence is called symbiotic. Symbiotic relationships may be compulsory, meaning that two beings are dependent on each other for survival, or maybe voluntary, meaning that there is an unnecessary connection between the two beings that is beneficial to both. In symbiotics, mutualism, commensalism, and parasitism are the most common relationships in nature. In a mutual relationship, symbiotic benefits both species. In a commensalism relationship, symbiotic is beneficial to one species and does not affect the other. In a parasitic relationship, symbiotic is to the advantage of one species and the detriment of the other. In the proposed algorithm, like other population-based algorithms, to finding the global optimal answer of a candidate population, the answers are used repeatedly in the promising areas of the search space to generate new answers for the next iteration. In the SOS algorithm, imitation of the biological interaction between two organisms of the ecosystem prevails in generating a new answer. Three phases are introduced that are similar to the biological event interaction model. Mutualism phase, Commensalism phase, and Parasitism phase. Each organism interacts with another organism in all stages. This process is repeated to meet the final criteria. Details of the different stages are as follows.

3.1.1 Mutualism phase

At this stage, which is an imitation of Mutualism, X_i is an organism of the ith organism of the existing ecosystem. Then another creature X_j, is randomly selected from the system to interact with X_i. The answers of the new candidate for X_i and X_j are calculated based on the symbiotic between the two organisms.

$$X_{inew} = X_i + rand(0,1) \times (X_{best} - Mutual\_Vector \times BF_1) \tag{10}$$

$$X_{jnew} = X_j + rand(0,1) \times (X_{best} - Mutual\_Vector \times BF_2) \tag{11}$$

$$Mutual\_Vector = \frac{X_i + X_j}{2} \tag{12}$$

In Eqs (10) and (11), the round (0,1) is a vector of random numbers, and the benefit factors BF_1 and BF_2 are randomly selected from 1 and 2. Equation (12) is a mutual vector that represents the relationship between X_i and X_j.

3.1.2 Commensalism phase

Like the Mutualism phase, in the Commensalism phase, one X_j is randomly selected from the ecosystem for X_i to interact. The answer of the new candidate X_i is calculated based on the symbiotic of the existing X_j and X_i, which is modeled in Eqs (13).

$$X_{inew} = X_i + rand(-1,1) \times (X_{best} - X_j) \tag{13}$$

In Eqs (13), $(X_{best} - X_j)$ represents the benefit provided by X_J to help X_i increase its survival advantage in the ecosystem.

3.1.3 Parasitism phase

In this phase, organism X_i plays a role similar to the anopheles mosquito by creating an artificial parasite called "Parasite_Vector" and propagating the organism X_i in the search space. The Parasite_Vector created using a random number to modify the randomly selected dimensions. The X_j organism is randomly selecting from the ecosystem and acts as the host of the Parasite_Vector. The Parasite_Vector tries to replace X_j in the ecosystem. Both are then evaluating for compatibility. If the value of the Parasite_Vector is greater than organism w, it will kill organism X_j and take its place in the ecosystem. But if the organism X_j is more compatible, the organism X_j will be safe against the Parasite_Vector, and the Parasite_Vector can no longer live in the ecosystem.

3.2 Review of Harmony search algorithm

The Harmony Search algorithm is inspired by music to achieve the best answer with the aim of coordinating the harmonies. Trying to find this harmony in music is like finding the optimal conditions in the optimization process. In fact, the harmonic search algorithm is the best strategy for transforming qualitatively examined processes into quantitative and tangible optimization processes. A process with some ideal rules that will result in turning beautiful pieces of music into a suitable solution for solving various optimization problems.

Step 1: initialization; In the first step, the HS algorithm has several parameters that are inquired to be adjusted to solve the optimization problem. Harmony Memory (HM), Harmony Memory Size (HMS), Harmony Memory Consideration Rate (HMCR), and Pitch Adjusting Rate (PAR). In this section, we must generate a population and store it in the HM.

$$HM = \begin{bmatrix} X^1 \\ . \\ . \\ X^{HMS} \end{bmatrix}_{HMS \times mg} \tag{14}$$

Step 2: Initialize a new harmony from the HM: The HMCR is within [0,1] and used for considering the HM and choosing a new vector from the previous value. And (1-HMCR) sets the rate of randomly choosing one value from a possible range of values.

$$X_i^K = \begin{cases} select\ from \left\{X_i^1, X_i^2, ..., X_i^{HMS}\right\} & w.p. HMCR \\ select\ from\ the\ possible\ range & w.p.\ \ (1-HMCR) \end{cases} \tag{15}$$

Step 3: Updating the harmony memory: If a new harmony vector is better than the worst harmony in the HM, judged in terms of the objective function value, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

Step 4: Terminating criterion controlling: Repeat Steps 2 and 3 until the terminating criteria are satisfied.

4. A DISCRETE HYBRID SOS AND HS ALGORITHM

The SOSHS hybrid algorithm consists of SOS and HS algorithms. First, the SOS algorithm will be implemented, if the stored optimal answers do not satisfy the convergence condition. In the second step, the HS algorithm will be applied, which has a high search power. If the prepared answers do not meet the final criteria, the process will be repeated until the best optimal answer is obtained. The discrete optimization process is as follows.

Discretization is a mapping from continuous to discrete space in which the search space is divided into a limited number of intervals. So far, a lot of researches have been done in the field of discrete analysis of structures. There are several methods for discretization, including Equal Frequency Discretization (EFD), Equal Width Discretization (EWD), Random Discretization (RD). In the EWD method, by considering the upper and lower bounds, the amplitude is limited to subdomains with specified width. All of the above methods given a good answer to problems with a small search space, but when the search space is significantly large. In order to achieve more appropriate solutions, it is necessary to use the above methods repeatedly, with discretization intervals that gradually converge.

After performing successive iterations, the optimal values obtained from different discretized sets gradually converge towards a single answer. In this case, the calculated standard deviation tends to zero. Therefore, the following conditions can be considered as stopping criteria.

$$\sigma_i^* = \varepsilon \qquad\qquad i = 1, 2, ..., n \tag{16}$$

Since σ_i^* is the standard deviation and ε is a very small number.

The hybrid optimization procedure including the following steps:

Step 1: As conventional SOS algorithm:

1- initialize a population of organisms; these organisms are like harmony in the HS algorithm.
2- find the best organism (Xbest) from the initial ecosystem.

Step 2: Mutualism phase:

1- select an organism X_j that is opposite to X_i.
2- Determine mutual relationship vector (Mutual_Vector) and benefit factor (BF)
3- Modify organism X_i and X_j based on their mutual relationship
4- If the new organism has better conditions than the previous one, upgrade it in the ecosystem.

Step 3: Commensalism Phase:

1- select an organism X_j that is opposite to X_i.
2- Modify organism X_i according to organism X_j
3- If the modified organism has better conditions than the previous one, upgrade it in the ecosystem.

Step 4: Parasitism Phase:

1- select an organism Xj that is opposite to Xi.
2- generate a Parasite (Parasite_Vector) from Organism Xi
3- If the Parasite_Vector has better conditions than Xi, then replace it with the Parasite_Vector.

Step 5: Harmony Search Algorithm:

$$X_{i,j} = \begin{cases} w.p.HMCR ==> select\ a\ new value\ from\ X_i^k \\ \qquad ==> w.p.(1-PAR)\ do\ nothing \\ \qquad ==> w.p.PAR\ choose\ neighboring\ value \\ w.p.(1-HMCR) ==> select\ a\ new value\ randomly \end{cases} \tag{17}$$

where X_i^k is a function which rounds the continuous value to nearest discrete value, $X_{i,j}$ is the jth variable of student i, the HMCR is varying within [0,1] which sets a rate of choosing a value from the historic values stored in the X_i^k, (1-HMCR) sets the rate of choosing one value from the possible list of values. The pitch adjusting process is performed only after a value is chosen from X_i^k. the value (1-PAR) sets the rate of doing nothing, A PAR (pitch adjusting rate) of 0.1 indicates that the algorithm will choose a neighboring value with 10%×HMCR probability.

5. NUMERICAL EXAMPLES

To illustrate the application of the new hybrid algorithm, we selected several numerical examples that have been discretized by other researchers for optimization. 25-member spatial truss with 8 design variables. 52-member planar truss with 12 design variables. 72-member spatial truss with 16 design variables that optimized in two cases. The results of the new algorithm are compared with the results of other researchers.

5.1 Twenty-five bar spatial truss

The 25-member truss is shown in Fig. 1. The loading conditions are presented in Table 1. The grouping of truss members as size variables is shown in Table 2. The modulus of elasticity is 10e 7 psi. And specific gravity of 0.1 lb / in^3 have been selected as the mechanical properties of the materials. The results of truss weight optimization in Table 3 are compared with other references. Also the stress in truss members is equal to ± 40ksi and displacement in truss nodes is equal to ± 0.35in. The discrete cross-sectional area within a range of 0.01 - 3.4 in^2. The truss weight convergence trend is also shown in Fig. 2.

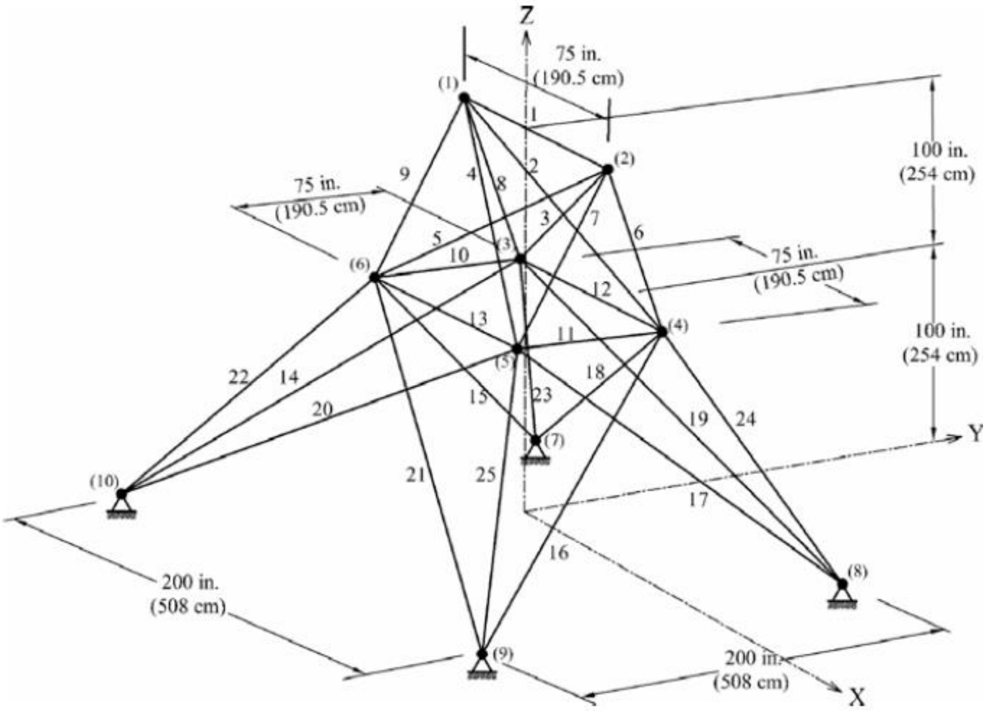


Figure 1. Topology of the 25-bar spatial truss

Table 1: Loading conditions for the 25-bar space truss

case	node	P_x (kips)	P_y (kips)	P_z (kips)
1	1	1.0	-10.0	-10.0
	2	0.0	-10.0	-10.0
	3	0.5	0.0	0.0
	6	0.6	0.0	0.0

Note: 1 in^2 = 6.452 cm^2 ; 1lb = 4.45 N

Table 2: Elements information

Group of elements							
1	2	3	4	5	6	7	8
1(1,2)	2:(1,4) 3:(2.3) 4:(1,5) 5:(2,6)	6:(2,4) 7:(2,5) 8:(1,3) 9:(1,6)	10(6.3) 11:(5,4)	12:(3,4) 13:(6,5)	14:(3,10) 15:(6,7) 16:(4,9) 17:(5,8)	18:(4,7) 19:(3,8) 20:(5,10) 21:(6,9)	22:(6,10) 23:(3,7) 24:(4,8) 25:(5,9)

The results of the hybrid SOS and other metaheuristic algorithms are listed in Table 3. As you see the best weight of the 25-bar spatial truss designed by the hybrid SOS is 481.17lb.

Table 3: performance comparison for 25-bar spatial truss with Discrete variables

variables		Cross-sectional area(in^2)					
Element group	members	GA Rajeev and Krishnamoorthy [1]	ACO Camp and Bichon [4]	PSOGA[13]	GA Cao [15]	BB–BC [16]	This work
1	1	0.10	0.10	0.10	0.10	0.10	0.10
2	2,3,4,5	1.80	0.30	0.50	0.50	0.30	0.30
3	6,7,8,9	2.30	3.40	2.30	3.40	3.40	2.30
4	10,11	0.20	0.10	0.10	0.10	0.10	0.10
5	12,13	0.10	2.10	1.50	1.90	2.10	1.50
6	14,15,16 ,17	0.80	1.00	0.70	0.90	1.00	0.70
7	18,19,20 ,21	1.80	0.50	0.90	0.50	0.50	0.90
8	22,23,24 ,25	3.00	3.40	3.10	3.40	3.40	3.10
Weight(lb)		546.01	484.85	482.25	485.05	484.85	481.17
W_{avg}(lb)		-	486.46	483.3.10	-	485.20	482.54
W_{stdv}(lb)		-	4.71	0.25	-	0.62	0.19
$N_{analysis}$		800	7,700	1,200	15,000	6,670	1,000

Note: 1 in^2 = 6.452 cm^2 ; 1lb = 4.45 N

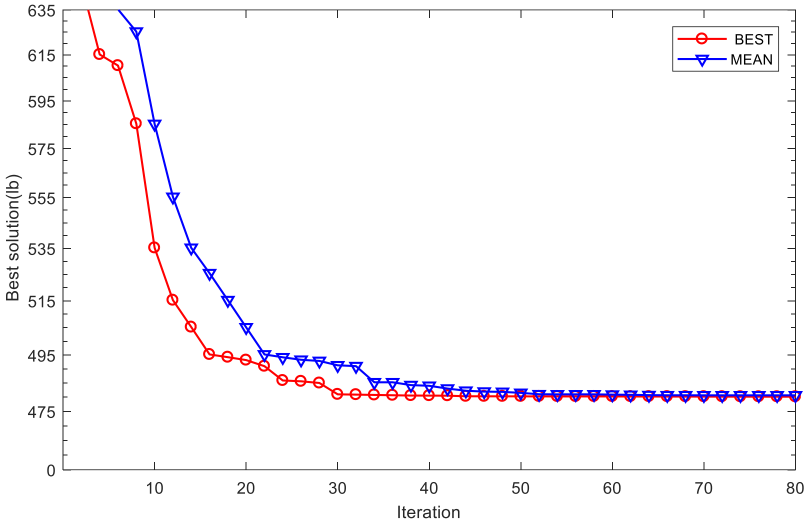


Figure 2. The convergence history of 25-bar spatial truss

with 1,000 searches. the best weight designed by GA standard [1] is 546.01lb with 800 searches, which more than the hybrid SOS algorithm also they didn't report any information about standard deviation. the best weight of the GA [15] algorithm is 485.05lb with 15,000 searches, also it didn't report any information about standard deviation. The best weight of the ACO [3] algorithm is 484.85lb after 7,700 searches and with a standard deviation of 4.71lb. the best weight of BB-BC [4] is 484.85lb with 6,670 searches, with a standard deviation of 0.62lb. The hybrid PSOGA [13] algorithm achieved the best weight 482.25lb after 1,200 searches with a standard deviation of 0.25lb. The BB-BC [16] algorithm achieved the best weight 484.85lb after 6,670 searches with a standard deviation of 0.62lb.

However, the result of the hybrid SOS is better than other metaheuristic algorithms. While the average weight of the hybrid SOS is 482.54 lb. with a standard deviation of 0.19lb. also, the hybrid SOS has a smaller required number of iteration for convergence.

5.2 Fifty-two bar planar truss

The 25-member truss is shown in Fig. 3. This structure is in the x-direction under load P_x = 100 KN and in the y-direction under load P_y = 200 KN. The modulus of elasticity is 2.05*10^5 MP. And specific gravity of 36.13*10-6 lb/in^3 have been selected as the mechanical properties of the materials. Also the stress in truss members is equal to ±180MP, and displacement in truss nodes is equal to ± 0.35in. The discrete variables are chosen from Table 4. The design variables are divided in to 12 groups as: (A1) 1-4, (A2) 5-10, (A3) 11-13, (A4) 14-17, (A5) 18-23, (A6) 24-26, (A7) 27-30, (A8) 31-36, (A9) 37-39, (A10) 40-43, (A11) 44-49, (A12) 50-52. The truss weight convergence trend is also shown in Fig. 4.

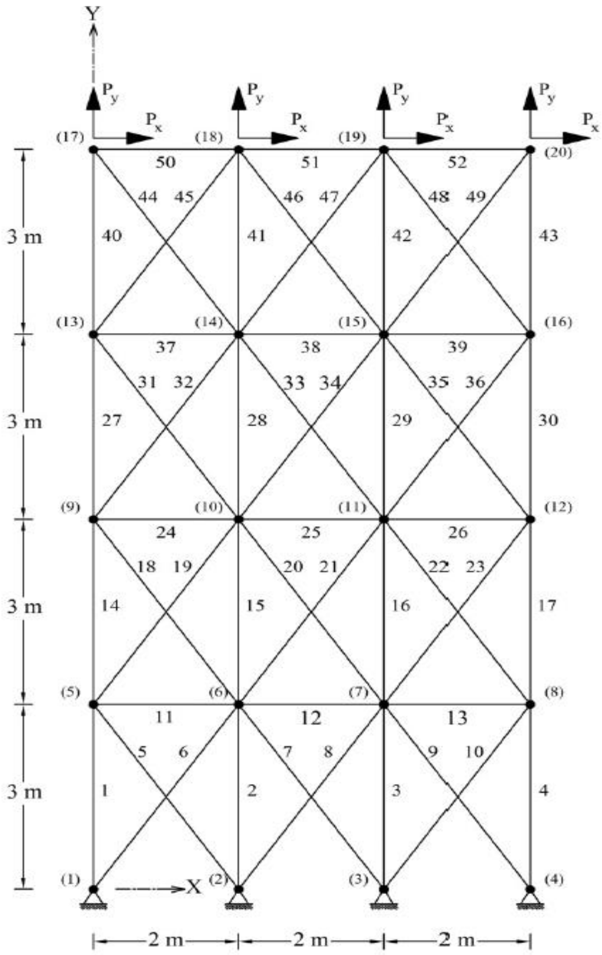


Figure 3. Topology of 52-bar planar truss

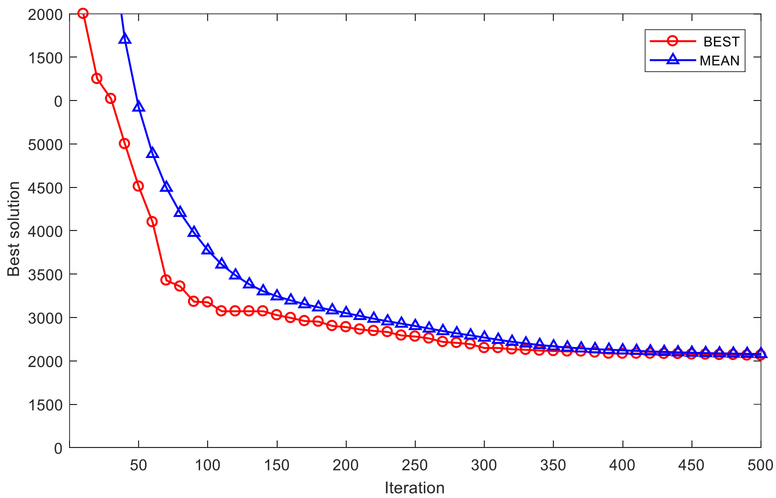


Figure 4. The convergence history of 52-bar truss

Table 4: the available cross-section areas of the AISC code

NO	in^2	mm	NO	in^2	mm
1	0.111	71.613	33	3.840	2477.414
2	0.141	90.968	34	3.870	2496.796
3	0.196	126.451	35	3.880	2503.221
4	0.250	161.290	36	4.180	2696.769
5	0.307	198.064	37	4.220	2722.575
6	0.391	252.258	38	4.490	2896.768
7	0.442	285.161	39	4.590	2961.284
8	0.563	363.225	40	4.800	3096.768
9	0.602	388.386	41	4.970	3206.445
10	0.766	494.193	42	5.120	3303.219
11	0.785	506.451	43	5.740	3703.218
12	0.994	641.289	44	7.220	4658.055
13	1.000	645.160	45	7.970	5141.925
14	1.228	792.256	46	8.530	5503.215
15	1.266	816.773	47	9.300	5999.988
16	1.457	393.998	48	10.850	6999.986
17	1.563	1008.385	49	11.500	7419.430
18	1.620	1045.159	50	13500	8709.660
19	1.800	1161.288	51	13.900	8967.724
20	1.990	1283.868	52	14.200	9161.272
21	2.130	1374.191	53	15.500	9999.980
22	2.380	1535.481	54	16.000	10322.560
23	2.620	1690.319	55	16.900	10903.204
24	2.630	1696.771	56	18.800	12129.008
25	2.880	1858.061	57	19.900	12838.684
26	2.930	1890.319	58	22.000	14193.520
27	2.090	1993.544	59	22.900	14774.164
28	1.130	729.031	60	24.500	15806.420
29	3.380	2180.641	61	26.500	17096.740
30	3.470	2238.705	62	28.000	18064.480
31	3.550	2290.318	63	30.000	19354.800
32	3.630	2341.931	64	33.500	21612.860

The results of the hybrid SOS and other metaheuristic algorithms are listed in Table 5. As you see the optimized weight of the 52-bar truss designed by the hybrid SOS is 1901.10kg. with 4,980 searches. The optimized weight of GA [17] is 1970.142kg. The best weight of

HS [18] is 1906.76kg. The optimized weight of HPSO [19] is 1905.49kg after 2000 iterations. The best weight designed by DHPSACO [9] is 1904.83kg. The best weight of the 52-bar truss designed by the hybrid PSO and GA [13] is 1901.35kg with 250 iterations and 5,000 searches. which in all cases are more than the hybrid SOS algorithm. However, the result of the hybrid SOS is better than other metaheuristic algorithms. Also, they didn't report any information about standard deviation, number of analyses, and average weight.

Table 5: performance comparison for 52-bar truss with Discrete variables

variables		Cross-sectional area(mm^2)					
Element group	members	DHPSACO [9]	PSOGA [13]	Wu and Chow GA [17]	Lee and Geem HS [18]	Li et al. [19] HPSO	This work
1	1-4	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055
2	5-10	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288
3	11-13	494.193	285.161	645.160	506.451	363.225	363.225
4	14-17	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219
5	18-23	1008.385	1045.159	1045.159	940.000	940.000	940.000
6	24-26	285.161	363.225	494.193	494.193	494.193	285.161
7	27-30	2290.318	2477.414	2477.414	2290.318	2238.705	2477.414
8	31-36	1008.385	1045.160	1045.159	1008.385	1008.385	1045.160
9	37-39	388.386	161.290	285.161	2290.318	388.386	161.290
10	40-43	1283.868	1283.868	1696.771	1535.481	1283.868	1283.868
11	44-49	1161.288	1161.288	1045.159	1045.159	1161.288	1161.288
12	50-52	506.451	506.451	641.289	506.451	729.256	506.451
Weight(kg)		1904.83	1901.35	1970.142	1906.76	1905.49	1901.10

Note: 1 in^2 = 6.452 cm^2 ; 1lb = 4.45 N

5.3 Seventy-two bar spatial truss

The geometry and more details of the 72-bar truss are shown in Fig. 5. The modulus of elasticity is 1e7 psi. the unit weight of the material is 0.1 lb/in^3. The members are subjected to the allowable stress limits of ±25ksi and the maximum displacement of each node is ±0.25 in through X, Y, and Z direction. There are 16 groups of design variables with a minimum 0.1in^2 and a maximum of 3.0 in^2: [A1] 1-4, [A2] 5-12, [A3] 13-16, [A4] 17-18, [A5] 19-22, [A6] 23-30, [A7] 31-34, [A8] 35-36, [A9] 37-40, [A10] 41-48, [A11] 49-52, [A12] 53-54, [A13] 55-58, [A14] 59-66, [A15] 67-70, [A16] 71-72. The structure is designed for two individual cases as:

Case 1: the structure in both cases subjected to multiple loading that listed in Table 6. The discrete design variables are chosen from the set {with minimum 0.1 and maximum 3.2, with interval 0.1} (in^2).

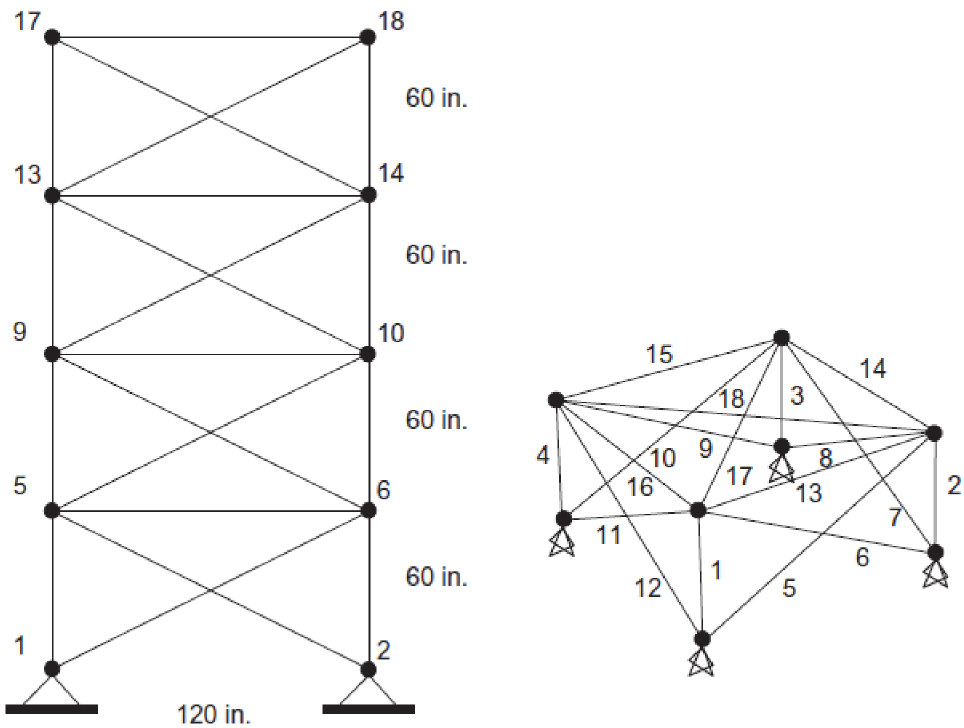


Figure 5. geometry and elements definition of 72-bar truss;(a) dimension and node numbering; (b) the pattern of element numbering.

Table 6: Multiple loading for the 72-bar truss

case	node	P_x (kips)	P_y (kips)	P_z (kips)
1	17	0.0	0.0	-5.0
	18	0.0	0.0	-5.0
	19	0.0	0.0	-5.0
	20	0.0	0.0	-5.0
2	17	5.0	5.0	-5.0

Note: 1 in^2 = 6.452 cm^2 ; 1lb = 4.45 N

Case 2: In this case, the design variables are chosen from Table 4.

The result of the metaheuristic algorithms in the first case is listed in Table 7. The best weight of hybrid SOS is 384.76lb. which better than other metaheuristic algorithms. The best weight of GA [17] is 400.66lb. The best weight designed by HS [18] is 387.94lb. The best weight of Li et al. [19] is 388.94lb, and the best weight designed by DHPSACO [9] is 385.54lb which in all cases are more than the result of the new hybrid SOS algorithm. They didn’t report any information about standard deviation and average weight.

Table 7: Performance comparison for 72-bar spatial truss with discrete variables (case 1)

Variables		Cross-sectional area(in^2)						
Element group	members	Wu and Chow GA [17]	Lee and Geem HS [18]	Li et al. [19] PSO	Li et al. [19] PSOPC	Li et al. [19] HPSO	DHPSACO [9]	This work
1	1-4	1.5	1.9	2.6	3.0	2.1	1.9	1.5
2	5-12	0.7	0.5	1.5	1.4	0.6	0.5	0.5
3	13-16	0.1	0.1	0.3	0.2	0.1	0.1	0.1
4	17-18	0.1	0.1	0.1	0.1	0.1	0.1	0.1
5	19-22	1.3	1.4	2.1	2.7	1.4	1.3	1.3
6	23-30	0.5	0.6	1.5	1.9	0.5	0.5	0.5
7	31-34	0.2	0.1	0.6	0.7	0.1	0.1	0.2
8	35-36	0.1	0.1	0.3	0.8	0.1	0.1	0.1
9	37-40	0.5	0.6	2.2	1.4	0.5	0.6	0.5
10	41-48	0.5	0.5	1.9	1.2	0.5	0.5	0.5
11	49-52	0.1	0.1	0.2	0.8	0.1	0.1	0.1
12	53-54	0.2	0.1	0.9	0.1	0.1	0.1	0.2
13	55-58	0.2	0.2	0.4	0.4	0.2	0.2	0.2
14	59-66	0.5	0.5	1.9	1.9	0.5	0.6	0.5
15	67-70	0.5	0.4	0.7	0.9	0.3	0.4	0.3
16	71-72	0.7	0.6	1.6	1.3	0.7	0.6	0.6
Weight(lb)		400.66	387.94	1089.88	1069.79	388.94	385.54	384.76

Note: 1 in^2 = 6.452 cm^2 ; 1lb = 4.45 N

The result of the metaheuristic algorithms in the second case is listed in Table 8. as you see the results of previous works the best weight obtained by DHPSACO [9] 393.38lb, which is better than other methods. But the result of hybrid SOS is 391.89 which is better than DHPSACO [9]. Fig. 6. shown the convergence history of the hybrid SOS algorithm for the 72-bar spatial truss.

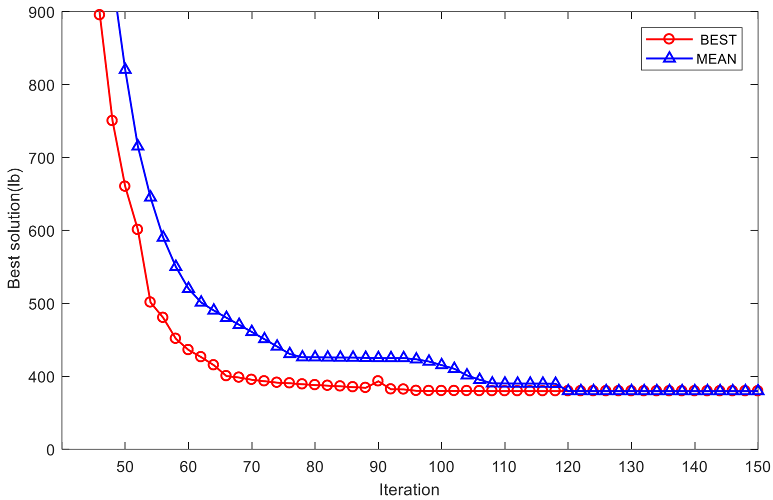


Figure 6. The convergence history of 72-bar spatial truss

Table 8: performance comparison for 72-bar spatial truss with discrete variables (case 2)

Variables		Cross-sectional area(in^2)					
Element group	members	Wu and Chow GA [17]	Li et al. [19] PSO	Li et al. [19] PSOPC	Li et al. [19] HPSO	DHPSACO [9]	This work
1	1-4	0.196	7.220	4.490	4.970	1.800	1.800
2	5-12	0.602	1.800	1.457	1.228	0.442	1.228
3	13-16	0.307	1.130	0.111	0.111	0.141	0.111
4	17-18	0.766	0.196	0.111	0.111	0.111	0.111
5	19-22	0.391	3.090	2.620	1.880	1.228	0.391
6	23-30	0.391	0.785	1.130	1.457	0.563	0.391
7	31-34	0.141	0.563	0.196	0.141	0.111	0.141
8	35-36	0.111	0.785	0.111	0.111	0.111	0.111
9	37-40	1.800	3.090	1.266	1.563	0.563	1.563
10	41-48	0.602	1.228	1.457	1.228	0.563	0.602
11	49-52	0.141	0.111	0.111	0.111	0.111	0.111
12	53-54	0.307	0.563	0.111	0.196	0.250	0.196
13	55-58	1.563	0.990	0.442	0.391	0.196	0.196
14	59-66	0.766	1.620	1.457	1.457	0.563	0.766
15	67-70	0.141	1.563	1.228	0.766	0.442	0.141
16	71-72	0.111	1.266	1.457	1.563	0.563	0.111
Weight(lb)		427.203	1209.48	941.82	933.09	393.38	391.89

Note: 1 in^2 = 6.452 cm^2 ; 1lb = 4.45 N

6. CONCLUSION

Optimization of continuous problems using methods that are basically presented for discrete problems, applying the discrete methods is avoidable. Providing an efficient approach that can streamline the process of achieving global optimal without getting caught up in local optimal has been a serious challenge to researchers. In recent decades, several optimization methods proposed by researchers. But since all of the metaheuristics are inspired by the life of creatures, they have weaknesses. In recent years, the idea of hybridizing metaheuristic algorithms has come to the attention of many researchers. In this work, we hybridized the SOS and HS algorithms, which are the best meta-heuristic algorithms, then designed and optimized several benchmark structures. The results show that this new algorithm is able to provide better performance in the optimization process by eliminating the weaknesses of previous algorithms.

REFERENCES

1. Rajeev S, Krishnamoorthy CS. Discrete optimization of structures using genetic algorithms, *J Struct Eng* 1992; **118**(5): 1233-50.
2. Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm, *Comput Struct* 2004; **82**(9–10): 781-98
3. Camp CV, Bichon BJ. Design of space trusses using ant colony optimization, *J Struct Eng* 2004; **130**(5): 741–51.
4. Camp CV. Design of space trusses using big bang–big crunch optimization, *J Struct Eng* 2007; **133**(7): 999–1008.
5. Kennedy J, Eberhart R. Particle swarm optimization, In: *Proceedings of IEEE International Conference on Neural Networks* 1995; pp. 1942–1948.
6. Rao RV, Savsani VJ, Vakharia DP. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, *Comput Aided Des* 2011; **43**(3): 303–15
7. Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search, *Acta Mech* 2010; **213**: 267-289.
8. Cheng M, Prayogo D. Symbiotic organisms search: A new metaheuristic optimization algorithm, *Comput Struct* 2014; **139**: 98-112.
9. Kaveh A, Talatahari S. A particle swarm ant colony optimization for truss structures with discrete variables, *J Construct Steel Res* 65 (2009); 1558-68.
10. Kaveh A, Talatahari S. A hybrid particle swarm and ant colony optimization for design of truss structures, *Asian J Civil Eng* 2008; **9**(4): 329-48.
11. Kaveh A, Talatahari S. Hybrid algorithm of harmony search, particle swarm and ant colony for structural design optimization, *Comput Struct* 2009; **3**: 642-03450
12. Shojaee S, Arjomand M, Khatibinia M. A hybrid algorithm for sizing and layout optimization of truss structures combining discrete pso and convex approximation, *Int J Optim Civil Eng* 2013; **3**(1): 57-83.
13. Omidinasab F, Goodarzimehr V. A hybrid particle swarm optimization and genetic

algorithm for truss structures with discrete variables, *J Appl Comput Mech* 2020; **6**(3): 593-604. doi: 10.22055/jacm.2019.28992.1531.
14. Talatahari S, Goodarzimehr V, Taghizadieh N. Hybrid teaching-learning-based optimization and harmony search for optimum design of space trusses, *J Optim Indust Eng* 2020; **13**(1): 177-194. doi: 10.22094/joie.2019.1866904.1649.
15. Cao G. Optimized Design of Framed Structures Using a Genetic Algorithm, PhD thesis, The University of Memphis, TN, 1996.
16. Kaveh A, Talatahari S. Size optimization of space trusses using big bang–big crunch algorithm, *Comput Struct* 2009; **87**(17-18): 1129-40.
17. Wu SJ, Chow PT. Steady-state genetic algorithms for discrete optimization of trusses, *Comput Struct* 1995; **56**(6): 979-91.
18. Lee KS, Geem ZW, Lee SH, Bae KW. The harmony search heuristic algorithm for discrete structural optimization, *Eng Optim* 2005; **37**(7): 663-84.
19. Li LJ, Huang ZB, Liu F, Wu QH . A heurstic particle swarm optimizer for optimization of pin connected structures, *Comput Struct* 2007; **85**(7–8): 340–9.