Smart Structures and Systems, *Vol. 20, No. 1 (2017) 99-114*
DOI: https://doi.org/10.12989/sss.2017.20.1.099

A new meta-heuristic optimization algorithm using star graph

Saeed Asil Gharebaghi*[1], Ali Kaveh[2] and Mohammad Ardalan Asl[1]

[1]Department of Civil Engineering, K. N. Toosi University of Technology, Tehran, Iran
[2]Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran

(Received November 7, 2016, Revised May 20, 2017, Accepted May 23, 2017)

Abstract. In cognitive science, it is illustrated how the collective opinions of a group of individuals answers to questions involving quantity estimation. One example of this approach is introduced in this article as Star Graph (SG) algorithm. This graph describes the details of communication among individuals to share their information and make a new decision. A new labyrinthine network of neighbors is defined in the decision-making process of the algorithm. In order to prevent getting trapped in local optima, the neighboring networks are regenerated in each iteration of the algorithm. In this algorithm, the normal distribution is utilized for a group of agents with the best results (guidance group) to replace the existing infeasible solutions. Here, some new functions are introduced to provide a high convergence for the method. These functions not only increase the local and global search capabilities but also require less computational effort. Various benchmark functions and engineering problems are examined and the results are compared with those of some other algorithms to show the capability and performance of the presented method.

Keywords: meta-heuristic algorithm**;** global optimization; graph theory; optimal design; truss structures; frame structures

1. Introduction

For decades, extensive researches have been conducted to develop methods for finding the optimum solution of engineering problems. The gradient based classical methods are not capable of solving new problems, especially when the derivatives of the corresponding fitness function do not exist. Therefore, the meta-heuristic methods which use natural concepts have been introduced as the new efficient solutions. These methods are inspired by certain laws from nature. Similar to the radar, which has been invented based on the behavior of a bat, new optimization methods are defined using such natural events (Yang 2010). Dorigo *et al*. (1996) used ant colony behavior, and Eberhart and Kennedy (1995) utilized birds' immigration to find the best solution in the search domain for mathematical problems.

The modeling of natural phenomenon in conjunction with stochastic laws is a common approach in developing the meta-heuristic algorithms (Lee and Geem 2005). These methods use a natural phenomenon as an idea to provide a new optimization algorithm. For example, Goldberg (1989) proposed Genetic Algorithm (GA), which is based on evolutionary biological process. Kennedy and Eberhart (1995) introduced Particle Swarm Optimization (PSO) according to the birds' migration. Kaveh and Khayatazad (2012) introduced Ray Optimization based on the Snell's light refraction law. In this method, agents are considered as rays of light and when light travels from a lighter medium to a darker medium, it refracts and its direction changes. Sadollah *et al.* (2012) proposed an optimization method derived from the explosion of mine bombs which is called mine blast algorithm (MBA).

Additionally, Kaveh and Mahdavi (2014) proposed a novel meta-heuristic algorithm called Colliding Bodies Optimization (CBO) which works based on one-dimensional collisions between bodies, and Mirjalili (2015) worked on a nature-inspired algorithm called Ant Lion Optimizer (ALO) which mimics the hunting mechanism of ant lions in nature. Also, a new technique of optimal analysis for applying in optimal design of cyclically repeated space trusses with frequency constraints is introduced by Kaveh and Zolghadr (2016).

Some new studies have been carried out to apply the graph theory in engineering optimization. Sharafi *et al.* (2014a,b) presented an intuitive procedure for the shape and sizing optimizations of open and closed thin-walled steel sections using graph theory. A bi-objective approach has been utilized in this article and the algorithm is going to find the shapes of optimum mass and strength. Wang (2015) studied a topology optimization problem by leveraging a regular network topology-circulant graph. Also, Liu and Kozan (2016) introduced two new algorithms based on network flow graph and conjunctive graph theory.

Recently, Kaveh and Ghafari (2016) worked on the optimum design of steel floor systems, also, Kaveh and Moradveisi (2016) studied the nonlinear analysis based optimal design of double-layer grids. Additionally, Kaveh and Shokohi (2016) introduced a new optimum design of laterally-supported castellated beams based on tug of war optimization algorithm. Recently, some meta-heuristic

∗Corresponding author, Assistant Professor
E-mail: asil@kntu.ac.ir

http://www.techno-press.com/journals/sss&subpage=7 ISSN: 1738-1584 (Print), 1738-1991 (Online)

methods have been applied to different types of structural engineering problems by Saka (2014), Akin and Saka (2015), and Saka *et al*. (2016).

The present article introduces a meta-heuristic algorithm based on an efficient dynamic neighboring pattern which is called Star Graph to find the global minimum of engineering problems. In each iteration of the algorithm, agents of the search space regenerate a new star graph and transmit information, based on this pattern, to provide a new feasible solution. This communication pattern has different manifestations in social and natural phenomenon. Communication pattern of individuals in an election competition and the pattern of co-operation in consulting groups are good examples of this approach.

In the following section, first the concept of Star Graph is introduced. Then, the flowchart and components of the algorithm are illustrated. At the end, some constrained and unconstrained benchmarks are optimized and the results of the algorithm are compared with those of some other meta-heuristics.

1.1 Definition of Star Graph

A graph is defined as a set of nodes and a set of edges together with a relation of incidence which associate a pair of nodes with an edge. The pattern of connections and the weight or the direction of the edges describes the characteristics of a graph. The edges of a graph can have directions or weights. There are different types of graphs, including Trees, Stars, Paths, and Cycle Graphs. A Star Graph is defined as a sub-graph (tree) with k+1 nodes with one node having vertex degree k and the other nodes having vertex degree 1 (Fig. 1). The degree of a node is the number of edge connected to that node.

In this article, the star graph is a weighted and directed graph, which describes the details of relation between neighbors and central node. Fig. 1 shows a graph, consisting of 19 nodes and their connections. The star graph of node 1 which consists of nodes 2, 3, 4, 5 and the incident edges to node 1 is shown in bold in the figure. This sub-graph can be called "k-star graph of node 1". The parameter k in this indicates the number of edges which are connected to the central node. Hence, in this case, the bolded sub-graph can be called "4-star graph of node 1". In this figure, it can be seen that each node has its own star graph. In fact, k is the number of neighbors which is connected to central agent.

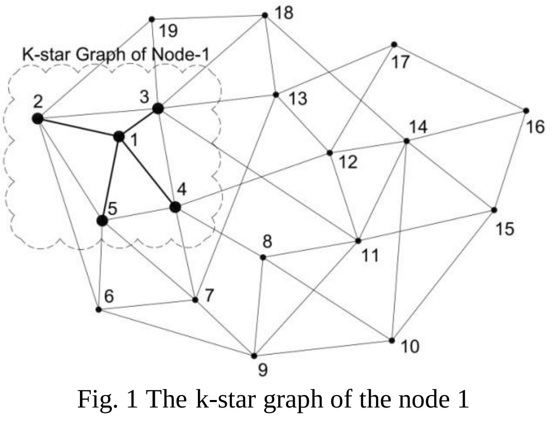

Fig. 1 The k-star graph of the node 1

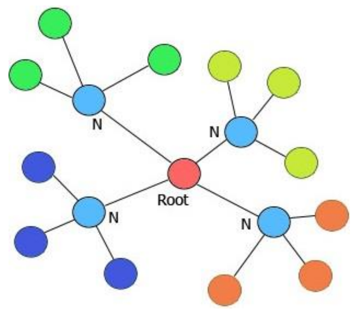

Fig. 2 Topology of a star graph

In an optimization problem, each node can be considered as a search agent and the edges describe the relations between these agents. In Star Graph approach, the central node is the i^{th} agent who is going to find the new solution. The other agents in the star graph are called Neighbors and are going to help the i^{th} agent in the search process.

The topology of the Star Graph method is such a tree (Fig. 2) which the i^{th} agent works as a root and the branches of the tree are the connections of the neighbors. In this method, the information is transmitted between neighbors and the central agent according to the pattern of Star Graph. In fact this group of agents can be considered as a consulting group to find a new solution for the optimization problem. The details of the process including selecting neighbors, gathering information and making a new decision are described in the next section.

In Fig. 2, nodes which are entitled "N" are the neighbors of the node which is called "Root". Similarly, each neighbor has its own n-star graph.

The star graph may be considered as a communication pattern among intellectual individuals. For example, the pattern of communication among the members of a society, during an election competition, introduces a model of effective collaboration between individuals who want to vote for candidates or parties in order to make a better future. Herein, this model is utilized to provide an efficient optimization algorithm.

In star graph method, each node has some direct connections with its neighbors and has indirect connection with other nodes. Therefore, if these nodes can transfer some information to others, each node will be able to be informed of all conditions of the search space provided by its neighbors. This concept will decrease the data transmission of the algorithm. Since, the neighbors have most similarity and compatibility with their central node, the convergence of the method will be improved.

After an agent receives data provided by some neighbors, the information should be analyzed providing a new result. If the new result is acceptable, the agent will send this information to agents who are associated with its star graph.

In optimization problems, the criterion of improvement can be considered as a decrease of the fitness function. The process of neighbors' selection, analysis and providing feasible solution should be repeated for all the agents in each iteration. In this algorithm, each agent may affect the convergence path of others even if they are not connected directly.

In the next sections, the components of the algorithm are illustrated and then some numerical examples are examined to show the efficiency and performance of the algorithm.

2. Star Graph algorithm

In the Star Graph algorithm, the agents are considered as nodes and relations among the agents are taken as the edges of the star graph. The concept of star graph is inspired as a decision making process in some social and natural populations.

At the first stage of the algorithm, search agents are randomly distributed in a valid search space. Next, the fitness function value of all agents is determined. The new solution can be considered as the i^{th} agent in the next iteration. Therefore, finding a new feasible location of the i^{th} agent for the next iteration is the purpose of this method. In each iteration of the algorithm, if the star graph method finds the new feasible solution, the algorithm may converge to the global minimum.

In the algorithm, each agent informs the neighbors about its location and its fitness function value for the guidance of the agents. The interaction between each agent and the population can only be transmitted by the neighbors. This interaction is in accordance with the following procedure:

- The neighbors send their information, including the location and the fitness function value, to the i^{th} agent.
- If the fitness function value of the j^{th} neighbor is less than its counterpart of the i^{th} agent, the i^{th} agent will tend to approach the j^{th} neighbor.
- Conversely, if the fitness function value of the j^{th} neighbor is greater than the i^{th} agent's, the i^{th} agent will tend to get away from the j^{th} neighbor.

In this way, the i^{th} agent moves from a location with a higher fitness function value to a location with a lower one. It indicates that the star graph is directed and the direction of the edges tends to decrease the fitness function value.

If the fitness function value of the j^{th} neighbor is considerably less than the other neighbors, the i^{th} agent should tend towards it with the corresponding magnitude. Therefore, the method is capable of moving the agent to the better location using the weighted function. This indicates that the edges of the star graph are weighted based on the fitness function value of neighbors. As a result, the j^{th} neighbor moves the i^{th} agent to a weighted path. The normalized resultant vector of weighted paths (the bold black arrow in Fig. 3) shows the new suggested path for the i^{th} agent.

In this method, the new location of an agent in each iteration depends on the history of process. It means the data of the current iteration participates in determining the new location of the agent in the next iteration. It decreases the sensitivity of the method to the variation of neighbors. Hence, the information of each iteration should be saved for the next iteration.

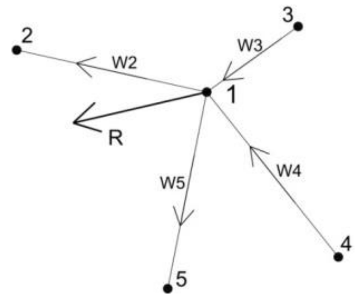

Fig. 3 Star graph of the 1st agent

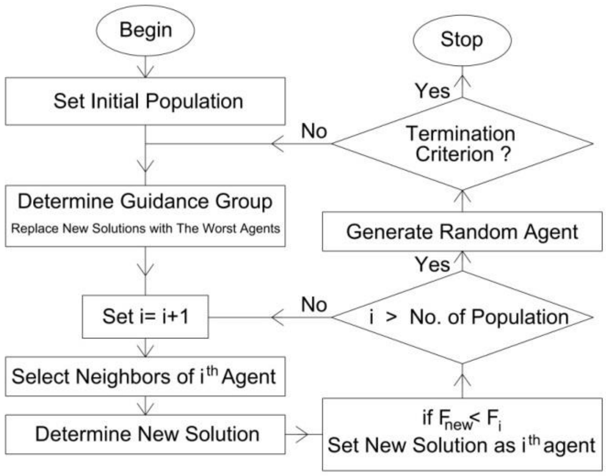

Fig. 4 Flowchart of the star graph algorithm

The star graph method is regenerated in each iteration of the algorithm and the new star graph of each agent is reselected in each iteration. The current version of the algorithm uses a weighted random function to select the new neighbors. In fact, the random selection approach decreases the probability of getting trapped in local minima. In other words, if the i^{th} agent is going to be trapped in local minima, selecting new neighbors and getting new information help to change the path and converge to the global minimum.

In Fig. 3, the star graph of 1^{st} agent and direction and weight of edges are shown. The direction of edges is intended to decrease the fitness function value. So, the resultant vector (R) indicates the appropriate direction of 1^{st} agent to decrease the fitness function value.

More precisely, in the first step of each iteration, the new neighbors of the *i*th agent are randomly selected. Then, the new location of the *i*th agent is obtained using fitness function value of the neighbors as aforementioned. Consequently, if the new location is better than the previous one, it will be considered as the new location of the *i*th agent in the next iteration. The flowchart of star graph algorithm is presented in Fig. 4.

In the next section, the components of the SG algorithm and its procedure are illustrated step by step. In brief, the presented algorithm consists of four subroutines or functions, which create an adaptive structure to conduct the agents to the global minimum efficiently.

2.1 Details of Star Graph algorithm

The procedure of the SG algorithm, in accordance with the flowchart of Fig. 4, can be presented as follows.

Step 1. Initial Population Generation

The number of agent population and neighbors could be changed in each step of the algorithm. A dynamic number of the population is required in the algorithm process. Actually, in some first iterations SG needs the maximum number of population; however, the number of population begins to decrease when the iteration number increases, or when the algorithm begins to converge to the final result. The dynamic size of population decreases the cost of calculation while the effects on the convergence of the algorithm are negligible. At its simplest form, a linear function is utilized to determine the number of population (N_n) and neighbors (N_b), based on the number of iterations. The first generation is distributed using the following equation

$$\overrightarrow{X^1} = \vec{a} + (\vec{b} - \vec{a}) \times rand \quad (1)$$

Where, $\vec{a}$ and $\vec{b}$ are vectors containing boundaries of variables, and *rand* is a uniform random number generator, which generate real numbers in the range of [0, 1]. The vector $\overrightarrow{X^1}$ is the location of the agent in the first iteration of the algorithm. The superscript shows the iteration number.

After first population generation, its corresponding fitness function should be saved in *F*.

Step 2. Guidance Group Function

The aim of this step is to utilize the pattern of those agents with the best results. This set of agents is called guidance group.

During the process of the algorithm, some agents have no acceptable results and could not converge to a better solution. They increase the cost of computation while have no positive effect on the convergence. A new function is introduced herein, which moves these agents to some suitable locations. This process improves the performance of the algorithm.

New locations are determined using the mean and standard deviation of the location of the guidance group. Although in the first step of the algorithm agents were randomly distributed using the uniform random distribution, in the second step the agents distribute according to the normal distribution data, which is based on the guidance group data. The guidance group (i_m) consists of *m* agents with the best results. The parameter *m* can be between 3 and 10. The new locations are used in order to relocate the agents with the worst results (i_{ma}) in the next iteration.

In simple words, in each iteration, the algorithm estimates the location of the agents with the best results to move the agents with the worst results to some better locations. The applied functions are shown in Eq. (2).

$$\left.\begin{array}{l} \mu = Mean\,[\overrightarrow{X^n}(I_m)] \\ \sigma = SD\,[\overrightarrow{X^n}(I_m)] \end{array}\right\} \rightarrow \quad (2)$$
$$\overrightarrow{X^n}(i_{ma}) = a_1 \times (2 \times rand - 1) \times \sigma + \mu$$

Where, $\overrightarrow{X^n}(I_m)$ shows the location (solution) of the guidance group. The *Mean* and *SD* are the mean and standard deviation functions respectively. Also, i_{ma} indicates the set of agents with the worst results. The results of the present study have shown that it is better to choose i_{ma} so that $i_{ma} < 0.05N_n$. The coefficient a_1 controls the distribution range of the new agents. In the present version of the algorithm, a_1 can be in the range of [1, 3].

Step 3. Neighbors Selection

In the third step, the neighbors of the ith agent are selected using a random selector with cumulative weighted function (*G*). The neighbors are selected from all the agent population (*R*) except the agent with the best result (i_m), the agents provided by guidance group (i_{ma}) and ith agent. The neighbors' selection is done using Eqs. (3) and (4). Besides, the weight of jth agent is as

$$w_j = \frac{F_{max} - F_j}{F_{max} - F_{min}} \quad , \qquad \alpha_j = \frac{w_j}{\sum_{t=1}^{Nn} w_t} \quad (3)$$
$$\rightarrow \qquad for \;\; j,\; t \in R - \{i, i_m, i_{ma}\}$$

Where, F_{max} and F_{min} are the maximum and minimum of the fitness function of the agents respectively.

The cumulative weighted function *G* is an array. Each element of this array contains the summation of the weight of jth agent and all its previous agents in the global numbering of agents. The array can be produced using Eq. (4)

$$G_1 = \alpha_1$$
$$G_j = G_{j-1} + \alpha_j \quad (4)$$
$$for \;\; j = 2, 3, \ldots \in R - \{i, i_m, i_{ma}\}$$

For selecting each neighbor of the ith agent, the minimum value of *j* should be determined in such a way that $rand \le G_j$. Here, $j = 1, 2, 3, \ldots \in R - \{i, i_m, i_{ma}\}$ and *rand* is a uniform random number generator in the range of [0, 1]. Indeed, for obtaining the neighbors of the ith agent (I_b), this procedure should be repeated N_b times.

The random selection of the neighbors decreases the probability of being trapped in local minima. In fact, when the ith agent is going to be trapped in local minima, changing the neighbors and getting new data help the agent to escape from the situation.

Step 4. New Agent Generation

In the fourth step, a new solution is proposed for the ith agent using the star graph method as aforementioned. If the fitness function value of the result is less than the previous one, this solution will be the new location of the ith agent in the next iteration.

This step consists of two subroutines or functions. The weighting function H increases the efficacy of the agents with better fitness function value to enhance the convergence of the algorithm. This function guides the local search of the algorithm to find the global minimum.

Moreover, the vector $\overrightarrow{X_C}$ is provided using the location of neighbors in the previous iteration. It helps to increase the search capability of the method. Thus, this vector plays the role of the global search of the algorithm.

The weighting function *H* is obtained by

$$W_j = \frac{F_{\max}}{F_j} \quad , \quad H_j = \frac{W_j}{\sum_{t=1}^{N_b} W_t} \qquad (5)$$

$$\rightarrow \quad for \quad j,\ t \in I_b$$

The attraction or repulsion caused by neighbors is applied along the line between each neighbor and the i^th^ agent. This direction can be represented by the unit vector $\vec{U}$ in accordance with the following

$$\overrightarrow{U_j} = \frac{\overrightarrow{X_i^n} - \overrightarrow{X_j^n}}{\left\| \overrightarrow{X_i^n} - \overrightarrow{X_j^n} \right\|} \times Sign(F_j - F_i) \qquad (6)$$

$$\rightarrow \quad for \quad j \in I_b$$

Where, *Sign*(*x*) indicates the sign of *x*. $\overrightarrow{X_i^n}$ shows the location of the ith agent in the nth iteration and $\left\| \overrightarrow{X_i^n} \right\|$ is the norm of the vector $\overrightarrow{X_i^n}$. I_b shows the neighbors of the ith agent. The vector $\overrightarrow{X_c}$ which represents the effects of pervious iteration is defined as Eq. (7).

$$\overrightarrow{X_c} = \sum_{j=1}^{N_b} H_j \ . \left\| \overrightarrow{X_i^{n-1}} - \overrightarrow{X_j^{n-1}} \right\| \ . \ \overrightarrow{U_j} \qquad (7)$$

$$\rightarrow \quad j \in I_b$$

Here, $\overrightarrow{X_i^{n-1}}$ is the location of the ith agent in the *n*-1th iteration and *n* shows the iteration number of the method. N_b indicates the number of the neighbors of the ith agent.

The direction vector $\vec{T_1}$ and step length T_2 are defined according to Eqs. (8) and (9).

$$\vec{T_1} = C_1 \times \sum_{j=1}^{N_b} H_j . \overrightarrow{U_j} + C_2 \times \frac{\overrightarrow{X_c}}{\left\| \overrightarrow{X_c} \right\|} \qquad (8)$$

$$\rightarrow \quad for \quad j \in I_b$$

$$T_2 = C_1 \times \sum_{j=1}^{N_b} H_j . \left\| \overrightarrow{X_i^n} - \overrightarrow{X_j^n} \right\| + C_2 \times \left\| \overrightarrow{X_c} \right\| \qquad (9)$$

$$\rightarrow \quad for \quad j \in I_b$$

The coefficients C_1 and C_2 are suggested as Eq. (10).

$$C_1 = a_2^{\frac{a_3 \times n}{N_i}} \qquad (10)$$

$$C_2 = a_2^{\frac{-a_4 \times n}{N_i}}$$

Where, N_i is the maximum number of iterations. a_2 is the base of the exponential function and a_3 and a_4 are some constant coefficients. All of these parameters can be assumed to be in the range of [1, 3]. For first estimation, it is suggested that $a_2 = a_3 = \frac{a_4}{2}$.

In these equations, $\vec{T_1}$ represents the direction vector and T_2 is the step length of the ith agent. It is obtained using the neighbors of the ith agent in the *n*th and *n*-1th iterations. The coefficients C_1 and C_2 help to convert the global search of the method to a local search. Hence, these values have a profound impact on the convergence of the algorithm. More accurately, C_1 is an incremental factor that amplifies the local search of the algorithm. On the other hand, C_2 is a decreasing factor and inspires the global search. In the algorithm procedure, the global search is gradually converted to the local search. Consequently, a new exponential function is suggested in order to improve the process.

Finally, the new suggested location of the ith agent for the next iteration is obtained as Eq. (11).

$$\overrightarrow{X_i^{n+1}} = \overrightarrow{X_i^n} + rand \times \frac{\vec{T}_1}{\left\| \vec{T}_1 \right\|} \times T_2 \qquad (11)$$

Where, $\overrightarrow{X_i^{n+1}}$ and $\overrightarrow{X_i^n}$ are the suggested and present location of the agent, respectively.

Notably, the location $\overrightarrow{X_i^{n+1}}$ will be the new location of the ith agent only if its fitness function value is less than the value of the present location $\overrightarrow{X_i^n}$. In each iteration, the process of neighbors' selection and finding the new location of each agent should be repeated.

Additionally, the process of the algorithm starts with the global search and tends to a local search. The global search needs the larger number of agents than the local one. Hence, the number of population is dynamic and may change during the algorithm. Therefore, the cost of calculation decreases without noticeable reduction in the convergence capability.

Step 5. Random Agent Generator

To prevent an agent from being trapped in local minima, a stochastic function is utilized to generate new random agents. This is what exactly happens in the fifth step of the SG algorithm. This function replaces the agents with the worst fitness function (I_x) with the new random agents according to

$$\overrightarrow{X^n}(I_x) = \vec{a} + (\vec{b} - \vec{a}) \times rand \qquad (12)$$

Here, $\vec{a}$ and $\vec{b}$ are vectors containing boundaries of variables in the search space.

In this way, the method utilizes a set of functions based on the star graph to converge into the global minimum.

Table 1 The specifications of benchmark problems (Kaveh and Talatahari 2010a)

Name	F(x)	Domain	F_{min}
Aluffi-Pentiny (AP)	$\frac{1}{4}x_1^4-\frac{1}{2}x_1^2+\frac{1}{10}x_1+\frac{1}{2}x_2^2$	$[-10,10]^2$	-0.352386
Bohachevsky-1 (Bf1)	$x_1^2+2x_2^2-\frac{3}{10}\cos(3\pi x_1)-\frac{4}{10}\cos(4\pi x_2)+\frac{7}{10}$	$[-100,100]^2$	0.000000
Bohachevsky-2 (Bf2)	$x_1^2+2x_2^2-\frac{3}{10}\cos(3\pi x_1)\cos(4\pi x_2)+\frac{3}{10}$	$[-50,50]^2$	0.000000
Becker and Lago	$(\lvert x_1\rvert-5)^2+(\lvert x_2\rvert-5)^2$	$[-10,10]^2$	0.000000
Branin	$(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos(x_1)+10$	$x_1\in[-5,10]$ $x_2\in[0,15]$	0.397887
Camel	$4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	$[-5,5]^2$	-1.031600
Cosine Mixture (CM)	$\sum_{i=1}^{n}x_i^2-\frac{1}{10}\sum_{i=1}^{n}\cos(5\pi x_i)$	$[-1,1]^4$	-0.400000
DeJoung	$x_1^2+x_2^2+x_3^2$	$[-5.12,5.12]^3$	0.000000
Exponential (EXP2, EXP4, EXP8)	$-\exp(-0.5\sum_{i=1}^{n}x_i^2)$	[-1,1] *n*= 2, 4, 8	-1.000000
Goldstein and price	$[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times$ $[30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	$[-2,2]^2$	3.000000
Griewank-2	$1+\frac{1}{200}\sum_{i=1}^{2}x_i^2-\prod_{i=1}^{2}\frac{\cos(x_i)}{\sqrt{i}}$	$[-100,100]^2$	0.000000
Hartman-3	$-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right)$ $a=\begin{bmatrix}3&10&30\\0.1&10&35\\3&10&30\\0.1&10&35\end{bmatrix};\ c=\begin{bmatrix}1\\1.2\\3\\33.2\end{bmatrix};\ p=\begin{bmatrix}0.3689&0.117&0.2673\\0.4699&0.4387&0.747\\0.1091&0.8732&0.5547\\0.03815&0.5743&0.8828\end{bmatrix}$	$[0,1]^3$	-3.862782
Hartman-6	$-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2\right)$ $a=\begin{bmatrix}10&3&17&3.5&1.7&8\\0.05&10&17&0.1&8&14\\3&3.5&1.7&10&17&8\\17&8&0.05&10&0.1&14\end{bmatrix};\ c=\begin{bmatrix}1\\1.2\\3\\3.2\end{bmatrix}$ $p=\begin{bmatrix}0.1312&0.1696&0.5569&0.0124&0.8283&0.5886\\0.2329&0.4135&0.8307&0.3736&0.1004&0.9991\\0.2348&0.1451&0.3522&0.2883&0.3047&0.6650\\0.4047&0.8828&0.8732&0.5743&0.1091&0.0381\end{bmatrix}$	$[0,1]^6$	-3.322368
Rastrigin	$x_1^2+x_2^2-\cos(18x_1)-\cos(18x_2)$	$[-1,1]^2$	-2.000000
Rosenbrock	$\sum_{i=1}^{n-1}\left(100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right)$	$[-30,30]^2$	0.000000
Zakharov (Z10, Z50)	$\sum_{j=1}^{n}x_j^2+(\sum_{j=1}^{n}0.5jx_j)^2+(\sum_{j=1}^{n}0.5jx_j)^4$	$[-5,10]^{10,50}$	0.000000
Levy	$\frac{\pi}{n}\left\{10\sin^2(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\right\}$	$[-10,10]^{20}$	0.000000
Brown	$\sum_{i=1}^{n-1}[(x_i^2)^{(x_{i+1}^2+1)}+(x_{i+1}^2)^{(x_i^2+1)}]$	$[-1,4]^{20}$	0.000000

Table 2 The average numbers of function evaluation for the benchmark problems

Function	GEN	GEN-S	GEN-S-M	GEN-S-M-LS	CSS	RO	SG Present Work
AP	1360 (0.99)	1360	1277	1253	804	331	**304**
Bf1	3992	3356	1640	1615	1187	677	**394**
Bf2	20234	3373	1676	1636	742	582	**353**
BL	19596	2412	2439	1436	423	**303**	314
Branin	1442	1418	1404	1257	852	463	**355**
Camel	1358	1358	1336	1300	575	332	**250**
CM	2105	2105	1743	1539	1563	802	**678**
Dejoung	9900	3040	1462	1281	630	452	**236**
Exp2	938	936	817	807	132	136	**75**
Exp4	3237	3237	2054	1496	867	382	**252**
Exp8	3237	3237	2054	1496	1426	1287	**820**
Goldstein and Price	1478	1478	1408	1325	682	451	**272**
Griewank	18838 (0.91)	3111 (0.91)	1764	1652 (0.99)	1551	1091 (0.98)	**1078**
Hartman3	1350	1350	1332	1274	860	N/A	**472**
Hartman6	2562 (0.54)	2562 (0.54)	2530 (0.67)	1865 (0.68)	1783	N/A	**1459**
Rastrigin	1533 (0.97)	1523 (0.97)	1392	1381	1402	**1013** (0.98)	1289
Rosenbrock	9380	3739	1675	1462	1452	N/A	**1132**

N/A: Not available

Finally, the algorithm continues until one of the following termination criteria is true

- The current iteration number of the algorithm exceeds the maximum number of iterations.
- The sequence of the best fitness function values converges so that the difference between the best values of two immediate iterations falls into a predefined tolerance value.

3. Engineering benchmark problems

In this section, the efficiency of the star graph algorithm is evaluated and the results of some different methods are compared with the present method.

3.1 Unconstrained problems

For unconstrained optimization problems, the algorithm is compared with different versions of known meta-heuristic methods (Kaveh and Talatahari 2010a). The benchmark problems are introduced in Table 1 and the compared results are shown in Tables 2 and 3.

It is valuable to note that Tsoulos (2008) studied some different versions of Genetic Algorithm which are introduced as GEN, GEN-S, GEN-S-M and GEN-S-M-LS. The method CSS (Kaveh and Talatahari 2010a) utilized the charged system principles to find global minimum. Similarly, the method RO (Kaveh and Khayatazad 2012) introduced a new meta-heuristic algorithm using the Snell's light refraction law.

Georgieva and Jordanov (2010) introduced a hybrid meta-heuristic technique for bound-constrained global optimization which combines the two techniques LPτO and NM. This approach provides a powerful hybrid optimization technique entitled LPτNM. Chelouah and Siarry (2003) worked on a hybrid method, called continuous hybrid algorithm (CHA), performing the exploration with a GA and the exploitation with a Nelder–Mead SS. Also, Chelouah and Siarry (2000) defined an adaptation of combinatorial Tabu Search (ECTS) which aims to follow Glover's basic approach. Additionally, Zheng *et al*. (2005) and Price *et al*. (2005) worked on a staged continuous tabu search (SCTS) and a differential evolution based algorithm (DE), respectively.

The perspective view and related contour lines of some benchmark functions are shown in Fig. 5.

The values in Tables 2 and 3 indicate the average numbers of function evaluation in 50 independent runs. The number in parenthesis represents the ratio of successful runs in which the method has found the global minimum. The predefined accuracy of the method is taken as $\varepsilon = |f_{\min} - f_{final}| = 10^{-4}$. The absence of the parentheses shows that the algorithm has been successful in all independent runs. In Table 2, it can be seen that only CSS and SG algorithms have been unconditionally successful in all fifty runs of all benchmark problems. Also, in benchmarks with higher number of variables, SG shows an efficient performance to find the global minimum point (Table 3).

Table 3 The average numbers of function evaluation for the benchmark problems

Function	LPτNM	ECTS	SCTS	CHA	DE	SG Present Work
Z50	N/A	63970	N/A	75520	N/A	**56597**
Z10	6826	4630	N/A	4291	34532	**3776**
Levy	10987	N/A	17443	N/A	29268	**8180**
Brown	11425	N/A	15142	N/A	28032	**8914**

N/A: Not available

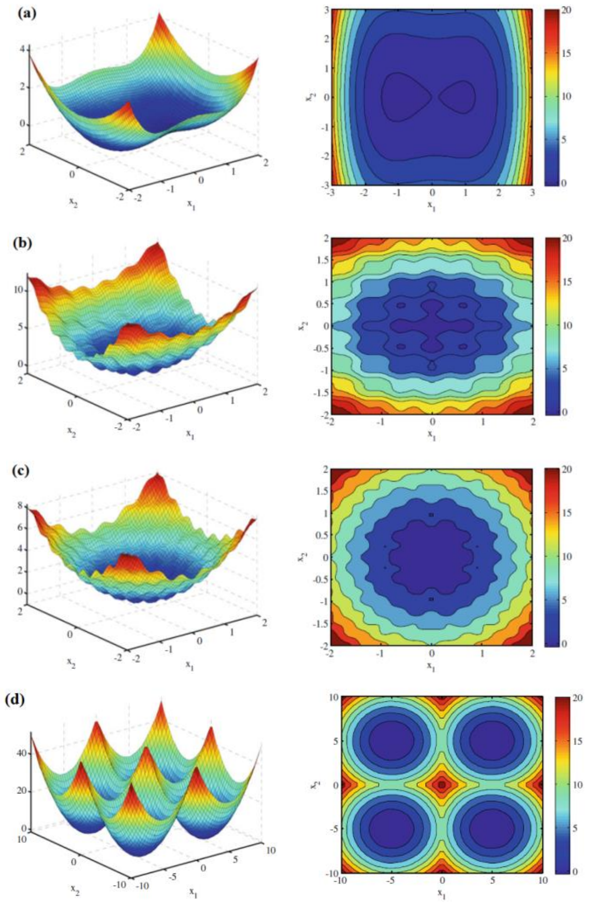

Fig. 5 A perspective view and related contour lines for some of functions in two-dimensional forms. (a) Aluffi-Pentiny, (b) Bohachevsky-1, (c) Bohachevsky-2 and (d) Becker and Lago

3.1.1 Comparison of the results

The method GEN-S-M-LS has better results than the other methods, which are based on GA. This method utilizes some auxiliary mechanisms such as an improved stopping law, the new mutation mechanism and an iterative approach in the local search. On the other hand, the methods CSS and RO improve the results more effectively than GA-based methods. As it can be seen, the Star Graph algorithm (SG) converges to the global minimum faster than RO, CSS and GA-based methods.

3.2 Constrained problems

3.2.1 A pressure vessel design problem

In this section, the optimal design of the cylindrical vessel, displayed in Fig. 6, is considered as a constrained optimization problem. The objective is to minimize the total cost including the cost of material, forming and welding (Sandgren 1988). This function is shown in Eq. (13).

$$f_{cost}(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (13)$$

Where, x_1 is the thickness of the shell (T_s), x_2 is the thickness of the head (T_h), x_3 the inner radius (R) and x_4 is the length of cylindrical section of the vessel (L). T_s and T_h are integer multiples of 0.0625 inches and R and L are real numbers.

The constraints and the design space can be stated as Eq. (14).

$$\begin{aligned} g_1(X) &= -x_1 + 0.0193x_3 \le 0 \\ g_2(X) &= -x_2 + 0.00954x_3 \le 0 \\ g_3(X) &= -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0 \\ g_4(X) &= x_4 - 240 \le 0 \\ 0 &\le x_1 \le 99 \\ 0 &\le x_2 \le 99 \\ 10 &\le x_3 \le 200 \\ 10 &\le x_4 \le 200 \end{aligned} \quad (14)$$

The constraints are applied to the algorithm using the penalty function method. The best results of various developed methods and corresponding statistical simulation results are shown in Tables 4 and 5, respectively. The results are obtained from ten independent runs of the methods. Although some methods such as Montes and Coello (2008) and Kaveh and Talatahari (2010a) have better results than the others, Star Graph method (SG) provides the best results. The standard deviation value of the SG algorithm is not the best, for instance in comparison to Coello (2000). However, the mean value of the SG algorithm has an error of about 0.6%, according to the minimum value of cost function, while Coello (2000) estimates the cost function with 3.9% of error. Besides, the values of mean and standard deviation of the SG algorithm are comparable to the other methods.

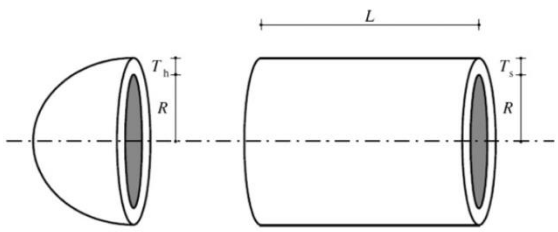

Fig. 6 Schematic shape of the pressure vessel

Table 4 Optimum results for the pressure vessel (Kaveh and Talatahari 2010a)

Methods	**Optimal Design Variable**				$\mathbf{f_{cost}}$
	$\mathbf{x_1}$ **($\mathbf{T_s}$)**	$\mathbf{x_2}$ **($\mathbf{T_h}$)**	$\mathbf{x_3}$ **(R)**	$\mathbf{x_4}$ **(L)**	
Sandgren (1988)	1.125000	0.625000	47.700000	117.701000	8129.1036
Kannan and Kramer (1994)	1.125000	0.625000	58.291000	43.690000	7198.0428
Deb and Gene (1997)	0.937500	0.500000	48.329000	112.679000	6410.3811
Coello (2000)	0.812500	0.437500	40.323900	200.000000	6288.7445
Coello and Montes (2002)	0.812500	0.437500	42.097398	176.654050	6059.9463
He and Wang (2007)	0.812500	0.437500	42.091266	176.746500	6061.0777
Montes and Coello (2008)	0.812500	0.437500	42.098087	176.640518	6059.7456
Kaveh and Talatahari (2010a)	0.812500	0.437500	42.098353	176.637751	6059.7258
SG (the current study)	0.812500	0.437500	42.098446	176.636596	**6059.1313**

Table 5 Statistical results of different methods for the pressure vessel (Kaveh and Talatahari 2010a)

Methods	**Best**	**Mean**	**Worst**	**Standard Deviation**
Sandgren (1988)	8129.1036	N/A	N/A	N/A
Kannan and Kramer (1994)	7198.0428	N/A	N/A	N/A
Deb and Gene (1997)	6410.3811	N/A	N/A	N/A
Coello (2000)	6288.7445	6293.8432	6308.1497	**7.4133**
Coello and Montes (2002)	6059.9463	6177.2533	6469.3220	130.9297
He and Wang (2007)	6061.0777	6147.1332	6363.8041	86.4545
Montes and Coello (2008)	6059.7456	6850.0049	7332.8798	426.0000
Kaveh and Talatahari (2010a)	6059.7258	6081.7812	6150.1289	67.2418
SG (the current study)	6059.1313	6093.2716	6203.7628	40.9574

3.2.2 A 10-bar planar truss

The optimal design of the 10-bar truss, shown in Fig. 7, is considered as another example of constrained optimization problem. More accurately, the weight of the truss is considered as the objective function. In this problem, the stress limit of the members is σ_0= ±172.37 MPa (25 ksi). The nodal displacements in the vertical direction are limited to ±5.08 cm (2.0 in) and the density of the material is ρ= 2767.99 kg/m^3 (0.1 lb/in^3). The minimum cross section and the modulus of elasticity are A_0= 0.6451 cm^2 (0.1 in^2) and E= 6.89×10^4 MPa (10^4 ksi), respectively.

Two different load cases are considered herein. In the first case, P_1= 444.82 KN (100 kips) and P_2= 0. Additionally, in the second case, P_1= 667.233 KN (150 kips) and P_2= 222.411 KN (50 kips). The constraints are applied to the algorithm using the penalty function method and search variables are considered as the cross-section areas of the truss members.

In Tables 6 and 7, the results of the suggested method are compared with various methods and the improvement of the results is shown. The results are obtained from twenty independent runs of the methods. Although, some methods such as Lamberti and Pappalettere (2003) and Sedaghati (2005), in Case 1, and Rizzi (1976) and John *et al*. (1987), in Case 2, approached the minimum weight of the truss, the present study provides the best value among the other methods.

3.2.3 A 25-bar spatial truss

The topology and nodal numbers of a 25-bar spatial truss structure are shown in Fig. 8. It is a very well-known test problem. In this case, the material density is considered as 0.1 lb/in^3 (2767.990 kg/m^3) and modulus of elasticity is taken as 10^4 ksi (68950 MPa). The range of cross-sectional areas varies from 0.01 to 3.4 in^2 (0.6452- 21.94 cm^2).

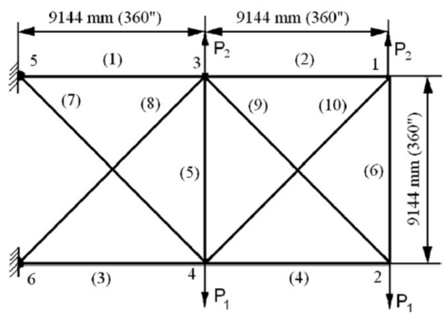

Fig. 7 A 10-bar planar truss (Schmit and Farshi 1974)

Table 6 Comparison of optimum designs of 10-bar truss (Case 1)

Member Number	**Schmit, Farshi (1974)**	**Schmit, Miura (1976)**	**Venkayya (1971)**	**Lamberti, Pappalettere (2003)**	**Sedaghati (2005)**	**Kaveh, Rahami (2006)**	**Li *et al*. (2007)**	**Farshi, Ziazi (2010)**	**SG Present Work**
1	33.430	30.670	30.4200	*	30.5210	30.6670	30.5690	30.5200	30.5280
2	0.1000	0.1000	0.1280		0.1000	0.1000	0.1000	0.1000	0.1000
3	24.2600	23.7600	23.4100		23.1990	22.8720	22.9740	23.2040	23.2050
4	14.2600	14.5900	14.9100		15.2220	15.3440	15.1480	15.2230	15.2180
5	0.1000	0.1000	0.1010		0.1000	0.1000	0.1000	0.1000	0.1000
6	0.1000	0.1000	0.1010		0.5510	0.4630	0.5470	0.5510	0.5510
7	8.3880	8.5780	8.6960		7.4570	7.4790	7.4930	7.4660	7.4570
8	20.7400	21.0700	21.0800		21.0360	20.9650	21.1590	21.0340	21.0360
9	19.6900	20.9600	21.0800		21.5280	21.7020	21.5560	21.5290	21.5220
10	0.1000	0.1000	0.1860		0.1000	0.1000	0.1000	0.1000	0.1000
Weight (lb)	5089.00	5076.85	5084.90	5060.88	5060.85	5061.90	5061.03	5061.40	5060.85

*Not mentioned

Table 7 Comparison of optimum designs of 10-bar truss (Case 2)

Member Number	**Schmit and Farshi (1974)**	**Schmit and Miura (1976)**	**Venkayya (1971)**	**Rizzi (1976)**	**John *et al.* (1987)**	**Li *et al.* (2007)**	**Farshi, Ziazi (2010)**	**SG Present Work**
1	24.2900	23.5500	25.1900	23.5300	23.5900	23.7430	23.5270	23.5300
2	0.1000	0.1000	0.3630	0.1000	0.1000	0.1010	0.1000	0.1000
3	23.3500	25.2900	25.4200	25.2900	25.2500	25.2870	25.2940	25.2900
4	13.6600	14.3600	14.3300	14.3700	14.3700	14.4130	14.3760	14.3680
5	0.1000	0.1000	0.4170	0.1000	0.1000	0.1000	0.1000	0.1000
6	1.9690	1.9700	3.1440	1.9700	1.9700	1.9690	1.9690	1.9690
7	12.6700	12.3900	12.0800	12.3900	12.3900	12.3620	12.4040	12.3980
8	12.5400	12.8100	14.6100	12.8300	12.8000	12.6940	12.8240	12.8520
9	21.9700	20.3400	20.2600	20.3300	20.3700	20.3230	20.3300	20.2960
10	0.1000	0.1000	0.5130	0.1000	0.1000	0.1030	0.1000	0.1000
Weight (lb)	4691.84	4676.96	4895.60	4676.92	4676.93	4677.70	4677.80	4676.89

Table 8 Loading conditions for the 25-bar spatial truss (Kaveh *et al*. 2014)

Node	Case 1			Case 2		
	P_x kips (kN)	P_y kips (kN)	P_z kips (kN)	P_x kips (kN)	P_y kips (kN)	P_z kips (kN)
1	0.0	20.0 (89)	-5.0 (22.25)	1.0 (4.45)	10.0 (44.5)	-5.0 (22.25)
2	0.0	-20 (89)	-5.0 (22.25)	0.0	10.0 (44.5)	-5.0 (22.25)
3	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0
6	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0

Table 9 Member stress limitation for the 25-bar spatial truss (Kaveh *et al*. 2014)

Element group		Compressive stress limitations ksi (Mpa)	Tensile stress limitations ksi (Mpa)
1	A_1	35.092 (241.96)	40.0 (275.80)
2	A_2-A_5	11.590 (79.913)	40.0 (275.80)
3	A_6-A_9	17.305 (119.31)	40.0 (275.80)
4	A_{10}-A_{11}	35.092 (241.96)	40.0 (275.80)
5	A_{12}-A_{13}	35.092 (241.96)	40.0 (275.80)
6	A_{14}-A_{17}	6.759 (46.603)	40.0 (275.80)
7	A_{18}-A_{21}	6.959 (47.982)	40.0 (275.80)
8	A_{22}-A_{25}	11.082 (76.410)	40.0 (275.80)

Table 10 Comparison of optimization results in the 25-bar tower problem

Element group		**HPSSO**	**PSO**	**MSPSO**	**PSOPC**	**HPSO**	**SG Present Work**
1	A_1	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
2	A_2-A_5	1.9907	1.9503	1.9848	1.9790	1.9700	1.9960
3	A_6-A_9	2.9881	3.0408	2.9956	3.0110	3.0160	2.9793
4	A_{10}-A_{11}	0.0100	0.0100	0.0100	0.1000	0.0100	0.0100
5	A_{12}-A_{13}	0.0100	0.0100	0.0100	0.1000	0.0100	0.0100
6	A_{14}-A_{17}	0.6824	0.6929	0.6852	0.6570	0.6940	0.6860
7	A_{18}-A_{21}	1.6764	1.6866	1.6778	1.6570	1.6810	1.6756
8	A_{22}-A_{25}	2.6656	2.6362	2.6599	2.6930	2.6430	2.6635
Best weight (lb)		545.164	545.220	545.160	545.270	545.190	545.150
Average weight (lb)		545.556	549.960	546.030	N/A	N/A	545.177
Worst weight (lb)		546.990	594.530	548.780	N/A	N/A	545.222
Standard deviation		0.432	9.910	0.800	N/A	N/A	0.0002
No. of analysis		13326	18400	10800	50000	50000	13000

N/A: Not available

Twenty five members are categorized into eight groups and also, the spatial truss is subjected to two loading conditions shown in Table 8. Maximum displacement limitations of ? .35 in (8.89 mm) are imposed on every node in every direction and the axial stress constraints vary for each group as shown in Table 9. The constraints are applied to the algorithm using the penalty function method and search variables are considered as the cross- sectional areas of truss members.

Table 10 indicates the results of optimal design of the frame using SG and results are compared with HPSSO (Kaveh *et al*. 2014), PSO (Talatahari *et al*. 2013), MPSO (Talatahari *et al*. 2013), PSOPC (Li *et al*. 2007) and HPSO (Li *et al*. 2007). Herein, statistical results and the average number of analysis are obtained using 20 independent runs of algorithm. The table shows the best, worst and average weight of the structure are improved than the other methods, although the number of analysis is similar. Also, standard deviation of SG shows better results than other methods.

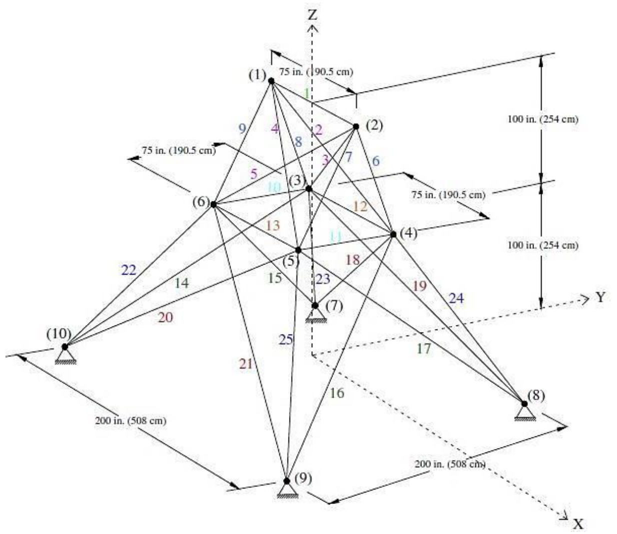

Fig. 8 Schematic of the spatial 25-bar tower (Kaveh *et al*. 2014)

3.2.4 A 3-bay 15-story frame problem

The optimal weight of 3-bay 15-story frame, shown in Fig. 9, is considered. In this figure, external loads and groups of elements are indicated. Assumptions for analysis and design of structure are considered as follows:

The modulus of elasticity and yielding stress of steel are considered 29 Msi (200 GPa) and 36 ksi (248.2MPa), respectively. The effective length factor of members in out-of-plane behavior is equal to $k_y = 1.0$ and in in-plane behavior $k_x \geq 0$ can be obtained using following equation (Dumonteil 1992)

$$k = \sqrt{\frac{1.6G_A G_B + 4.0(G_A + G_B) + 7.5}{G_A + G_B + 7.5}} \tag{15}$$

Where, G_A and G_B are ratio of the bending stiffness of column to connected beams in two end joints, A and B, in each column. Unbraced length of column is equal to whole of column length and for beam is equal to one fifth of beam length.

Displacement and resistance constraints of problem accordance with AISC (2001) are as follows:

a) Maximum lateral displacement

$$\frac{\Delta_T}{H} - R \leq 0 \tag{16}$$

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

b) In-story maximum displacement

$$\frac{d_i}{h_i} - R_I \leq 0 \qquad i = 1, 2, ..., ns \tag{17}$$

Where, d_i is the difference between displacements of two stories, h_i is the height of ith story, ns is the total number of stories and R_I is the maximum drift between stories equal to 1/300.

c) Resistance constraints

$$\begin{cases} \dfrac{P_u}{2\phi_c P_n} + \dfrac{M_u}{\phi_b M_n} - 1 \leq 0, & for\ \dfrac{P_u}{\phi_c P_n} < 0.2 \\ \dfrac{P_u}{\phi_c P_n} + \dfrac{8M_u}{9\phi_b M_n} - 1 \leq 0, & for\ \dfrac{P_u}{\phi_c P_n} \geq 0.2 \end{cases} \tag{18}$$

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

The nominal tensile resistance for yielding of member

$$P_n = A_g . F_y \tag{19}$$

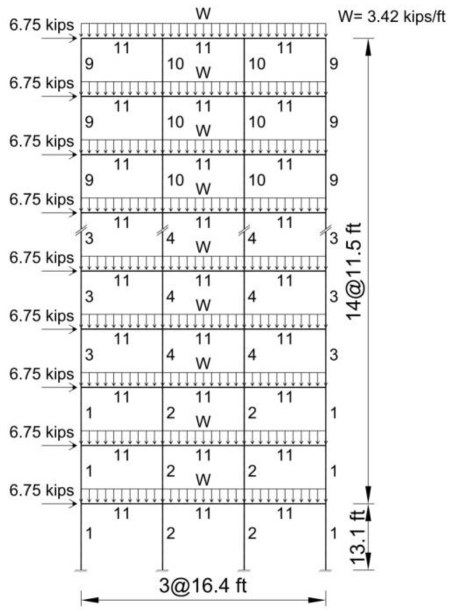

Fig. 9 Schematic of the 3-bay 15-story frame

Also, the nominal compressive resistance of member

$$P_n = A_g . F_{cr} \tag{20}$$

Where

$$\begin{cases} F_{cr} = (0.658^{\lambda_c^2}) F_y, & \text{for } \lambda_c \leq 1.5 \\ F_{cr} = (\dfrac{0.877}{\lambda_c^2}) F_y \quad , & \text{for } \lambda_c > 1.5 \end{cases} \tag{21}$$

$$\lambda_c = \frac{kl}{r\pi}\sqrt{\frac{F_y}{E}} \tag{22}$$

Where, A_g is the cross section of member and k is the effective length factor of member. The constraints are applied to the algorithm using the penalty function method and search variables are considered as the cross-sectional areas of frame members.

In Table 11, the optimal result of the method is compared with some new method of literature. The optimum weight of SG is equal to 84747 lb with 15560 as the average number of analysis. The best result of other methods is obtained by ECBO (Kaveh and Ilchi Ghazaan 2015), which is equal to 86986 lb with 9000 analysis of structure.

In the table, the optimum weight 95850 lb is obtain by HPSACO (Kaveh and Talatahari 2009), 97689 lb by HBB-BC (Kaveh and Talatahari 2010b), 93846 lb by ICA (Kaveh and Talatahari 2010c), 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) and 91248 lb by DSOS (Talatahari 2016). Although the number of analysis is increased in this method, the best and average of optimal weights and standard deviation of the method are improved than other methods.

Table 11 Optimization results obtained for the 3-bay 15 story frame

Element group	**HPSACO**	**HBB-BC**	**ICA**	**CSS**	**ECBO**	**ES-DE**	**DSOS**	**SG Present Work**
1	W21´111	W24x117	W24x117	W21x147	W14x99	W18x106	W16x100	W27X102
2	W18´158	W21x132	W21x147	W18x143	W27x161	W36x150	W32x152	W30X124
3	W10x88	W12x95	W27x84	W12x87	W27x84	W12x79	W12x79	W14X82
4	W30x116	W18x119	W27x114	W30x108	W24x104	W27x114	W27x114	W24X104
5	W21x83	W21x93	W14x74	W18x76	W14x61	W30x90	W21x93	W21X62
6	W24x103	W18x97	W18x86	W24x103	W30x90	W10x88	W12x79	W18X71
7	W21x55	W18x76	W12x96	W21x68	W14x48	W18x71	W21x55	W14X61
8	W27x114	W18x65	W24x68	W14x61	W14x61	W18x65	W14x61	W12X53
9	W10x33	W18x60	W10x39	W18x35	W14x30	W8x28	W14x22	W14X43
10	W18x46	W10x39	W12x40	W10x33	W12x40	W12x40	W14x43	W14X43
11	W21x44	W21x48	W21x44	W21x44	W21x44	W21x48	W21x48	W21X44
Best Weight (lb)	95850	97689	93846	92723	86986	93315	91248	84747
Average Weight (lb)	N/A	N/A	N/A	N/A	88410	98531	N/A	88236
Standard Deviation	N/A	N/A	N/A	N/A	N/A	3294	N/A	2312
No. of analyses	6800	9900	6000	5000	9000	10000	N/A	15560

N/A: Not available

3.2.5 A 3-bay 24-story frame problem

The Optimum weight of 3-bay 24-story frame, shown in Fig. 10, is considered. As shown, the members are categorized into 20 groups, including 16 column and 4 beam members. The section of beams is selected form 267W-shapes, while columns are selected from W14 sections. The modulus of elasticity and yielding stress of steel are equal to $E = 29.732$ Msi (205 GPa) and $f_y = 33.4$ ksi (230.3 MPa), respectively.

The effective length factor of members for out-of-plane behavior is $k_y = 1.0$ and for in-plane behavior of the frame $k_x \geq 0$. For all members, the unbraced length is equal to the total length of element. Herein, the resistance and displacement constraints are similar to the previous problem and specifications of AISC-LRFD are considered (AISC 2001).

This problem is optimized by GA (Saka and Kameshki 1998), ACO (Camp *et al*. 2005), HS (Degertekin 2008), CSS (Kaveh and Talatahari 2012), ECBO (Kaveh and Ilchi Ghazaan 2015), ES-DE (Talatahari *et al*. 2015) and DSOS (Talatahari 2016) and corresponding results are shown in Table 12. The optimal weight of frame in the present method is equal to 201489 lb and the average number of analysis is 18330. The best result of the other methods belongs to ECBO (Kaveh and Ilchi Ghazaan 2015) with 201618 lb and 15360 analysis of structure. The comparison of results shows, the best and average and the standard deviation of optimal weight are improved in the present method, although the number of analysis is briefly increased.

4. Conclusions

A new meta-heuristic algorithm is introduced in this article based on Star Graph. This method represents a special pattern of communication among the agents to find an appropriate solution. Herein, some efficient weighting functions are presented to improve the capability and performance of the algorithm in local and global search, which are based on the fitness function values of the neighbors.

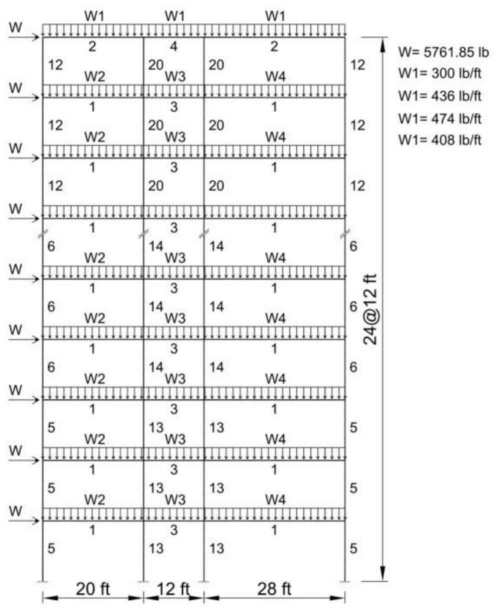

Fig. 10 Schematic of the 3-bay 24-story frame

Table 12 Optimization results for the 3-bay 24-story frame

Element Group	GA	ACO	HS	CSS	ECBO	ES-DE	DSOS	SG Present Work
1	838x292x194UB	W30x90	W30x90	W30x90	W30x90	W30x90	W30x90	W30X90
2	305x102x25UB	W8x18	W10x22	W21x50	W6x15	W21x55	W21x62	W8X13
3	457x191x82UB	W24x55	W18x40	W21x48	W24x55	W21x48	W21x48	W24X55
4	305x102x25UB	W8x21	W12x16	W12x19	W6x8.5	W10x45	W21x55	W6X8.5
5	305x102x25UC	W14x145	W14x176	W14x176	W14x145	W14x145	W14x176	W14X145
6	305x368x129UC	W14x132	W14x176	W14x145	W14x132	W14x109	W14x109	W14X132
7	305x305x97UC	W14x132	W14x132	W14x109	W14x99	W14x99	W14x120	W14X99
8	356x368x129UC	W14x132	W14x109	W14x90	W14x90	W14x145	W14x82	W14X90
9	305x305x97UC	W14x68	W14x82	W14x74	W14x74	W14x109	W14x61	W14X74
10	203x203x71UC	W14x53	W14x74	W14x61	W14x38	W14x48	W14x99	W14X38
11	305x305x118UC	W14x43	W14x34	W14x34	W14x38	W14x38	W14x34	W14X38
12	152x152x23UC	W14x43	W14x22	W14x34	W14x22	W14x30	W14x38	W14X22
13	305x305x137UC	W14x145	W14x145	W14x145	W14x99	W14x99	W14x120	W14X99
14	305x305x198Uc	W14x145	W14x132	W14x132	W14x99	W14x132	W14x109	W14X99
15	356x368x202UC	W14x120	W14x109	W14x109	W14x99	W14x109	W14x90	W14X99
16	356x368x129UC	W14x90	W14x82	W14x82	W14x82	W14x68	W14x90	W14X82
17	356x368x129UC	W14x90	W14x61	W14x68	W14x68	W14x68	W14x82	W14X68
18	356x368x153UC	W14x61	W14x48	W14x43	W14x61	W14x68	W14x38	W14X61
19	203x203x60UC	W14x30	W14x30	W14x34	W14x30	W14x61	W14x38	W14X30
20	254x254x89UC	W14x26	W14x22	W14x22	W14x22	W14x22	W14x22	W14X22
The Best Weight (lb)	251547	220465	214860	212364	201618	212492	209795	201489
The Average Weight (lb)	N/A	229555	222620	215226	209644	N/A	N/A	208212
The Standard Deviation	N/A	4561	N/A	2448	N/A	N/A	N/A	2201
No. of analyses	30000	15500	13924	5500	15360	12500	7500	18330

N/A: Not available

Also, the normal distribution is utilized for the guidance group to replace the existing infeasible solutions. Additionally, the dynamic regeneration of neighboring network in each iteration of the algorithm reduces the probability of being trapped in local minima. Moreover, the small size of neighboring groups decreases the computational cost. This algorithm is examined for various constrained and unconstrained benchmark functions, and the results of the method are compared with those of the other meta-heuristic counterparts. In constrained benchmark problems, it is shown that SG is unconditionally successful for fifty independent runs while some other methods failed in specific benchmark problems. Furthermore, in constrained problems, it is shown that the SG algorithm not only is capable of finding the global minimum but its standard deviation, in a statistical analysis, is better than some other existing methods.

References

Akin, A. and Saka, M.P. (2015), "Harmony search algorithm based optimum detailed design of reinforced concrete plane frames subject to ACI 318-05 provisions", *Comput. Struct.*, **147**, 79-95.

American Institute of Steel Construction (AISC), (2001), *Manual of steel construction: load and resistance factor design*.

Camp, C.V., Bichon, B.J. and Stovall, S. (2005), "Design of steel frames using ant colony optimization", **131**(3), *J. Struct. Eng*. – *ASCE*, 369-379.

Chelouah, R. and Siarry, P. (2000), "Tabu search applied to global optimization", *Eur. J. Oper. Res*., **123**(2), 256-270.

Chelouah, R. and Siarry, P. (2003), "Genetic and Nelder–Mead algorithms hybridised for a more accurate global optimisation of continuous multidimensional functions", *Eur. J. Oper. Res*., **148**(2), 335-348.

Coello, C.A.C. (2000), "Use of a self-adaptive penalty approach for engineering optimization problems", *Comput. Indust*., **41**(2), 113-127.

Coello, C.A.C. and Montes, E.M. (2002), "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection", *Adv. Eng. Inf.*, **16**(3), 193-203.

Deb, K. and Gene, A.S. (1997), "A robust optimal design technique for mechanical component design", (Eds., Dasgupta, D. and Michalewicz, Z.), *Evolutionary Algorithms in Engineering Applications*, Springer, Berlin, 497-514.

Degertekin, S.O. (2008), "Optimum design of steel frames using harmony search algorithm", *Struct. Multidiscip. O.*, **36**(4), 393-401.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996), "The ant system: optimization by a colony of cooperating agents", *IEEE T. Syst. Man Cy. B*, **41**, 26-29.

Dumonteil, P. (1992), "Simple equations for effective length factors", *Eng. J. AISC*, **29**(3), 111-115.

Eberhart, R.C. and Kennedy, J. (1995), "A new optimizer using particle swarm theory", *Proceedings of the 6th international symposium on micro machine and human science*, Nagoya, Japan.

Farshi, B. and Ziazi, A.A. (2010), "Sizing optimization of truss structures by method of centers and force formulation", *Int. J. Solids Struct*., **47**(18-19), 2508-2524.

Georgieva, A. and Jordanov, I. (2010), "A hybrid meta-heuristic for global optimization using low-discrepancy sequences of points", *Comput. Oper. Res*., **37**(3), 456-469.

Goldberg, D. (1989), *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Reading, MA.

He, Q. and Wang, L. (2007), "An effective co-evolutionary particle swarm optimization for constrained engineering design problems", *Eng. Appl. Artif. Intel*., **20**(1), 89-99.

John, K.V., Ramakrishnan, C.V. and Sharma, K.G. (1987), "Minimum weight design of trusses using improved move limit method of sequential linear programming", *Comput. Struct*., **27**(5), 583-591.

Kannan, B.K. and Kramer, S.N. (1994), "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design", *J. Mech. Des.- ASME DC*, **116**(2), 318-320.

Kaveh, A. and Talatahari, S. (2009), "Hybrid algorithm of harmony search, particle swarm and ant colony for structural design optimization", *Studies Comput. Intell*., **239**, 159-198.

Kaveh, A. and Ghafari, M.H. (2016), "Optimum design of steel floor system: effect of floor division number, deck thickness and castellated beams", *Struct. Eng. Mech.*, **59**(5), 933-950.

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 Khayatazad, M. (2012), "A new meta-heuristic method: Ray Optimization", *Comput. Struct.*, **112-113**, 283-294.

Kaveh, A. and Mahdavi, V.R. (2014), "Colliding bodies optimization: A novel meta-heuristic method", *Comput. Struct.*, **139**, 18-27.

Kaveh, A. and Moradveisi, M. (2016), "Nonlinear analysis based optimal design of double-layer grids using enhanced colliding bodies optimization method", *Struct. Eng. Mech.*, **58**(3), 555-576.

Kaveh, A. and Rahami, H. (2006), "Analysis, design and optimization of structures using force method and genetic algorithm", *Int. J. Numer. Meth. Eng*., **65**(10), 1570-1584.

Kaveh, A. and Shokohi, F. (2016), "Optimum design of laterally-supported castellated beams using tug of war optimization algorithm", *Struct. Eng. Mech*., **58**(3), 533-553.

Kaveh, A. and Talatahari, S. (2010a), "A novel heuristic optimization method: charged system search", *Acta Mech*., **289**, 213-267.

Kaveh, A. and Talatahari, S. (2010b), "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. (2010c), "Optimum design of skeletal structure using imperialist competitive algorithm", *Comput. Struct*., **88**(21-22), 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. and Zolghadr, A. (2016), "Optimal analysis for optimal design of cyclically repeated space trusses with frequency constraints, Smart Structures and Systems", *Smart Struct. Syst.*, **18**(4), 733-754.

Kaveh, A., Bakhshpoori, T. and Afshari, E. (2014), "An efficient hybrid particle swarm and swallow swarm optimization algorithm", *Comput. Struct*., **143**, 40-59.

Kennedy, J. and Eberhart, R. (1995), "Particle swarm optimization", *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia.

Lamberti, L. and Pappalettere, C. (2003), "Move limits definition in structural optimization with sequential linear programming, Parts I & II", *Comput. Struct*., **81**(4), 197-238.

Lee, K.S. and Geem, Z.W. (2005), "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice", *Comput. Method. Appl. M*., **194**(36-38), 3902-3933.

Li, L.J., Huang, Z.B., Liu, F. and Wu, Q.H. (2007), "A heuristic particle swarm optimizer for optimization of pin connected structures", *Comput. Struct*., **85**(7-8), 340-349.

Liu, S.Q. and Kozan, E. (2016), "New graph-based algorithms to efficiently solve large scale open pit mining optimization problems", *Exp. Syst. Appl.*, **43**, 59-65.

Mirjalili, S. (2015), "The ant lion optimizer", *Adv. Eng. Softw*., **83**, 80-98.

Montes, E.M. and Coello, C.A.C. (2008), "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems", *Int. J. Gen. Sys*., **37**(4), 443-473.

Price, K., Storn, R. and Lampinen, J. (2005), *Differential evolution–a practical approach to global optimization*, Berlin, Springer.

Rizzi, P. (1976), "Optimization of multi-constrained structures based on optimality criteria", *Proceedings of the AIAA/ASME/SAE 17th Structures, Structural Dynamics, and Materials Conference*, King of Prussia, PA.

Sadollah, A., Bahreininejad, A., Eskandar, H. and Hamdi, M. (2012), "Mine blast algorithm for optimization of truss structures with discrete variables", *Comput. Struct.*, **102-103**, 49–63.

Saka, M.P. (2014), "Shape and topology optimization design of skeletal structures using metaheuristic algorithms: A review", *Comput. Technol. Rev.*, **9**, 31-68.

Saka, M.P. and Kameshki, E.S. (1998), "Optimum design of multi-story sway steel frames to BS 5950 using a genetic algorithm", *Proceedings of the 4th International Conference on Computational Structures Technology*, Edinburgh, Scotland, UK.

Saka, M.P., Hasançebi, O. and Geem, Z.W. (2016), "Metaheuristics in structural optimization and discussions on harmony search algorithm", *Swarm Evol. Comput*., **28**, 88-97.

Sandgren, E. (1988), "Nonlinear integer and discrete programming in mechanical design", *Proceedings of the ASME design technology conference*, Kissimine, F.L.

Schmit, L.A. and Farshi, B. (1974), "Some approximation concepts for structural synthesis", *AIAA J*., **12**(5), 692-699.

Schmit, L.A. and Miura, H. (1976), *Approximation concepts for efficient structural synthesis*, NASA-CR-2552, NASA, Washington, DC.

Sedaghati, R. (2005), "Benchmark case studies in structural design optimization using the force method", *Int. J. Solids Struct*., **42**(21-22), 5848-5871.

Sharafi, P., The, L.H. and Hadi, M.N.S., (2014a), "Shape optimization of thin-walled steel sections using graph theory

and ACO algorithm", *J. Constr. Steel Res.*, **101**, 331-341.
Sharafi, P., Teh, L.H. and Hadi, M.N.S. (2014b), "Conceptual design optimization of rectilinear building frames: a knapsack problem approach", *Eng. Optim*., **47**(10), 1303-1323.
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.
Talatahari, S., Kheirollahi, M., Farahmandpour, C. and Gandomi, A.H. (2013), "A multi-stage particle swarm for optimum design of truss structures", *Neural Comput. Appl*., **23**(5), 1297-1309.
Tsoulos, I.G. (2008), "Modifications of real code genetic algorithm for global optimization", *Appl. Math. Comput*., **203**(2), 598-607.
Venkayya, V.B. (1971), "Design of optimum structures", *Comput. Struct*., **1**(1-2), 265-309.
Wang, D. (2015), "Bandwidth-efficiency-oriented topology optimization for integrated switching systems based on circulant graphs", *Comput. Netw.*, **83**, 199-216.
Yang, X.S. (2010), "A new metaheuristic bat-inspired algorithm", *Nature Inspired Cooperative Strategies for Optimization* (NISCO, 2010) (Eds., Gonzalez, J.R. *et al*.), *Studies in Computational Intelligence*, Springer, Berlin: Springer.
Zheng, R., Ngo, N., Shum, P. and Tjin, S. (2005), "A staged continuous tabu search algorithm for the global optimisation and its applications to the design of fibre Bragg gratings", *Comput. Optim. Appl*., **30**(3), 319-335.

CC