

A new hybrid optimization algorithm based on path projection

Saeed Asil Gharebaghi* and Mohammad Ardalan Asl^a

Department of Civil Engineering, K. N. Toosi University of Technology, Tehran, Iran

(Received November 18, 2017, Revised January 12, 2018, Accepted January 16, 2018)

Abstract. In this article, a new method is introduced to improve the local search capability of meta-heuristic algorithms using the projection of the path on the border of constraints. In a mathematical point of view, the Gradient Projection Method is applied through a new approach, while the imposed limitations are removed. Accordingly, the gradient vector is replaced with a new meta-heuristic based vector. Besides, the active constraint identification algorithm, and the projection method are changed into less complex approaches. As a result, if a constraint is violated by an agent, a new path will be suggested to correct the direction of the agent's movement. The presented procedure includes three main steps: (1) the identification of the active constraint, (2) the neighboring point determination, and (3) the new direction and step length. Moreover, this method can be applied to some meta-heuristic algorithms. It increases the chance of convergence in the final phase of the search process, especially when the number of the violations of the constraints increases. The method is applied jointly with the authors' newly developed meta-heuristic algorithm, entitled Star Graph. The capability of the resulted hybrid method is examined using the optimal design of truss and frame structures. Eventually, the comparison of the results with other meta-heuristics of the literature shows that the hybrid method is successful in the global as well as local search.

Keywords: hybrid optimization; global search; local search; meta-heuristic methods; classic methods

1. Introduction

In the recent decade, attempts to improve meta-heuristic methods using various ideas have been increased. Accordingly, some researchers have introduced new powerful meta-heuristic algorithms, while others have applied a combination of several methods to create a new algorithm. The latter approach is more popular than the former because the resulted mixed algorithm attempts to inherit the strengths of its parents and simultaneously resolve their issues. For example, Soleimani and Kannan (2015) considered improving closed-loop supply chain network optimization processes. Indeed, two popular meta-heuristic algorithms were deemed to develop a new elevated hybrid algorithm: The Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Later, a new hybrid method, which combines the extensive area search ability of the GA and the local search capacity of the PSO, was introduced by Wanga *et al.* (2015). Mostly these hybrid methods are established, based on some well-developed meta-heuristic methods, such as Simulated Annealing (SA) (Kirkpatrick *et al.* 1983), Big Bang–Big Brunch (BB-BC) (Erol and Eksin 2006), and Symbiotic Organisms Search (SOS) (Cheng and Prayogo 2014).

Epitropakis *et al.* (2012) developed a hybrid framework that combines PSO and the Differential Evolution Algorithm (DEA). Additionally, Wang *et al.* (2013) proposed a hybrid PSO algorithm, called DNSPSO, which

employs a diversity enhancing mechanism and neighborhood search strategies to achieve a trade-off between exploration and exploitation abilities. Recently, a hybrid Harmony Search PSO with global dimension selection for improving the performance of PSO has been presented by Ouyang *et al.* (2016).

Giftson Samuel and Christober Asir Rajan (2015) presented two optimization methods: A Hybrid Particle Swarm Optimization based on Genetic Algorithm, and a Hybrid Particle Swarm Optimization based on Shuffled Frog Leaping Algorithm for solving the long-term generator maintenance scheduling problem. Besides, Nayanatara *et al.* (2016) presented a hybrid algorithm which, includes Fuzzy-Genetic Algorithm (FGA). This method is used to optimize the Distributed Generation (DG) parameters in a deregulated power system. A new hybrid optimization technique was proposed by Kelner *et al.* (2008) that merges GA with a local search strategy based on the Interior Point method. Moreover, Tuba and Bacanin (2014) introduced modifications to the Seeker Optimization Algorithm to control exploitation/exploration balance and hybridized it with the elements of the Firefly Algorithm that improved its exploitation capabilities.

An efficient hybrid optimization strategy was used for determining the parameters of radial basis function neural networks by Wu *et al.* (2015). This method incorporates the adaptive optimization of PSO into GA, namely HPSOGA. Cheng *et al.* (2016) presented a new variant of the Harmony Search (HS) algorithm. While retaining Harmony Memory and pitch adjustment functions, this variant replaces the HS algorithm randomization function with Global-best PSO search and neighborhood search. Also, Shao *et al.* (2016) presented a hybrid discrete optimization algorithm based on

*Corresponding author, Ph.D.

E-mail: asil@kntu.ac.ir

^aPh.D. Student

Teaching-Probabilistic Learning mechanism (HDTPL), which consists of four components: discrete teaching phase, discrete probabilistic learning phase, population reconstruction, and neighborhood search. Besides, Ma *et al.* (2014) proposed several hybrid evolutionary algorithms by combining some recently developed algorithms with a biogeography-based hybridization strategy.

A new hybrid heuristic approach that combines the Quantum PSO technique with a local search method to solve the Multi-Dimensional Knapsack Problem was proposed by Haddar *et al.* (2016). Besides, Hoseini and Shayesteh (2013) proposed a hybrid algorithm including GA, Ant Colony Optimization (ACO), and Simulated Annealing (SA) for increasing the contrast of images. The effect of elite pool in three hybrid population-based meta-heuristics was investigated by Jaradat *et al.* (2016). This method consists of an elite pool of a hybrid Elitist-Ant System, a hybrid Big Bang-Big Crunch optimization and a hybrid Scatter Search. Jeslin Drusila Nesamalar *et al.* (2016) presented a Hybrid Nelder-Mead-Fuzzy Adaptive PSO (HNM-FAPSO) for a Multi-Line Congestion Management (MLCM) problem. Also, the objective of hybridizing the Nelder-Mead (NM), Fuzzy Adaptive PSO (FAPSO) is to blend their unique advantages as well as their efficacy. Liu *et al.* (2016) introduced two hybrid methods with two different strategies to combine limited memory BFGS (L-BFGS) with Greedy Diffusion Search (GDS).

Some new hybrid methods have been developed and utilized in the optimal design of structures. For instance, Kaveh and Talatahari (2009c) have introduced a new hybrid approach, based on PSO, ACO and HS to optimize truss structures. Also, Cheng and Prayogo (2017) have developed a new Fuzzy Adaptive Teaching-Learning-Based Optimization method (FATLBO) for the optimal design of different types of structures. Furthermore, some other methods have been developed by Li *et al.* (2009), Prayogo *et al.* (2018), Kaveh, A. and Shahrouzi, M. (2008), Rahami *et al.* (2011), Kaveh and Laknejadi (2013), Kaveh and Mahdavi (2013), Kaveh and Javadi (2014) and Kaveh and Ilchi Ghazaan (2018).

Asil Gharebaghi *et al.* (2017) have introduced a new meta-heuristic method, based on the concept of Star Graph (SG). The method has been applied to some constrained and unconstrained engineering problems. Although the meta-heuristic algorithms are good in the global search process, they need to be improved when a local search is required. In this article, the local search ability of SG is improved using Path Projection method. Ultimately, the resulted hybrid method is utilized in the optimal design of truss and frame structures.

This paper consists of five sections, which are described after a brief description of the basis of the method.

2. Basics of the hybrid method

The meta-heuristic methods, including PSO, have a good capacity to solve the engineering optimization problems. But in the local convergence, they may get into trouble because of the fast diversity reduction (Wang *et al.* 2013). To improve the performance of these methods some

different hybrid approaches have been developed. As aforementioned, in some cases, two or more meta-heuristic methods are combined (Soleimani and Kannan 2015, Ouyang *et al.* 2016, Nayanatara *et al.* 2016, Wu *et al.* 2015, Cheng *et al.* 2016, Shao *et al.* 2016, Jaradat *et al.* 2016, Jeslin Drusila Nesamalar *et al.* 2016, Kaveh *et al.* 2015, Abbasnia *et al.* 2014). In other cases, a classic method is joined with a meta-heuristic algorithm (Wang *et al.* 2015, Epitropakis *et al.* 2012, Tuba and Bacanin 2014). In some other cases, the new hybrid method is created by adding a new part to an available meta-heuristic method. Mostly, the new part is not independent of the main algorithm (Wang *et al.* 2013). This part is applied only to reinforce the local search ability in the meta-heuristic algorithm. The presented method in this article contains this pattern to improve the capability of some meta-heuristic methods.

Although the simultaneous usage of constraints and fitness function to identify the optimal results is common in the classic methods, it has not been sufficiently addressed in meta-heuristic methods. In these methods, the application of constraints is limited to the penalty function, while constraints consist of valuable information for the local search procedure. In other word, unlike meta-heuristic methods, the interaction between fitness function and constraints is the basis of classic methods. As an example, in Linear Programming, the agents move on the border of constraints to reach the final result (Luenberger 1973).

In the case of violation of constraints in meta-heuristic methods, the penalty of fitness function stops the progress of the agent. Therefore, the global search ability of meta-heuristics is stronger than their local search. Thus, if the information of constraints is utilized to determine the search path, more convergence rate could be reached in the meta-heuristic methods. This method is independent of the global search procedure, and it can be utilized as a practical tool along with meta-heuristics.

This article utilizes the concept of Gradient Projection Method (Rosen 1960, Rosen 1961) to employ constraints and fitness function simultaneously, but some new practical features are applied to improve the procedure. Gradient Projection Method has been based on the gradient calculation of the fitness function and constraints. In the first step of this method, active constraints are determined using Lagrangian function. In the next step, the projection of the gradient of fitness function on the border of the active constraints is determined. The resulted projection is applied as a direction of the local search. Identification of constraints and gradient vector projection on constraints' border are the primary functions of this method. Similar to the other classical methods, in Gradient Projection Method, there are limitations in the calculation of the gradient of the fitness function and constraints.

In this article, the concept of the gradient projection is utilized with a different aspect, so that imposed limitations, such the gradient calculation, are eliminated. Accordingly, the gradient vector is replaced with a new meta-heuristic vector, and the determination of active constraint and the projection process are changed into some new and low-cost methods. In the one hand, the concept of gradient projection is preserved; on the other hand, limitations and the cost of the process are decreased.

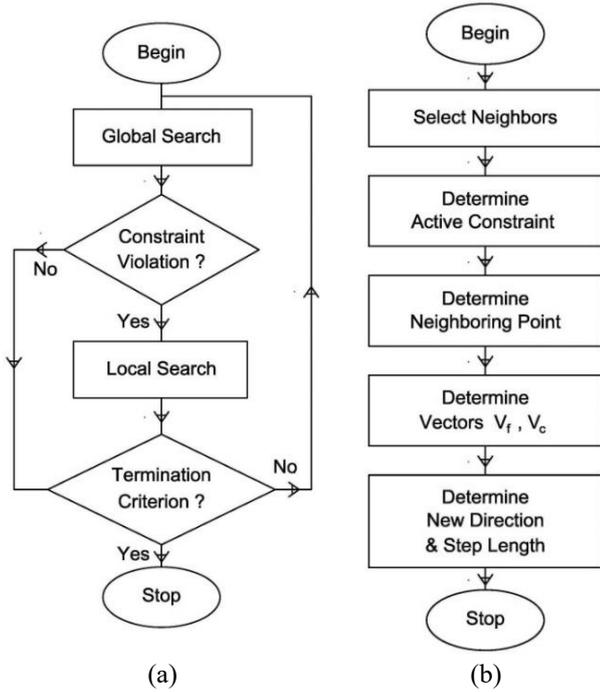


Fig. 1(a) The local search in interaction with the global search (meta-heuristics), (b) the flowchart of the presented local search

The presented method is inspired by Gradient Projection Method, while the costly functions such the gradient calculations are removed and some low-cost features are replaced. In Gradient Projection method, a matrix containing the gradient vectors of constraints should be generated. It makes some difficulties in calculation process, but in the present work, some simple geometric vectors are replaced instead of the gradient vectors. Although, in both methods, the movement along the constraints' border is utilized as a basic concept of the local search, the present method is more compatible with the implicit nonlinear constraints.

As it was mentioned before, the presented method can be utilized as the main tool in the exploitation stage of some meta-heuristic methods. For this purpose, in the case of constraint violation, as shown in Fig. 1(a), the present method can be applied to perform a more accurate local search. The utilized information may consist of the vector of location, the fitness function and constraints value of agents. The presented method, as depicted in Fig. 1(b), gets the input data and suggests a new location based on the fitness function and constraints of the problem.

In this article, Star Graph method is selected to be used as a global search engine in Fig. 1(a). The capability of this method in the optimal design of different structures has been previously demonstrated (Asil Gharebaghi *et al.* 2017). However, it is desired to know whether or not SG is capable of being used in companion with Path Projection method to make a hybrid algorithm. In the present work, the global features of SG, and local capabilities of Path Projection are used to enhance the rate of convergence. Finally, the presented hybrid method is examined for weight optimization of structures, as well.

3. The local search algorithm

In this section, the algorithm is described in detail. In the case of constraint violation in a meta-heuristic method, the information of current and previous step of an agent can be utilized in the following algorithm. This local search algorithm will correct the path and suggest a new feasible location for the agent. The flowchart of this algorithm, as shown in Fig. 1(b), consists of four primary functions. The details of the algorithm are described as following:

Step 1: Neighbors Selection. In the first step, a set of agents is selected as the neighbors of the i^{th} agent. A random selector determines the neighbors using a cumulative weighted function (G). The weight function is calculated using Eq. (1). Later, the cumulative weighted function could be computed from Eq. (2).

$$\alpha_j = \frac{w_j}{\sum_{t=1}^{N_b} w_t}, \quad w_j = \frac{F_{\max} - F_j}{F_{\max} - F_{\min}} \quad (1)$$

Where, F_{\max} and F_{\min} are the maximum and minimum of the fitness function value in all the agents of the search space. The cumulative weighted function G is an array, which its j^{th} element includes the sum of the weight of the j^{th} agent and all previous agents in the global numbering system. This array can be mentioned as Eq. (2)

$$G_j = G_{j-1} + \alpha_j, \quad G_1 = \alpha_1 \quad (2)$$

For the selection of each neighbor of the i^{th} agent, the minimum integer value of j should be considered in such a way that $rand \leq G_j$. Where, $rand$ is a random number generator with a uniform distribution in a range of $[0, 1]$. Indeed, to determine all the neighbors of the i^{th} agent (I_b), this process should be repeated for N_b times. N_b is the number of neighbors, who belong to the i^{th} agent.

This approach is important in two aspects: first, by using this selection method, the probability of the selection of an agent with a suitable location is more than the others. As a result, it helps to accelerate the convergence of the algorithm. Second, the weighting method does not restrict the neighbor selection in a small local zone. Thus, all agents have a chance of being selected. It helps to increase the diversity of the method and prevents from being trapped in a local minimum.

Step 2: Active Constraint and Neighboring Point Determination. If a meta-heuristic algorithm violates a single constraint, the violated constraint will be known as an active constraint. However, if the new result violates more than one constraint, the active violated constraint should be determined to apply for the path correction procedure. The new correction path will go along the border of the active constraint.

Assuming that the new location of an agent violates one or more constraint(s), a line is drawn from the previous (admissible) agent to the new (inadmissible) one. The first violated constraint will be known as the active constraint. One can utilize the line equation in vector space to determine the corresponding active constraint. The schematic presentation of the process is illustrated in Fig. 2. In Fig. 2, the path of the i^{th} agent violates three constraints

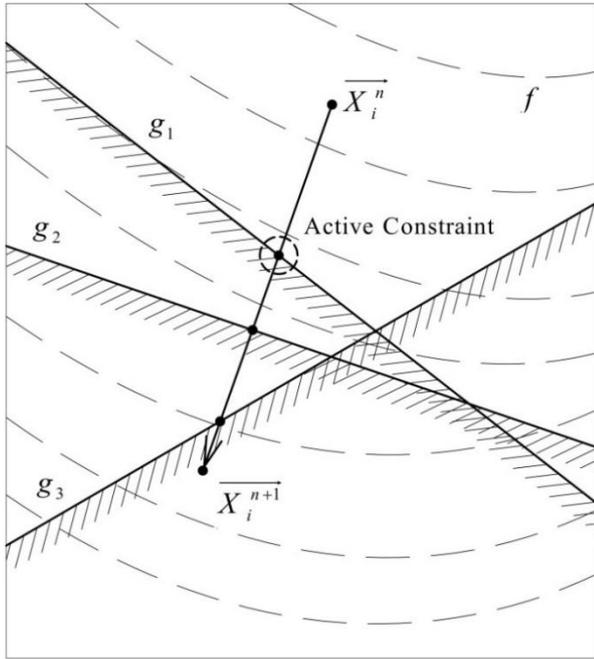


Fig. 2 The schematic presentation of the particle path and the active constraint determination

g_1 , g_2 , and g_3 . Since the constraint g_1 is the first constraint, which is intersected by the aforementioned line, it will be defined as the active constraint. This constraint probably has the greatest impact on the path of the agent, so it is especially utilized in the presented method. In Fig. 2, the hatched areas on the constraints border indicate the unauthorized or inadmissible zone of the search space. The intersection of the agent path and the border of the active constraint could be determined as a neighboring point of the active constraint (\vec{X}_0). This location may be found by a linear interpolation on the agent path (Appendix A), based on the values of the active constraint in current and previous steps. According to Appendix A, the neighboring point of the active constraint (g) for the i^{th} agent in steps n and $n+1$ (\vec{X}_i^n and \vec{X}_i^{n+1}) is as Eq. (3)

$$\vec{X}_0 = [x_{0,1} \quad x_{0,2} \quad \dots \quad x_{0,n}] \quad (3)$$

The neighboring point is a location near to the active constraint. Analytical results indicate that the accuracy of this location has a limited impact on the final result of the method. Hence, a conservative assumption could be considered in the calculation of this location, so that the neighboring point is located in the valid zone of search space and near to the active constraint. Therefore, the previous relation can be changed to the following conservative equation

$$X_0^* = X_i^n + \alpha \times |X_0 - X_i^n| \times \frac{X_0^{n+1} - X_i^n}{|X_0^{n+1} - X_i^n|} \quad (4)$$

Where, the factor α can be equal to 0.95, conservatively.

Step 3: Vectors \vec{V}_f and \vec{V}_c Determination. Two new

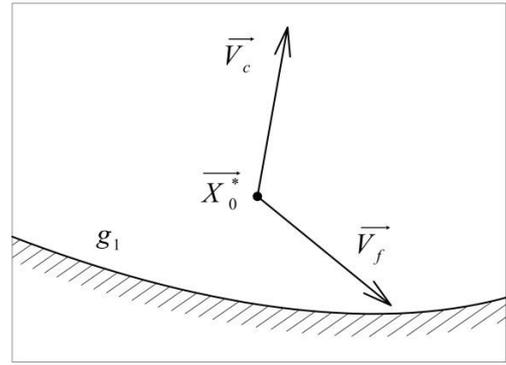


Fig. 3 The neighboring point \vec{X}_0^* and the vectors \vec{V}_f and \vec{V}_c

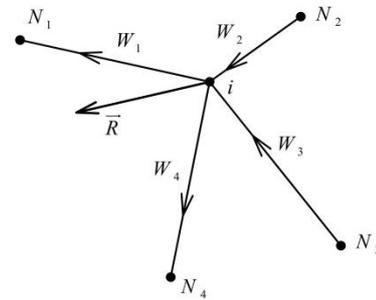


Fig. 4 The weighted directed graph of i^{th} agent

vectors are introduced herein to define the correction path of the i^{th} agent using corresponding neighbors. First, a vector to assess the fitness function devaluation \vec{V}_f and the second vector, which is almost perpendicular to the active constraint, is \vec{V}_c . These vectors are not an exact definition of the gradient estimation; however, they could be a numerical primary estimation of the nature of the problem in the border of the active constraint.

In the determination of these vectors, a weighted directed graph including the i^{th} agent and its neighbors is utilized. Each edge of the graph has a weight corresponding to the fitness function value (f) and the active constraint value (g) of its nodes. The vectors \vec{V}_f and \vec{V}_c are the resultant of the directions of these weighted edges. A typical graph of the i^{th} agent, which consists of nodes (agents) N_1 , N_2 , N_3 , and N_4 is shown in Fig. 4.

In this figure, the vector \vec{R} indicates the resultant of the weighted directions of the edges of the graph. The weighting function H , according to the definition of the graph, is as Eq. (5)

$$H_{f,j} = \frac{W_{f,j}}{\sum_{j=1}^{N_s} W_{f,j}}, \quad H_{g,j} = \frac{W_{g,j}}{\sum_{j=1}^{N_s} W_{g,j}} \quad (5)$$

$$W_{f,j} = \frac{f_{\max}}{f_j}, \quad W_{g,j} = \frac{g_{\max}}{g_j}$$

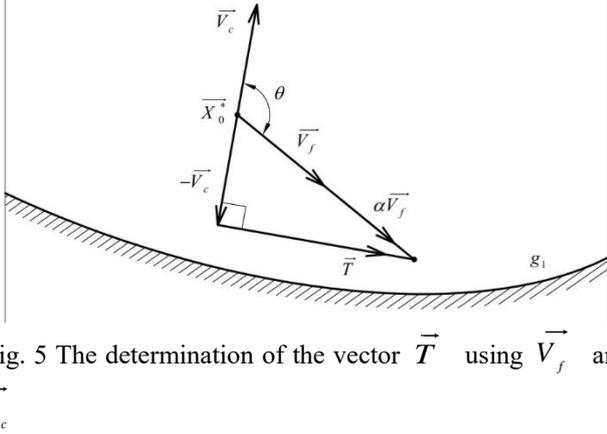


Fig. 5 The determination of the vector \vec{T} using \vec{V}_f and \vec{V}_c

Where, f_{max} and g_{max} are the maximum of the fitness function and the active constraint value of the neighbors of the i^{th} agent (I_b), respectively.

Vectors \vec{U}_f and \vec{U}_g , which show the direction of the edges of the graph can be described by Eq. (6)

$$\begin{aligned}\vec{U}_{f,i} &= \frac{\vec{X}_i^n - \vec{X}_j^n}{\|\vec{X}_i^n - \vec{X}_j^n\|} \times \text{Sign}(f_j - f_i) \\ \vec{U}_{g,i} &= \frac{\vec{X}_i^n - \vec{X}_j^n}{\|\vec{X}_i^n - \vec{X}_j^n\|} \times \text{Sign}(g_i - g_j)\end{aligned}\quad (6)$$

Where, $\text{Sign}(x)$ indicates the sign of variable x , \vec{X}_i^n is the location of the i^{th} agent in the n^{th} iteration and $\|\vec{X}_i^n\|$ is the norm of the vector \vec{X}_i^n . Vectors \vec{V}_f and \vec{V}_c can be calculated by applying the weighting function to the previous equations as

$$\begin{aligned}\vec{V}_{f,i} &= \frac{\vec{V}_{f,i}^*}{\|\vec{V}_{f,i}^*\|}, \quad \vec{V}_{f,i}^* = \sum_{j=1}^{N_b} H_{f,j} \cdot \vec{U}_{f,j} \\ \vec{V}_{c,i} &= \frac{\vec{V}_{c,i}^*}{\|\vec{V}_{c,i}^*\|}, \quad \vec{V}_{c,i}^* = \sum_{j=1}^{N_b} H_{g,j} \cdot \vec{U}_{g,j}\end{aligned}\quad (7)$$

Where, the index i refers to the corresponding vectors of the i^{th} agent.

Step 4. The New Direction and Step Length Determination. In this article, the path of the agent is determined based on two principles: first, the agent moves along the border of the active constraint; second, it moves in a direction, which decreases the fitness function value.

The active constraint and the neighboring point have been determined so far. Herein, a correction path, which is based on the mentioned rules and vectors \vec{V}_f and \vec{V}_c is introduced. The path selection in an N-dimensional system could be partly difficult; however, the new suggested geometric method can solve the problem quickly.

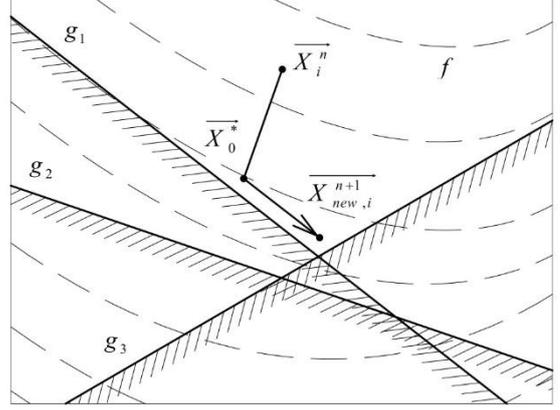


Fig. 6 The suggested path of i^{th} particle in the local search process

As it has been depicted in Fig. 3, it is considered that the neighboring point \vec{X}_0^* is located close to the active constraint g_l and vectors \vec{V}_f and \vec{V}_c are determined according to the fitness function and the active constraint value of the neighbors. The determination of path which moves along the border of the active constraint and decreases the fitness function is illustrated in Fig. 5. The vector \vec{T} can be defined using a right triangle, which consists of $-\vec{V}_c$ and a scaled version of \vec{V}_f . At this point, the value of the scale is unknown. To have a clear view, in a three-dimensional search space, the vector \vec{T} is the projection of the vector \vec{V}_f on the plane that is normal to the vector \vec{V}_c . It is noted that the vector \vec{V}_f could decrease the fitness function and the vector \vec{V}_c is almost normal to the active constraint. To form the right triangle, the scale factor α can be calculated from Eq. (8)

$$\alpha = \frac{\|\vec{V}_f\|^2}{\vec{V}_c \cdot \vec{V}_f}\quad (8)$$

Where, the dot sign (\cdot) indicates the inner product and $\|\vec{V}_f\|$ shows the norm of the vector \vec{V}_f . Accordingly, the path vector \vec{T} can be expressed using Eq. (9)

$$\vec{T} = \alpha \cdot \vec{V}_f - \vec{V}_c\quad (9)$$

Also, the new suggested location of the i^{th} agent ($\vec{X}_{new,i}^{n+1}$) can be computed as

$$\vec{X}_{new,i}^{n+1} = \vec{X}_0^* + \lambda \cdot \vec{S}, \quad \vec{S} = \frac{\vec{T}}{\|\vec{T}\|}\quad (10)$$

Where, λ is the step length that is defined in Eq. (11).

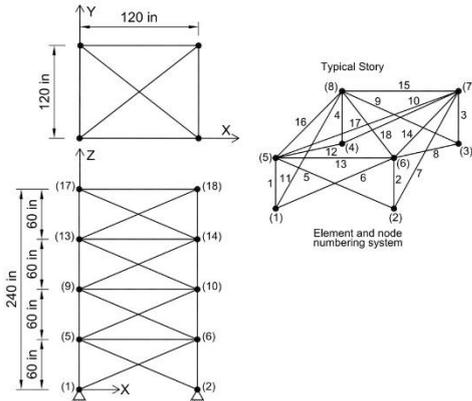


Fig. 7 The geometry of 72-member space truss

$$\lambda = \beta \times \sum_{j=1}^{N_b} H_{f,j} \cdot \left| \vec{X}_i - \vec{X}_j \right| \quad (11)$$

In which, β is a random number generator with a uniform distribution in the range of [0, 1]. The final path of the i^{th} agent has been shown in Fig. 6.

In Fig. 6, the point \vec{X}_0^* acts as a mediator and exerts the effects of the constraints on the path of the agent. After the determination of the new modified location of the agent, if it is in a valid search space and the fitness function is decreased, the results will be recorded; otherwise, the process will return to the global search procedure.

The presented method has used some simplifying assumptions, which definitely results in a computational error, especially in \vec{V}_c and \vec{V}_f . Thus, the new path may not be parallel to the border of the active constraint. However, the numerical results show that the movement in the near of the borders satisfactorily meets the requirements of the algorithm.

3.1 The Kuhn-Tucker conditions

To control the possibility of convergence, the path correction process should satisfy Kuhn-Tucker conditions (Hanson 1981). The presented method is subjected to check whether or not it satisfies Kuhn-Tucker conditions. In the case of failure, the method should get the agent far from the border of the constraints. As a result, approaching to the borders will decrease the chance of finding the solution. In such circumstances, to solve the problem, one can replace

the vector $\vec{S} = \frac{\vec{V}_g}{\|\vec{V}_g\|}$ in Eq. (10). This approach helps to

improve the convergence of the method and prevents the agent from searching in a wrong direction. This approach is implemented to solve the numerical problems of the article.

4. Numerical problems

In this section, some practical constrained optimization

Table 1 The design load combinations of 72-member space truss

Node	Case 1			Case 2		
	P_x	P_y	P_z	P_x	P_y	P_z
	kips (kN)					
17	5.0 (22.25)	5.0 (22.25)	-5.0 (22.25)	0.0	0.0	-5.0 (22.25)
18	0.0	0.0	0.0	0.0	0.0	-5.0 (22.25)
19	0.0	0.0	0.0	0.0	0.0	-5.0 (22.25)
20	0.0	0.0	0.0	0.0	0.0	-5.0 (22.25)

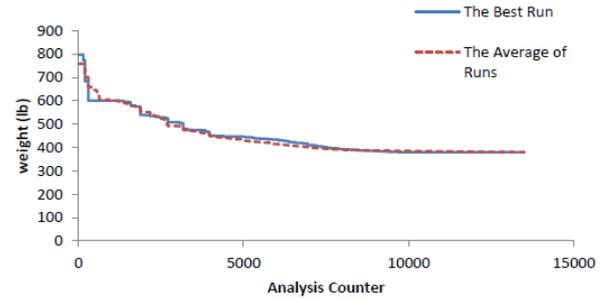


Fig. 8 Convergence graph of the 72-member space truss problem

problems are presented. The problems have been taken from the structural engineering literature. The results of the present work are compared to their counterparts of other methods, and the efficiency of the method is approved.

4.1 A 72-member space truss problem

For the first example, the optimal design of a space truss with 72 members, as shown in Fig. 7, is considered. The assigned load combinations are presented in Table 1. The cross sections of the members are assumed as the variables of design. The constraints of the problem are presented as:

- The allowable cross section of the members is between 0.1 to 2.5 in².

- The maximum allowable stress (tensile and compressive) is equal to 25.0 ksi for all groups of the members.

- The maximum allowable node displacement in X and Y directions for both load combinations is restricted to ± 0.25 in (Kaveh *et al.* 2014).

In Table 2, the results of the optimal design of the method are compared with the methods GA (Erbatur *et al.* 2000), PSO (Perez and Behdinan 2007), BB-BC (Kaveh and Talatahari 2009b), SAHS (Degertekin 2012), CSP (Kaveh *et al.* 2014), and SG (Asil Gharebaghi *et al.* 2017). The best-obtained result of these methods belongs to BB-BC with 379.66 lb and an average of 13200 analyses for 20 independent runs of the algorithm (Kaveh and Talatahari 2009b). The optimum weight of the presented method is equal to 379.63 lb and an average of 13542 analyses for 20 independent runs of the algorithm. In this case, the critical value of constraints is limited to 2.7×10^{-6} and no constraint violation is occurred. Also, the convergence graph of this method is presented in Fig. 8. Star Graph algorithm (SG),

Table 2 The comparison of optimal design results of 72-member space truss

Element group	Optimal cross-sectional areas (in ²)							Present work
	GA	PSO	BB-BC	SAHS	CSP	SG		
1	A ₁ -A ₄	1.755	1.743	1.9042	1.860	1.94459	0.156089	1.8726
2	A ₅ -A ₁₂	0.505	0.518	0.5162	0.521	0.50260	0.553373	0.5093
3	A ₁₃ -A ₁₆	0.105	0.100	0.1000	0.100	0.10000	0.440694	0.1000
4	A ₁₇ -A ₁₈	0.155	0.100	0.1000	0.100	0.10000	0.549534	0.1001
5	A ₁₉ -A ₂₂	1.155	1.308	1.2582	1.271	1.26757	0.522068	1.2574
6	A ₂₃ -A ₃₀	0.585	0.519	0.5035	0.509	0.50990	0.519134	0.5107
7	A ₃₁ -A ₃₄	0.100	0.100	0.1000	0.100	0.10000	0.100168	0.1000
8	A ₃₅ -A ₃₆	0.100	0.100	0.1000	0.100	0.10000	0.100145	0.1000
9	A ₃₇ -A ₄₀	0.460	0.514	0.5178	0.485	0.50674	1.296045	0.5252
10	A ₄₁ -A ₄₈	0.530	0.546	0.5214	0.501	0.51651	0.520015	0.5206
11	A ₄₉ -A ₅₂	0.120	0.100	0.1000	0.100	0.10752	0.100047	0.1000
12	A ₅₃ -A ₅₄	0.165	0.109	0.1007	0.100	0.10000	0.100179	0.1002
13	A ₅₅ -A ₅₈	0.155	0.161	0.1566	0.168	0.15618	1.796383	0.1563
14	A ₅₉ -A ₆₆	0.535	0.509	0.5421	0.584	0.54022	0.503644	0.5486
15	A ₆₇ -A ₇₀	0.480	0.497	0.4132	0.433	0.42229	0.100002	0.4156
16	A ₇₁ -A ₇₂	0.520	0.562	0.5756	0.520	0.57941	0.100043	0.5713
Best weight (lb)		383.12	381.91	379.66	380.62	379.97	379.89	379.63
Average weight (lb)		N/A*	N/A*	381.85	382.42	381.56	381.70	380.20
St. dev. (lb)		N/A*	N/A*	1.912	1.380	1.803	0.761	0.088
No. of analyses		N/A*	N/A*	13200	13742	10500	13500	13542

*N/A: Not available

which is a meta-heuristic method introduced by authors, is applied in this article as a global search engine. The statistical results indicate that the best, average and standard deviation of results are improved utilizing the present method. Also, the number of required analysis is comparable with other methods. Although it is not reasonable to compare the results of a non-hybrid method with a hybrid algorithm, the comparison of the present hybrid method with non-hybrid methods shows the advantages of using the hybrid approach.

In this problem, 20 independent runs of the algorithm are executed. The results of the best run, which approaches to the best final weight, are shown in Table 2. Besides, the convergence graph of the best run, entitled “The Best Run”, is shown in Fig. 8. In this figure, the convergence graph, entitled “The Average of Runs”, shows the average of the weight of different runs in each iteration. The figure shows that the result of the best run in some iterations can be worse/better than the average value. It seems that at least there are two reasons: the numerical approximation of \vec{V}_f , \vec{V}_c , which results in some local errors, and the step length, λ , which not only depends on the random variable, β , but on \vec{V}_f , and \vec{V}_c . Such numerical approximations may arise some estimation errors, as was mentioned in the third step

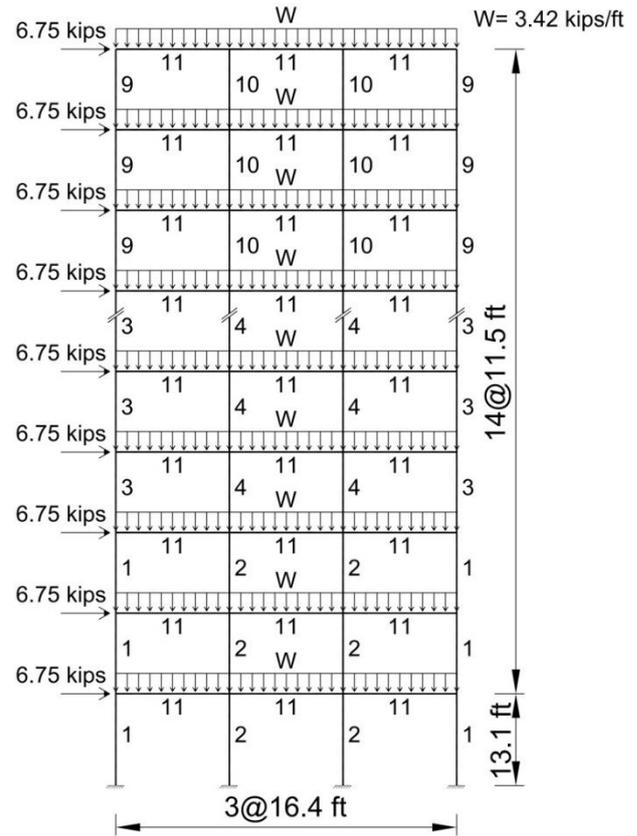


Fig. 9 Schematic of the 3-bay 15-story frame

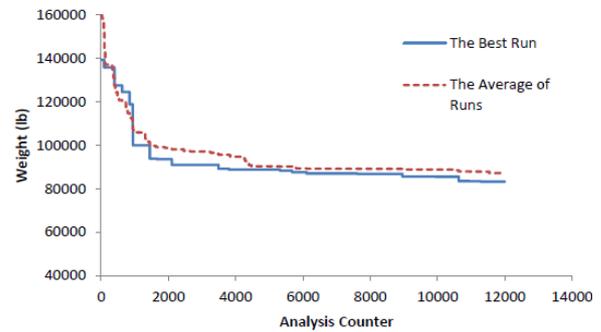


Fig. 10 Convergence graph of the 3-bay 15-story frame problem

of the local search algorithm in Section 3. The best run may be affected by such errors, as well. However, the results show that the method is capable of decreasing such errors and approaching to a better solution in comparison with the other algorithms, considered in this article.

4.2 A 3-bay 15-story frame problem

The optimal weight design of a 3-bay 15-story frame, as depicted in Fig. 9, is considered as the second example (Kaveh and Talatahari 2012). Fig. 9 also shows the external forces and member grouping of the structure. The assigned assumptions of the model are as follows:

- i. The modulus of elasticity and yielding stress of steel are equal to 29 Msi (200 GPa) and 36 ksi (248.2 MPa), respectively.
- ii. The effective length factor of the frame members for

the out-of-plane action is $k_y=1.0$ and for the in-plane action is $k_x \geq 0$, which could be presented in the following equation (Dumonteil 1992)

$$k = \sqrt{\frac{1.6G_A G_B + 4.0(G_A + G_B) + 7.5}{G_A + G_B + 7.5}} \quad (12)$$

Where, G_A and G_B are the ratios of the flexural stiffness of the columns to the stiffness of the beams at each end-joints of the column. The whole length of the columns is assumed as unbraced, but the unbraced length of beams is equal to one-fifth of their length.

iii. The displacement and resistance constraints are considered in accordance with AISC (2001) as the following relations:

a. The maximum lateral displacement

$$\frac{\Delta_T}{H} - R \leq 0 \quad (13)$$

Where, Δ_T is the maximum lateral displacement, H is the total height of the structure. The maximum drift is equal to $1/300$.

b. The lateral displacement between stories

$$\frac{d_i}{h_i} - R_i \leq 0 \quad i = 1, 2, \dots, n_s \quad (14)$$

Where, d_i is the difference between the lateral displacements of two stories, h_i is the height of the i^{th} story, n_s is the total number of stories and R_i is allowable relative drift, which is equal to $1/300$.

c. The resistance constraints

$$\begin{cases} \frac{P_u}{2\phi_c P_n} + \frac{M_u}{\phi_b M_n} - 1 \leq 0, & \text{for } \frac{P_u}{\phi_c P_n} < 0.2 \\ \frac{P_u}{\phi_c P_n} + \frac{8M_u}{9\phi_b M_n} - 1 \leq 0, & \text{for } \frac{P_u}{\phi_c P_n} \geq 0.2 \end{cases} \quad (15)$$

Where, P_u is the axial required strength (in tension and compression), P_n expresses the nominal axial resistance (in tension and compression), ϕ_c is the axial resistance reduction factor ($\phi_c = 0.9$ for tension and $\phi_c = 0.85$ for compression), M_u represents the required flexural strength, M_n is the nominal flexural resistance and ϕ_b indicates the flexural resistance reduction factor, which is equal to $\phi_b = 0.9$.

The nominal tensile strength, in agreement with the yielding of member section, is

$$P_n = A_g \cdot F_y \quad (16)$$

Also, the nominal compressive strength is equal to

$$P_n = A_g \cdot F_{cr} \quad (17)$$

Where

Table 3 Optimization results obtained for the 3-bay 15 story frame

El. group	HPSAC O	HBB-BC	ICA	CSS	ECBO	ES-DE	DSOS	SG	Present Work
1	W21×11	W24×11	W24×11	W21×14	W14×99	W18×10	W16×10	W27×10	W27×102
2	1	7	7	7	6	0	2	2	W18×15
3	8	2	7	3	1	0	2	4	W21×13
4	W10×88	W12×95	W27×84	W12×87	W27×84	W12×79	W12×79	W14×82	W14×82
5	W30×11	W18×11	W27×11	W30×10	W24×10	W27×11	W27×11	W24×10	W27×102
6	6	9	4	8	4	4	4	4	W21×83
7	W21×83	W21×93	W14×74	W18×76	W14×61	W30×90	W21×93	W21×62	W21×62
8	W24×10	W18×97	W18×86	W24×10	W30×90	W10×88	W12×79	W18×71	W21×68
9	3	3	3	3	3	3	3	3	W21×55
10	W21×55	W18×76	W12×96	W21×68	W14×48	W18×71	W21×55	W14×61	W14×61
11	W27×11	W18×65	W24×68	W14×61	W14×61	W18×65	W14×61	W12×53	W12×53
12	4	4	4	4	4	4	4	4	W10×33
13	W10×33	W18×60	W10×39	W18×35	W14×30	W8×28	W14×22	W14×43	W14×43
14	W18×46	W10×39	W12×40	W10×33	W12×40	W12×40	W14×43	W14×43	W8×28
15	W21×44	W21×48	W21×44	W21×44	W21×44	W21×48	W21×48	W21×44	W21×44
Best W. (lb)	95850	97689	93846	92723	86986	93315	91248	84537	83247
Ave. W. (lb)	N/A*	N/A*	N/A*	N/A*	88410	98531	N/A*	88715	87143
St. Dev.	N/A*	N/A*	N/A*	N/A*	N/A*	3294	N/A*	3210	2240
No. Analyses	6800	9900	6000	5000	9000	10000	N/A	12000	12000

*N/A: Not available

$$\begin{cases} F_{cr} = (0.658^{\lambda_c^2}) F_y & \text{for } \lambda_c \leq 1.5 \\ F_{cr} = \left(\frac{0.877}{\lambda_c^2}\right) F_y & \text{for } \lambda_c > 1.5 \end{cases} \quad (18)$$

$$\lambda_c = \frac{kl}{r\pi} \sqrt{\frac{F_y}{E}} \quad (19)$$

In the above equations, A_g is the gross area of the member and k is the effective length factor.

In Table 3, the final result of the present method is compared with some other methods of the literature. The presented method gained the lowest weight of 83247 lb with an average of 12000 analyses in 20 independent runs of the algorithm. In this case, the critical value of constraints is limited to 7.5×10^{-6} and no constraint violation is occurred. Also, the convergence graph of the method is presented in Fig. 10. Star Graph algorithm (SG), which is a meta-heuristic method introduced by authors, is applied in this article as a global search engine. The minimum weight among the other methods is obtained by ECBO (Kaveh and Ilchi Ghazaan 2015), which is equal to 86986 lb with an average of 9000 analyses in 20 independent runs of the algorithm. Table 3 shows that a design weight of 95850 lb has been obtained by HPSACO (Kaveh and Talatahari 2009a), 97689 lb by HBB-BC (Kaveh and Talatahari 2010a), 93846 lb by ICA (Kaveh and Talatahari 2010b), 92723 lb by CSS (Kaveh and Talatahari 2012), 86986 lb by ECBO (Kaveh and Ilchi Ghazaan 2015), 93315 lb by ES-DE (Talatahari *et al.* 2015), 91248 lb by DSOS (Talatahari 2016), and 84537 lb by SG (Asil Gharebaghi *et al.* 2017). The final results indicate that the best and average weight of the structure and the standard deviation of the results are improved using the present method rather than other hybrid and non-hybrid methods.

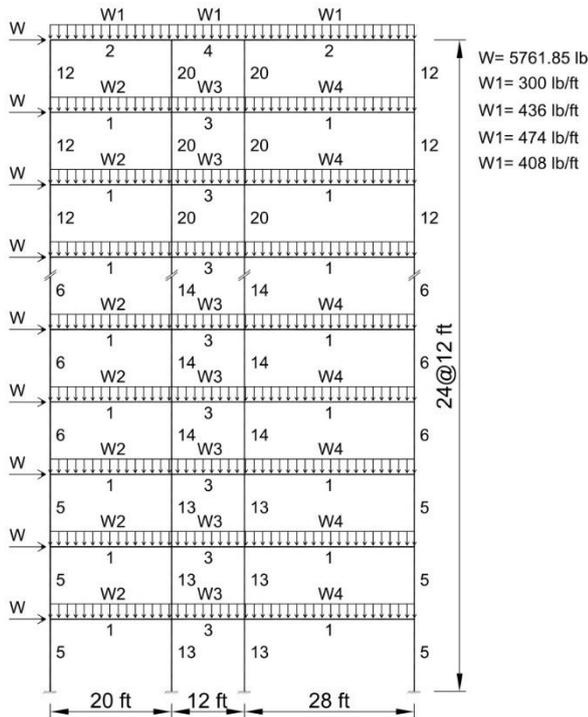


Fig. 11 Schematic of the 3-bay 24-story frame

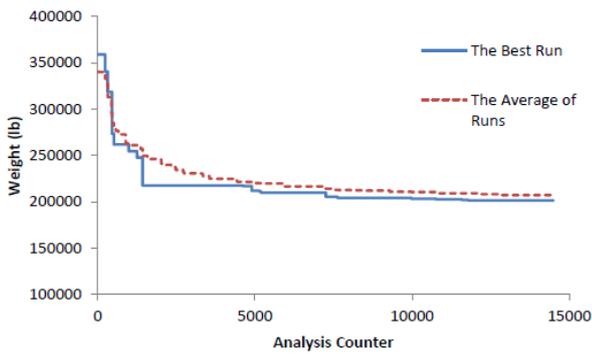


Fig. 12 Convergence graph of the 3-bay 24-story frame problem

4.3 A 3-bay 24-story frame problem

The third example deals with the optimal weight design of a 3-bay 24-story frame, as shown Fig. 11 (Kaveh and Talatahari 2012). The members of the structure are divided into 20 groups: 16 groups for the columns and 4 groups for the beams. Beam sections should be considered as 267W-shapes; however, column sections are limited to W14. The modulus of elasticity and the yield stress of the material are 29.732 Msi (205 GPa) and 33.4 ksi (230.3 MPa), respectively.

The effective length factor of the members for out-of-plane action k_y , is equal to one, and for in-plane action for a sway frame is $k_x \geq 0$. For all members, the unbraced length is equal to the whole length of the element. The resistance and displacement constraints are similar to the previous problem, and the design requirements should be in compliance with AISC (2001) specification.

As shown in Table 4, the problem has been examined by GA (Saka 1998), ACO (Camp *et al.* 2005), HS (Degertekin

Table 4 Optimization results for the 3-bay 24-story frame

El. Group	GA	ACO	HS	CSS	ECBO	ES-DE	DSOS	SG	Present Work
1	838×292×194 UB	W30×90							
2	305×102×25U B	W8×18	W10×22	W21×50	W6×15	W21×55	W21×62	W8×18	W8×13
3	457×191×82U B	W24×55	W18×40	W21×48	W24×55	W21×48	W21×48	W21×48	W24×55
4	305×102×25U B	W8×21	W12×16	W12×19	W6×8.5	W10×45	W21×55	W6×8.5	W6×8.5
5	305×102×25U C	W14×14	W14×17	W14×17	W14×14	W14×14	W14×17	W14×17	W14×145
6	305×368×129 UC	W14×13	W14×17	W14×14	W14×13	W14×10	W14×10	W14×14	W14×132
7	305×305×97U C	W14×13	W14×13	W14×10	W14×99	W14×99	W14×12	W14×99	W14×99
8	356×368×129 UC	W14×13	W14×10	W14×90	W14×90	W14×14	W14×14	W14×82	W14×90
9	305×305×97U C	W14×68	W14×82	W14×74	W14×74	W14×10	W14×61	W14×82	W14×74
10	203×203×71U C	W14×53	W14×74	W14×61	W14×38	W14×48	W14×99	W14×38	W14×38
11	305×305×118 UC	W14×43	W14×34	W14×34	W14×38	W14×38	W14×34	W14×30	W14×38
12	152×152×23U C	W14×43	W14×22	W14×34	W14×22	W14×30	W14×38	W14×26	W14×22
13	305×305×137 UC	W14×14	W14×14	W14×14	W14×99	W14×99	W14×12	W14×90	W14×99
14	305×305×198 UC	W14×14	W14×13	W14×13	W14×99	W14×13	W14×10	W14×99	W14×99
15	356×368×202 UC	W14×12	W14×10	W14×10	W14×99	W14×10	W14×90	W14×99	W14×99
16	356×368×129 UC	W14×90	W14×82	W14×82	W14×82	W14×68	W14×90	W14×90	W14×82
17	356×368×129 UC	W14×90	W14×61	W14×68	W14×68	W14×68	W14×82	W14×61	W14×68
18	356×368×153 UC	W14×61	W14×48	W14×43	W14×61	W14×68	W14×38	W14×61	W14×61
19	203×203×60U C	W14×30	W14×30	W14×34	W14×30	W14×61	W14×38	W14×34	W14×30
20	254×254×89U C	W14×26	W14×22						
Best W. (lb)	251547	220465	214860	212364	201618	212492	209795	202227	201489
Ave. W. (lb)	N/A*	229555	222620	215226	209644	N/A*	N/A*	212734	207170
St. Dev.	N/A*	4561	N/A*	2448	N/A*	N/A*	N/A*	3450	2212
No. analyses	30000	15500	13924	5500	15360	12500	7500	14000	14500

*N/A: Not available

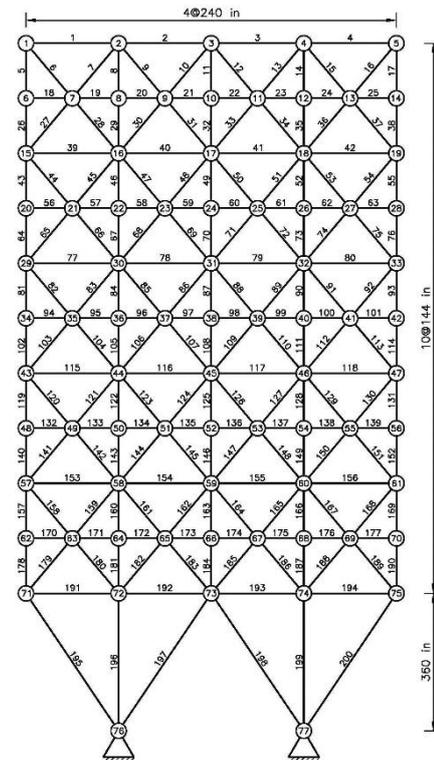


Fig. 13 Schematic of the planar 200-bar truss

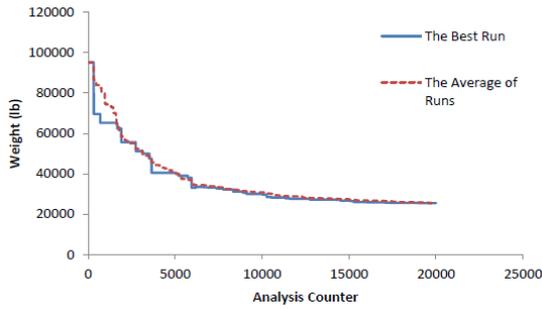


Fig. 14 Convergence graph of the planar 200-bar truss problem

Table 5 Optimization results for the planar 200-bar truss

Groups	Members	Optimal cross-sectional areas (in ²)				
		CSP	TLBO	SAHS	SG	Present work
1	1, 2, 3, 4	0.1480	0.1460	0.1540	0.1480	0.1470
2	5, 8, 11, 14, 17	0.9460	0.9410	0.9410	0.9460	0.9400
3	19, 20, 21, 22, 23, 24	0.1010	0.1000	0.1000	0.1000	0.1000
4	18, 25, 56, 63, 94, 101, 132, 139, 170, 177	0.1010	0.1010	0.1000	0.1000	0.1000
5	26, 29, 32, 35, 38	1.9461	1.9410	1.9420	1.9460	1.9440
6	6, 7, 9, 10, 12, 13, 15, 16, 27, 28, 30, 31, 33, 34, 36, 37	0.2979	0.2960	0.3010	0.2979	0.2966
7	39, 40, 41, 42	0.1010	0.1000	0.1000	0.1001	0.1000
8	43, 46, 49, 52, 55	3.1072	3.1210	3.1080	3.1060	3.1059
9	57, 58, 59, 60, 61, 62	0.1010	0.1000	0.1000	0.1010	0.1000
10	64, 67, 70, 73, 76	4.1062	4.1730	4.1060	4.1052	4.1050
11	44, 45, 47, 48, 50, 51, 53, 54, 65, 66, 68, 69, 71, 72, 74, 75	0.4049	0.4010	0.4090	0.4049	0.4033
12	77, 78, 79, 80	0.1944	0.1810	0.1910	0.1869	0.1921
13	81, 84, 87, 90, 93	5.4299	5.4230	5.4280	5.4288	5.4277
14	95, 96, 97, 98, 99, 100	0.1010	0.1000	0.1000	0.1009	0.1000
15	102, 105, 108, 111, 114	6.4299	6.4220	6.4270	6.4288	6.4277
16	82, 83, 85, 86, 88, 89, 91, 92, 103, 104, 106, 107, 109, 110, 112, 113	0.5755	0.5710	0.5810	0.5748	0.5738
17	115, 116, 117, 118	0.1349	0.1560	0.1510	0.1349	0.1339
18	119, 122, 125, 128, 131	7.9747	7.9580	7.9730	7.9718	7.9718
19	133, 134, 135, 136, 137, 138	0.1010	0.1000	0.1000	0.1010	0.1
20	140, 143, 146, 149, 152	8.9747	8.9580	8.9740	8.9747	8.9718
21	120, 121, 123, 124, 126, 127, 129, 130, 141, 142, 144, 145, 147, 148, 150, 151	0.7064 8	0.7200	0.7190	0.7060	0.7055
22	153, 154, 155, 156	0.4225	0.4780	0.4220	0.4221	0.421
23	157, 160, 163, 166, 169	10.868 5	10.897 0	10.892 0	10.866 2	10.8665
24	171, 172, 173, 174, 175, 176	0.1010	0.1000	0.1000	0.1010	0.1000
25	178, 181, 184, 187, 190	11.868 4	11.897 0	11.887 0	11.866 1	11.8665
26	158, 159, 161, 162, 164, 165, 168, 179, 180, 182, 183, 185, 186, 188, 189	1.0359 99	1.0800	1.0400	1.0358	1.0348
27	191, 192, 193, 194	6.6859	6.4620	6.6460	6.6858	6.6848
28	195, 197, 198, 200	10.8111 0	10.799 0	10.804 0	10.8111 0	10.81
29	196, 199	13.846 49	13.922 0	13.870 0	13.845 0	13.845
Best W. (lb)		25467.9	25488.15	25491.05	25461.05	25452.98
Ave. W. (lb)		25547.6	25533.14	25610.2	25537.87	25482.32
St. Dev.		135.09	27.44	141.85	23.59	20.12
No. Analyses		31700	28059	19670	20000	20000

and Ilchi Ghazaan 2015), ES-DE (Talatahari *et al.* 2015), DSOS (Talatahari 2016), and SG (Asil Gharebaghi *et al.* 2017). The optimum weight of the present method is equal to 201489 lb with an average of 14500 analyses for 20 independent runs of the algorithm. In this case, the critical value of constraints is limited to 1.2×10^{-5} and no constraint violation is occurred. The convergence graph of the method is presented in Fig. 12. The best result among the other methods belongs to ECBO (Kaveh and Ilchi Ghazaan 2015). Its optimal weight is equal to 201618 lb with an average of 15360 analyses. This indicates that the best and average weight of the frame and the standard deviation of the method are improved using the results of the present research.

4.4 A Planar 200-bar truss problem

The optimal design of a planar 200-bar truss, shown in Fig. 13, is considered. The elements of the truss are divided into 29 groups as shown in Table 5. The material density and modulus of elasticity are considered as 0.283 lb/in³ and 30 Msi, respectively. The stress should not exceed 10 ksi in the compressive or the tensile elements. There are three independent loading conditions: (1) 1.0 kip acting in the positive x-direction at nodes 1, 6, 15, 20, 29, 34, 43, 48, 57, 62, and 71; (2) 10 kips acting in the negative y-direction at nodes 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22, 24, ... , 71, 72, 73, 74, and 75; and (3) conditions 1 and 2 acting together.

In Table 5, the optimal results of CSP (Kaveh *et al.* 2014), TLBO (Degertekin 2013), SAHS (Degertekin 2012) and SG (Asil Gharebaghi *et al.* 2017) are compared with the presented method. The best result, among all the other methods, belongs to SG with weight of 25461.05 lb, and an average of 20,000 analyses of the structure. However, in the present method, the weight of 25452.98 lb has been obtained using the same number of analyses. In this case, the critical value of constraints is limited to 1.2×10^{-6} and no constraint violation is occurred. The convergence graph of this method is presented in Fig. 14. Besides, the improvement of the results of Path Projection method in comparison with SG and the other meta-heuristic methods indicates the performance of the resulted hybrid algorithm.

5. Conclusions

In this article, a new hybrid method has been introduced to reinforce the local search ability of some meta-heuristic algorithms. Herein, this method is applied to a newly developed meta-heuristic algorithm, entitled Star Graph (SG). In the case of constraint violation, Path Projection Method suggests a new path, which effectively improves the convergence process. Conceptually, the classic methods have a significant capacity in the local search; hence, the hybrid classic-meta heuristic method can be successful in both local and global searches. Path Projection Method has been developed based on the concept of Gradient Projection Method. However, the algebraic equations have been replaced with some new approximated geometrical counterparts. Therefore, not only the capability of the

classic method has been maintained, but the calculation cost has been decreased. Also, the limitations such as difficulties in the gradient calculation have been eliminated. Path Projection Method consists of three main steps: (1) the identification of the active constraint, (2) finding the neighboring point, and (3) the determination of the direction and step length of the agent's movement. In this study, the suggested method has been applied to the optimal design of truss and frame structures. The results are compared with some hybrid and non-hybrid methods of the literature. Although it is not good to compare a hybrid method with a non-hybrid algorithm, the comparison shows that the presented hybrid method has a noticeable performance. In fact, utilizing hybrid features may improve the performance of non-hybrid methods. The results show that the presented algorithm outperforms some new hybrid method.

References

- Abbasnia, R., Shayanfar, M. and Khodam, A. (2014), "Reliability-based design optimization of structural systems using a hybrid genetic algorithm", *Struct. Eng. Mech.*, **52**, 1099-1120.
- AISC (2001), *Manual of Steel Construction: Load and Resistance Factor Design*, American Institute of Steel Construction, U.S.A.
- Asil Gharebaghi, S., Kaveh, A. and Ardalan Asl, M. (2017), "A new meta-heuristic optimization algorithm using star graph", *Smart Struct. Syst.*, **20**(1), 99-114.
- Camp, C.V., Bichon, B.J. and Stovall, S. (2005), "Design of steel frames using ant colony optimization", *J. Struct. Eng.*, **131**(3), 369-379.
- Cheng, M.Y. and Prayogo, D. (2014), "Symbiotic organisms search: A new metaheuristic optimization algorithm", *Comput. Struct.*, **139**, 98-112.
- Cheng, M.Y. and Prayogo, D. (2017), "A novel fuzzy adaptive teaching-learning-based optimization (FATLBO) for solving structural optimization problems", *Eng. Comput.*, **33**(1), 55-69.
- Cheng, M.Y., Prayogo, D., Wu, Y.W. and Lukito, M.M. (2016), "A hybrid harmony search algorithm for discrete sizing optimization of truss structure", *Automat. Constr.*, **69**, 21-33.
- Degertekin, S.O. (2008), "Optimum design of steel frames using harmony search algorithm", *Struct. Multidiscipl. Optim.*, **36**(4), 393-401.
- Degertekin, S.O. (2012), "Improved harmony search algorithms for sizing optimization of truss structures", *Comput. Struct.*, **92-93**, 229-241.
- Degertekin, S.O. (2013), "Sizing truss structures using teaching-learning-based optimization", *Comput. Struct.*, **119**, 177-188.
- Dumonteil, P. (1992), "Simple equations for effective length factors", *Eng. J.*, **29**(3), 111-115.
- Epitropakis, M.G., Plagianakos, V.P. and Vrahatis, M.N. (2012), "Evolving cognitive and social experience in particle swarm optimization through differential evolution: A hybrid approach", *Inf. Sci.*, **216**, 50-92.
- Erbatur, F., Hasançebi, O., Tütüncü, I. and Kiliç, H. (2000), "Optimal design of planar and space structures with genetic algorithms", *Comput. Struct.*, **75**, 209-224.
- Erol, O.K. and Eksin, I. (2006), "A new optimization method: big bang-big crunch", *Adv. Eng. Softw.*, **37**(2), 106-111.
- Giftson Samuel, G. and Christofer Asir Rajan, C. (2015), "Hybrid: Hybrid: Particle swarm optimization-genetic algorithm and particle swarm optimization-shuffled frog leaping algorithm for long-term generator maintenance scheduling", *J. Electr. Pow. Energy Syst.*, **65**, 432-442.
- Haddar, B., Khemakhem, M., Hanafi, S. and Wilbaut, C. (2016), "A hybrid quantum particle swarm optimization for the multidimensional knapsack problem", *Eng. Appl. Artif. Intell.*, **55**, 1-13.
- Hanson, M.A. (1981), "On sufficiency of the kuhn-tucker conditions", *J. Math. Anal. Appl.*, **80**(2), 545-550.
- Hoseini, P. and Shayesteh, M.G. (2013), "Efficient contrast enhancement of images using hybrid ant colony optimisation, genetic algorithm, and simulated annealing", *Digit. Sign. Proc.*, **23**(3), 879-893.
- Jaradat, G., Ayob, M. and Almarashdeh, I. (2016), "The effect of elite pool in hybrid population-based meta-heuristics for solving combinatorial optimization problems", *Appl. Soft Comput.*, **44**, 45-56.
- Jeslin Drusila Nesamalar, J., Venkatesh, P. and Charles Raja, S. (2016), "Managing multi-line power congestion by using hybrid Nelder-Mead-Fuzzy adaptive particle swarm optimization (HNM-FAPSO)", *Appl. Soft Comput.*, **43**, 222-234.
- Kaveh, A. and Ilchi Ghazaan, M. (2015), "A comparative study of CBO and ECBO for optimal design of skeletal structures", *Comput. Struct.*, **153**, 137-147.
- Kaveh, A. and Ilchi Ghazaan, M. (2018), "A new hybrid meta-heuristic algorithm for optimal design of large-scale dome structures", *Eng. Optim.*, **50**(2), 235-252.
- Kaveh, A. and Javadi, S.M. (2014), "An efficient hybrid particle swarm strategy, ray optimizer, and harmony search algorithm for optimal design of truss structures", *Period. Polytech.*, **58**(2), 65-81.
- Kaveh, A. and Laknejadi, K. (2013), "A hybrid evolutionary graph based multi-objective algorithm for layout optimization of truss structures", *Acta Mech.*, **224**, 343-364.
- Kaveh, A. and Mahdavi, V.R. (2013), "Optimal design of structures with multiple natural frequency constraints using a hybridized BB-BC/Quasi-Newton algorithm", *Period. Politech.*, **57**(1), 1-12.
- Kaveh, A. and Shahrouzi, M. (2008), "Dynamic selective pressure using hybrid evolutionary and ant system strategies for structural optimization", *J. Numer. Meth. Eng.*, **73**(4), 544-563.
- Kaveh, A. and Talatahari, S. (2009a), "Hybrid algorithm of harmony search, particle swarm and ant colony for structural design optimization", *Stud. Comput. Intellig.*, **239**, 159-198.
- Kaveh, A. and Talatahari, S. (2009b), "Size optimization of space trusses using big bang-big crunch algorithm", *Comput. Struct.*, **87**, 1129-1140.
- Kaveh, A. and Talatahari, S. (2009c), "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures", *Comput. Struct.*, **87**(5), 267-283.
- Kaveh, A. and Talatahari, S. (2010a), "A discrete big bang-big crunch algorithm for optimal design of skeletal structures", *Asian J. Civil Eng.*, **11**(1), 103-122.
- Kaveh, A. and Talatahari, S. (2010b), "Optimum design of skeletal structures using imperialist competitive algorithm", *Comput. Struct.*, **88**, 1220-1229.
- Kaveh, A. and Talatahari, S. (2012), "Charged system search for optimal design of frame structures", *Appl. Soft Comput.*, **12**(1), 382-393.
- Kaveh, A., Bakhshpoori, T. and Afshari, E. (2015), "Hybrid PSO and SSO algorithm for truss layout and size optimization considering dynamic constraints", *Struct. Eng. Mech.*, **54**(3), 453-474.
- Kaveh, A., Sheikholeslami, R., Talatahari, S. and Keshvari-Ilkhichi, M. (2014), "Chaotic swarming of particles: A new method for size optimization of truss structures", *Adv. Eng. Softw.*, **67**, 136-147.
- Kelner, V., Capitanescu, F., Leonard, O. and Wehenkel, L. (2008), "A hybrid optimization technique coupling an evolutionary and a local search algorithm", *J. Comput. Appl. Math.*, **215**(2), 448-

- 456.
- Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983), "Optimization by simulated annealing", *Sci.*, **220**(4598), 671-680.
- Li, L.J., Huang, Z.B. and Liu, F. (2009), "A heuristic particle swarm optimization method for truss structures with discrete variables", *Comput. Struct.*, **87**(7), 435-443.
- Liu, J., Zhang, S., Wu, C., Liang, J., Wang, X. and Teo, K.L. (2016), "A hybrid approach to constrained global optimization", *Appl. Soft Comput.*, **47**, 281-294.
- Luenberger, D.G. (1973), *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, Mass.
- Ma, H., Simon, D., Fei, M., Shu, X. and Chen, Z. (2014), "Hybrid biogeography-based evolutionary algorithms", *Eng. Appl. Artif. Intell.*, **30**, 213-224.
- Nayanatara, C., Baskaran, J. and Kothari, D.P. (2016), "Hybrid optimization implemented for distributed generation parameters in a power system network", *J. Electr. Pow. Energy Syst.*, **78**, 690-699.
- Ouyang, H., Gao, L., Kong X., Li, S. and Zou, D. (2016), "Hybrid harmony search particle swarm optimization with global dimension selection", *Inf. Sci.*, **346**, 318-337.
- Perez, R.E. and Behdinan, K. (2007), "Particle swarm approach for structural design optimization", *Comput. Struct.*, **85**, 1579-1588.
- Prayogo, D., Cheng, M.Y., Wu, Y.W., Herdany, A.A. and Prayogo, H. (2018), "Differential big bang-big crunch algorithm for construction-engineering design optimization", *Automat. Constr.*, **85**, 290-304.
- Rahami, H., Kaveh, A., Aslani, M. and Najian Asl, R. (2011), "A hybrid modified genetic-nelder mead simplex algorithm for large-scale truss optimization", *J. Optim. Civil Eng.*, **1**(1), 29-46.
- Rosen, J.B. (1960), "The gradient projection method for nonlinear programming. Part I. Linear constraints", *J. Soc. Industr. Appl. Math.*, **8**(1), 181-217.
- Rosen, J.B. (1961), "The gradient projection method for nonlinear programming. Part II. Nonlinear constraints", *J. Soc. Industr. Appl. Math.*, **9**(4), 514-532.
- Saka, M.P. and Kameshki, E.S. (1998), "Optimum design of multi-story sway steel frames to bs5950 using genetic algorithm", *Proceedings of the 4th International Conference on Computational Structures Technology*, Edinburgh, Scotland, U.K.
- Shao, W., Pi, D. and Shao, Z. (2016), "A hybrid discrete optimization algorithm based on teaching-probabilistic learning mechanism for no-wait flow shop scheduling", *Know-Bas. Syst.*, **107**, 219-234.
- Soleimani, H. and Kannan, G. (2015), "A hybrid particle swarm optimization and genetic algorithm for closed-loop supply chain network design in large-scale networks", *Appl. Math. Modell.*, **39**(14), 3990-4012.
- Talatahari, S. (2016), "Symbiotic organisms search for optimum design of and grillage system", *Asian J. Civil Eng.*, **17**(3), 299-313.
- Talatahari, S., Gandomi, A.H., Yang, X.S. and Deb, S. (2015), "Optimum design of frame structures using the eagle strategy with differential evolution", *Eng. Struct.*, **91**, 16-25.
- Tuba, M. and Bacanin, N. (2014), "Improved seeker optimization algorithm hybridized with firefly algorithm for constrained optimization problems", *Neurocomput.*, **143**, 197-207.
- Wang, H., Sun, H., Li, C., Rahnamayan, S. and Pan, J.S. (2013), "Diversity enhanced particle swarm optimization with neighborhood search", *Inf. Sci.*, **223**, 119-135.
- Wang, J., Yuan, W. and Cheng, D. (2015), "Hybrid genetic-particle swarm algorithm: An efficient method for fast optimization of atomic clusters", *Comput. Theoret. Chem.*, **1059**, 12-17.
- Wu, J., Long, J. and Liu, M. (2015), "Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm", *Neurocomput.*, **148**, 136-142.

CC

Appendix A. The root of line equation in vector space

The index i is the counter of variables of the vector \vec{X} .

For simplification purpose only, if $\vec{X}(x, y, z)$ is a vector in three-dimensional space, the line equation will be expressed as Eq. (A1)

$$\vec{PX} = \lambda_0 \vec{U} \tag{A1}$$

$$(x - x_0, y - y_0, z - z_0) = \lambda_0 (u_1, u_2, u_3) \tag{A2}$$

Where, $A(x_0, y_0, z_0)$ is a given point on the line. The \vec{U} is direction cosine of the line and can be defined by two given points $(A(x_0, y_0, z_0), B(x_1, y_1, z_1))$

$$\begin{aligned} \vec{U} = (u_1, u_2, u_3) &= \frac{\vec{P}_1 - \vec{P}_0}{|\vec{P}_1 - \vec{P}_0|} = \\ &= \frac{(x_1 - x_0, y_1 - y_0, z_1 - z_0)}{|x_1 - x_0, y_1 - y_0, z_1 - z_0|} \end{aligned} \tag{A3}$$

To find a point on the line which satisfies $z = 0$. Eq. (A3) is substituted in Eq. (A2)

$$z = 0 \rightarrow \lambda_0 = -\frac{z_0}{u_3} \tag{A4}$$

Thus, this point can be easily obtained

$$(x, y, z) = (x_0 - \frac{u_1}{u_3} z_0, y_0 - \frac{u_2}{u_3} z_0, 0) \tag{A5}$$

Using Eqs. (A3) and (A5)

$$(x, y, z) = (x_0 - (\frac{x_1 - x_0}{z_1 - z_0}) z_0, y_0 - (\frac{y_1 - y_0}{z_1 - z_0}) z_0, 0) \tag{A6}$$

Assuming that the points A and B are defined in vector space $\vec{S} = [x_1 \ x_2 \ \dots \ x_n \ z]$ with $n + 1$ variables

$$S_A = [x_1^A \ x_2^A \ \dots \ x_n^A \ z^A] \tag{A7}$$

$$S_B = [x_1^B \ x_2^B \ \dots \ x_n^B \ z^B] \tag{A8}$$

Where, z can be the value of a function on the variable $\vec{X} = [x_1 \ x_2 \ \dots \ x_n]$ in the vector space. The root of this function ($z = 0$) can be expressed as Eq. (A9)

$$x_i = x_i^A - \frac{z^A}{m_i} \tag{A9}$$

$$m_i = \frac{z^B - z^A}{x_i^B - x_i^A} \tag{A10}$$