Applied Soft Computing 19 (2014) 177–187

Contents lists available at ScienceDirect

Applied Soft Computing

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

Exchange market algorithm

Naser Ghorbani, Ebrahim Babaei *

Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

ARTICLE INFO

Article history:
Received 27 October 2012
Received in revised form 2 February 2014
Accepted 3 February 2014
Available online 18 February 2014

Keywords:
Optimization
Evolutionary algorithm
Exchange market algorithm
Stock market

ABSTRACT

This paper proposes a new evolutionary algorithm for continuous non-linear optimization problems. This optimization algorithm is inspired by the procedure of trading the shares on stock market and it is called exchange market algorithm (EMA). Evaluation of how the stocks are traded on the stock market by elites has formed this evolutionary as an optimization algorithm. In the proposed method there are two different modes in EMA. In the first mode, there is no oscillation in the market whereas in the second mode, the market has oscillation. It is noticeable that at the end of each mode, the individuals' finesses are evaluated. For the first mode, the algorithm's duty is to recruit people toward successful individuals, while in the second case the algorithm seeks optimal points. In this algorithm, the generation and organization of random numbers are performed in the best way due to the existence of two absorbent operators and two searching operators leading to high capability in global optimum point extraction. To evaluate the performance of the proposed algorithm, this algorithm has been implemented on 12 different benchmark functions with 10, 20, 30 and 50 dimension variables. The results obtained by 30 dimension variables are compared with the results obtained by the eight new and efficient algorithms. The results indicate the ability of the proposed algorithm in finding the global optimum point of the functions for each run of the program.

1. Introduction

Optimization is a process of finding the best solution for a problem. In optimization of the problems, the global optimum is the maximum/minimum of a function [1]. In solving optimization problems, the mathematical methods are capable to optimize various problems in comparison with the meta-heuristic algorithms with greater accuracy in less time [2]. The meta-heuristic algorithms, unlike mathematical methods, can obtain the output of a function by selecting random numbers in input regardless of the complexity of the problems and constraints. This advantage has increased the usage of heuristic algorithms in optimization of several complex and applied functions in the real world which cannot be solved by mathematical methods [3–5]. Nowadays, due to the complexity of the engineering problems, the existence of different constraints in these problems and various objective functions in a problem, it is required to use the meta-heuristic algorithms for optimizing these practical problems [6,7]. In this respect, extensive researches have been conducted to improve the algorithms for optimization of engineering and practical problems [8,9].

For all heuristic algorithms, there is a searching operator and an absorbing operator to find out optimal values. The operators inspired by natural or law-based processes are used to produce and organize some random numbers. For instance, the genetic algorithm (GA) uses operators inspired by the natural genetic variation and natural selection [10–12]. The particle swarm optimization (PSO) algorithm uses the operators inspired by social behavior of bird flocking or fish schooling [13–15]. The ant colony optimization (ACO) is another evolutionary optimization algorithm. This stochastic optimization algorithm is inspired by the pheromone trail laying behavior of real ant colonies [16–18]. The imperialistic competitive algorithm (ICA) is one of the optimization algorithms which unlike the above-mentioned algorithms, is inspired by human behaviors in this context [19–21]. All of these algorithms have different searching and absorbent operators resulting in advantages or limitations in comparison with each other. Investigating the results of the functions optimization through the above mentioned approaches, it seems that the trapping in local optimum points and consequently the early convergence (i.e., exploration problem) or insufficient ability in finding adjacent points of the optimum point (i.e., exploitation problem) and convergence to non-similar solutions are some of the problems of these meta-heuristic algorithms.

This paper proposes a new type of evolutionary algorithms with two efficient and powerful searching operators and two absorbent

* Corresponding author. Tel.: +98 411 3300819; fax: +98 411 3300819.
E-mail addresses: naser.ghorbani@yahoo.com (N. Ghorbani), e-babaei@tabrizu.ac.ir, babaeiebrahim@yahoo.com (E. Babaei).

http://dx.doi.org/10.1016/j.asoc.2014.02.006

operators. As a result, the random numbers' generation and organization are performed in a best form in this algorithm leads to highly improvement of the other algorithms' limitations. This algorithm is suitable for continuous nonlinear optimization problems. The proposed algorithm is inspired by shares traded on the stock market and it is called the EMA. In the stock market, the price of a stock is increased by increasing the demand and decreases with declining the demand. Shareholders in the market try to buy and sell stocks in all sorts in order to achieve the most possible benefit. Due to political and economic measures taken by organizations and countries, there are always price oscillations. Buying and selling shares in an oscillated market have higher risk in comparison with no oscillated market which may be highly profitable or harmful [22].

In a stock market, the shares trading manner is completely sophisticated, different and unique according to mental conditions of several individuals [23]. In the proposed algorithm, it is assumed that the people who are active in the exchange market act similar to the elite stock dealers such as Warren Buffett [24]. In this algorithm, in each market mode the fitness of each individual is reviewed, ranked and sorted according to their properties values. In the EMA, all of the shareholders try to introduce themselves as the most successful individuals to market and then the individuals have less fitness tend to do greater risks. Similar to real stock market, in this algorithm, each individual carries unique trade and risks. Shareholders in the EMA are arranged according to their rank after each fitness test. After sorting the population, the individuals at the beginning, middle and end of shareholders are known as first, second and third groups. The individuals in the first group as successful people in the market remain unchanged in all stages of the market. The second and third groups trade with separate equations. In a non-oscillated market, the individuals in second and third groups select stocks which are same or close to the shares of the first group. In other words, the algorithm has the duty to recruit members toward the elite members. For an oscillated market, the individuals in second and third groups trade with separate relationship at high risk, in other words, the algorithm searches for unknown points. In the proposed algorithm, the search area or the value of shares traded are easily adjustable, thus this algorithm has the ability to optimize any function in the best possible way.

The proposed algorithm is successfully implemented on 12 different standard benchmark functions with dimension of 10, 20, 30 and 50 variables. The results with dimension of 30 variables are compared with real coded genetic algorithm (RCGA) [25], PSO [25], differential evolution (DE) [26], artificial bee colony (ABC) [27], biogeography-based optimization (BBO) [28], harmony search algorithm (HSA) [29], gravitational search algorithm (GSA) [25] and cuckoo search algorithm (CSA) [30]. The obtained results indicate the high ability of the proposed algorithm in finding the global optimum points in comparison with the other algorithms.

2. Exchange market formation

When some European traders faced with loss in their trades and tried to find a way to prevent or minimize their detriment, the stock formation thought was formed. Therefore, they shared some traders in their trades to divide the probable profit and loss. This successful experience convinced the traders to continue their economic activities through this approach. Over time, this experience was lawful and reformed to establishment of corporations [31].

The first experience of foundation of a corporation flashes back to 1553 AD in Russia, in which some traders provided the required stock and shared in corporation's profit and loss in proportion with their shares in order to decrease the probable loss of the trade. Therefore, by developing the trade in Europe it was necessary to have agencies to establish the relationship between the investors. These types of agencies were established and were called stock markets. The world's first stock market was founded in Amsterdam, Netherlands in the 17th century [31]. Nowadays, the stock markets are considered as one of the economic bases of any country or financial organization, in which people trade several types of shares and try to achieve the maximum possible benefit. In the stock market, there exist the specific numbers of several types of organizations and agencies shares and the shareholders buy and sell shares according to their own experiences and financial policies. In these markets, people usually buy different types of shares in order to gain benefit and decrease the loss risk [32].

In a stock market, sometimes the market faces with non-oscillations and sometimes it faces with oscillations due to the financial and economic policies of organizations and countries. Under balanced conditions, the market is easier to anticipate and people can increase their shares' profit to some extent without doing any unusual risks. Under oscillated conditions, selecting shares to buy and sell holds some risks and it is difficult to anticipate the market status and the activities of shareholders can be profitable or detrimental [22]. However, the stock market has always its own complexity and the behavior of the shareholders in encountering with different conditions is very complex and unique [23].

2.1. Stock market in the proposed algorithm

The proposed algorithm is inspired by the stock market, in which the shareholders buy and sell several types of shares in the virtual stock market under different market conditions. It is assumed that the shareholders compete to introduce themselves as the most successful dealers in the ranking list. In addition, it is assumed that the lower ranked people tend to deal with logical risks to gain more benefit and it is supposed that the intelligent shareholders perform similar to be the successful individuals of stock market such as Warren Buffett, the most successful shareholder of the recent 50 years. The main characteristics of a successful shareholder are as follows [33]:

(a) Capital preservation is always their top priority.
(b) They buy shares as more as they can.
(c) They avoid investing in sectors not match their criteria.
(d) They accept their mistakes and correct them immediately.
(e) They make their mistakes as a learning experience.
(f) They follow the experiences and the performance of the successful shareholders.

In the proposed algorithm, each person of the exchange market is an answer of the problem. Here, there exist a specific number of shares (the variable of the optimization problem), where any person attempts to intelligently buy some of the shares (initializing the problem variables). At the end of each period, they are intelligently proceed to gain the maximum possible profit in the market by calculating the value of their total shares.

The exchange market encounters different market conditions due to the political and financial decisions of organizations and countries. In the proposed algorithm, it is assumed that in the exchange market there are two general market states. In the first state, the market is in its normal condition with no considerable oscillation and the shareholders try to utilize the experiences of the successful shareholders to gain more benefit without performing non-market risks. In the second state, the exchange market encounters different oscillations and the shareholders try to intelligently risk by identifying the market conditions, in order to better use the created conditions in increasing their finance. It is obvious that any additional information about market conditions or the investigated problem can make these risks more targeted and beneficial.

3. Exchange market algorithm

3.1. The exchange market under balanced conditions

In this section, the market is assumed in normal state without any considerable oscillation and the shareholders try to gain the maximum possible benefits without performing non-market risks by using the experiences of the successful shareholders. Therefore, they compete with each other. Here, any person is ranked in three groups by numbering the shares from any type according to the fitness function. The groups are called the primary, mean, and the final persons of shareholders population.

3.1.1. The first group: shareholders with high rank

This group forms the highest ranked people of the list, who do not change their shares with any risk and trade to keep their rank in population. These people are 10–30% of total population. The members of this group are the elite members of the exchange market and the best answers are required without facing any variation.

3.1.2. The second group: shareholders with mean rank

This group forms the middle ranked people and utilize the differences of the different first group people's shares amounts. The people of this group are in charge of searching around the optimum point in the optimization problems. This group of shareholders forms 20–50% of exchange market dealers. To compare the individuals' shares in the first group, it is required that the comparison is done at least between two individuals. For instance, if one person in the first group shares x with value of a unit and an another person in the same group has the same share with the value b, the jth individual in the second group uses the difference of two shareholder in the first group and selects share x with a value between a and b. Selection of these two persons from first group is randomly done. Accordingly, to evaluate the differences among all shares of these two shareholders in the first and second groups' individuals, change their shares' value of any types based on (1) to reach more benefits:

$$pop_j^{group(2)} = r \times pop_{1,i}^{group(1)} + (1-r) \times pop_{2,i}^{group(1)}$$
$$i = 1, 2, 3, \ldots n_i \text{ and } j = 1, 2, 3, \ldots, n_j \quad (1)$$

where n_i is the nth person of the first group, n_j is the nth person of the second group and r is a random number in interval $[0, 1]$. $pop_{1,i}^{group(1)}$ and $pop_{2,i}^{group(1)}$ are the ith person of the first group and $pop_j^{group(2)}$ is the jth person of the second group.

3.1.3. The third group: shareholders with weak rank

The individuals in this group have less fitness in comparison with the second group. Then to obtain more benefits they select shares similar to the first group and with higher risk in comparison with the second group. Unlike second group individuals, they utilize the differences of share values of the first group as well as their share values' differences compared to the first group individuals and change their shares. To compare the first group's shares, it is required that the comparison is done between two shareholders. To increase this group's individuals risk in comparison with the second group, two random factors are used in place of factor 1 which its average value is nearly equal to one. Consequently, this group's individuals change their shares' value of any types based on (2) to reach more benefits:

$$S_k = 2 \times r_1 \times (pop_{i,1}^{group(1)} - pop_k^{group(3)}) + 2 \times r_1 \times (pop_{i,2}^{group(1)} - pop_k^{group(3)}) \quad (2)$$

$$pop_k^{group(3),new} = pop_k^{group(3)} + 0.8 \times S_k \quad (3)$$

where r_1 and r_2 are random numbers in interval $[0, 1]$ and n_k is the nth member of the third group. $pop_k^{group(3)}$ is the kth member and S_k is the share variations of the kth member of the third group.

The members of this group actually search the optimum points around the desired optimum point in wider range than that of the second group members. This group contains 20–50% of the total market population.

3.2. The exchange market in oscillation state

Sometimes, the stock market falls in oscillation state due to political and economical behaviors of financial organizations or countries. In this condition, after reassessment of shareholders and ranking the members in groups, the shareholders make their best to accomplish some severe but very intelligent risks in order to gain the maximum possible profit and to achieve high market rankings from fitness function viewpoint. Here, each member takes a different financial policy depends on the gained profit and accomplishes several risks to surpass the market best member. The members can be classified into three groups according to their performance.

3.2.1. The first group: shareholders with high ranks

This part of the population forms the elite members of the exchange market or the best answers of the optimization problem. This group forms 10–30% of the total market population.

3.2.2. The second group: shareholders with mean ranks

These members try to find the best cost by changing their shares amounts. The risk percentage of these shareholders is different and increases as their ranks decreases. In this group, the total share values of the members are constant and just a part of the shares value increase and the other part faces with decreasing in a way that the total share value of any member does not finally vary. Initially, the number of some shares of each member increases as follows:

$$\Delta n_{t1} = n_{t1} - \delta + (2 \times r \times \mu \times \eta_1) \quad (4)$$

$$\mu = \frac{t_{pop}}{n_{pop}}. \quad (5)$$

$$n_{t1} = \sum_{y=1}^{n} s_{ty}, \quad y = 1, 2, 3, \ldots, n \quad (6)$$

$$\eta_1 = n_{t1} \times g_1 \quad (7)$$

$$g_1^k = g_{1,\max} - \frac{g_{1,\max} - g_{1,\min}}{iter_{\max}} \times k \quad (8)$$

where Δn_{t1} is the amount of shares should be added randomly to some shares, n_{t1} is total shares of tth member before applying the share changes. S_{ty} is the shares of the tth member, δ is the information of exchange market. r is a random number in interval $[0, 1]$. η_1 is risk level related to each member of the second group, t_{pop} is the number of the tth member in exchange market. n_{pop} is the number of the last member in exchange market, μ is a constant coefficient for each member and g_1 is the common market risk amount which decreases as iteration number increases. $iter_{\max}$ is the last iteration number and k is the number of program iteration. $g_{1,\max}$ and $g_{1,\min}$ indicate the maximum and minimum values of risk in market, respectively.

The operation is in a way that a percentage of Δn_{t1} value is added randomly to one of the shares using rand command and this continues until Δn_{t1} is completely added to one or more shares. It is obvious that this action is performed just on shares whose value can be increased considering the constraints and limitations of exchange market. As mentioned, δ is the market information and its existence is very important in optimization problems and results in intelligence and fast convergence of the function to its desired value. Two different conditions exist to select δ based on market information.

3.2.2.1. Some information about market situation is in hand. This case is obtained in problems that the penalty factor is utilized. The δ value equals to the expected value or the problem data. In this condition, there is no need to use penalty factor and the search process falls just in the logical domain (applicable in solving problems such as economic load dispatch where the load amount is specified). The presence of $\delta = n_{t1}$ causes each member's total share value equal to the problem's constant value (total values of the problem variations is constant).

3.2.2.2. No information about market situation is in hand. In this condition, δ equals to total shares value of each member in no market oscillation condition. In other words, δ equals to Δn_{t2} (this condition is utilized in standard functions optimization).

μ is the risk increase coefficient and makes lower ranked shareholders from fitness function viewpoint perform more risks in compare with the more successful members to increase their finance. g_1 is the variable risk coefficient and determines shareholders should change what percentage of their shares. The amount of this parameter varies in different problems and its variation range generally falls in $[0, 1]$. In the primary iterations, g_1 has its maximum value and decreases as the iteration number increases. This coefficient can be designed in a way that it does not depend on the iteration and decreases with a constant or exponential slop.

In the second part of this section, it is necessary that each member's total share amount is similar to the previous state. Therefore, each member should sell the same amount of his shares he has previously bought to equalize each member's share amount with that of the primary state. It is required that each member finally decrease Δn_{t2} of his share amount. Here Δn_{t2} is as follows:

$$\Delta n_{t2} = n_{t2} - \delta \tag{9}$$

where Δn_{t2} is the amount of shares should be decreased randomly from some shares and n_{t2} is the total share amount of tth member after applying the share variations.

As it is obvious, if the market faces with oscillation and there is no information about the market situation, some members sell randomly some parts of their shares and some buy randomly new shares equal to the sold amount without any change in total market share amount. In other words, the selection of a stock need to be sold is performed in random. Selection of the number of shares that should be reduced is also done randomly. In this case, each member buys some shares from any type randomly and sells the equal amount of the bought amount from any share type in continuous. This enables shareholders to change some of their shares without changing their total share value.

3.2.3. The third group: shareholders with low ranks

These members try to find better cost by changing their total share amounts. The risk percentage of these members is different and increases as their ranking decrease. In this group, the total share number of members varies and is comprised of a single part, unlike the previous section. Here, the shareholders try to find out a new and unknown combination of shares and they vary the number of some of their shares as follows:

$$\Delta n_{t3} = (4 \times r_s \times \mu \times \eta_2) \tag{10}$$

$$r_s = (0.5 - rand) \tag{11}$$

$$\eta_2 = n_{t1} \times g_2 \tag{12}$$

where Δn_{t3} is the share amount should be randomly added to the shares of each member, r_s is a random number in $[-0.5, 0.5]$ and η_2 is the risk coefficient related to each member of the third group. g_2 is the variable risk of the market in the third group and μ is the risk increase coefficient which forces lower ranked shareholders from fitness function viewpoint to perform more risk in compare with successful competitors to increase their finance. g_2 is the variable risk coefficient of the market and determines what percentage of shares should be changed by shareholders. The value of g_2 is within $[0, 1]$. In the primary iterations, g_2 has its maximum value and decreases as iteration number increases.

In this section, the members trade a part of their shares randomly. As it is obvious, the trade of members in this section does not depend on the cost or profit of each member and depends just on the number of shares. Therefore, the members trade their shares until the last iteration and the algorithm finds out the better optimum points even with small variations. The ideal state will be found out by selecting proper market initial values and proper algorithm iteration number.

4. Exchange market algorithm implementation pattern

4.1. Step 1: selecting initial values and attributing stock to initial shareholders

In this step, the shares number, shares values, initial shareholders number and the required iteration number are determined according to the optimization problem type and the shareholders take a number of any type of shares randomly.

4.2. Step 2: calculating the shareholders costs and ranking

In this step, the members are evaluated and sit in three separate groups according to the validity of their total shares in order to specify the different shareholder groups. It is noticeable that the group means the high, middle and low ranked members of the shareholders and is not separated from the total population. It is named as group just to apply some specific variations on high, middle and low ranked members of population.

4.3. Step 3: applying changes on the shares of the second group in balance market condition

In this step, the members of the first group or the high ranked and elite members remain with no variation where the mean members of shareholders or members of second group change some of the shares according to (1).

4.4. Step 4: applying changes on shares of third group members in balance market condition

The members of this group are the low ranked members from fitness function viewpoint and change their shares values from any type according to (3).

4.5. Step 5: calculating the shareholders cost and ranking

Until this step, it was aimed to search around the optimum point and the market was in non-oscillated condition. In this step, the main population of shareholders is evaluated from fitness function viewpoint and is arranged based on the member's value. Then, the members sit in separate groups according to the variations accomplished in mean and low ranked members' shares.

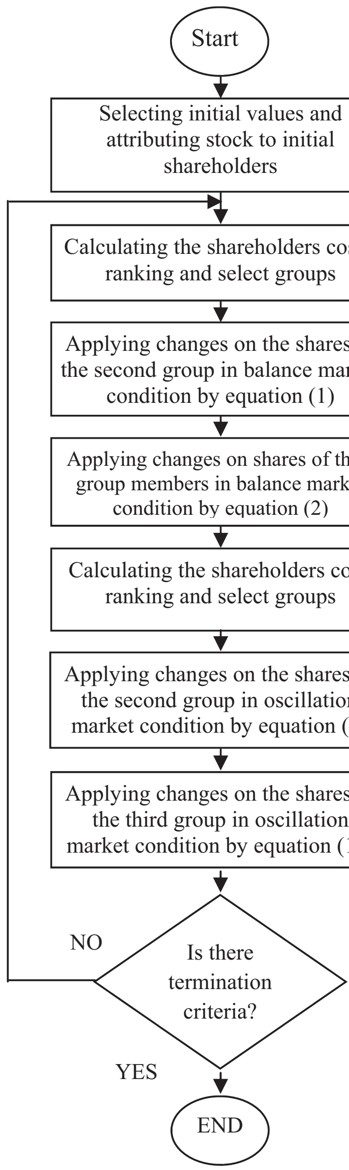

Fig. 1. General principle of EMA.

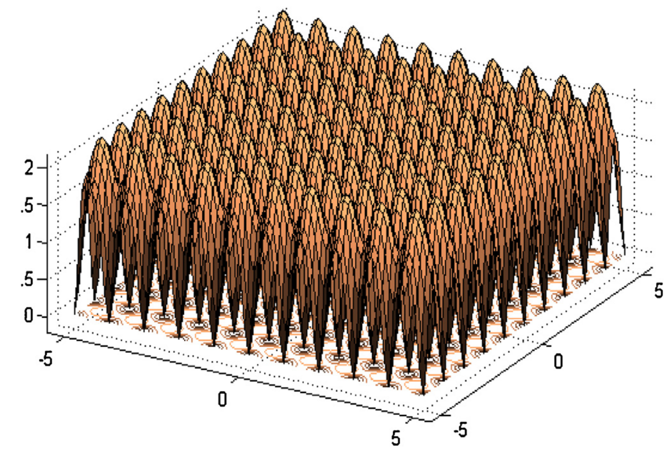

Fig. 2. The two-dimensional version of f_1.

4.6. *Step 6: the trade in shares of second group members using (4) under market oscillation condition*

In this step, the members of first group or the elite members remain without variations. The mean members or the members of the second group trade their shares according to (4) and change some of their shares.

4.7. *Step 7: the trade in shares of third group members using (10) under market oscillation condition*

In this step, unlike the previous step, shareholders trade some shares according to (10) with no attention to their total share amount.

4.8. *Step 8: go to step 2 if the program's end conditions are not satisfied*

In this step, the market oscillation condition is finished and the program starts to operate in order to evaluate the shareholders from step 2 if end up conditions are not satisfied. If end up conditions are satisfied, the program operation is ended up.

The exchange market proposed algorithm flowchart is illustrated in Fig. 1.

5. Numerical results

The proposed algorithm is implemented on 12 standard benchmark functions with 10, 20, 30 and 50 dimensions in order to evaluate the exchange market algorithm's performance in facing with several types of problems. The required data of benchmark functions [25] is presented in Table 1. For each function, 50 independent experiments were conducted to compare the quality of solution and convergence properties. In all experiments, the initial population number is 50. In addition, the number of individuals in the first, second and third groups in balanced market conditions are 25, 25 and 50% of the initial population and the pattern for the oscillated market conditions are equal to 20, 60 and 20% of initial population, respectively. The individuals' percentage in three groups has approximately constant values with the mentioned optimum values. The main adjustable parameters of the proposed algorithm are risk factors of the second and third groups in oscillated market which its optimum value for each test function are given in Table 2. The adjustable parameters for RCGA, PSO and GSA used are available in [25].

In the benchmark functions used to investigate the capabilities of the proposed algorithm, it is tried to assess different states of the functions characteristics. As it is obvious, f_1, f_2, f_3 and f_4 functions have multimodal and non-separable characteristics, where f_7, f_8, f_9 and f_{10} are unimodal and non-separable. f_5, f_{11} and f_{12} are unimodal and separable and f_6 is multimodal and separable. In all functions, the optimum function value is zero. In order to investigate the experimented benchmark functions properties, the Ackley function with two variables within $[-5, 5]$ region is shown in Fig. 2. This function is proposed by Ackley (1987) and is generalized by Back (1993). The exponential function of this relation covers the surface of the function with considerably large number of local minimum points. The algorithm, search region of which operates in descending slop sticks in these local points and just the strategies capable of analyzing a wide range can pass through the local optimum points. In order to obtain proper responses for this function, the search strategy should have efficient exploratory and exploitative components. As mentioned before, this function has multimodal and non-separable characteristics. The multimodal functions are often a difficult test for optimization due to the existence of large amounts of local optimum points. In the accomplished experimentations,

N. Ghorbani, E. Babaei / Applied Soft Computing 19 (2014) 177–187

Table 1
Benchmark functions used in experiments.

No.	Name function	C	Range	Formulation
1	Ackley	MN	[−32, 32]	$f_1(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$
2	Griewank	MN	[−600, 600]	$f_2(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
3	Penalized function 1	MN	[−50, 50]	$f_3(x) = \frac{\pi}{n}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})] + (y_n-1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4) \quad y_i = 1 + (x_i+1)/4 \quad u(x_i, a, k, m) = \left\{\begin{matrix} k(x_i-a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i-a)^m x_i < -a \end{matrix}\right\}$
4	Penalized function 2	MN	[−50, 50]	$f_4(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})] + (x_n-1)^2[1+\sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$
5	Quartic	US	[−1.28, 1.28]	$f_5(x) = \sum_{i=1}^{n} i x_i^4 + random[0, 1)$
6	Rastrigin	MS	[−5.12, 5.12]	$f_6(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$
7	Raosenbrock	UN	[−30, 30]	$f_7(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
8	Schwefel's 1.2	UN	[−100, 100]	$f_8(x) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$
9	Schwefel's 2.21	UN	[−100, 100]	$f_9(x) = \max_i\left\{\lvert x_i\rvert, 1 \le i \le n\right\}$
10	Schwefel's 2.22	UN	[−10, 10]	$f_{10}(x) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$
11	Sphere	US	[−100, 100]	$f_{11}(x) = \sum_{i=1}^{n} x_i^2$
12	Step function	US	[−200, 200]	$f_{12}(x) = \sum_{i=1}^{n}\left(\lfloor x_i + 0.5\rfloor\right)^2$

C, characteristic; U, unimodal; M, multimodal; S, separable; and N, non-separable.

Table 2
The tunable parameters of EMA to accomplish numerical experiments.

g_2	g_1	Functions
[0.1 0.05]	[0.1 0.05]	F1
[0.1 0.05]	[0.1 0.05]	F2
[0.04 0.00]	[E−12 0]	F3
[0.04 0.00]	[E−12 0]	F4
[0.2 0.1]	[0.2 0.1]	F5
[0.1 0.07]	[E−5 E−6]	F6
[0.02 0.005]	[0.01 0.005]	F7
[0.1 0.05]	[0.01 0.005]	F8
[0.05 0.01]	[0.1 0.05]	F9
[0.05 0.02]	[0.1 0.05]	F10
[0.05 0.02]	[0.1 0.05]	F11
[0.05 0.02]	[0.1 0.05]	F12

Table 3
The mean errors of EMA in different dimensions.

Optimal	50	30	20	10	Dimensions
0	0	0	0	0	Ackley
0	0	0	0	0	Griewank
0	3E−16	0	0	0	Penalty #1
0	1E−18	0	0	0	Penalty #2
0	0	0	0	0	Quartic
0	0	0	0	0	Rastrigin
0	6E−3	2E−7	6E−10	1E−25	Rosenbrock
0	0	0	0	0	Schwefel 1.2
0	0	0	0	0	Schwefel 2.21
0	0	0	0	0	Schwefel 2.22
0	0	0	0	0	Sphere
0	0	0	0	0	Step

this function is evaluated by EMA with the expansion and integration of 10, 20, 30 and 50 variables in [−32, 32] domain. The detailed information about the standard benchmark function's characteristics and properties applied in Table 1 are presented in [34,35].

The results obtained from solving the above problems with 10, 20, 30 and 50 variables are shown in Table 3. The numbers less than 10^{-32} are neglected in the calculations. In all performed experiments, the global optimum point of all test functions is equal to zero, thus the obtained results show this algorithm error. The results in Table 3 show the average error value of each function after 50 runs program.

As it is obvious from Table 3, the EMA is able to find out exactly the global optimum point in each program implementation in nine of the functions with dimension of 10, 20, 30 and 50 variables. In three remained functions, the obtained answer is close to the optimum point. Generally, each function is experimented in four dimensions and the EMA is able to find out the global optimum point in 42 of experimentations and in 6 experiments the results are very close to the optimum point.

Table 5
Comparison of mean run times of EMA, GSA, PSO and RCGA.

Function	GSA	PSO	RCGA	EMA
F1	91.12	41.60	28.91	32.13
F2	92.91	40.78	28.09	32.24
F3	105.54	56.71	39.58	39.05
F4	104.35	43.06	36.48	38.89
F5	91.45	38.48	27.99	31.1
F6	90.98	37.89	28.45	30.91
F7	98.30	42.91	40.11	38.2
F8	95.77	38.86	27.86	31.85
F9	90.51	42.18	28.35	31.99
F10	90.08	40.74	27.41	31.25
F11	90.09	37.59	28.68	30.89
F12	90.21	36.96	26.51	27.58
Average time	94.27	41.48	30.70	33.06
Average T/I	0.0094	0.0041	0.0030	0.0033

In Table 4, the average error obtained by EMA in 30-variable dimensions are compared with the average error of DE, PSO, RCGA and some new algorithms such as CSA, GSA, BBO, HS and ABC. Overall, the average error of each algorithm is presented at the end of Table 4. Considering several conducted comparisons, some of the algorithms such as ABC algorithm have large error in one or more functions and better results in the other functions in comparison with the other algorithms in order to have more accurately assessment and comparison. The errors of the algorithms in any of functions are compared to each other and they are ranked from 1 to 9 and the average ranks of each algorithm are presented at the end of the Table 4. In Table 5, the average run time of the methods for each benchmark functions provided for 10,000 iterations. Average execution time of each algorithm and the average time to iteration (T/I) for each algorithm are also presented in Table 5. Fig. 3 compares the convergence characteristic of EMA, RCGA, PSO and GSA.

As shown in Table 4, the EMA could extract global optimum point in 11 test functions without error in each program run and in optimization of Rosenbrock function; it could extract global optimum point with the average error of 2×10^{-7} in average run of 50. As seen in Table 4, each of algorithms has the high ability to solve some functions while has not suitable performance in optimization of the other functions. For example, GSA in optimization of functions f_4 and f_5 with ranking of 2 and 7 has suitable and weak performances, respectively, and the average rank of GSA is 4.08. The average error of this algorithm is also 1641.3 which its 1600 units, is due to ineffective performance in optimization of function f_5. For another example, ABC algorithm had good performance in optimization of function f_{12} with rank 1 and functions f_1, f_2 and f_3 with rank 2, while had the worst performance in optimization of functions f_8 and f_9 in comparison with the other algorithms. Table 4 shows the average ranking for algorithms in optimization of

Table 4
Comparison of mean errors obtained by different algorithms.

Functions (D = 30)	HS	ABC	BBO	CSA	DE	GSA	PSO	RCGA	EMA	Best performance by
Ackley	2.94E−03	8.14E−06	3.48E−01	3.47E−02	3.1E−3	1.1E−5	2E−02	2.15	0	EMA
Griewank	5E−01	2.48E−04	4.82E−01	2.4E−03	1E−3	2.9E−01	5.5E−2	1.16	0	EMA
Penalty #1	1.32E−1	3.34E−13	5.29E−03	3.62E−4	1.2E−01	4.2E−13	2.6E+2	5.3E−2	0	EMA
Penalty #2	2.21E−04	6.17E−13	1.42E−01	1.7E−13	1.7E−25	3.2E−32	7.1E+2	8.1E−2	0	EMA
Quartic	3.75E−02	0.2089	1.9E−02	1.29E−02	1.28E−2	5.33E−01	1.04	5.6E−01	0	EMA
Rastrigin	4.27E−02	4.66E−02	8.5E−02	16.1867	30.12	15.32	72.8	5.92	0	EMA
Rosenbrock	7.64E+01	1.1836	9.14E+01	3.98E−01	2.4E−01	25.16	1.7E+3	1.1E+3	2E−7	EMA
Schwefel 1.2	3.66E+03	11,173	4.16E+02	3.14E−10	1.1E−01	1.6E+03	2.9E+3	5.6E+3	0	EMA
Schwefel 2.21	3.77	40.737	7.76E−01	2.8952	1.3E−02	8.5E−06	23.6	11.78	0	EMA
Schwefel 2.22	9.75E−03	5.86E−07	2.42E−01	9.8E−12	9E−01	6.09E−05	2.0	1.07	0	EMA
Sphere	5.14E−04	4.6E−12	8.86E−01	1.71E−16	11.78	2.1E−10	5E−02	23.45	0	EMA
Step	2E−02	0	2.8E−01	0	1.27E−03	2.1E−10	2E−02	24.52	0	–
Ave. Rank	5.57	4.16	5.83	3.66	4.49	4.08	7.41	7.58	1	EMA
Sum error	3740.914	11,215.17	510.6653	19.5309	43.3	1641.30	5669.58	6770.74	2E−7	EMA

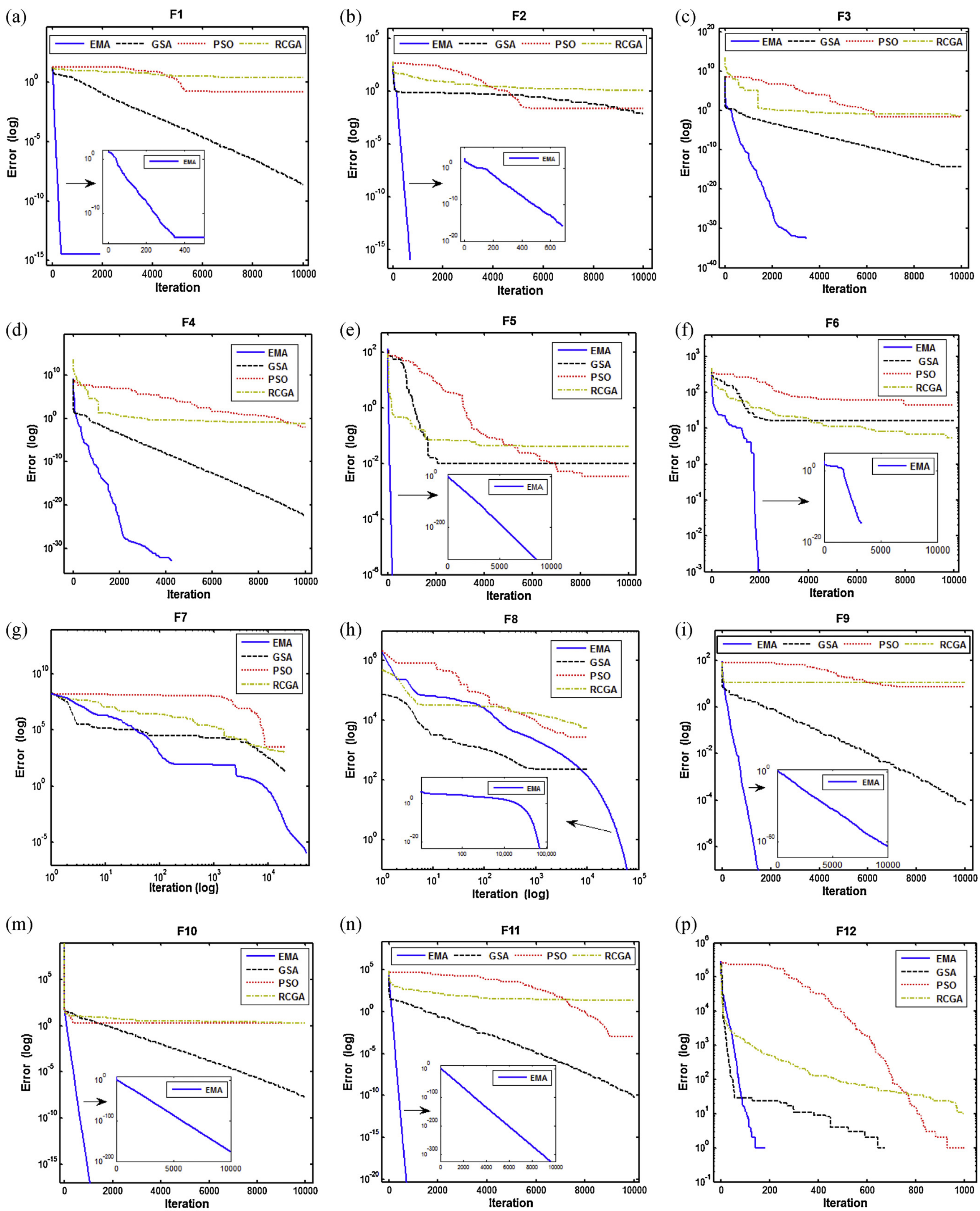

Fig. 3. Comparison of convergence characteristic of EMA, RCGA, PSO and GSA.

different functions. The results indicate that after EMA, CSA and GSA with the average rank of 3.66 and 4.08, respectively, had high ability in optimization of various problems in comparison with the other algorithms. Interestingly, as can be seen in Table 4, EMA has high capability in finding the optimal point for all functions. This is due to have two searching operators and the ability of EMA in selecting the search area. As can be seen in Table 1, EMA has different risk percentage for some functions and searches particular area. This advantage does not exist in some algorithms, such as PSO. For example, in PSO algorithm due to its structure the inertia coefficient has relatively constant value and thus the algorithm always searches for particular areas [36]. Total error value in EMA for 50 times runs of all functions is 2×10^{-7}.

The average run time for RCGA, PSO, GSA and EMA are given in Table 5. In calculation of the program run time, the RCGA, PSO and GSA are optimized in terms of time and their programming trends are performed in the easiest and most suitable form. As it can be seen, RCGA with run time of 30.7 s and 10,000 iterations has the lowest program executive time in comparison with the other algorithms. The average executive time for EMA is more than RCGA by 2.3 s, while it is less than PSO by 20% and three times less than GSA.

As it was mentioned earlier, Ackley function has infinite optimum points due to its exponential function. Fig. 3(a) depicts the convergence characteristics of EMA and other algorithms in optimization of function f_1. As it is obvious, EMA could find optimum point in first 200 iterations with the error of 3×10^{-10} and further in iteration 1800 it could extract global optimum point without error.

Fig. 3(b) shows the EMA convergence in optimization of function f_2. To have clear observation of EMA convergence, zoomed figure is provided. The convergence characteristics of EMA for optimization of functions f_3 and f_4 are shown in Fig. 3(c) and (d), respectively. For both of functions, EMA could extract global optimum point in average 4000 iterations. As it was stated, trades of stocks in EMA are a fraction of sum absolute value of problem variables. For functions f_1 and f_2, when all variables are zero the sum of all variables will be zero, as a result the global optimum point is obtained, whereas for functions f_3 and f_4 the sum variables are not zero. By comparing the results of these four functions, it can be concluded that if all of the variables have no value of zero but the sum value is zero, this algorithm will be not trapped in local optimum point because trades are performed as a fraction of shares absolute value. Moreover, this algorithm does not want to make the shares of shareholders equal to zero.

In oscillation market, the individuals in the second group make trade without changes in their total shares and the individuals in the third group change their total shares by buying/selling shares. In optimization of functions f_3 and f_4, at first, the individuals in the third group have the most impact on finding optimum points because they change share value from ±32 close to 1 by buying/selling shares. Afterward the second group individuals have the most effect on finding global optimum points and by trading shares all of the variables go toward 1. The existence of two search operators (the second and third groups' individuals) which are complementary to each other makes EMA to be able in finding global optimum point.

The convergence characteristic of function f_5 is shown in Fig. 3(e). Due to the weak results of the other algorithms in comparison with EMA, it is not possible to depict EMA curve totally and convergence figure is shown in $[5 \times 10^2, 1 \times 10^{-6}]$. As shown in Fig. 3(e), EMA could find global optimum point in iteration 100 with error of 1×10^{-4}, in iteration 1000 with error of 1×10^{-35}, in iteration 2000 with error of 1×10^{-72}, in iteration 5000 with error of 1×10^{-189}, in iteration 8500 with error of 1×10^{-345} and finally, in iteration more than 8500 with error of zero. It should be noted that the convergence characteristics of the EMA is similar to a straight line and indicates the algorithm could find better optimum point after each iteration verifying the efficacy of search parameters. Another advantage of this algorithm is escaping from global optimum point with small error. As it is obvious, even this algorithm could find global optimum point in iteration 8500 with error of 1×10^{-345}.

The convergence characteristics of function f_7 are shown in Fig. 3(g). Due to the need of EMA to large number of iterations in comparison with the other algorithms, the number of iterations is depicted in logarithm range. This function is uniquely complex and because of variables' changes in inside of many groups, the global optimum point extraction needs more iteration by EMA. The convergence pattern for this algorithm is depicted for twice the other algorithms' iterations. For the same iteration number by other algorithms, the EMA could find global optimum point with the error of 1×10^{-2}, and by twice the other algorithms' iteration with 2×10^{-7}, respectively. The reason why the other algorithms are not depicted for large number of iteration is that they were trapped in local minima and they cannot acquire better results. The global optimum point of this function is obtained for the value 1 for the variables and EMA can find the value 0.99999 for all variables. As shown in Fig. 3(g), in iteration 4×10^4, the EMA is continuing to decrease the error of this function and not trapped in local minima. The EMA could find the global optimum point with the error of 1×10^{-10} in iteration 2×10^5, and with the error of 1×10^{-14} in iteration 6×10^5. However, due to the need for higher iteration number and small changes in error, the results obtained in iteration 4×10^4 are considered as the result of optimization.

Function f_8 is another complex benchmark function, in this function each variable has a membership in 2–30 groups (number of variables) and optimal change in each variable requires simultaneous change in the other variables. As shown in Table 4, the efficient algorithms such as ABC, HS, GSA and BBO could find the global optimum point for this function with the errors of 1.1×10^4, 3.6×10^3, 1.6×10^3 and 4.1×10^2, respectively. As shown in Fig. 3(h), although EMA needs higher number of iteration in comparison with the other algorithms, but it is able to find out the extract global optimum point for this function. The reason why the other algorithms are not depicted for the same number of iteration is that they were trapped in local minima and they cannot acquire better results in large number of iteration. The convergence characteristics of functions f_9–f_{12} are depicted in Fig. 3(i) and (p). For all of these functions, the EMA is able to find global optimum point in each program run.

As mentioned, one of the advantages of the proposed algorithm is the ability to choose the search area and to select the value of each share's changes. This advantage causes this algorithm to find global optimum points for various functions, and then appropriate selection of search area in this algorithm is indispensable. In addition, this algorithm can find global optimum points by a risk value less than 0.2, so the search area should be adjusted properly to find global optimum point with the lowest error in any iteration and with the lowest number of iterations. The following items have great impact on fast and appropriate selection of individuals' risk or search area:

(1) The values of start risks of the second and third groups in oscillation market are usually less than 0.2 and mostly between 0.05 to 0.1.
(2) The value of risk for the third group's individuals is usually $0.5 \times g_1 \leq g_2 \leq 2 \times g_1$.
(3) In functions such as f_3, f_4 and f_7 because the sum of variables' values does not sharply decrease near the global optimum point, so the value of risk decreases by the increase in iteration number.
(4) In functions such as f_1, f_2 and f_{11} because the sum of variables' values decreases near the global optimum point, so there is

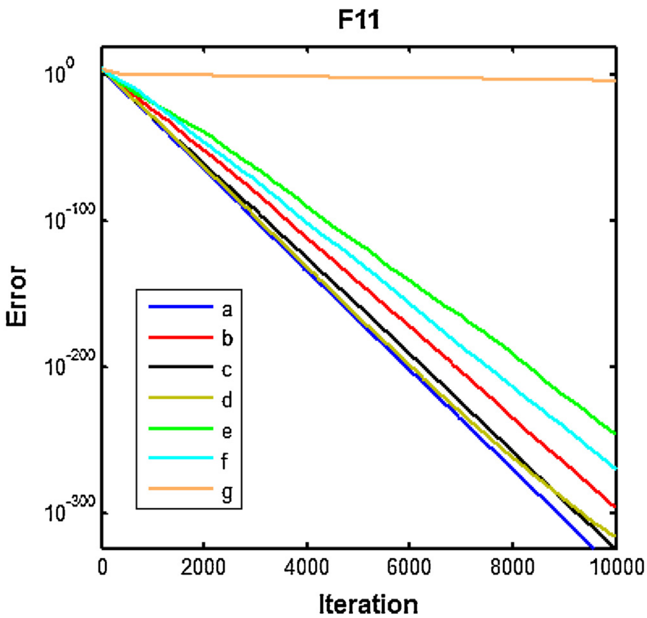

Fig. 4. Comparison of convergence characteristic of EMA with different risk values.

no need to highly decrease the risk value by the increase in iteration number.

(5) In functions such as f_3, $f_4$and f_8 because the variables are depending on each other and changing in group, so it is better to have small changes.

To observe convergence trend and to evaluate the impact of risk on the proposed algorithm's performance and to choose the value of risk the following tests have been done on function f_{11}.

(a) According to (1), the start risk of the second group is 0.1 and based on (4) the risk in the last iteration is half of the initial value. Based on (2), the third group's individuals risk value is half of the second group ($g_1 = [0.1, 0.05]$ and $g_2 = [0.05, 0.02]$).
(b) The second group's individuals risk is similar to state *a*, and third group's individuals risk is twice as many as the second group ($g_1 = [0.1, 0.05]$ and $g_2 = [0.2, 0.1]$).
(c) The second group's individuals risk is similar to state *a*, and third group's individuals risk is similar to the 2nd group ($g_1 = [0.1, 0.05]$ and $g_2 = [0.1, 0.05]$).
(d) The initial risk of the second and third groups' individuals are similar to state *a* and the risk value tend to zero by increasing the iteration ($g_1 = [0.1, 0]$ and $g_2 = [0.05, 0]$).
(e) The value of risk for second group is more than 0.2, and the risk value for third group's individuals is same as to state *a* ($g_1 = [0.3, 0.2]$ and $g_2 = [0.05, 0.02]$).
(f) The value of risk for 3rd group is more than 0.2, and the risk value for 3rd group's individuals is same as to state *a*($g_1 = [0.1, 0.05]$ and $g_2 = [0.3, 0.2]$).
(g) The value of risk for individuals is more than 0.3, ($g_1 = [0.5, 0.3]$ and $g_2 = [0.5, 0.3]$).

Convergence characteristics of the proposed algorithm for states *a* to *g* are shown in Fig. 4.

As shown in Fig. 4, the fast convergence and global optimum point extraction without error requires appropriate risk selection for various groups. For lower risks than 0.2, the proposed algorithm has good performance. For state *a* which owes the best risk percentage, the EMA could find global optimum point in iteration 9300 without error. For state *b*, the global optimum point is found in iteration 9300 without error. For the other states, finding global optimum point has small errors. As it is obvious, by increasing the risk value more than 0.2, the algorithm's error increases and the maximum error for the algorithm is for the state *g*. Noticeable point in Fig. 4 is the convergence trend of state *d*. In this state, in the final iterations, the individuals' risk value of both groups are zero, however, as can be seen the algorithm could still find optimum points without searching operator. The main reason for this search is the 3rd group's individuals in balanced market. As stated earlier, this group's individuals not only absorb individuals toward elite people but also search the neighborhood of optimum point.

6. Conclusions

This paper presents a new algorithm for optimizing continuous problems. The proposed algorithm is inspired by human intelligence and the experience of elite individuals in stock market. Exchange market algorithm has two absorbent operators which, absorbs the individuals toward elite members (members of 2nd and 3rd groups in balanced market condition) and two searching operators (members in 2nd and 3rd groups in oscillated market condition) which are complement to each other. These operators cause random numbers generation and organization to be performed in best state. As shown in Tables 3 and 4, the proposed algorithm obviates the other algorithms' limitations such as trapping in local minima, premature convergence, inability to find neighborhood points of optimum point and convergence to non-equal solutions in any iteration.

The proposed algorithm's performance is such that the required changes are selected based on sum of shares of shareholders. Thus, the search area has an intelligent relation with the obtained results. In EMA, the searching procedure is independent of the cost but depends only on shares value which is taken by each individual. Then, individuals make trades until the last iteration and the algorithm find better optimum points by the time even when the changes are negligible; this fact is shown in Fig. 3.

As shown in Table 2, in EMA the individuals' risks are easily controlled. Therefore, the proposed algorithm could select the search area. This advantage causes the proposed algorithm to optimize various types of functions. The mentioned advantage is not seen in comparison with algorithms shown in Table 4 because they have good performance on some functions, while bad performance on the others. The execution time for the proposed algorithm is lower because of doing no calculation for 1st group's individuals (consisting of 20% of population) and also change in one or some shares of each person. The execution time for the proposed algorithm is equal to GA and less than the other compared algorithms in Table 5.

The advantages for EMA are less execution time, effective random numbers generation and organization because of having two absorbent operators and two efficient searching operators, the ability in selecting search area and in turn the ability for optimization of various problems, convergence to the absolutely identical solutions in each program iteration, not trapping in local minima and therefore high ability in extraction of global optimum points and needing no penalty factor in some problems. The results proved the robustness and effectiveness of the proposed algorithm and show that it could be used as a reliable tool for solving real optimization problems.

References

[1] R.L. Haupt, S.E. Haupt, Practical Genetic Algorithms, second ed., John Wiley & Sons, New Jersey, 2004.
[2] J.B. Park, Y.W. Jeong, J.R. Shin, K.Y. Lee, An improved particle swarm optimization for nonconvex economic dispatch problems, IEEE Transactions on Power System 25 (1) (2010) 156–166.
[3] D.Y. Lera, D. Sergeyev, Lipschitz and Holder global optimization using space-filling curves, Applied Numerical Mathematics 6 (2010) 115–129.

[4] K.Y. Lee, M.A. El-Sharkawi (Eds.), Modern Heuristic Optimization Techniques with Applications to Power Systems, IEEE Power Engineering Society (02TP160), NewYork, 2002.
[5] S. Akpınar, G.M. Bayhan, A. Baykasoglu, Hybridizing ant colony optimization via genetic algorithm for mixed-model assembly line balancing problem with sequence dependent setup times between tasks, Applied Soft Computing 13 (2013) 574–589.
[6] D. Aydın, S. Ozyon, Solution to non-convex economic dispatch problem with valve point effects by incremental artificial bee colony with local search, Applied Soft Computing 13 (2013) 2456–2466.
[7] A. Azadeh, M. Hosseinabadi Farahani, H. Eivazy, S. Nazari-Shirkouhi, G. Asadipour, A hybrid meta-heuristic algorithm for optimization of crew scheduling, Applied Soft Computing 13 (2013) 158–164.
[8] Y.F. Liu, S.Y. Liu, A hybrid discrete artificial bee colony algorithm for permutation flowshop scheduling problem, Applied Soft Computing 13 (2013) 1459–1463.
[9] Y. Marinakis, G.R. Iordanidou, M. Marinaki, Particle swarm optimization for the vehicle routing problem with stochastic demands, Applied Soft Computing 13 (2013) 1693–1704.
[10] D.E. Goldberg, Genetic Algorithms in Search Optimization and Machine Learning, Addison Wesley, Boston, MA, USA, 1989.
[11] S. Siva Sathya, M.V. Radhika, Convergence of nomadic genetic algorithm on benchmark mathematical functions, Applied Soft Computing 13 (2013) 2759–2766.
[12] N.F. Wang, X.M. Zhang, Y.W. Yang, A hybrid genetic algorithm for constrained multi-objective optimization under uncertainty and target matching problems, Applied Soft Computing 13 (2013) 3636–3645.
[13] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948.
[14] B. Bolat, O. Altun, P. Cortes, A particle swarm optimization algorithm for optimal car-call allocation in elevator group control systems, Applied Soft Computing 13 (2013) 2633–2642.
[15] M. Joorabian, E. Afzalan, Optimal power flow under both normal and contingent operation conditions using the hybrid fuzzy particle swarm optimization and Nelder–Mead algorithm (HFPSO-NM), Applied Soft Computing 14 (2014) 623–633.
[16] M. Mavrovouniotis, Ant colony optimization with immigrants schemes for the dynamic traveling salesman problem with traffic factors, Applied Soft Computing 13 (2013) 4023–4037.
[17] M. Dorigo, T. Stutzle, Ant Colony Optimization, MIT Press, Cambridge, MA, 2004.
[18] S. Khan, A.R. Baig, W. Shahzad, A novel ant colony optimization based single path hierarchical classification algorithm for predicting gene ontology, Applied Soft Computing 16 (2014) 34–39.
[19] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, IEEE Congress on Evolutionary Computation (CEC2007), Singapore, 2007, pp. 4661–4667.
[20] F. Ramezani1, S. Lotfi, M.A. Soltani-Sarvestani, A Hybrid Evolutionary Imperialist Competitive Algorithm (HEICA), vol. 7196, Springer-Verlag, Berlin/Heidelberg, 2012, pp. 359–368.
[21] M.A. Ahmadi, M. Ebadi, A. Shokrollahi, S.M.J. Majidi, Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir, Applied Soft Computing 13 (2013) 1085–1098.
[22] M.H. Birnbaum, New paradoxes of risky decision making, Psychological Review 115 (2) (2008) 463–501.
[23] A. Shleifer, Inefficient Markets: An Introduction to Behavioral Finance, Oxford University Press, New York, 1999.
[24] Heller, Warren Buffet, The Man Who Made Billion With a Unique Investment Strategy, Dorling Kindersley, London, 2000.
[25] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, Sciences 179 (13) (2009) 2232–2248.
[26] E. Cuevas, M. González, D. Zaldivar, M. Pérez-Cisneros, G. García, An algorithm for global optimization inspired by collective animal behavior, Discrete Dynamics in Nature and Society 2012 (2012) 1–24, http://dx.doi.org/10.1155/2012/638275.
[27] X. Wan-li, A. Mei-qing, An efficient and robust artificial bee colony algorithm for numerical optimization, Computers & Operations Research 40 (2013) 1256–1265.
[28] W. Gong, Z. Cai, C.X. Ling, H. Li, A real-coded biogeography-based optimization with mutation, Applied Mathematics and Computation 216 (2010) 2749–2758.
[29] C. Worasucheep, A harmony search with adaptive pitch adjustment for continuous optimization, International Journal of Hybrid Information Technology 4 (4) (2011) 13–23.
[30] X. Li, J. Wang, M. Yin, Enhancing the performance of cuckoo search algorithm using orthogonal learning method, Neural Computing & Application 22 (2013) 1–15.
[31] The History of Stock Exchange, 2012, http://en.wikipedia.org/wiki/Stock_exchange (retrieved 26.10.12).
[32] M. Statman, Foreign stocks in behavioural portfolios, Financial Analysis Journal (March/April 1999) 12–16, http://dx.doi.org/10.1002/etep.1766, in press.
[33] M. Tier, The Winning Investment Habits of Warren Buffett & George Soros, Martin Press, USA, 2006.
[34] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Applied Mathematics and Computation 214 (2009) 108–132.
[35] D. Ortiz-Boyer, C. Hervas-Martinez, N. Garcıa-Pedrajas, A crossover operator for evolutionary algorithms based on population features, Journal of Artificial Intelligence Research 24 (2005) 1–48.
[36] N. Ghorbani, S. Vakili, E. Babaei, A. Sakhavati, Particle swarm optimization with smart inertia factor for solving nonconvex economic load dispatch problems, International Transactions on Electrical Energy Systems (2013).