ℙ periodica polytechnica

Civil Engineering
57/2 (2013) 97–112
doi: 10.3311/PPci.7166
http://periodicapolytechnica.org/ci
Creative Commons Attribution 🅭

RESEARCH ARTICLE

An improved ray optimization algorithm for design of truss structures

Ali Kaveh / *Majid* Ilchi Ghazaan / *Taha* Bakhshpoori

Received 2012-09-19, revised 2013-09-17, accepted 2013-09-30

Abstract

The ray optimization algorithm is a recently developed meta-heuristic algorithm which was conceptualized using the relationship between the angles of incidence and fraction based on Snell's law. In ray optimization, each agent is modeled as a ray of light that moves in the search space in order to find the global or near-global optimum solution. This paper develops an Improved Ray Optimization (IRO) algorithm for solving optimization problems. IRO employs a new approach for generating new solution vectors which has no limitation on the number of variables, so in the process of algorithm there is no need to divide the variables into groups like RO. The procedure which returns the violated agents into feasible search space is also modified. These improvements enhance the accuracy and convergence rate of the RO. The Simulation results of the IRO for benchmark mathematical optimization problems and truss structures are compared to those of the standard RO and some well-known meta-heuristic algorithms, respectively. Numerical results indicate the effectiveness and robustness of the proposed IRO.

Keywords

Modified ray optimization · Snell's refraction law · optimal design · truss structures · local best memory

Ali Kaveh

Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Tehran, P.O. Box 16846-13114, Iran
e-mail: alikaveh@iust.ac.ir

Majid Ilchi Ghazaan
Taha Bakhshpoori

Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Tehran, P.O. Box 16846-13114, Iran

1 Introduction

One class of optimization is based on traditional mathematical methods which use the gradient information to search the optimal solutions with drawbacks such as complex derivatives, sensitivity to initial values, and the large amount of enumeration memory required [1]. In recent years, the other class of optimization techniques, namely stochastic optimization algorithms inspired by natural mechanisms, are developed for overcoming these disadvantages.

In recent years, the investigation of various kinds of meta-heuristic algorithms for discrete and continuous structural optimization problems such as Genetic Algorithms (GA) [2], Simulated Annealing (SA) [3], Ant Colony Optimization (ACO) [4], Particle Swarm Optimizer (PSO) [5], Harmony Search (HS) algorithm [1], Big Bang-Big Crunch (BB-BC) algorithm [6], Charged System Search (CSS) method [7], Imperialist Competitive Algorithm (ICA) [8], and Magnetic Charged System Search (MCSS) method [9] has attracted much attention and many possible applications. A comprehensive review on structural design optimization based on stochastic optimization algorithms has been presented recently by Lambert and Pappalettere [10]. Optimum design of the truss structures is known as benchmark in the field of optimization problems due to the presence of many design variables, large size of the search space, and many constraints. Therefore this can be considered as a suitable means to investigate the efficiency of the new algorithms.

Recently, a new optimization method is developed that is based on the transition of ray from one medium to another from physics, Kaveh and Khayatazd [11]. The transition of the ray is utilized for finding the global or near-global solutions. This algorithm is called Ray Optimization (RO) and uses the Snell's refraction law of the light. RO is good at identifying the high performance regions of the solution space at a reasonable time in relatively complicated problems, but it is not good in performing local search for complex problems. In order to create a solution vector, a new technique is added to the IRO that provides a better balance between exploration and exploitation. Furthermore it applies an increasing function that helps the algorithm in constraint handling.

In order to show the efficiency and robustness of proposed method, IRO is applied to three standard truss structures in addition to some mathematical functions. Numerical results reveal that the proposed algorithm is a suitable stochastic search technique for various optimization problems and performs well in terms of accuracy and the number of objective function evaluations compared to the standard RO.

The remaining sections of this paper are organized as follows: The improved version of the RO algorithm together with a brief overview of the standard RO are provided in Section 2. Some benchmark mathematical functions and three benchmark truss structures are optimized in Section 3 for both discrete and continuous search spaces. Finally the paper is concluded in Section 4.

2 Improved ray optimization algorithm

This section describes the proposed improved ray optimization (IRO) algorithm. First, a brief overview of the standard RO is provided, and then the modifications are described forming the proposed IRO algorithm.

2.1 The basic principles of the ray optimization algorithm

The Ray Optimization (RO) algorithm proposed by Kaveh and Khayatazad [11] like other multi-agent methods has a number of particles consisting of the variables of the problem. These agents are considered as rays of light. Based on the Snell's light refraction law when light travels from a lighter medium to a darker one, it refracts and its direction changes. This behavior helps the agents to explore the search space in early stages of the optimization process and to make them converge in the final stages. This law is the main tool of the RO [11].

Considering n_d and n_t as the refraction indexes of the lighter material and darker one respectively, the Snell's law can be expressed as:

$$n_d \cdot \sin(\theta) = n_t \cdot \sin(\phi) \tag{1}$$

where θ and ϕ are the angles between the normal of two surfaces (n) with incoming ray vector (d), and the refracted ray vector (t) respectively, as shown in Figure 1.

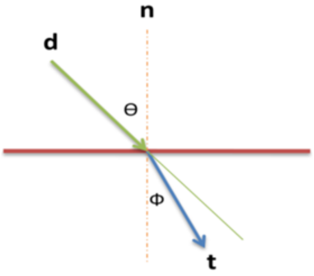

Fig. 1. Incident and refracted rays and their specifications

Having the direction of the incoming ray vector and the index of the refraction of the light and darker mediums, the direction of the refracted ray vector can be found.

For finding $\boldsymbol{t}$ in a 2-dimensional space, we have

$$\boldsymbol{t} = -\boldsymbol{n} \cdot \sqrt{1 - \frac{\boldsymbol{n}_d^2}{\boldsymbol{n}_t^2} \cdot \sin^2(\theta)} + \frac{n_d}{n_t} \cdot (\boldsymbol{d} - (\boldsymbol{d} \cdot \boldsymbol{n}) \cdot \boldsymbol{n}) \tag{2}$$

where $\boldsymbol{t}$, $\boldsymbol{d}$ and $\boldsymbol{n}$ are unit vectors.

In a 3-dimensional space, $\boldsymbol{d}$ and $\boldsymbol{n}$ are stated in a new coordinate system as:

$$\boldsymbol{n}^* = (1, 0), \quad \boldsymbol{d}^* = (\boldsymbol{d} \cdot \boldsymbol{i}^*, \boldsymbol{d} \cdot \boldsymbol{j}^*) \tag{3}$$

where $\boldsymbol{i}^*$ and $\boldsymbol{j}^*$ are two normalized vectors from Table 1, which *norm* is a function provides the length of a vector. Then $\boldsymbol{t}^* = (t_1^*, t_2^*)$ is calculated in 2-dimensional space, now $\boldsymbol{t}$ can be obtained in a 3-dimensional space:

$$\boldsymbol{t} = t_1^* \cdot \boldsymbol{i}^* + t_2^* \cdot \boldsymbol{j}^* \tag{4}$$

If the number of variables is more than 3, first the variables must be divided in 2-variables and 3-variables groups (for example a solution vector containing seven design variables divided into two 2-variables groups and one 3-variables group), thereafter prior formulas are applied to each group.

Tab. 1. The components of the new coordinate system

	$-0.05 \leq \boldsymbol{n} \cdot \boldsymbol{d} \leq 0.05$	$0.05 \leq \boldsymbol{n} \cdot \boldsymbol{d} \leq 1$	$-1 \leq \boldsymbol{n} \cdot \boldsymbol{d} \leq -0.05$
$\boldsymbol{i}^*$	$\boldsymbol{n}$	$\boldsymbol{n}$	$\boldsymbol{n}$
$\boldsymbol{j}^*$	$\boldsymbol{d}$	$\boldsymbol{j}^* = \frac{(\boldsymbol{n} - \frac{\boldsymbol{d}}{\boldsymbol{n} \cdot \boldsymbol{d}})}{norm(\boldsymbol{n} - \frac{\boldsymbol{d}}{\boldsymbol{n} \cdot \boldsymbol{d}})}$	$\boldsymbol{j}^* = \frac{(\boldsymbol{n} + \frac{\boldsymbol{d}}{-\boldsymbol{n} \cdot \boldsymbol{d}})}{norm(\boldsymbol{n} + \frac{\boldsymbol{d}}{-\boldsymbol{n} \cdot \boldsymbol{d}})}$

In the RO, the agents move to their new positions by the aid of the movement vectors. These movement vectors similarly must be divided into appropriate groups. In the first iteration of the optimization process, the positions and movement vectors are generated randomly. By adding the positions of each agent with their movement vectors, the agents move to the new positions. In this transition some agents violate the boundary conditions so their movement vector should be changed by a new one. The new movement vector is a vector with the same direction and a length equal to 0.9 times of the length between the current agent position and the boundary intersection caused by the prior movement vector.

Each agent must be moved to its new position, and first the point to which each particle moves must be determined. This point is named origin and it is specified by

$$\boldsymbol{O}_i^K = \frac{(ite + k) \cdot \boldsymbol{GB} + (ite - k) \cdot \boldsymbol{LB}_i}{2 \cdot ite} \tag{5}$$

Where $\boldsymbol{O}_i^k$ is the origin of the ith agent for the kth iteration, ite is the total number of iterations for the optimization process, and $\boldsymbol{GB}$ and $\boldsymbol{LB}_i$ are the global best and local best of the ith agent, respectively.

The normal is selected as a vector whose origin is $\boldsymbol{O}$ and its end is the current position of agent. Now, the direction of the new movement vector can be calculated based on Eq. (2), and after that the final form of the movement vector after finding the new direction is given by:

$$\boldsymbol{V}_{i,l} = \boldsymbol{V}'_{i,l} \cdot norm(\boldsymbol{X}_{i,l} - \boldsymbol{O}_{i,l}) \tag{6}$$

where $\boldsymbol{V}'_{i,l}$, $\boldsymbol{X}_{i,l}$, $\boldsymbol{O}_{i,l}$, and $\boldsymbol{V}_{i,l}$ are the normalized movement vector, current position of the agent, the origin, and refined movement vector of the *i*th agent, respectively for the *l*th group.

If the origin and its current position become identical, so the direction of the normal cannot be obtained. For solving this problem, the direction can be obtained utilizing the following equation:

$$\boldsymbol{V}_{i,l}^{k+1} = \frac{\boldsymbol{V}_{i,l}^{k}}{norm(\boldsymbol{V}_{i,l}^{k})} \cdot rand \cdot 0.001 \tag{7}$$

In this equation, $\boldsymbol{V}_{i,l}^{k}$ is the movement vector of the *k*th iteration that belongs to the *l*th group of the *i*th agent, and $\boldsymbol{V}_{i,l}^{k+1}$ is the movement vector of the $(k + 1)$th iteration. Finally, *rand* is a random number between 0 and 1.

With probability like ***stoch*** a movement vectors must be changed and considered as

$$\boldsymbol{V}_{ijl}^{(k+1)'} = -1 + 2 \cdot rand \tag{8}$$

where $\boldsymbol{V}_{ijl}^{(k+1)'}$ is *j*th component of *l*th group that belongs to *i*th agent in $(k + 1)$th iteration. But the magnitude of this vector should be refined. Therefore

$$\boldsymbol{V}_{il}^{K+1} = \frac{\boldsymbol{V}_{il}^{(k+1)'}}{norm(\boldsymbol{V}_{il}^{(k+1)'})} \cdot \frac{a}{d} \cdot rand \tag{9}$$

a is also calculated by the following relationship:

$$a = \sqrt{\sum_{i=1}^{n} (\boldsymbol{X}_{i,\max} - \boldsymbol{X}_{i,\min})^2} \quad n = \begin{cases} 2 \text{ for two variable groups} \\ 3 \text{ for three variable groups} \end{cases} \tag{10}$$

Where $\boldsymbol{X}_{i,max}$ and $\boldsymbol{X}_{i,min}$ are the maximum and minimum limits of variables that belong to the *i*th component of the movement vector, respectively. *d* is a number that divides *a* into smaller length for effective search. For further details, the reader may refer to Kaveh and Khayatazad [11].

2.2 Proposed improved ray optimization algorithm

In the following steps some definitions and contents are exactly the same as the standard RO and others which has improved the algorithm, are modified.

Step 1. The initial positions of the agents are determined randomly in the search space and after that the goal function is evaluated for each agent as

$$\boldsymbol{X}_{ij} = \boldsymbol{X}_{j,\min} + rand \cdot (\boldsymbol{X}_{j,\max} - \boldsymbol{X}_{j,\min}) \quad j = 1, 2, \ldots, n \tag{11}$$

Where $\boldsymbol{X}_{ij}$ is the *j*th variable of the *i*th agent. $\boldsymbol{X}_{j,min}$ and $\boldsymbol{X}_{j,max}$ are the minimum and maximum limits of the *j*th variable, and *n* is the number of variables.

Step 2. A memory which saves some or all the best positions as local best memory (***LBM***) is considered, and the position of the best agent is saved as the global best (***GB***). If the positions of all agents (especially when the number of agents is large) are saved, the computational cost grows. In this paper, if the number of agents are more than or equal to 25, the size of the local best memory is considered as 25, otherwise the size of the LBM is taken as half number of agents.

Step 3. The initial movement vectors of agents are stated as:

$$\boldsymbol{V}_{ij} = -1 + 2 \cdot rand \quad j = 1, 2, \ldots, n \tag{12}$$

Where $\boldsymbol{V}_{ij}$ is the initial value of the *j*th variable for the *i*th agent. Now, by adding the positions of each agent with their movement vectors, new positions are resulted.

Step 4. There is a possibility of boundary violation when an agent moves to its new position, so if any component of each agent violates a boundary, it must be regenerated by the following formula

$$\boldsymbol{X}_{ij}^{k+1} = \boldsymbol{X}_{ij}^{k} + 0.9(\boldsymbol{Int}_{ij} - \boldsymbol{X}_{ij}^{k}) \tag{13}$$

where $\boldsymbol{X}_{ij}^{k+1}$ and $\boldsymbol{X}_{ij}^{k}$ are the refined component and component of the *j*th variable for the *i*th agent in $(k + 1)$th and *k*th iteration, respectively. $\boldsymbol{Int}_{ij}$ is the intersection point (If an agent violates a boundary, it intersects the boundary at a specified point, because of having definite movement vector) of the *j*th variable for the *i*th agent. Instead of changing all the components of violating agents, only the components that violate the boundary is refunded This improvement is more effective when the number of variables is large. At the end of this step, the goal function is evaluated for each agent and thereafter the ***LBM*** and ***GB*** are updated.

Step 5 . Consider the origin as a point which each agent wants to move toward it and specified this point by

$$\boldsymbol{O}_{i}^{K} = \frac{(ite + k) \cdot \boldsymbol{GB} + (ite - k) \cdot \boldsymbol{LB}}{2 \cdot ite} \tag{14}$$

Where $\boldsymbol{O}_{i}^{k}$ is the origin of the *i*th agent for the *k*th iteration, *ite* is the total number of iterations for the optimization process, ***GB*** is the global best and ***LB*** is considered randomly from local best memory. Now for each agent target a vector (***Tv***) is defined as:

$$\boldsymbol{Tv}_{i} = \boldsymbol{O}_{i} - \boldsymbol{X}_{i} \tag{15}$$

Where $\boldsymbol{Tv}_{i}$ and $\boldsymbol{X}_{i}$ is target vector and current position of the *i*th agent, respectively.

Step 6. The main idea of RO is approximating the new movement vector with a normal vector. To achieve this aim, if the number of variables was more than 3, the proposed formula cannot be applied directly and first the main problem must be divided into some sub-problems and after the calculation, merge the results of the sub-problems to evaluate the goal function. When the number of variables is large the computational cost grows considerably. Instead of this approach the following formula (which has no limit on the number of variables) is applied to calculate the direction of the new movement vector that is defined according to prior movement vector and target vector as illustrated in Figure 2. Thus we have:

$$\boldsymbol{V}_i^{k+1} = \alpha \cdot \boldsymbol{Tv}_i^k + \beta \cdot \boldsymbol{V}_i^k \tag{16}$$

Where $\boldsymbol{V}_i^{k+1}$ and $\boldsymbol{V}_i^k$ are movement vectors for the *i*th agent in $(k+1)$th and *k*th iteration, respectively. α and β are the factors that control the exploitation and exploration respectively as shown in Figure 3. An efficient optimization algorithm should perform good exploration in early iterations and good exploitation in final iterations. Thus, α and β are increasing and decreasing functions respectively, and are defined as:

$$\alpha = 1 + (\frac{k}{ite}) \tag{17}$$

$$\beta = 1 - 0.5(\frac{k}{ite}) \tag{18}$$

Finally all the $\boldsymbol{V}_i^{k+1}$ vectors should be normalized.

Step 7 The magnitude of movement vectors must be calculated because in the previous steps only the direction of the movement vector is defined.

One of the important features of each meta-heuristic algorithm is its ability to escape from the trap when agents fall into a local minimum, so in the RO there is a possibility like ***stoch*** that specifies whether a movement vector must be changed or not, therefore we have

a. with probability like ***stoch***,

$$\boldsymbol{V}_{ij}^{k+1} = -1 + 2 \cdot rand \tag{19}$$

$$\boldsymbol{V}_i^{K+1} = \frac{\boldsymbol{V}_i^{k+1}}{norm(\boldsymbol{V}_i^{k+1})} \cdot \frac{a}{d} \cdot rand \tag{20}$$

b. with probability like ***(1-stoch)***,

If $norm(\boldsymbol{O}_i^k - \boldsymbol{X}_i^k) = 0$,

$$\boldsymbol{V}_i^{k+1} = \frac{\boldsymbol{V}_i^k}{norm(\boldsymbol{V}_i^k)} \cdot rand \cdot 0.001 \tag{21}$$

Otherwise,

$$\boldsymbol{V}_i^{k+1} = \boldsymbol{V}_i^{k+1} \cdot norm(\boldsymbol{X}_i^k - \boldsymbol{O}_i^k) \tag{22}$$

In the case of problems that have function constraints (behavior constraints), the following formulas are utilized instead of Eq. (22)

$$\boldsymbol{V}_i^{K+1} = \boldsymbol{V}_i^{k+1} \cdot \frac{a}{d} \tag{23}$$

$$d = d + r \cdot d \cdot (\frac{k}{ite}) \tag{24}$$

Where *r* is a constant factor

When the number of iterations rises the value of *d* increase and it help the algorithm to handle the constraint well.

If one of the subsequent steps is not fulfilled, by adding each movement vector with their current position they move to their new positions and the process of optimization will continue from Step 4.

Step 8 The terminating criterion is one of the following:

1. Maximum number of iterations: the optimization process is terminated when number of iterations is equal to predefined one.

2. Number of iterations without improvement: the optimization process is terminated if after number of specific iterations the objective function did not change.

3. Minimum goal function error: when the algorithm has found the global minimum with a predefined accuracy, the optimization process is terminated.

The flowchart of the IRO algorithm is illustrated in Figure 4.

3 Mathematical and Structural Design Examples

3.1 Standard mathematical functions

To test the ability of the proposed algorithm and to compare its results with those of the standard RO, some benchmark mathematical functions are considered which are tabulated in Table 2. Each of these functions tests the optimization algorithm in special conditions, identifying the weak points of the optimization algorithm. These functions are selected by Tsoulos [12] for evaluating modifications of Genetic Algorithm and utilized by Kaveh and Khayatazad [11] for investigating the standard RO.

The IRO method with different number of agents has been tested for each of the above functions. Assigning the number of agents as 50 in the CM, Griewank and Rastring functions and as 10 for other ones shows a better performance. We also have tried to tune other parameters of algorithm. From our simulations it is recommended to set parameters as 0.35 and 700 for ***Stoch*** and *d*, respectively. After implementing IRO algorithm using Matlab, it has been run independently 50 times to carry out meaningful statistical analysis. The algorithm stops when the variations of function values are less than a given tolerance as 10^{-4}. Table 3 reports the performance of GEN_S_M_LS as the best modificationof GA [12], the standard RO [11] and proposed IRO respectively, where the numbers are in the format: average number of evaluations ± one standard deviation (success rate). Considering this table, the standard RO and the improved RO show better performances in terms of the required number of analyses and success rate.

3.2 Continuous and discrete trusses

Optimum design of the truss structures is known as benchmark in the field of optimization problems. In this section, common truss optimization examples as benchmark problems are optimized with the IRO algorithm. The final results are compared to the solutions of other methods to demonstrate the efficiency of the IRO. We have tried to vary the number of agents and other parameters. From our simulations, setting the number of agents and ***Stoch*** 25 and 0.35 are efficient for design examples, respectively. Table 4 tabulates other parameters for each

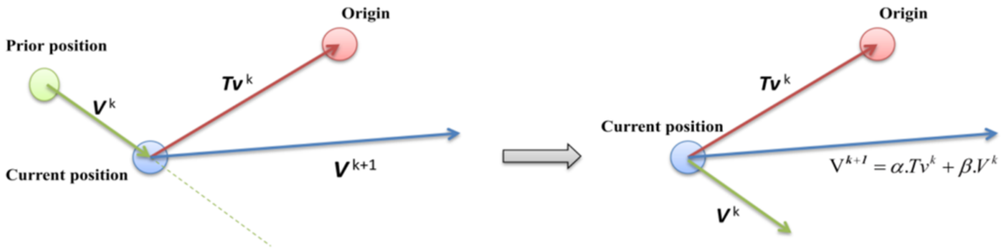

Fig. 2. Generation of the new movement vector

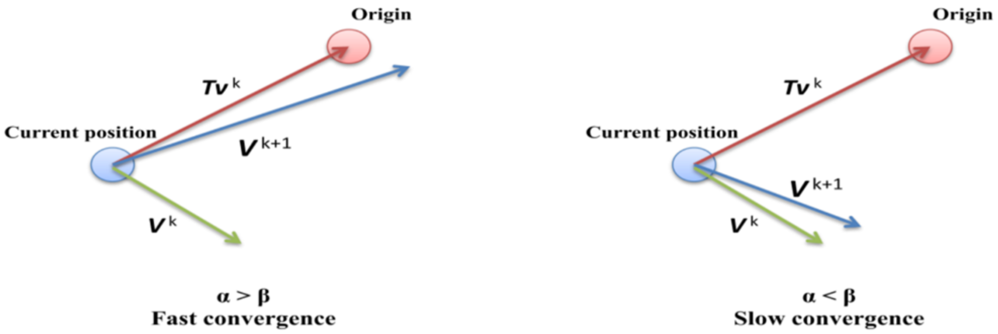

Fig. 3. Influence of the α and β parameters

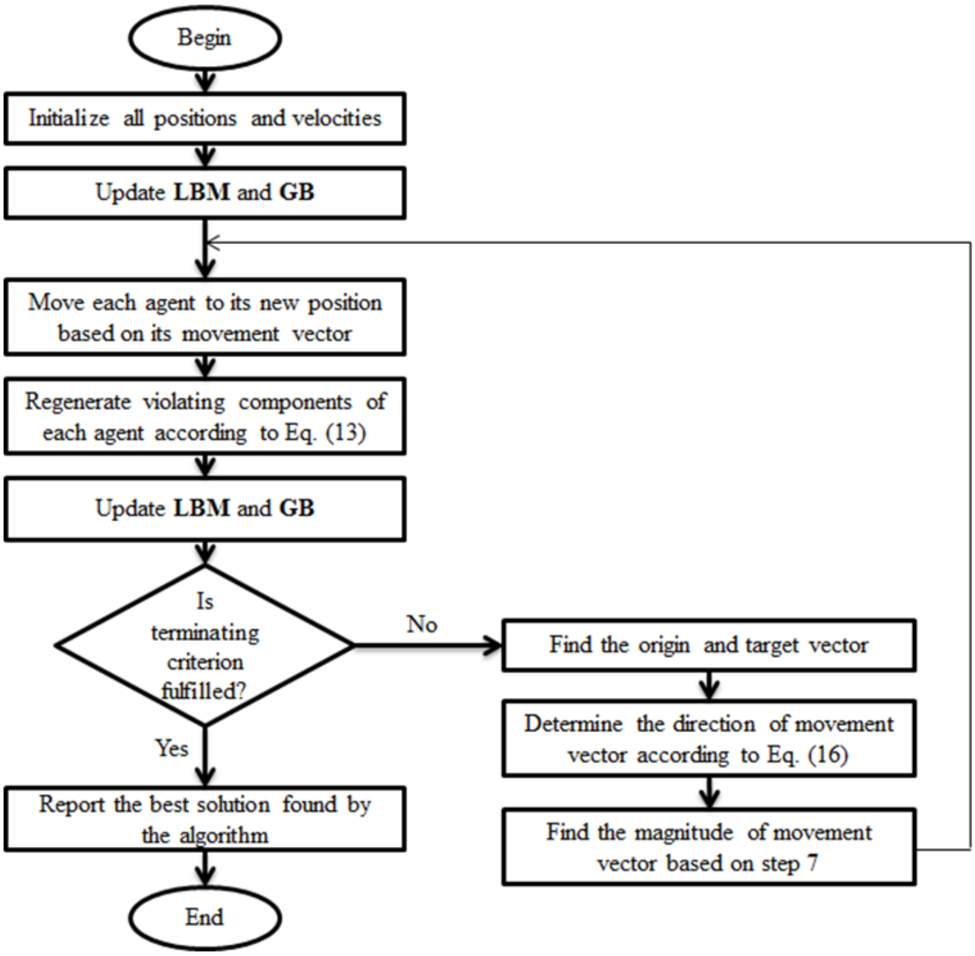
Begin
Initialize all positions and velocities
Update LBM and GB
Move each agent to its new position based on its movement vector
Regenerate violating components of each agent according to Eq. (13)
Update LBM and GB
Is terminating criterion fulfilled?
No
Find the origin and target vector
Determine the direction of movement vector according to Eq. (16)
Find the magnitude of movement vector based on step 7
Yes
Report the best solution found by the algorithm
End

Fig. 4. The flowchart of the IRO

Tab. 2. Specifications of the benchmark problems

Function name	Interval	Function	Global minimum
Aluffi-Pentiny	$\boldsymbol{X} \in [-10, 10]^2$	$f(X) = \frac{1}{4}x_1^4 - \frac{1}{2}x_1^2 + \frac{1}{10}x_1 + \frac{1}{4}x_2^2$	-0.352386
Bohachevsky 1	$\boldsymbol{X} \in [-100, 100]^2$	$f(X) = x_1^2 + 2x_2^2 - \frac{3}{10}\cos(3\pi x_1) - \frac{4}{10}\cos(4\pi x_2) + \frac{7}{10}$	0.0
Bohachevsky 2	$\boldsymbol{X} \in [-50, 50]^2$	$f(X) = x_1^2 + 2x_2^2 - \frac{3}{10}\cos(3\pi x_1)\cos(4\pi x_2) + \frac{3}{10}$	0.0
Becker and Lago	$\boldsymbol{X} \in [-10, 10]^2$	$f(X) = (\lvert x_1\rvert - 5)^2 + (\lvert x_2\rvert - 5)^2$	0.0
Branin	$0 \le x_2 \le 15 \quad -5 \le x_1 \le 10$	$f(X) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$	0.397887
Camel	$\boldsymbol{X} \in [-5, 5]^2$	$f(X) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	-1.0316
Cb3	$\boldsymbol{X} \in [-5, 5]^2$	$f(X) = 2x_1^2 - 1.05x_1^4 + \frac{1}{6}x_1^6 + x_1x_2 + x_2^2$	0.0
Cosine mixture	$n = 4, \quad \boldsymbol{X} \in [-1, 1]^n$	$f(X) = \sum_{i=1}^{n} x_i^2 - \frac{1}{10}\sum_{i=1}^{n}\cos(5\pi x_i)$	-0.4
De Joung	$\boldsymbol{X} \in [-5.12, 5.12]^3$	$f(X) = x_1^2 + x_2^2 + x_3^2$	0.0
Exponential	$n = 2, 4, 8, 16 \quad \boldsymbol{X} \in [-1, 1]^n$	$f(X) = -\exp(-0.5\sum_{i=1}^{n} x_i^2)$	-1.0
Goldstein and Price	$\boldsymbol{X} \in [-2, 2]^2$	$f(X) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2]$ $\cdot[30 + (2x_1 - 3x_2)^2(18 - 32x_1 - 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	3.0
Griewank	$\boldsymbol{X} \in [-100, 100]^2$	$f(X) = 1 + \frac{1}{200}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{2}\cos(\frac{x_i}{\sqrt{i}})$	0.0
Rastrigin	$\boldsymbol{X} \in [-1, 1]^2$	$f(X) = \sum_{i=1}^{n}(x_i^2 - \cos(18x_i))$	-2.0

Tab. 3. Performance comparison for the benchmark problems

FUNCTION	GEN_S_M_LS	Ray Optimization	Present work (IRO)
AP	1,253	331	253± 38.7985(100)
Bf1	1,615	677	438± 48.4636(100)
Bf2	1,636	582	395± 45.9674(100)
BL	1,436	303	194± 31.2733(100)
Branin	1,257	463	312±81.0515(100)
Camel	1,300	332	184± 21.0855(100)
Cb3	1,118	262	247± 36.4549(100)
CM	1,539	802	1290± 65.3543(100)
Dejoung	1,281	452	213± 26.3344(100)
EXP2	807	136	90± 20.5115(100)
EXP4	1,496	382	220± 50.5624(100)
EXP8	1,496	1,287	512± 97.7743(100)
EXP16	1945	17,236(0.46)	1141± 142.76(100)
GRIEWANK	1,652(0.99)	1,091(0.98)	1383± 100.3458(100)
RASTRIGIN	1,381	1,013(0.98)	1662± 202.3105(100)
Goldstein and Price	1,325	451	361± 59.0105(100)
TOTAL	22537(99.94)	25800(96.38)	8895(100)

case. In the sequel, formulation of optimum truss design problem is briefly overviewed at the first subsection then the examples are presented. The examples contain a 25-bar transmission tower and a 72-bar spatial truss with both discrete and continuous design variables and a dome shaped space truss with continuous search space.

3.2.1 Optimum design of truss structures

The aim of optimizing a structure is to find a set of design variables corresponding to the minimum weight satisfying certain constraints. This can be expressed as [8]:

$$\begin{array}{ll} \text{Find} & \{X\} = \left[x_1, x_2, \ldots, x_{ng}\right], \quad x_i \in D \\ \text{To minimize} & W(\{X\}) = \sum\limits_{i=1}^{ng} x_i \sum\limits_{j=1}^{nm(i)} \rho_j \cdot L_j \\ \text{Subject to:} & g_j(\{X\}) \leq 0 \quad j = 1, 2, \ldots, n \end{array} \qquad (25)$$

Where $\{X\}$ is the set of design variables; ng is the number of member groups in structure (number of design variables), the grouping of members is applied according to symmetry in topology of truss; D is the cross-sectional areas available for members in the truss which are restricted to a discrete or continuous range of values, bounded by an upper and lower limit; $W(\{X\})$ presents weight of the structure; $nm(i)$ is the number of members for the ith group; ρ_i and L_j denotes the material density and the length of the jth member for ith group, respectively; $g_j(\{X\})$ denotes design constraints; and n is the number of the constraints.

In order to handle the constraints, a penalty approach is utilized. In this method, the aim of the optimization is redefined by introducing the cost function as:

$$f_{\cos t}(\{X\}) = (1 + \varepsilon_1 \cdot \upsilon)^{\varepsilon_2} \times W(\{X\}), \quad \upsilon = \sum_{j=1}^{n} \max\left[0, g_j(\{X\})\right] \qquad (26)$$

Where n represents the number of evaluated constraints for each individual design, and υ denotes the sum of the violations of the design. The constants ε_1 and ε_2 are selected considering the exploration and the exploitation rate of the search space. Here, ε_1 is set to unity, ε_2 is selected in a way that it decreases the penalties and reduces the cross-sectional areas. Thus, in the first steps of the search process, ε_2 is set to 1.5 and ultimately increased to 3.

The constraint conditions for truss structures are briefly explained in the following. The stress limitations of the members are imposed according to the provisions of ASD-AISC [13] as follows:

$$\begin{cases} \sigma_i^+ = 0.6F_y & \text{for } \sigma_i \geq 0 \\ \sigma_i^- & \text{for } \sigma_i < 0 \end{cases} \qquad (27)$$

$$\sigma_i^- = \begin{cases} \left[\left(1 - \frac{\lambda_i^2}{2c_c^2}\right)F_y\right] / \left(\frac{5}{3} + \frac{3\lambda_i}{8c_c} + \frac{\lambda_i^3}{8c_c^3}\right) & \text{for } \lambda_i \geq c_c \\ \frac{12\pi^2 E}{23\lambda_i^2} & \text{for } \lambda_i < c_c \end{cases} \qquad (28)$$

Where E is the modulus of elasticity; F_y is the yield stress of steel; C_c denotes the slenderness ratio(λ_i) dividing the elastic and inelastic buckling regions ($c_c = \sqrt{2\pi^2 E/Fy}$); λ_i = the slenderness ratio ($\lambda_i = kl_i/r_i$); k = the effective length factor; L_i = the member length; and r_i = the radius of gyration. The radius of gyration (r_i) can be expressed in terms of cross-sectional areas as $r_i = aA_i^b$. Here, a and b are the constants depending on the types of sections adopted for the members such as pipes, angles, and tees. In this study, pipe sections ($a = 0.4993$ and $b = 0.6777$) are adopted for bars [1].

The other constraint corresponds to the limitation of the nodal displacements:

$$\delta_i - \delta_i^u \leq 0 \quad i = 1, 2, \ldots, nn \qquad (29)$$

Where δ_i is the nodal deflection; δ_i^u is the allowable deflection of node i; and nn is the number of nodes.

3.2.2 The 25-bar space truss

The 25-bar transmission tower is used widely in structural optimization to verify various meta-heuristic algorithms. The topology and nodal numbering of the truss is shown in Figure 5, Ref. [7]. The material density is considered as 0.1 lb/in^3 (2767.990 kg/m^3) and the modulus of elasticity is taken as 10^7 psi (68,950 MPa). Twenty-five members are categorized into eight groups, as follows: (1) A_1, (2) A_2 –A_5, (3) A_6–A_9, (4) A_{10}–A_{11}, (5) A_{12}–A_{13}, (6) A_{14}–A_{17}, (7) A_{18}–A_{21}, and (8) A_{22}–A_{25}. In this example, designs for both a single and multiple load cases using both discrete and continuous design variables are performed.

3.2.3 Design of the 25-bar truss utilizing discrete variables

In the first design of the 25-bar truss, a single load case {(kips) (kN)}is applied to the structure, at nodes 1, 2, 3 and 4 as follows: 1{(0, −10, −10) (0, −44.5, −44.5)}, 2{(1, −10, −10) (4.45, −44.5, −44.5)}, 3{(0.6, 0, 0) (2.67, 0, 0)}and 4{(0.5, 0, 0) (2.225, 0, 0)}. The allowable stresses and displacements are respectively ± 40 ksi (275.80 MPa) for each member and ± 0.35 in (± 8.89 mm) for each node in the x, y and z directions. The range of discrete cross-sectional areas is from 0.1 to 3.4 in^2 (0.6452 to 21.94 cm^2) with 0.1 in^2 (0.6452 cm^2) increment (resulting in 34 discrete cross sections) for each of the eight element groups [6].

Table 5 presents the performance of the IRO and other algorithms. The IRO algorithm achieves the best solution weighted by 484.85 lb (219.92 kg), after 925 analyses. Although, this is identical to the best design developed using BB-BC algorithm [6] and a multiphase ACO procedure [4], it perform better than others when the number of analyses and average weight for 50 runs are compared.

3.2.4 Design of the 25-bar truss utilizing continuous variables

In the second design of the 25-bar truss, the structure is subjected to two load cases listed in Table 6. Maximum displacement limitations of ± 0.35 in (± 8.89 mm) are imposed on every node in every direction and the axial stress constraints vary for

Tab. 4. Algorithm parameters for truss examples

Parameter	25-bar (discrete)	25-bar (continuous)	72-bar (discrete)	72-bar (continuous)	120-bar
d	15	5	15	10	10
r	7	4	7	20	20

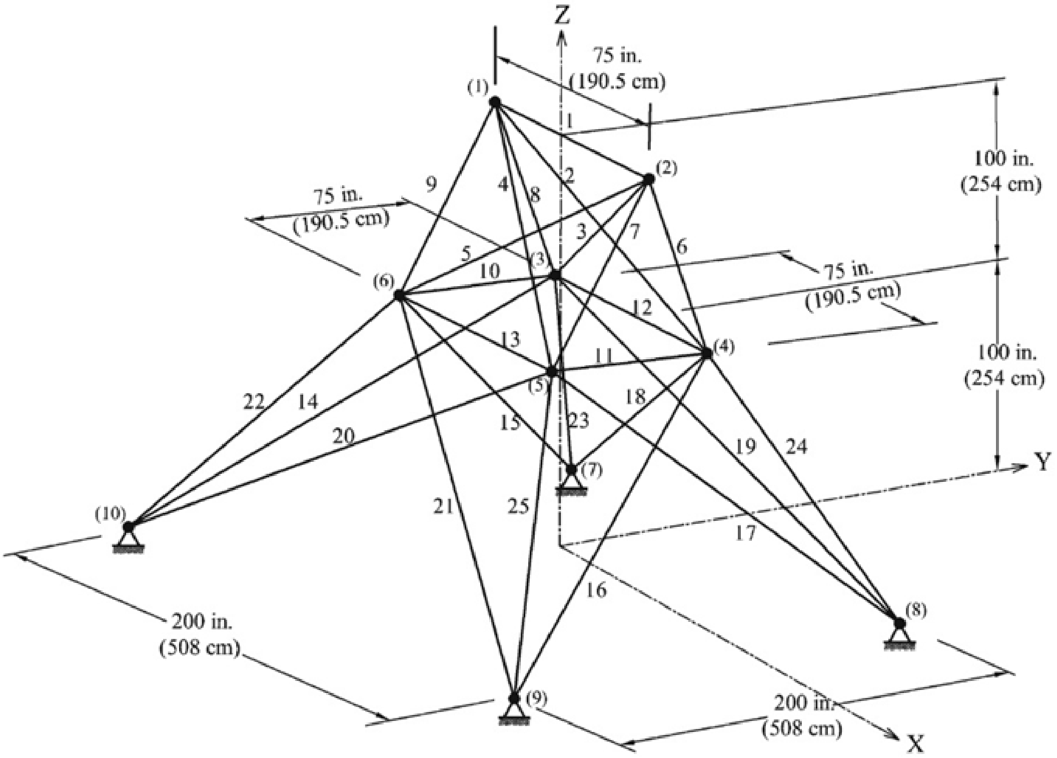

Fig. 5. The 25-bar space truss

Tab. 5. Performance comparison for the 25-bar spatial truss under single load case

Element group		Optimal cross-sectional areas (in^2)					
		GA	GA	ACO	BB-BC phases 1,2	Present work	
		[6]	[6]	[4]	[6]	in^2	cm^2
1	A_1	0.10	0.10	0.10	0.10	0.10	0.645
2	A_2–A_5	1.80	0.50	0.30	0.30	0.30	1.935
3	A_6–A_9	2.30	3.40	3.40	3.40	3.40	21.935
4	A_{10}–A_{11}	0.20	0.10	0.10	0.10	0.10	0.645
5	A_{12}–A_{13}	0.10	1.90	2.10	2.10	2.10	13.548
6	A_{14}–A_{17}	0.80	0.90	1.00	1.00	1.00	6.452
7	A_{18}–A_{21}	1.80	0.50	0.50	0.50	0.50	3.226
8	A_{22}–A_{25}	3.00	3.40	3.40	3.40	3.40	21.935
Best weight (lb)		546.01	485.05	484.85	484.85	484.85	219.92 (kg)
Average weight (lb)		N/A	N/A	486.46	485.10	484.90	219.94 (kg)
Number of analyses		800	15,000	7700	9000	925	

each group as shown in Table 7. The range of cross-sectional areas varies from 0.01 to 3.4 in^2 (0.06452 to 21.94 cm^2) [7].

Table 8 shows the best solution vectors, the corresponding weights, average weights and the required number of analyses for present algorithm and some other meta-heuristic algorithms. The best result obtained by IACS algorithm [14] in the aspects of low weight and number of analyses. The IRO-based algorithm needs 12200 analyses to find the best solution while this number is equal to 9596, 7000, 9875, 12,500 and 13880 analyses for a PSO-based algorithm [7], the CSS algorithm [7], a combination algorithm based on PSO, ACO and HS [15], an improved BB–BC method using PSO properties [16] and RO algorithm [11], respectively. The difference between the results of the IRO and these algorithms are very small, but the average weight obtained by the IRO algorithm for 50 runs is better than others. The convergence history for the best result and average penalized weight of 50 runs are shown in Figure 6. The important point is that although the IRO requires 12200 analyses to achieve the weight of 545.19 lb (247.29 kg), it can achieve the 547 lb (248.12 kg) after 6000 analyses. Convergence speed in IRO is acceptable and step-like movements in diagram of IRO performance exhibit how it escapes from local minimum points in order to find a better optimum point.

Tab. 6. Loading conditions for the 25-bar spatial truss

Case	Node	F_x kips (kN)	F_y kips (kN)	F_z kips (kN)
1	1	1.0 (4.45)	10.0 (44.5)	−5.0 (−22.25)
	2	0.0	10.0	−5.0 (−22.25)
	3	0.5 (2.225)	0.0	0.0
	6	0.5 (2.225)	0.0	0.0
2	1	0.0	20.0 (89)	−5.0 (−22.25)
	2	0.0	−20.0 (−89)	−5.0 (−22.25)

3.2.5 The 72-bar space truss

For the 72-bar spatial truss structure shown in Figure 7 [16], the material density is 0.1 lb/in^3 (2767.990 kg/m^3) and the modulus of elasticity is 10^7 psi (68,950 MPa). The 72 structural members of this spatial truss are categorized into 16 groups using symmetry: (1) A_1–A_4, (2) A_5–A_{12}, (3) A_{13}–A_{16}, (4) A_{17}–A_{18}, (5) A_{19}– A_{22}, (6) A_{23}–A_{30}, (7) A_{31}–A_{34}, (8) A_{35}–A_{36}, (9) A_{37}–A_{40}, (10) A_{41}–A_{48}, (11) A_{49}–A_{52}, (12) A_{53}–A_{54}, (13) A_{55}–A_{58}, (14) A_{59}–A_{66} (15), A_{67}– A_{70}, and (16) A_{71}–A_{72}. In this example, designs for a multiple load cases using both discrete and continuous design variables are performed. The values and directions of the two load cases applied to the 72-bar spatial truss for both discrete and continuous designs are listed in Table 9. The members are subjected to the stress limits of ± 25 ksi (± 172.375 MPa) for both discrete and continuous designs. Maximum displacement limitations of ± 0.25 in (± 6.35 mm), are imposed on every node in every direction and on the uppermost nodes in both x and y directions respectively for discrete and continuous cases.

Tab. 7. Member stress limitation for the 25-bar space truss

Element group	Compression ksi (MPa)	Tension 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)

3.2.6 Design of the 72-bar truss using discrete variables

In this case, the discrete variables are selected from 64 discrete values from 0.111 to 33.5 in^2 (71.613 to 21612.860 mm^2). For more information, the reader can refer to Table 2 in Kaveh and Talatahari [7].

Table 10 shows the best solution vectors, the corresponding weights and the required number of analyses for present algorithm and some other meta-heuristic algorithms. The IRO algorithm can find the best design among the other existing studies. Although the number of required analyses by the IRO algorithm is more than ICA algorithm, but the best weight of the IRO algorithm is 389.33lb (176.60 kg) that is 3.51lb (1.59 kg) lighter than the best result obtained by ICA algorithm [8].The convergence history of the best result and the average penalized weight of 50 runs are shown in Figure 8. Convergence speed in IRO is acceptable and step-like movements in diagram of IRO performance exhibit how it escapes from local minimum points, to find a better optimum point. It is important to note that this case has an expended search space than is requisite. The performance of the IRO decreased from 389.87± 1.1643lb (176.84± 0.5281 kg) to 408.17± 71.2108 lb (185.14± 32.3007 kg) considering 47 and all 50 independent runs, respectively. In the other words, IRO yields to unexpected designs in just three of 50 independent runs. Unfortunately comprehensive statistical study of this case is not available in optimization literature.

3.2.7 Design of the 72-bar truss using continuous variables

In this case the minimum value for the cross-sectional areas is 0.1 in^2 (0.6452 cm^2) and the maximum value is limited to 4.00 in^2 (25.81 cm^2).

Table 11 summarizes the obtained results of IRO and other methods are available in optimization literature. The IRO algorithm achieves the best result among other algorithms in the aspect of average weight of 50 runs. Performance of the IRO for 50 independent runs is observed as an average weight of 380.55 lb (172.61 kg) with a standard deviation of 1.5234. The difference between the optimum design obtained by IRO and variants of BB-BC is very small.

Tab. 8. Performance comparison for the 25-bar spatial truss under multiple load cases

Element group		Optimal cross-sectional areas (in^2)							
		PSO	CSS	IACS	HPSACO	HBB–BC	RO	Present work	
		[7]	[7]	[14]	[15]	[16]	[11]	in^2	cm^2
1	A_1	0.010	0.010	0.010	0.010	0.010	0.016	0.0112	0.0722
2	A_2–A_5	2.121	2.003	2.042	2.054	1.993	2.022	1.9766	12.7522
3	A_6–A_9	2.893	3.007	3.001	3.008	3.056	2.932	3.0099	19.4187
4	A_{10}–A_{11}	0.010	0.010	0.010	0.010	0.010	0.010	0.0100	0.0645
5	A_{12}–A_{13}	0.010	0.010	0.010	0.010	0.010	0.011	0.0100	0.0645
6	A_{14}–A_{17}	0.671	0.687	0.684	0.679	0.665	0.656	0.6842	4.4142
7	A_{18}–A_{21}	1.611	1.655	1.625	1.611	1.642	1.679	1.6783	10.8277
6	A_{22}–A_{25}	2.717	2.660	2.672	2.678	2.679	2.716	2.6571	17.1425
Best weight (lb)		545.21	545.10	545.03	544.99	545.16	544.66	545.19	247.29(kg)
Average weight (lb)		546.84	545.58	545.74	545.52	545.66	546.69	545.35	247.37(kg)
Number of analyses		9596	7000	3254	9875	12500	13880	12200	

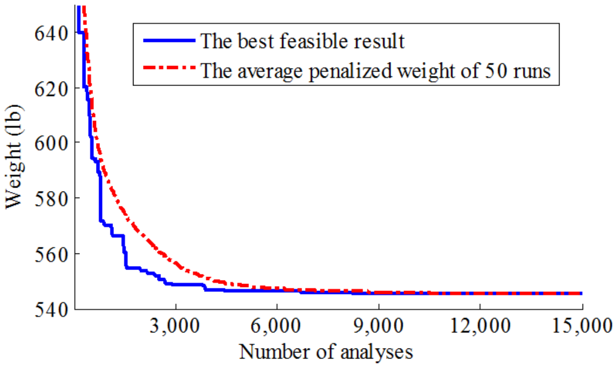

Fig. 6. Convergence history of the 25-bar space truss under multiple load cases

Tab. 9. Multiple loading conditions for the 72-bar truss

Case	Node	F_x kips (kN)	F_y kips (kN)	F_z kips (kN)
1	17	0.0	0.0	−5.0 (−22.25)
	18	0.0	0.0	−5.0 (−22.25)
	19	0.0	0.0	−5.0 (−22.25)
	20	0.0	0.0	−5.0 (−22.25)
2	17	5.0 (22.25)	5.0 (22.25)	−5.0 (−22.25)

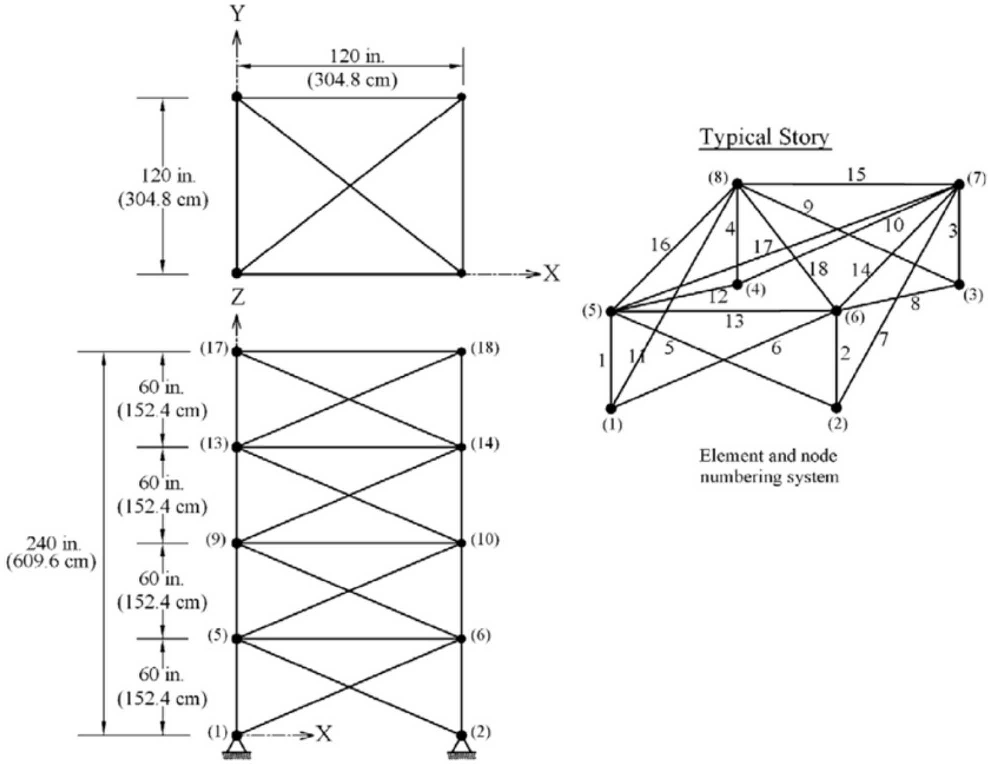

Fig. 7. A 72-bar space truss

Tab. 10. Performance comparison for the 72-bar spatial truss with discrete variables

Element group		Optimal cross-sectional areas (in^2)						
		GA	PSOPC	HPSO	HPSACO	ICA	Present work	
		[8]	[8]	[8]	[5]	[8]	in^2	cm^2
1	A_1–A_4	0.196	4.490	4.970	1.800	1.99	1.99	12.839
2	A_5–A_{12}	0.602	1.457	1.228	0.442	0.442	0.563	3.632
3	A_{13}–A_{16}	0.307	0.111	0.111	0.141	0.111	0.111	0.716
4	A_{17}–A_{18}	0.766	0.111	0.111	0.111	0.141	0.111	0.716
5	A_{19}–A_{22}	0.391	2.620	2.880	1.228	1.228	1.228	7.923
6	A_{23}–A_{30}	0.391	1.130	1.457	0.563	0.602	0.563	3.632
7	A_{31}–A_{34}	0.141	0.196	0.141	0.111	0.111	0.111	0.716
8	A_{35}–A_{36}	0.111	0.111	0.111	0.111	0.141	0.111	0.716
9	A_{37}–A_{40}	1.800	1.266	1.563	0.563	0.563	0.563	3.632
10	A_{41}–A_{48}	0.602	1.457	1.228	0.563	0.563	0.442	2.852
11	A_{49}–A_{52}	0.141	0.111	0.111	0.111	0.111	0.111	0.716
12	A_{53}–A_{54}	0.307	0.111	0.196	0.250	0.111	0.111	0.716
13	A_{55}–A_{58}	1.563	0.442	0.391	0.196	0.196	0.196	1.265
14	A_{59}–A_{66}	0.766	1.457	1.457	0.563	0.563	0.563	3.632
15	A_{67}–A_{70}	0.141	1.228	0.766	0.442	0.307	0.391	2.523
16	A_{71}–A_{72}	0.111	1.457	1.563	0.563	0.602	0.563	3.632
Weight (lb)		427.203	941.82	933.09	393.380	392.84	389.33	176.60 (kg)
Number of analyses		N/A	150,000	50,000	5330	4500	17925	

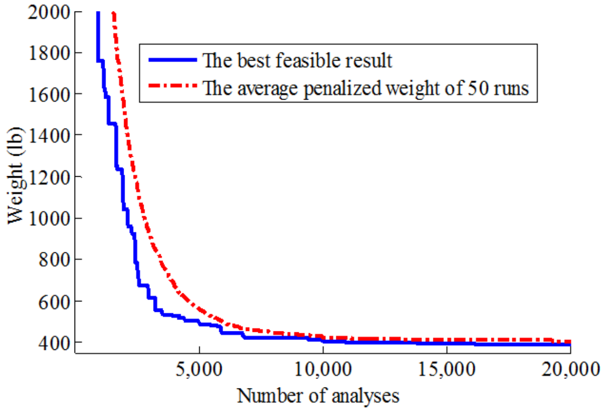

Fig. 8. Convergence history of the 72-bar space truss under multiple load cases

Tab. 11. Performance comparison for the 72-bar spatial truss with continuous variables

Element group		Optimal cross-sectional areas (in^2)						
		GA	ACO	PSO	BB–BC	HBB–BC	Present work	
		[16]	[4]	[16]	[6]	[16]	in^2	cm^2
1	A_1–A_4	1.755	1.948	1.7427	1.8577	1.9042	1.8378	11.8568
2	A_5–A_{12}	0.505	0.508	0.5185	0.5059	0.5162	0.5261	3.3942
3	A_{13}–A_{16}	0.105	0.101	0.1000	0.1000	0.1000	0.1000	0.6452
4	A_{17}–A_{18}	0.155	0.102	0.1000	0.1000	0.1000	0.1000	0.6452
5	A_{19}–A_{22}	1.155	1.303	1.3079	1.2476	1.2582	1.2668	8.1729
6	A_{23}–A_{30}	0.585	0.511	0.5193	0.5269	0.5035	0.5249	3.3864
7	A_{31}–A_{34}	0.100	0.101	0.1000	0.1000	0.1000	0.1000	0.6452
8	A_{35}–A_{36}	0.100	0.100	0.1000	0.1012	0.1000	0.1006	0.6490
9	A_{37}–A_{40}	0.460	0.561	0.5142	0.5209	0.5178	0.5164	3.3316
10	A_{41}–A_{48}	0.530	0.492	0.5464	0.5172	0.5214	0.5090	3.2839
11	A_{49}–A_{52}	0.120	0.100	0.1000	0.1004	0.1000	0.1012	0.6529
12	A_{53}–A_{54}	0.165	0.107	0.1095	0.1005	0.1007	0.1000	0.6452
13	A_{55}–A_{58}	0.155	0.156	0.1615	0.1565	0.1566	0.1568	1.0116
14	A_{59}–A_{66}	0.535	0.550	0.5092	0.5507	0.5421	0.5445	3.5129
15	A_{67}–A_{70}	0.480	0.390	0.4967	0.3922	0.4132	0.3918	2.5277
16	A_{71}–A_{72}	0.520	0.592	0.5619	0.5922	0.5756	0.5850	3.7742
Weight (lb)		385.76	380.24	381.91	379.85	379.66	379.86	172.30 (kg)
Average weight (lb)		N/A	383.16	N/A	382.08	381.85	380.55	172.61 (kg)
Number of analyses		N/A	18,500	N/A	19,621	13,200	15350	

Tab. 12. Performance comparison for the 120-bar dome shaped truss with continuous variables

Element group		Optimal cross-sectional areas (in^2)							
		PSOPC	PSACO	HPSACO	HBB–BC	CSS	ICA	Present work	
		[15]	[15]	[15]	[16]	[7]	[8]	in^2	cm^2
1	A_1	3.040	3.026	3.095	3.037	3.027	3.0275	3.0252	19.5174
2	A_2	13.149	15.222	14.405	14.431	14.606	14.4596	14.8354	95.7121
3	A_3	5.646	4.904	5.020	5.130	5.044	5.2446	5.1139	32.9928
4	A_4	3.143	3.123	3.352	3.134	3.139	3.1413	3.1305	20.1967
5	A_5	8.759	8.341	8.631	8.591	8.543	8.4541	8.4037	54.2173
6	A_6	3.758	3.418	3.432	3.377	3.367	3.3567	3.3315	21.4935
7	A_7	2.502	2.498	2.499	2.500	2.497	2.4947	2.4968	16.1084
Best weight (lb)		33481.2	33263.9	33248.9	33287.9	33251.9	33256.2	33256.48	15084.89(kg)
Average weight(lb)		N/A	N/A	N/A	N/A	N/A	N/A	33280.85	15095.94(kg)
Number of analyses		150,000	32,600	10,000	10,000	7000	6000	18300	

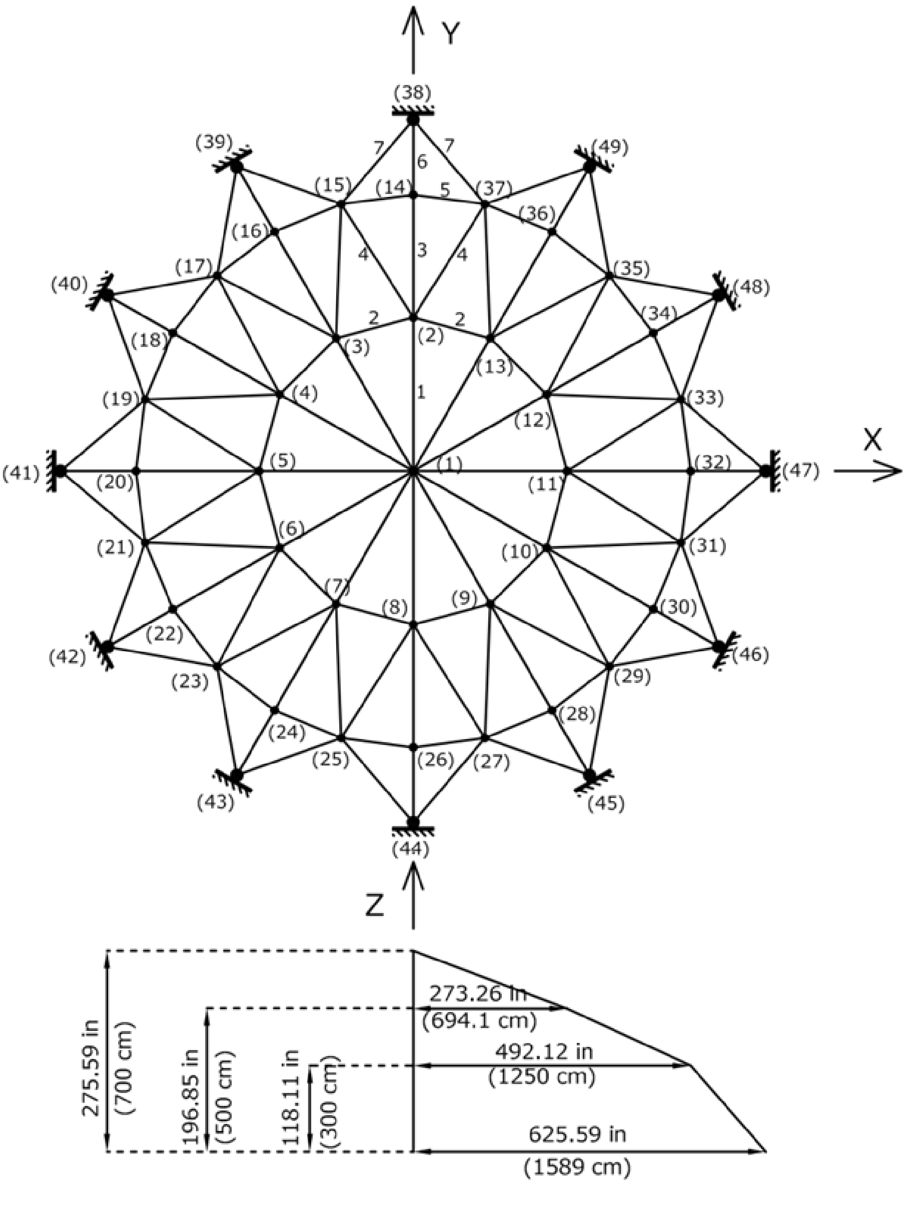

Fig. 9. A 120-bar dome shaped truss

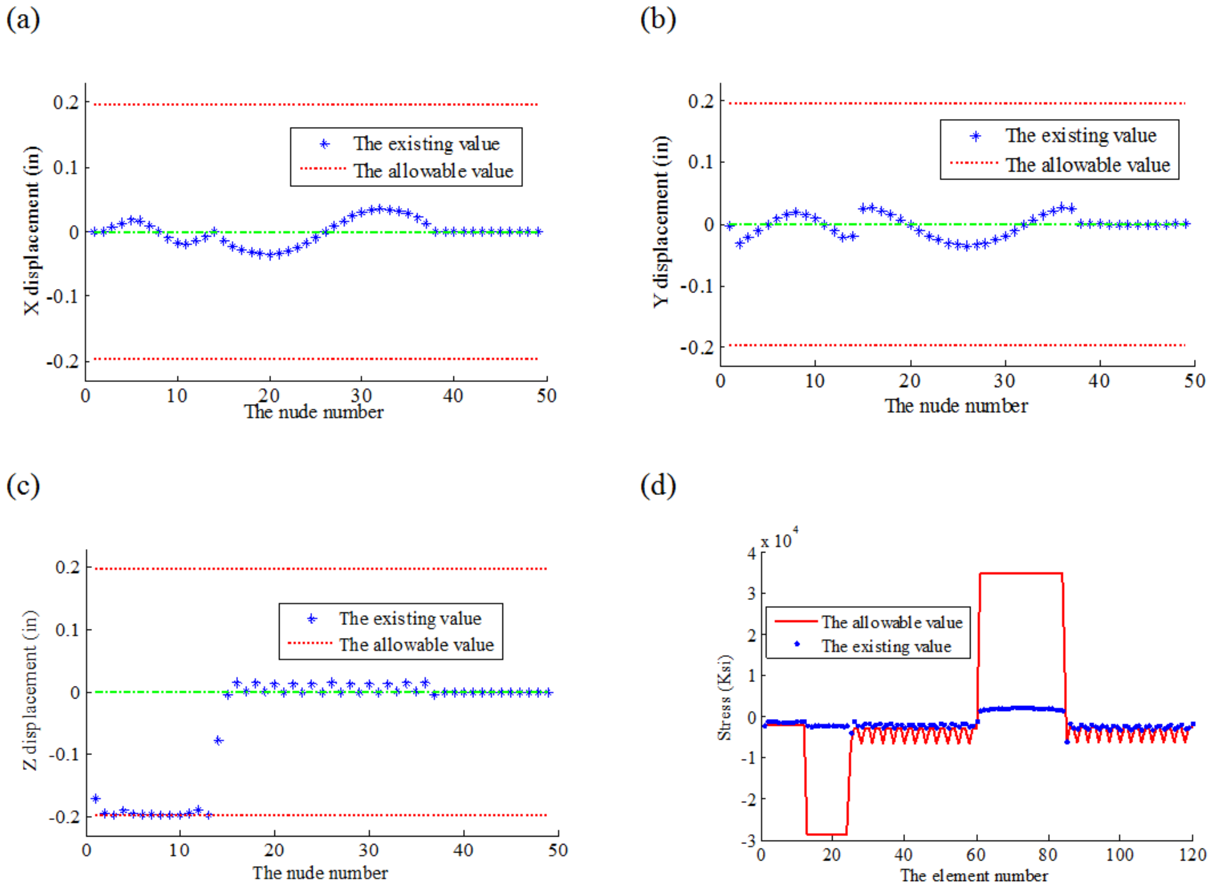

Fig. 10. Comparison of the allowable and existing constraints for the 120-bar dome shaped truss using the IRO (a) Displacement in the x direction, (b) Displacement in the y direction, (c) Displacement in the z direction, (d) Stresses

3.2.8 Design of the 120-bar dome shaped truss

The topology, nodal numbering and element grouping of the 120-bar dome truss are shown in Figure 9. For clarity, not all the element groups are numbered in this figure. The 120 members are categorized into seven groups, because of symmetry. Other conditions of problem are as follows [8], the modulus of elasticity is 30,450 ksi (210,000 MPa) and the material density is 0.288 lb/in^3 (7971.810 kg/m^3). The yield stress of steel is taken as 58.0 ksi (400 MPa). The dome is considered to be subjected to vertical loading at all the unsupported joints. These loads are taken as −13.49 kips (−60 kN) at node 1, −6.744 kips (−30 kN) at nodes 2 through 14, and −2.248 kips (−10 kN) at the rest of the nodes. The minimum cross-sectional area of all members is 0.775 in^2 (5 cm^2) and the maximum cross-sectional area is taken as 20.0 in^2 (129.032 cm^2). The constraints are stress constraints (as defined by Eqs. (27) and (28)) and displacement limitations of ± 0.1969 in (± 5 mm),imposed on all nodes in x, y and z directions.

Table 12 shows the best solution vectors, the corresponding weights and the required number of analyses for convergence of the present algorithm and some other meta-heuristic algorithms. The IRO-based algorithm needs 18300analyses to find the best solution while this number is equal to 150,000, 32,600, 10,000, 10,000, 7000 and 6000 analyses for a PSO-based algorithm [15], a PSO and ACO hybrid algorithm [15], a combination algorithm based on PSO, ACO and HS [15], an improved BB–BC method using PSO properties [16], the CSS algorithm [7] and the ICA algorithm [9], respectively. As a result, the IRO optimization algorithm only has better convergence rates than PSOPC and PSACO algorithms. Comparing the final results of the IRO and those of the other meta-heuristics shows that IRO finds the so nearly optimum design to the best results of other efficient methods while the difference between the result of the IRO and that obtained by the HPSACO [10], as the first best result, is 9 lbs. A comparison of the allowable and existing stresses and displacements of the 120-bar dome truss structure using IRO is shown in Figure 10. The maximum value for displacement is equal to 0.1969 in (5 mm) and the maximum stress ratio is equal to 99.99%.

4 Concluding remarks

A new meta-heuristic algorithm called IRO is developed to improve the performance of the RO algorithm. The basic idea of the RO is the refraction of light described by Snell's law. Upon this phenomenon, when a light ray passes from a lighter medium to a darker or denser one, its direction movement vector refracts. Due to the nature of the RO, by dividing the solution vector into two or three-variable groups, one can join relevant variable to each other and solve any large variable problems. In order to generate new solution vectors in the IRO, dynamic parameters are utilized which do not need categorization, furthermore these make a better balance between exploration and exploitation. By considering a memory, modifying violating agent rule and defining increasing function, the stochastic and handling constraints abilities of the algorithm are improved.

Sixteen mathematical functions with three standard engineering optimization trusses are considered to show the efficiency and robustness of the proposed method. Numerical results indicate that the IRO is an acceptable stochastic search technique for various optimization problems and performs better in terms of accuracy and the number of objective function evaluations than the standard RO and some other well-known meta-heuristic algorithms.

Acknowledgement

The first author is grateful to the Iran National Science Foundation for the support.

References

1 **Lee K, Geem W**, *A new structural optimization method based on the harmony search algorithm*, Computers and Structures, **82**, (2004), 781–798, DOI 10.1016/j.compstruc.2004.01.002.

2 **Rajeev S, Krishnamoorthy C**, *Discrete optimization of structures using genetic algorithms*, Journal of Structural Engineering, **118**, (1992), 1233–1250, DOI 10.1061/(ASCE)0733-9445(1992)118:5(1233).

3 **Lamberti L**, *An efficient simulated annealing algorithm for design optimization of truss structures*, Computers and Structures, **86**, (2008), 1936–1953, DOI 10.1016/j.compstruc.2008.02.004.

4 **Camp C, Bichon B**, *Design of space trusses using ant colony optimization*, Journal of Structural Engineering, **130**, (2004), 741–751, DOI 10.1061/(ASCE)0733-9445(2004)130:5(741).

5 **Kaveh A, Talatahari S**, *A particle swarm ant colony optimization for truss structures with discrete variables*, Journal of Constructional Steel Research, **65**, (2009), 1558–1568, DOI 10.1016/j.jcsr.2009.04.021.

6 **Camp C**, *Design of space trusses using big bang-big crunch optimization*, Journal of Structural Engineering, **133**, (2007), 999–1008, DOI 10.1061/(ASCE)0733-9445(2007)133:7(999).

7 **Kaveh A, Talatahari S**, *Optimal design of skeletal structures via the charged system search algorithm*, Structural and Multidisciplinary Optimization, **41**, (2010), 893–911, DOI 10.1007/s00158-009-0462-5.

8 **Kaveh A, Talatahari S**, *Optimum design of skeletal structures using imperialist competitive algorithm*, Computers and Structures, **88**, (2010), 1220–1229, DOI 10.1016/j.compstruc.2010.06.011.

9 **Kaveh A, Motie Share M, Moslehi M**, *Magnetic charged system search: a new meta-heuristic algorithm for optimization*, Acta Mechanica, **224**, (2013), 85–107, DOI 10.1007/s00707-012-0745-6.

10 **Lamberti L, Pappalettere C**, *Metaheuristic design optimization of skeletal structures: a review*, Computational Technology Reviews, **4**, (2011), 1–32.

11 **Kaveh A, Khayatazad M**, *A new meta-heuristic method: ray optimization*, Computers and Structures, **112–113**, (2012), 283–294, DOI 10.1016/j.compstruc.2012.09.003.

12 **Tsoulos I**, *Modifications of real code genetic algorithm for global optimization*, Applied Mathematics and Computation, **203**, (2008), 598–607, DOI 10.1016/j.amc.2008.05.005.

13 *Manual of steel construction–allowable stress design*, 9th edn., American Institute of Steel Construction (AISC); Chicago, USA, 1989.

14 **Kaveh A, Talatahari S**, *Ant colony optimization for design of space trusses*, International Journal of Space Structures, **23**, (2008), 167–181, DOI 10.1260/026635108786260956.

15 **Kaveh A, Talatahari S**, *Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss*

structures, Computers and Structures, **87**, (2009), 267–283, DOI 10.1016/j.compstruc.2009.01.003.

16 **Kaveh A, Talatahari S**, *Size optimization of space trusses using Big Bang-Big Crunch algorithm*, Computers and Structures, **87**, (2009), 1129–1140, DOI 10.1016/j.compstruc.2009.04.011.