Computers and Structures 112-113 (2012) 283–294

Contents lists available at SciVerse ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

A new meta-heuristic method: Ray Optimization

A. Kaveh *, M. Khayatazad

Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran 16, Iran

ARTICLE INFO

Article history:
Received 24 February 2012
Accepted 12 September 2012
Available online 12 October 2012

Keywords:
Meta-heuristic
Optimization
Ray Optimization (RO)
Snell's law
Truss structures

ABSTRACT

In this paper a new meta-heuristic method, so-called Ray Optimization, is developed. Similar to other multi-agent methods, Ray Optimization 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 medium, 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 Ray Optimization algorithm.

1. Introduction

Mathematical programming and meta-heuristics are well-known methods to optimize mathematical and engineering problems. In the mathematical programming there are several ways to perform optimization that can be listed as: linear programming, homogenous linear programming, non-linear programming, integer programming and dynamic programming. These methods of optimization have a faster convergence and high accuracy, however, because of gradient requirements, dependency on a suitable initial starting point, and other properties these have given their place to the meta-heuristic methods. The meta-heuristic algorithms can obtain global or near-global optimum solutions by means of their exploration and exploitation capabilities. Though the solutions obtained from these methods are not necessarily absolute optimum, but they can be found in an affordable time. These stochastic methods do not need derivatives of the objective function and constraints, in contrary to the mathematical programming approaches.

Nature has always been the human's teacher. As an example, the invention of RADAR was possible by means of inspiration from the behavior of the bats. Nowadays engineers utilize the guidance of their teacher in finding optimum solutions. In fact inspiration from various phenomena of nature is the key for solution of the optimization problems. Examples are Dorigo et al. [1], who utilized the behavior of a colony of ants in finding food in a very fast way, and Eberhart and Kennedy [2], who used migration of birds for finding the global best solution in the search space. Among many such procedures, one can list: Evolutionary Algorithm (EA) by Fogel et al. [3], De Jong [4] and Koza [5], Genetic Algorithm by Holland [6] and Goldberg [7], Taboo search (TA) by Glover [8], Gravitational Search Algorithm by Rashedi [9], charged system search by Kaveh and Talatahari [10] and so on. Some of these are inspired by biological evolutionary processes, some by the behavior of animals, and some others adopted from physical phenomena.

The present method is inspired by transition of ray from one medium to another from physics. Here the transition of ray is utilized for finding the global or near-global solution. This algorithm is called Ray Optimization (RO) and uses the Snell's refraction law of the light.

2. Definitions and concepts from ray theory

2.1. Background

The contents of this section and Section 2.2 mainly follow the definitions and concepts presented in Ref. [11].

Certain transparent materials so-called dielectric, refract the light. As the light travels through these materials, its path is changed according to the Snell's refraction law. Each transparent material has an index of refraction. Showing the index of the refraction of the lighter material by n_d, and denoting the index of the refraction of the darker material by n_t, 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, $\boldsymbol{n}$, with incoming ray vector and the angle between the normal with the refracted ray vector, respectively, as shown in Fig. 1. Having the

* Corresponding author. Tel.: +98 21 44202710; fax: +98 21 77240398.
E-mail address: alikaveh@iust.ac.ir (A. Kaveh).

0045-7949/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compstruc.2012.09.003

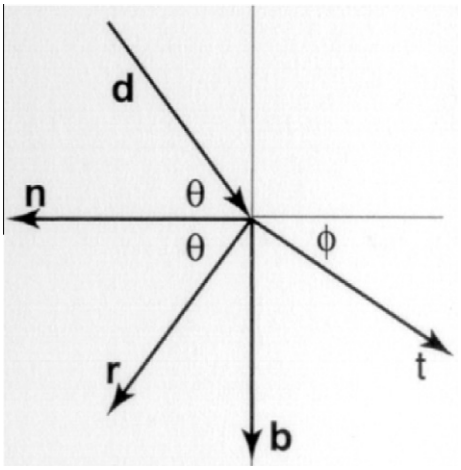

Fig. 1. Incident and refracted rays and their specifications.

direction of incoming ray vector and the index of refraction of lighter and darker mediums, one can find the direction of refracted ray vector $\boldsymbol{t}$.

2.2. Tracing ray in a two dimensional space

The computation for finding $\boldsymbol{t}$ is rather lengthy, but not difficult. To help us along the way, we use Fig. 1, where the vectors $\boldsymbol{d}$, $\boldsymbol{n}$ and $\boldsymbol{b}$ are unit vectors. This makes the formula slightly less complicated.

1. We express $\boldsymbol{t}$ in terms of $\boldsymbol{n}$ and $\boldsymbol{b}$.

   Let $\boldsymbol{t}_n$ be the component of $\boldsymbol{t}$ along $\boldsymbol{n}$ and $\boldsymbol{t}_b$ be the component of $\boldsymbol{t}$ along $\boldsymbol{b}$. Then, since $\boldsymbol{d}$ is a unit vector, we have

   $$\cos(\phi) = \frac{\|\boldsymbol{t}_n\|}{\|\boldsymbol{t}\|} = \|\boldsymbol{t}_n\|. \tag{2}$$

   Similarly,

   $$\sin(\phi) = \|\boldsymbol{t}_b\|. \tag{3}$$

   Now

   $$\boldsymbol{t}_n = -\|\boldsymbol{t}_n\|.n, \tag{4}$$

   and

   $$\boldsymbol{t}_b\| = \|\boldsymbol{t}_b\|.\boldsymbol{b}. \tag{5}$$

   Therefore,

   $$\boldsymbol{t} = -\cos(\phi).\boldsymbol{n} + \sin(\phi).\boldsymbol{b}. \tag{6}$$

1. Express $\boldsymbol{b}$ in terms of the known quantities using the fact that $\boldsymbol{b}$ is the unit length vector parallel to the projection of $\boldsymbol{d}$ onto the perpendicular to $\boldsymbol{n}$. Let $\boldsymbol{d}_n$ be the component of $\boldsymbol{d}$ along $\boldsymbol{n}$ and $\boldsymbol{d}_b$ be the component of $\boldsymbol{d}$ along $\boldsymbol{b}$, the direction perpendicular to $\boldsymbol{n}$. Then

   $$\boldsymbol{d} = \boldsymbol{d}_n + \boldsymbol{d}_b. \tag{7}$$

   Thus

   $$\boldsymbol{d}_b = \boldsymbol{d} - \boldsymbol{d}_n = \boldsymbol{d} - (\boldsymbol{d}.\boldsymbol{n}).\boldsymbol{n}. \tag{8}$$

   Also, since $\boldsymbol{d}$ is a unit vector, we have

   $$\sin(\theta) = \frac{\|\boldsymbol{d}_b\|}{\|\boldsymbol{d}\|} = \|\boldsymbol{d}_b\|. \tag{9}$$

   Thus

   $$\boldsymbol{b} = \frac{\boldsymbol{d}_b}{\|\boldsymbol{d}_b\|} = \frac{\boldsymbol{d} - (\boldsymbol{d}.\boldsymbol{n}).\boldsymbol{n}}{\sin(\boldsymbol{\theta})}. \tag{10}$$

1. Using Eq. (1), one can express everything in terms of $\boldsymbol{n}$, $\boldsymbol{d}$, $\boldsymbol{n_d}$ and $\boldsymbol{n_t}$.
   Therefore

   $$\begin{aligned}\boldsymbol{t} &= -\cos(\phi).\boldsymbol{n} + \sin(\phi).\boldsymbol{b}\\ &= -\cos(\phi).\boldsymbol{n} + \sin(\phi).\frac{\boldsymbol{d} - (\boldsymbol{d}.\boldsymbol{n}).\boldsymbol{n}}{\sin(\theta)}\\ &= -\cos(\phi).\boldsymbol{n} + \frac{n_d}{n_t}.(\boldsymbol{d} - (\boldsymbol{d}.\boldsymbol{n}).\boldsymbol{n})\end{aligned} \tag{11}$$

   Now we express $\cos(\phi)$ in terms of known quantities. Using the identity, we have

   $$\cos(\phi) = \sqrt{1 - \sin^2(\phi)}, \tag{12}$$

   and employing Eq. (1), we obtain:

   $$\cos(\phi) = \sqrt{1 - \frac{\boldsymbol{n}_d^2}{\boldsymbol{n}_t^2}.\sin^2(\theta).} \tag{13}$$

   Finally, we have

   $$\boldsymbol{t} = -\boldsymbol{n}.\sqrt{1 - \frac{\boldsymbol{n}_d^2}{\boldsymbol{n}_t^2}.\sin^2(\theta)} + \frac{n_d}{n_t}.(\boldsymbol{d} - (\boldsymbol{d}.\boldsymbol{n}).\boldsymbol{n}) \tag{14}$$

   Here, $\boldsymbol{t}$ is a normalized vector.

2.3. Tracing ray in a 3-dimensional space

In tracing a ray in a 2-dimesional space, $\boldsymbol{d}$, $\boldsymbol{t}$, $\boldsymbol{n}$ were placed in $z = 0$ plane. In a 3-dimensional space, it is clear that one can pass a plane through two vectors like $\boldsymbol{n}$ and $\boldsymbol{d}$, which intersect each other at one point. Thus the ray tracing in 3-dimensional spaces is a special state of ray tracing in 2-dimensional spaces which occurs in a plane with an arbitrary orientation. Now if one can find two normalized vectors that are perpendicular to each other, like $\boldsymbol{i}^*$ and $\boldsymbol{j}^*$, then $\boldsymbol{n}$ and $\boldsymbol{d}$ can be rewritten in terms of these unit vectors. Finally, after finding $\boldsymbol{t}$ in the new coordinate system, it can be rearranged based on the primary coordinate system. One of these new unit components can be $\boldsymbol{n}$ as $\boldsymbol{i}^*$. The other one can be found by the following relationship:

$$\boldsymbol{n}.\boldsymbol{d} = \|\boldsymbol{n}\|.\|\boldsymbol{d}\|.\cos(\omega) = \cos(\omega) \tag{15}$$

where ω is the angle between $\boldsymbol{n}$ and $\boldsymbol{d}$. Now if

$$\boldsymbol{n}.\boldsymbol{d} = 0 \tag{16}$$

Then $\omega = \pi/2$ and $\boldsymbol{d}$ will be $\boldsymbol{j}^*$, and if

$$0 < \boldsymbol{n}.\boldsymbol{d} \leq 1 \tag{17}$$

then the direction of $\boldsymbol{j}^*$ will be obtained by $(\boldsymbol{n} - \frac{\boldsymbol{d}}{\boldsymbol{n}.\boldsymbol{d}})$, since

$$\boldsymbol{n} \cdot \left(\boldsymbol{n} - \frac{\boldsymbol{d}}{\boldsymbol{n} \cdot \boldsymbol{d}}\right) = \boldsymbol{n} \cdot \boldsymbol{n} - \frac{\boldsymbol{n} \cdot \boldsymbol{d}}{\boldsymbol{n} \cdot \boldsymbol{d}} = 1 - 1 = 0 \tag{18}$$

And finally

$$\boldsymbol{j}^* = \frac{(\boldsymbol{n} - \frac{\boldsymbol{d}}{\boldsymbol{n}.\boldsymbol{d}})}{norm(\boldsymbol{n} - \frac{\boldsymbol{d}}{\boldsymbol{n}.\boldsymbol{d}})}. \tag{19}$$

where norm is function in MATLAB that provides the magnitude of a vector. Similarly if

$$-1 \leq \boldsymbol{n}.\boldsymbol{d} < 0 \tag{20}$$

$\boldsymbol{j}^*$ can be obtained by:

$$\boldsymbol{j}^* = \frac{(\boldsymbol{n} + \frac{\boldsymbol{d}}{-\boldsymbol{n}.\boldsymbol{d}})}{norm(n + \frac{-\boldsymbol{d}}{-\boldsymbol{n}.\boldsymbol{d}})}. \tag{21}$$

Table 1
The component of the new coordinate system.

	$-0.05 \leqslant \boldsymbol{n}.\boldsymbol{d} \leqslant 0.05$	$0.05 \leqslant \boldsymbol{n}.\boldsymbol{d} \leqslant 1$	$-1 \leqslant \boldsymbol{n}.\boldsymbol{d} \leqslant -0.05$
$\boldsymbol{i}^*$	$\boldsymbol{n}$	$\boldsymbol{n}$	$\boldsymbol{n}$
$\boldsymbol{j}^*$	$\boldsymbol{d}$	$\boldsymbol{j}^* = \frac{\left(n - \frac{d}{n.d}\right)}{norm\left(n - \frac{d}{n.d}\right)}$	$\boldsymbol{j}^* = \frac{\left(n + \frac{d}{n.d}\right)}{norm\left(n + \frac{d}{n.d}\right)}$

Now, in the new form, $\boldsymbol{d}$ and $\boldsymbol{n}$ are stated as:

$$\boldsymbol{n}^* = (1, 0), \tag{22}$$

and

$$\boldsymbol{d}^* = (\boldsymbol{d}.\boldsymbol{i}^*, \boldsymbol{d}.\boldsymbol{j}^*) \tag{23}$$

Therefore after calculating $\boldsymbol{t}^* = (t_1^*,\ t_2^*)$ in a two dimensional space, $\boldsymbol{t}$ in a three dimensional space is obtained as

$$\boldsymbol{t} = t_1^*.\boldsymbol{i}^* + t_2^*.\boldsymbol{j}^* \tag{24}$$

It should be mentioned that for avoiding singularity in MATLAB, some changes are considered as shown in Table 1.

3. Ray Optimization method

3.1. Scattering and evaluation step

Like every other meta-heuristic method, the RO has also a number of agents containing the variables of the design problem. As it was mentioned before, the ray tracing which is the main base of the RO was addressed in two and three dimensional spaces but it is necessary to introduce a procedure for performing the steps of the algorithm in higher dimensional spaces.

Suppose a solution vector has 4 design variables. In the first step, the goal function for this solution vector is determined. Now, this solution vector must be moved to a new position in the search space based on the RO algorithm. To achieve this, the solution vector is divided into two groups each group having 2 members and then the corresponding agents are moved to their new positions. In other word, we move the first group to the new position based on a 2D space and also the second group is moved to the new position in another 2D space. Now, we have a new solution vector with four variables which is ready for determining the goal function.

If a solution vector for another problem has seven variables, then we divide it into two groups of two variables and one group of three variables, and repeat the above procedure in two 2D spaces and one 3D space for the movement. After the movements, we join them together to have a unit solution vector. For any other number of variables the grouping into two and three elements can be performed. Therefore by using this approach, the problem of higher dimensions is dealt with and one can return to the first step of the algorithm. In this step, the agents must be scattered in the search space, and this requirement is provided by:

$$\boldsymbol{X}_{ij} = \boldsymbol{X}_{j,\min} + rand.(\boldsymbol{X}_{j,\max} - \boldsymbol{X}_{j,\min}). \tag{25}$$

where $\boldsymbol{X}_{ij}$ is the jth variable of the ith agent. $\boldsymbol{X}_{j,\min}$ and $\boldsymbol{X}_{j,\max}$ are the minimum and maximum limits of the jth variable, and *rand* is a random number with its range being between 0 and 1. At the end of this step, after the evaluation of the goal function for each agent, the position of the best agent is saved as the global best and the position of each agent is saved as its local best.

3.2. Movement vector and motion refinement step

For each of the above mentioned agents, a groups of movement vectors should be assigned according to their division, and if the agent has a one 3-variable group and two 2-variable groups, it must have a one 3-variable movement vector group and two 2-variable movement vector groups, respectively. For the first movement, these vectors are obtained by:

$$\boldsymbol{V}_{ij} = -1 + 2.rand. \tag{26}$$

where $\boldsymbol{V}_{ij}$ is the jth component of the ith agent and it may belong to a 2-variable or a 3-variable group. After finding the components of each 2 or 3-variable group, these must be converted to normalized vectors. The reason of this action will be presented in the subsequent steps. Now by adding the movement vector of each agent, they move to their new positions, but there is a possibility of boundary violation, so they must be refined. This objective is fulfilled by the following procedure:

If an agent violates a boundary, it intersects the boundary at a specified point, because of having definite movement vector. Now using this point and initial position of the agent, one can make a new vector whose direction is the same as the prior movement vector and its length is a multiple of the distance which is between the initial position and the boundary intersection. This multiple should naturally be less than 1. Since the best answer, especially in engineering problems, is close to the boundaries [12], it is locked on 0.9. During the implementation of this stage, it is found that when a movement vector is very close to a unit one-component vector such (1,0) and (0,1) for 2-dimensional spaces and (1,0,0), (0,1,0) and (0,0,1) for 3-dimensional spaces, it causes singularity in the process of finding the intersection point at the boundary. For solving this problem, after normalizing the movement vector, if one component of this vector is equal or greater than 0.95, the other component(s) are not considered and upon this solitary component, the new movement vector is made.

After motion refinement and evaluation of the goal function, again the so-far best agent at this stage is selected as the global best and for each agent, the so-far best position by this stage, is selected as its local best.

3.3. Origin making and convergent step

Now 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 (cockshy) and it is specified by:

$$\boldsymbol{O}_i^k = \frac{(ite + k).\boldsymbol{GB} + (ite - k).\boldsymbol{LB}_i}{2.ite}. \tag{27}$$

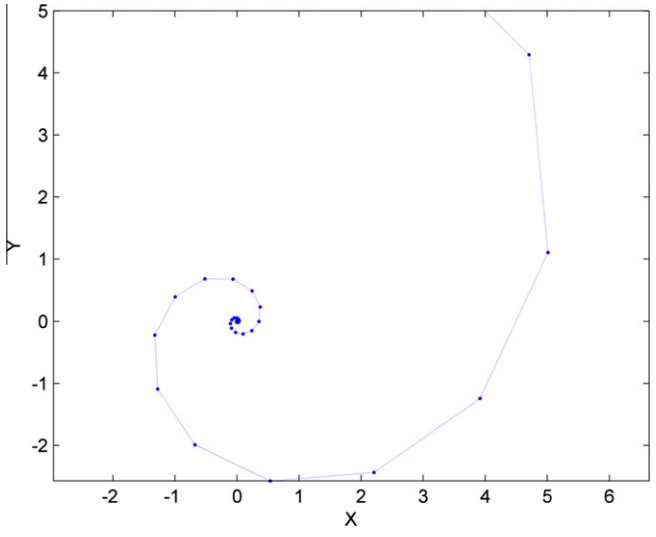

Fig. 2. The movement of an agent to the origin.

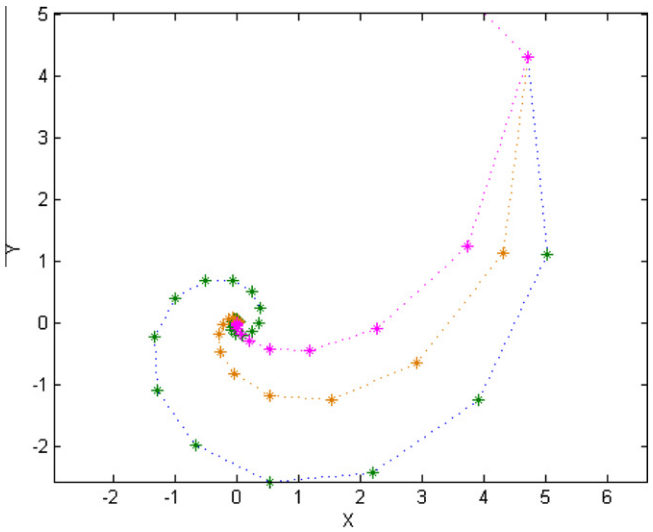

Fig. 3. Ratio of the index refraction of the magnet line, brown line and blue line are 0.5, 0.65 and 0.85, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $\boldsymbol{O}_i^k$ is the origin of ith agent for the kth iteration, ite is the total number of iteration for the optimization process, and $\boldsymbol{GB}$ and $\boldsymbol{LB}_i$ are the global best and local best of the ith agent, respectively. As Eq. (27) implies, at the beginning of iterations, the origin is approximately at the middle of the local best and global best. Thus exploration which is an important parameter in optimization is achieved. By progressing the iterations there is a balance between exploration and second important parameter, known as the exploitation. By passing the mid steps of iterations, the origin will be close to global best and the exploration is empowered.

As it was mentioned before, the ray tracing is used for the movement and convergent step. Each ray of light has a normalized vector with which passes the medium. In the optimization, the movement vector is a positive multiple of this vector. When the ray comes to a new darker medium, the direction of its vector will be converted upon its angle between the initial direction and normal of surfaces of mediums and the ratio of refraction index. After refraction, the new direction will be closer to the normal than the initial direction, so one can say it converges to the normal. Therefore if in the process of optimization, the normal is selected as a vector whose origin is O and its end is the current position of agent, it should be anticipated that the agent will converge to O. With refinement of O during the optimization, the agents approach to best result. Now, the direction of the new movement vector can be created because $\boldsymbol{n}$ and incoming ray vector, $\boldsymbol{d}$, which is the last movement vector, are obtained. But it is necessary to notice that, $\boldsymbol{n}$ and the last movement vector might not be a unit vector and based on Eq. (14), these vectors must be normalized.

In order to show how the agents converge the to a point, consider an agent with arbitrary position and movement vector in the two dimensional space, like (4,5) and (0.707, −0.707), respectively. If this particle wants to move to a predefined point like the origin, it can be moved toward this point as illustrated in Fig. 2. It should be mentioned that, some restrictions are imposed for this convergence. These consist of a fixed ratio 0.6 as the index refraction and 200 times as the number of iterations. In order to show how the ratio of the index refraction affects the search procedure, see Fig. 3 where the above problem is solved with different values of the index refraction ratios. As can be seen, when the index refraction ratio is a number close to 1, the exploration is increased, but by decreasing this value the convergence occurs rapidly.

Now, the direction of the new movement vector is determined. According to Eq. (14), it is a normalized vector and it requires a logical coefficient. Thus the final form of the movement vector after finding the new direction is given by:

$$\boldsymbol{V}_{i,l} = \boldsymbol{V}'_{i,l}.norm(\boldsymbol{X}_{i,l} - \boldsymbol{O}_{i,l}). \tag{28}$$

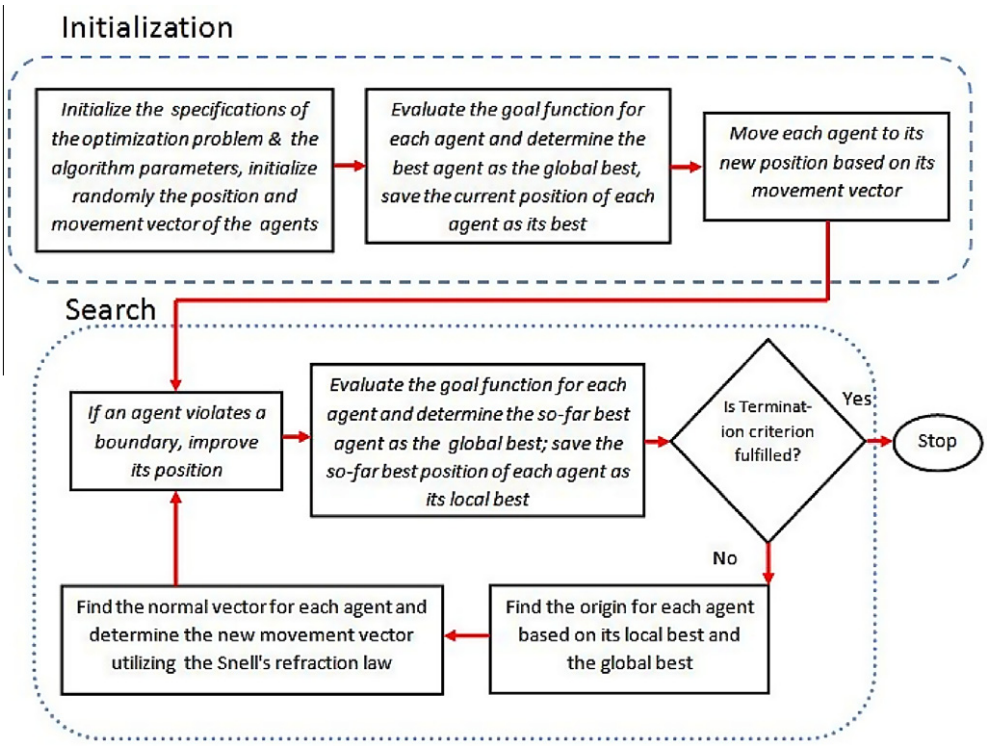

Fig. 4. The flowchart of the RO.

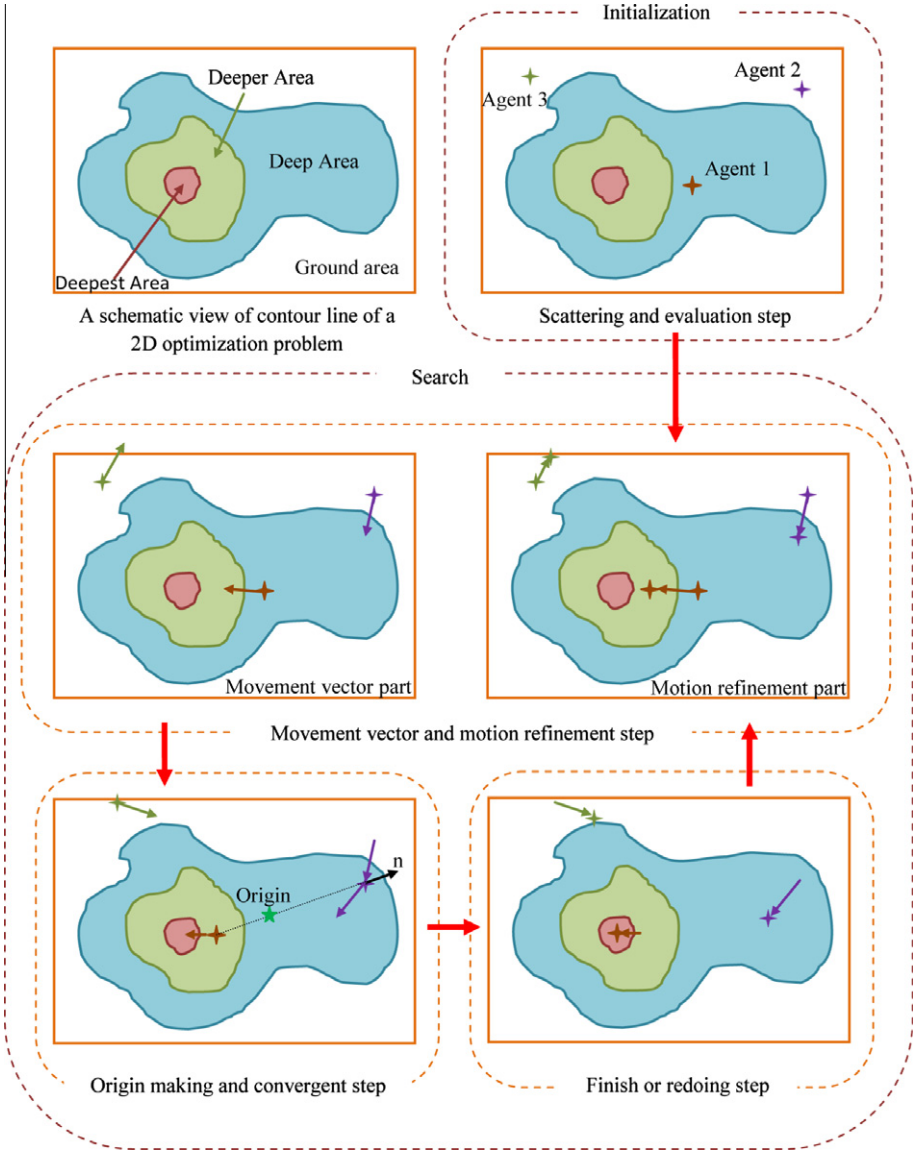

Fig. 5. The schematic view of the agent movement.

Table 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	$0 \leqslant x_2 \leqslant 15\ -5 \leqslant x_1 \leqslant 10$	$f(X) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$	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	$n = 4, \boldsymbol{X} \in [-1, 1]^n$	$f(X) = \sum_{i=1}^n x_i^2 - \frac{1}{10}\sum_{i=1}^n \cos(5\pi x_i)$	−0.4
De Joung	$\boldsymbol{X} \in [-5.12,5.12]^2$	$f(X) = x_1^2 + x_2^2 + x_3^2$	0.0
Exponential	$n = 2, 4, 8, \boldsymbol{X} \in [-1,1]^n$	$f(X) = -\exp(-0.5\sum_{i=1}^n x_i^2)$	−1.0
Goldstein And price	$\boldsymbol{X} \in [-2,2]^2$	$f(X) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2]$ $\times[30 + (2x_1 - 3x_2)^2(18 - 32x_1 - 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2]$	3.0
Griewank	$\boldsymbol{X} \in [-100,100]^2$	$f(X) = 1 + \frac{1}{200}\sum_{i=1}^n x_i^2 - \prod_{i=1}^2 \cos\left(\frac{x_i}{\sqrt{i}}\right)$	0.0
Rastrigin	$\boldsymbol{X} \in [-1,1]^2$	$f(X) = \sum_{i=1}^n (x_i^2 - \cos(18x_i))$	−2.0

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 ith agent, respectively, that belong to lth group.

In some cases it is possible that for an agent, $\boldsymbol{O}_{i,l}$ and its current position are the same, so the direction of normal cannot be obtained. This problem occurs when the agent is the so-far best one. Therefore, it is logical to permit it to move in the same direction because of finding a more adequate answer, but the length of this vector should be changed according to:

$$\boldsymbol{V}_{i,l}^{k+1} = \frac{\boldsymbol{V}_{i,l}^{k}}{norm(\boldsymbol{V}_{i,l}^{k})}.rand.0.001. \tag{29}$$

In this equation, $\boldsymbol{V}_{i,l}^{k}$ is the movement vector of the kth iteration that belongs to lth group of the ith agent, and $\boldsymbol{V}_{i,l}^{k+1}$ is the movement vector of the$(k+1)$th iteration. Also, for a fine and stochastic search, the initial normalized vector is multiplied by 0.001 and a random number between 0 and 1 is utilized.

One of the important features of each meta-heuristic algorithm is that, it should have a stochastic nature to find the best answer. Here, this feature is added to the RO by adding a random change to the movement vector. In other word, there is a possibility like ***stoch*** that specifies whether a movement vector must be changed or not. If this occurs, a new movement vector will be made considered as

$$\boldsymbol{V}_{ijl}^{(k+1)} = -1 + 2.rand. \tag{30}$$

where $\boldsymbol{V}_{ijl}^{(k+1)}$ is the jth component of the lth group that belongs to the ith agent in $(k+1)$th iteration. However, the length of this vector should be refined. Therefore the following relationship is considered:

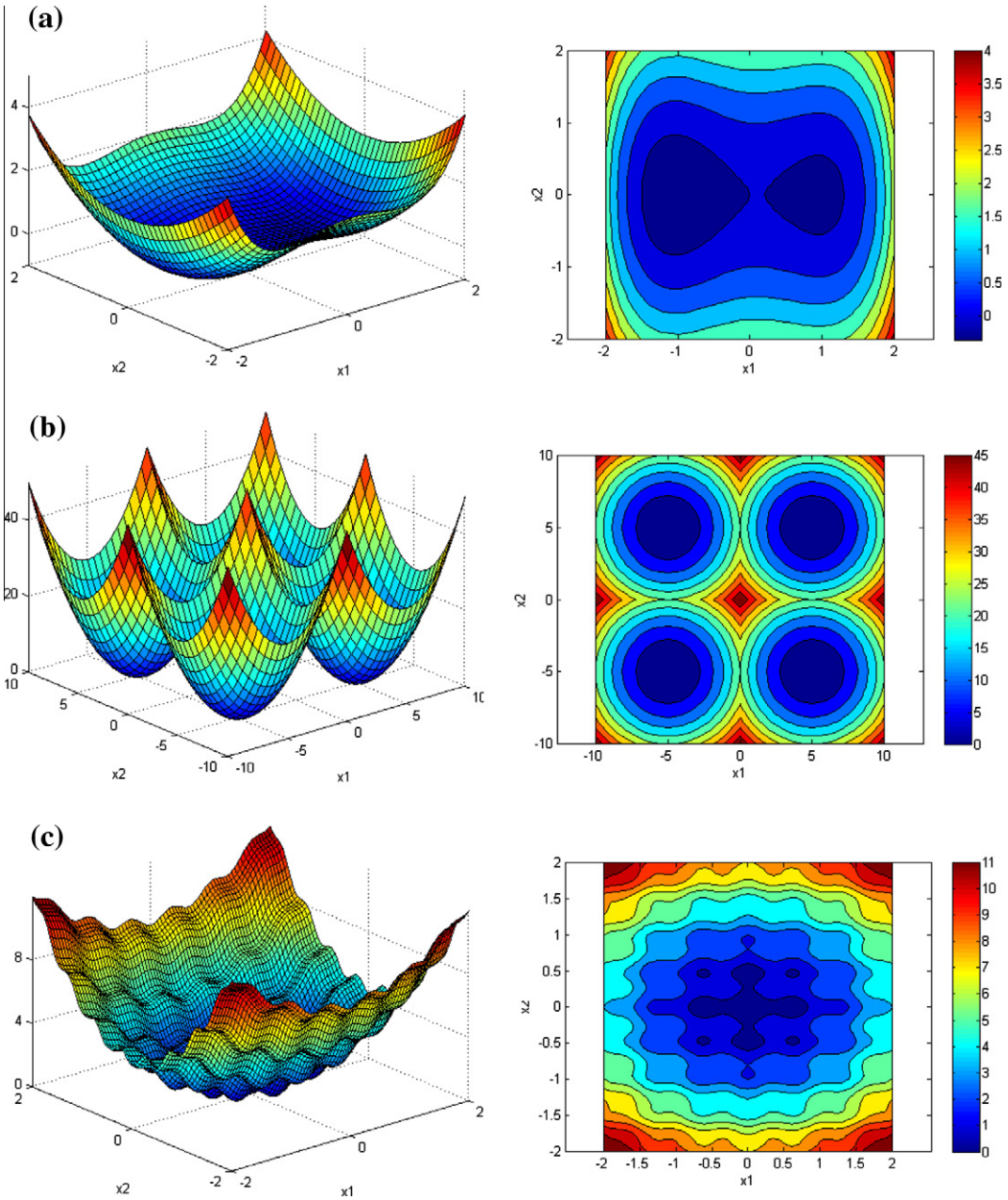

Fig. 6. A perspective view and the related contour lines for some of function when $n = 2$. a. Aluffi-Pentiny, b. Becker and lago, c. Bohachevsky 1.

$$\boldsymbol{V}_{il}^{k+1} = \frac{\boldsymbol{V}_{il}^{(k+1)'}}{norm(\boldsymbol{V}_{il}^{(k+1)'})} \cdot \frac{a}{d} . rand. \tag{31}$$

Here, a is calculated by the following relationship:

$$a = \sqrt{\sum_{i=1}^{n} (\boldsymbol{X}_{i,\max} - \boldsymbol{X}_{i,\min})^2} \quad n = \left\{ \begin{matrix} 2 & \text{for two variable groups} \\ 3 & \text{for three variable groups} \end{matrix} \right\} \tag{32}$$

where $\boldsymbol{X}_{i,\max}$ and $\boldsymbol{X}_{i,\min}$ are the maximum and minimum limits of variables that belongs to ith component of the movement vector, and d is a number that divides a into smaller segments for effective search. It is found that if d and *stoch* are chosen as 7.5 and 0.35, the best result for optimization process will be obtained.

3.4. Finish or redoing step

By approaching to a pre-specific criterion, the process of optimization ceases. Some criteria are considered as

- Maximum number of iteration: by approaching to a predefined number of iteration, the process of optimization will be terminated.
- Number of ineffective iteration: if by passing a predefined number of iteration, there is no improvement in the goal function, the process of optimization will be ceased.
- Approaching to a minimum goal function error: sometimes the best answer of a goal function is specified, like mathematical benchmarks, so if an answer is found with a predefined error compared to real answer, the process of optimization will be terminated.

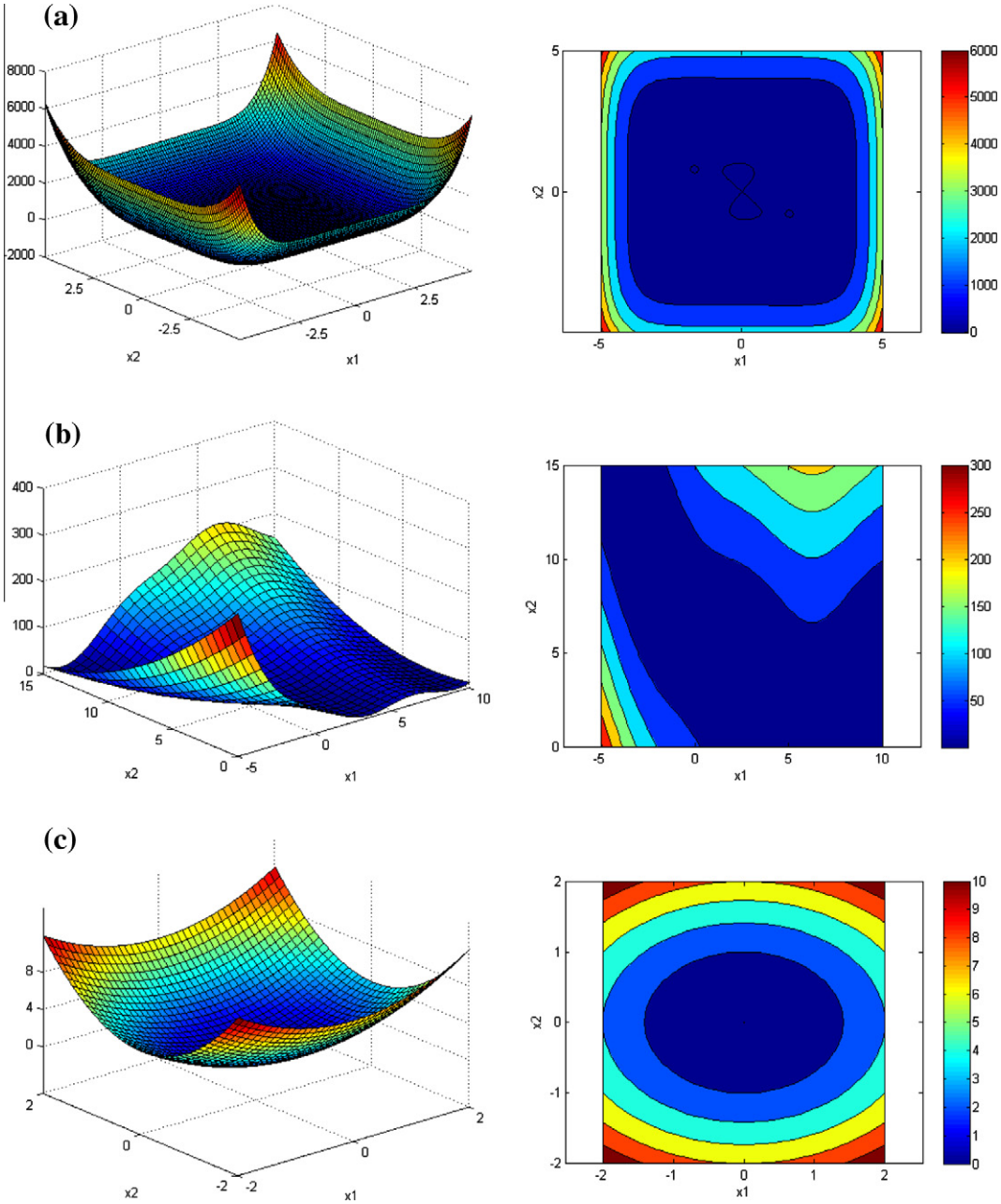

Fig. 7. A perspective view and the related contour lines for some of function when $n = 2$. a. Camel, b. Branin, c. Bohachevsky 2.

However if one of these criteria is not fulfilled, the process of optimization will continue and with new movement vector, the agents will move to their new positions. This cycle is continued until a predefined criterion is fulfilled.

The flowchart of the optimization is illustrated in Fig. 4. The movement of a typical agent is shown in Fig. 5. As can be seen, in the origin making and convergent step, because of similar global best position and local best position that belong to agent 1, this agent moves in the same direction but with a smaller length. The movement vector for the agent 2 is determined based on the light refraction law. Finally, for showing the stochastic behavior of RO, agent 3 moves in a random direction with a controlled length.

4. Validation of the Ray Optimization

In order to validate the efficiency of the RO, some mathematical examples are considered from literature. These examples are investigated in Section 4.1. For further investigation, in Section 4.2, some well-studied engineering design problems are also studied from the literature.

4.1. Mathematical optimization problems

In order to verifying the efficiency of the RO algorithm, some benchmark functions are optimized by this algorithm, Table 2. Additional functions SHEKEL5, SHEKEL7, SHEKEL10 are presented

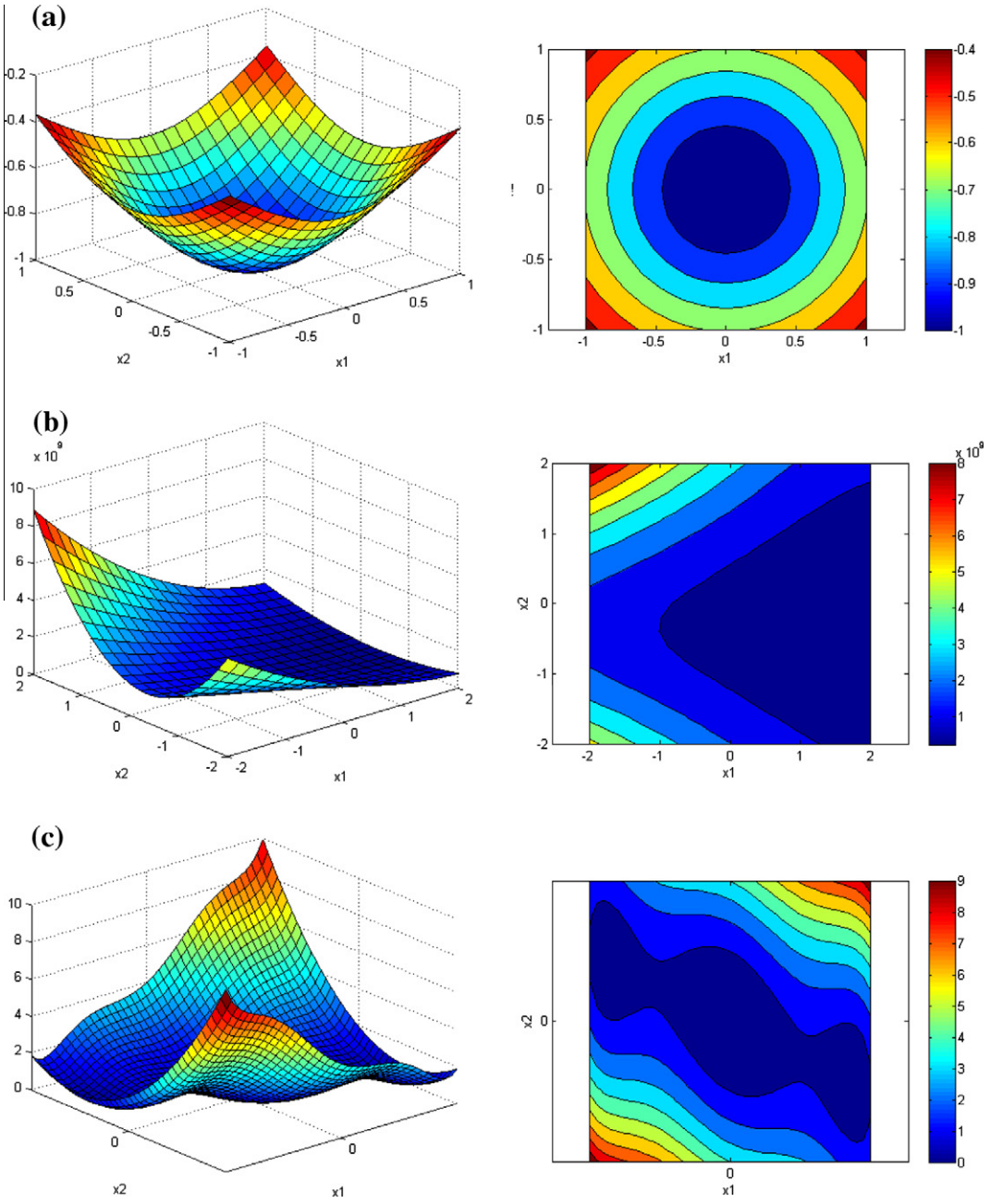

Fig. 8. A perspective view and the related contour lines for some of function when $n = 2$. a. Exponential, b. Goldstein and price, c. Cb3.

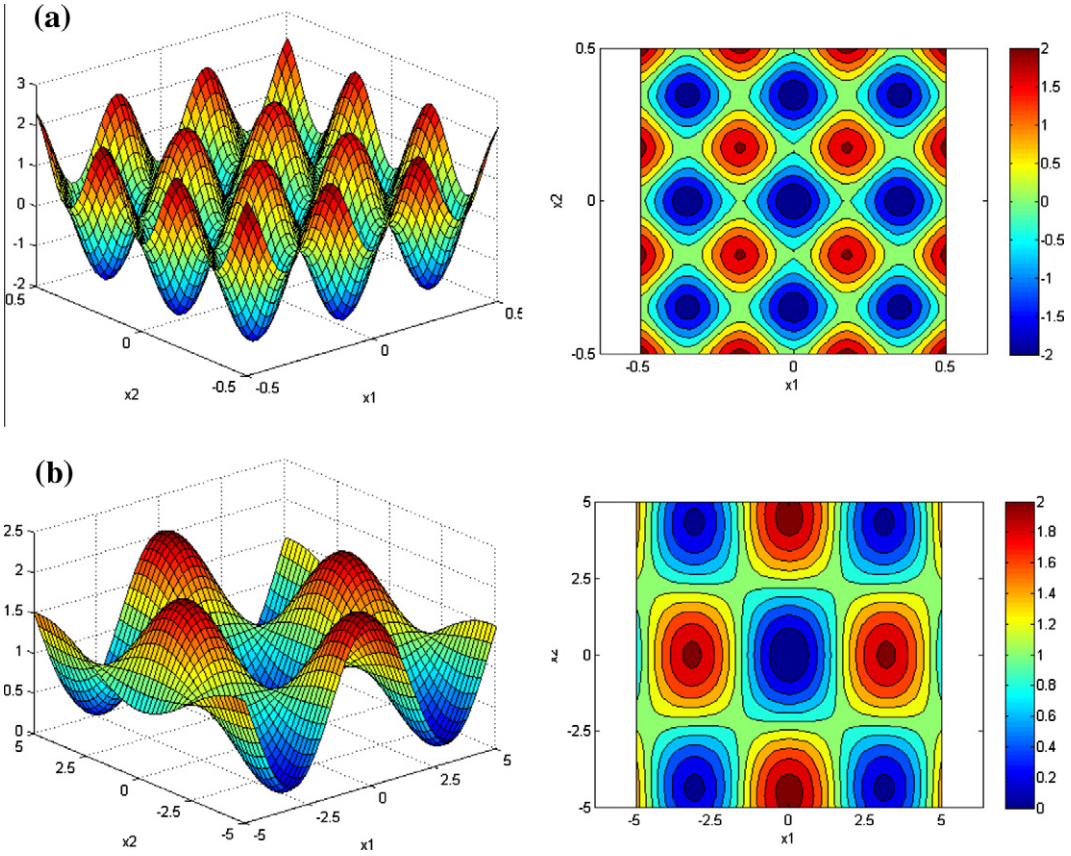

Fig. 9. A perspective view and the related contour lines for some of function when $n = 2$. a. Griewank, b. Rastrigin.

Table 3
Performance comparison for the benchmark problems.

FUNCTION	GEN	GEN_S	GEN_S_M_LS	Ray Optimization
AP	1360(0.99)	1360	1253	331
Bf1	3992	3356	1615	677
Bf2	20,234	3373	1636	582
BL	19,596	2412	1436	303
Branin	1442	1418	1257	463
Camel	1358	1358	1300	332
Cb3	9771	2045	1118	262
CM	2105	2105	1539	802
Dejoung	9900	3040	1281	452
EXP2	938	936	807	136
EXP4	3237	3237	1496	382
EXP8	3237	3237	1496	1287
EXP16	8061	8061	1945	17,236(0.46)
GRIEWANK	18,838(0.91)	3111(0.91)	1652(0.99)	1091(0.98)
RASTRIGIN	1533(0.97)	1523(0.97)	1381	1013(0.98)
Goldstein and Price	1478	1478	1325	451
SHEKEL5	2527(0.61)	2507(0.61)	2049(0.67)	3401(0.52)
SHEKEL7	2567(0.72)	2500(0.72)	2032(0.75)	3459(0.76)
SHEKEL10	2641(0.71)	2598(0.71)	2141(0.76)	4152(0.66)
TOTAL	114,815(94.26)	49,655(94.31)	28,759(95.63)	36,812(91.36)

are examined which can be found in Ref. [13]. A perspective view and the corresponding contour lines for these benchmarks are depicted in Figs. 6–9), where the number of variables is taken as 2. Like any other meta-heuristics, considering a large number of agent causes vigorous search, but it increases the time and cost of the optimization. Also considering a small number of agents leads to a weak search. Therefore, for optimizing the mathematical problems, the number of agents is taken as 20. However it should mentioned we used 100 agents for the EXP16, SHEKEL5, SHEKEL7, SHEKEL10. Table 3 shows the result of the optimization by the present approach. In this table, the numbers in columns specify the average number of function evaluations. In this paper for providing the stochastic behavior of meta-heuristic algorithms, the number of independent runs is chosen as 50. The numbers in the parentheses represent the ratio of successful runs in which the algorithm has found the global minimum with a predefined accuracy, which is taken as $\varepsilon = f_{\min} - f_{\text{final}} = 10^{-4}$. The absence of the parentheses shows that the algorithm has been successful in

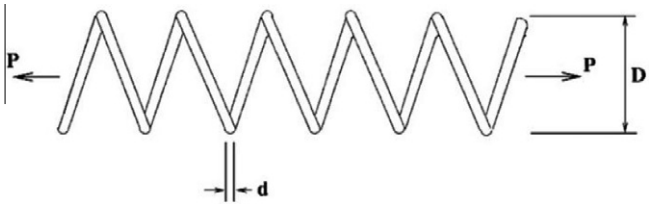

Fig. 10. A tension/compression spring.

Table 4
Specifications of the tension/compression spring problem.

Cost function	$f(X) = (x_3 + 2)x_2x_1^2$
Constraint functions	$g_1(X) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \le 0$
	$g_2(X) = \frac{4x_2^4 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0$
	$g_3(X) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0$
	$g_4(X) = \frac{x_1+x_2}{1.5} - 1 \le 0$
Variable region	$0.05 \le x_1 \le 2$
	$0.25 \le x_2 \le 1.3$
	$2 \le x_3 \le 15$

Table 5
Optimum results for the tension/compression spring design.

	Optimal design variable			
Methods	$X_1(d)$	X_2(D)	X_3(N)	f_{cost}
Belegunda [14]	0.050000	0.315900	14.250000	0.0128334
Arora [15]	0.053396	0.399180	9.185400	0.0127303
Present work	0.051370	0.349096	11.76279	0.0126788

all independent runs. The results show that the present algorithm has a higher efficiency than GA and some of its variants for these benchmark functions.

4.2. Engineering design problems

In this section, three engineering design problems are presented to show the efficiency of the RO. For the sake of simplicity, the penalty approach is used for constraint handling. In using the penalty function, if the constraints are not violated, the penalty will be zero; otherwise, the value of the penalty is calculated by dividing the violation of the allowable limit to the limit itself. It should be mentioned that in the RO, the use of this type of constraint handling is not a necessity and any other type of constraint handling approach can be employed.

Table 6
Specifications of a welded beam design problem.

Cost function	$f(\boldsymbol{X}) = 1.1047x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$
Constraint functions	$g_1(X) = \tau(X) - \tau_{max} \le 0$
	$g_2(X) = \sigma(X) - \sigma_{max} \le 0$
	$g_3 = x_1 - x_4 \le 0$
	$g_4(\boldsymbol{X}) = 0.1047x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0$
	$g_5(\boldsymbol{X}) = 0.125 - x_1 \leqslant 0$
	$g_6(\boldsymbol{X}) = \delta(\boldsymbol{X}) - \delta_{max} \leqslant 0$
	$g_7(\boldsymbol{X}) = P - P_c(X) \leqslant 0$
	$\tau(\boldsymbol{X}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$
	$\tau' = \frac{P}{\sqrt{2}x_1x_2} \quad \tau'' = \frac{MR}{J}$
	$M = P(L + \frac{x_2}{2}), \quad R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2}$
	$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + (\frac{x_1+x_3}{2})^2\right]\right\}$
	$\sigma(\boldsymbol{X}) = \frac{6PL}{x_4x_3^2}, \quad \delta(\boldsymbol{X}) = \frac{4PL^3}{Ex_4x_3^3}$
	$P_c(\boldsymbol{X}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$
	$P = 6000$ lb, $L = 14$ in
	$E = 30 \times 10^6$ psi $G = 12 \times 10^6$ psi
	$\tau_{max} = 13.6 \times 10^3$ psi, $\sigma_{max} = 30 \times 10^3$ psi, $\delta_{max} = 0.25$ in
Variable regions	$0.1 \le x_{1,4} \le 2$
	$0.1 \le x_{2,3} \le 10$

Table 7
Optimum results for the design of welded beam.

Methods	Optimal design variables				
	$X_1(h)$	$X_2(l)$	$X_3(t)$	$X_4(b)$	f_{cost}
APPROX	0.2444	6.2189	8.2915	0.2444	2.3815
DAVID	0.2434	6.2552	8.2915	0.2444	2.3841
SIMPLEX	0.2792	5.6256	7.7512	0.2796	2.5307
RANDOM	0.4575	4.7313	5.0853	0.6600	4.1185
Deb [16]	0.248900	6.173000	8.178900	0.253300	2.433116
Present work	0.203687	3.528467	9.004233	0.207241	1.735344

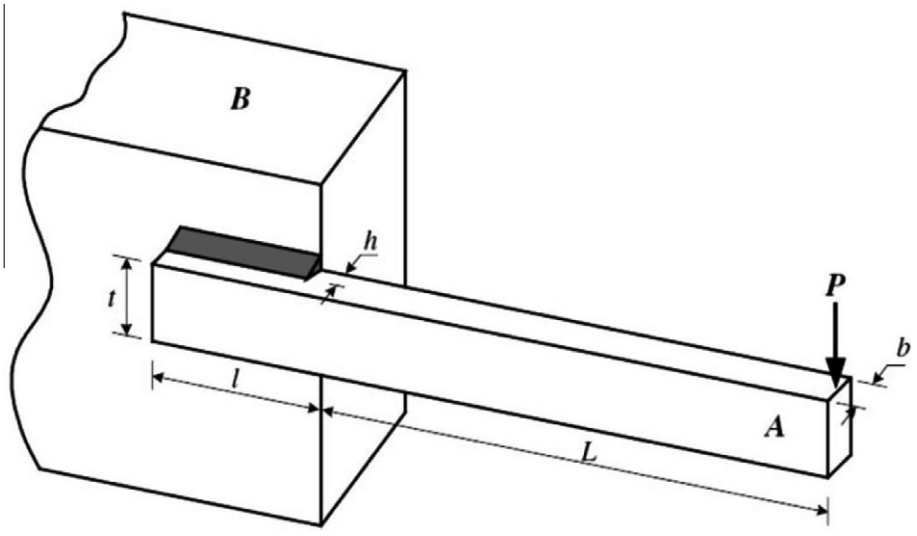

Fig. 11. A welded beam system.

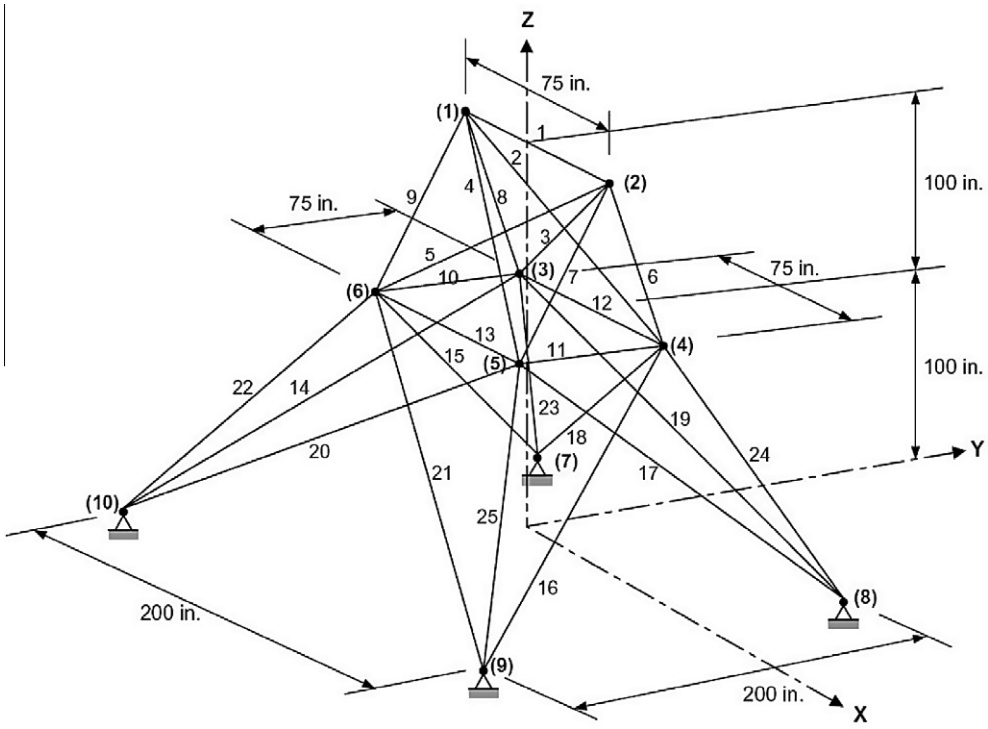

Fig. 12. A twenty five-bar spatial truss.

Table 8
Loading conditions for the 25-bar spatial truss.

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	−20.0(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
4	0.0	0.0	0.0	0.5(2.22)	0.0	0.0

Table 9
Member stress limitations for the 25-bar spatial truss.

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

4.2.1. A tension/compression spring design problem

Belegunda [14] and Arora [15] described this problem for the first time. The goal of this problem is to minimize the weight of a tension/compression spring subjected to constraints on shear stress, surge frequency, and minimum deflection as shown in Fig. 10. Here, the design variables are the mean coil diameter D; the wire diameter d, and the number of active coils. Table 4 shows the cost function, constraints and the bounds of the design space.

Belegunda [14] has solved this problem using eight different mathematical optimization techniques (only the best results are shown). Also, it has been solved by Arora [15] using a numerical optimization technique called constraint at the constant cost. Finally, this problem is solved by RO using 40 agents, and the results of optimization are shown in Table 5. Considering these results, it can be concluded that the RO performs better than the above-mentioned methods. The average value and the standard deviation for 50 independents runs are 0.13547 and 0.001159, respectively.

4.2.2. A welded beam design problem

The other well-studied engineering design problem is the welded beam design problem that is often used as a benchmark problem [16]. The goal is to find the minimum fabricating cost of the welded beam subjected to constraints on shear stress (τ), bending stress (σ), bulking load (P_c), end deflection (δ) and side constraint. The design variables are $h(=x_1)$, $L(=x_2)$, $t(=x_3)$ and $b(=x_4)$,

Table 10
Performance comparison for the 25-bar spatial truss.

	Element group	Optimal cross-sectional areas (in^2)				
		Rajeev and Krishnamoorthy GA [18]	Schutte and Groenwold PSO [19]	Lee and Geem HS [20]	in^2	Present work cm^2
1	A1	0.10	0.010	0.047	0.015713	0.101374
2	A2–A5	1.80	2.121	2.022	2.021744	13.043483
3	A6–A9	2.30	2.893	2.950	2.931913	18.915530
4	A10–A11	0.20	0.010	0.010	0.010194	0.065768
5	A12–A13	0.10	0.010	0.014	0.01086	0.070064
6	A14–A17	0.80	0.671	0.688	0.656255	4.233895
7	A18–A21	1.80	1.611	1.657	1.679326	10.834340
8	A22–A25	3.0	2.717	2.663	2.716261	17.524229
	Best weight (lb)	546	545.21	544.38	544.6565	247.05 kg
	Average weight (lb)	N/A	546.84	N/A	546.6891	247.97 kg
	Std Dev (lb)	N/A	1.478	N/A	1.61248	
	No. of analyses	N/A	9596	15,000	13,880	

Fig. 11. Table 6 shows the cost function, constraints and the bounds of the design space.

This problem has been solved by Radgsdell and Philips [16] and Deb [17]. Radgsdell and Philips compared optimal results of different optimization methods which were mainly based on mathematical optimization algorithms. These methods are APPROX (Griffith and Stewart's Successive linear approximation), DAVID (Deviation-Fletcher-Powell with a penalty function), SIMPLEX (Simplex method with a penalty function), and RANDOM (Richardson's random method) algorithms. Finally, RO solved this problem by using 40 agents in 50 independents runs. The results are collected in Table 7 indicating that the RO has a better solution than the above mentioned methods. The mean of the results and the standard deviation of 50 independent runs are 1.9083 and 0.173744, respectively.

4.2.3. A 25-bar spatial truss

A 25-bar spatial truss structure is considered with the topology and nodal numbering shown in Fig. 12. Table 8 shows two load cases for which the design is performed. In this example, the material density is considered as 0.1 lb/in^3 (2767.990 kg/m^3) and the modulus of elasticity is taken as 10,000 ksi (68,950 MPa). Twenty five bars are classified into eight groups, as follows:

$$(1)A_1(2)A_2\_A_5, (3)A_6\_A_9, (4)A_{10}\_A_{11}, (5)A_{12}\_A_{13}, (6)A_{14}\_A_{17},$$
$$(7)A_{18}\_A_{21}, \text{and}(8)A_{22}\_A_{25}.$$

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 each group as shown in Table 9. The range of the cross-sectional areas varies from 0.01 to 3.4 in^2 (0.6452–21.94 cm^2).

In this problem, 50 agents are used. After analysis, the best weight is achieved by RO as 544.6565 lb (247.05 kg) which is less than those achieved by Rajeev and Krishnamoorthy [18] and Schutte and Groenwold [19], but the best weight of Lee and Geem's method [20] is better. Also, the average weight for the RO is 546.6891 lb (247.97 kg) that is less than Schutte and Groenwold's method whose average weight is 546.84 lb (248.04 kg). The standard deviation and number of analyses of the RO, are 1.61248 lb and 13,880, respectively, which are bigger than those of Schutte and Groenwold, but the number of analyses of the RO is smaller than that of the method of Lee and Geem. Table 10 shows the results in detail.

5. Discussion and conclusions

In this paper, a new physics-based algorithm, so-called RO, is presented for finding the near-global optimum in optimization problems. Like other multi-agent algorithms, RO has a number of agents that move in the search space. While the optimization process progresses, these agents converge to a zone in which near-global optimum is situated. In order to provide a mean for convergence, the Snell's refraction law of light is utilized.

The results of some benchmark and well-studied engineering problems show that the RO has a good efficiency and it can be utilized for structural optimization problems.

Acknowledgement

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

References

[1] Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B 1996;26:29–41.
[2] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, Nagoya, Japan; 1995.
[3] Fogel LJ, Owens AJ, Walsh MJ. Artificial Intelligence Through Simulated Evolution. Chichester: Wiley; 1996.
[4] K. De Jong, Analysis of the behavior of a class of genetic adaptive systems. Ph.D. thesis, University of Michigan, Ann Arbor, MI; 1975.
[5] J.R. Koza, Genetic programming: a paradigm for genetically breeding populations of computer programs to solve problems. Report No. STAN-CS-90-1314, Stanford University, Stanford, CA; 1990.
[6] Holland JH. Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan Press; 1975.
[7] Goldberg DE. Genetic Algorithms in Search Optimization and Machine Learning. Boston: Addison-Wesley; 1989.
[8] Glover F. Heuristic for integer programming using surrogate constraints. Decis Sci 1971;8:156–66.
[9] Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci 2009;179:2232–48.
[10] Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search. Acta Mech 2010;213:267–89.
[11] B. Laval Philippe, Mathematics for Computer Graphics – Ray Tracing III. Kennesaw State University, November 12, 2003.
[12] Kaveh A, Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput Struct 2009;87:267–83.
[13] Tsoulos IG. Modifications of real code genetic algorithm for global optimization. Appl Math Comput 2008;203:598–607.
[14] A.D. Belegundu, A study of mathematical programming methods for structural optimization. Ph.D. thesis, Department of Civil and Environmental Engineering, University of Iowa, Iowa, USA; 1982.
[15] Arora JS. Introduction to Optimum Design. New York: McGraw-Hill; 1989.
[16] Ragsdell KM, Phillips DT. Optimal design of a class of welded structures using geometric programming. ASME J Eng Ind Ser B 1976;98:1021–5.
[17] Deb K. Optimal design of a welded beam via genetic algorithms. Am Inst Aeronaut Astronaut J 1991;29:2013–5.
[18] Rajeev S, Krishnamoorthy CS. Discrete optimization of structures using genetic algorithms. J Struct Eng ASCE 1992;118:1233–50.
[19] Schutte JJ, Groenwold AA. Sizing design of truss structures using particle swarms. Struct Multidisc Optim 2003;25:261–9.
[20] Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm. Comput Struct 2004;82:781–98.