International Academic Institute for Science and Technology

International Academic Journal of Innovative Research
Vol. 2, No. 2, 2015, pp. 60-73.
ISSN 2454-390X

International Academic Journal of Innovative Research

www.iaiest.com

A discrete binary version of the Forest Optimization Algorithm

Shadi Mahmoudi[a] , Chia Lailypour[b]

[a] *Computer Engineering Department, Boukan branch, Islamic Azad Univesity Boukan, Iran.*

[b] *Electrical Engineering Department Of Urmia University, Urmia, Iran.*

Abstract

: In this paper Binary Forest Optimization Algorithm (BFOA) is introduced. Recently introduced FOA has proven its excellent capabilities, such as faster convergence and better global minimum achievement. As the introduced FOA is originally designed to solve continuous optimization problems, in this paper some of FOA components are modified in a way that it can be applied on problems with discrete and binary nature. To prove the BFOA performance and illustrate capability of the proposed binary version, it is applied to solve wide range of optimization problems. The comparative results between BFOA, Binary Genetic Algorithm (BGA), Binary Particle Swarm Optimization (BPSO), Binary Gravitational Search Algorithm (BGSA) and Binary Cuckoo Optimization Algorithm (BCOA) show that Binary Forest Optimization Algorithm (BFOA) has much faster convergence, similar to its continuous version.

Keywords: forest optimization algorithm, binary optimization, evolutionary algorithm

Introduction:

Optimization has a key role in solution of many modern world things. Domain of this scientific subject is being transpired from the limit of engineering fields of study. So, in recent years, evolutionary algorithms because of their abilities are being enjoyed from specific attention in solution of optimization problems [1]. Evolutionary algorithms are from random algorithms that start seeking with a set of random responses that called primary population and with use of competition and cooperation of population members, they can find optimized response. These ways, often find the optimized response faster than traditional optimizers.

Optimization problems have categorized from different attitudes. One of the most important of these dimensions is being continuous and discrete of them. According to this view, since the response of optimization problems is being discussed in continuous or discrete spaces, evolutionary algorithms that being introduced, should be able to solve the problems in both of spaces. From algorithms that have this ability, can say the genetic algorithms [2, 3], particle swarm optimization [4], differential algorithm [5, 6], imperialist competitive algorithm [7], ant colony [8], firefly algorithm [9], lion algorithm [10], cuckoo search algorithm[11] [12] and cuckoo optimization algorithm[13]. In fact, most of evolutionary algorithms in their emergence early, dictate in continuous form, but there are many problems that have binary and discrete nature. However, there are discrete problems that can solve in continuous space but, in other hand there are things that don't have this capability. So, in such a condition, need to discrete and binary algorithms feels. So, the algorithms like PSO and ICA algorithms that were been dictated in continuous form, immediately the discrete version of them was been presented.

In this paper, one of the strongest and the most modern evolutionary algorithms, we introduced the forest optimization algorithm[14]. The abilities of the forest optimization algorithm in continuous space and importance of solution of discrete optimization problems, provided this motive that in this research, we can suggest some ideas for the forest algorithm binary version.

This algorithm is inspired from a few of forest trees that can live for few decades. While the other trees can remain for a short period. FOA is inspired from trees seeding procedure that some of seeds fall on the land just under the tree, while, other seeds in widespread areas by natural procedures like water and wind movement and animals that feed on seeds or fruits and scatter the seeds to the different places that spread out the forest trees territory.

The rest of this paper is organized as follows: in section 2, the basic Forest Optimization Algorithm (FOA) is reviewed and its different parts are studied in glance. The idea to use basic FOA in binary optimization problems is described in section 3. The proposed algorithm is tested with different benchmark functions in Section 4. Finally the conclusions are presented in Section 5.

1. Forest Optimization Algorithm

This algorithm includes 3 main operator:

a) Trees local seeding

b) Population limiting

c) Trees global seeding

Fig.1 shows the forest optimization algorithm. The general way of FOA as next cases is dictated in section 2.1 and 2.6 that in continue of this discussion we investigate the details of this algorithm. FOA like other evolutionary algorithms beings with primary population from trees. So each tree includes a potential solution for solving of problems. A tree besides the number of variables has a part related to age of related tree. Age of a tree at first is being considered 0. After starting from trees, local seeding will generate some young trees. Then all of trees except those have been produced recently, being old and their age rise to 1.

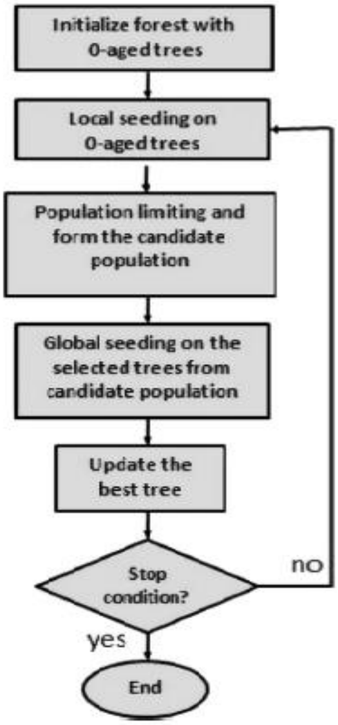


Fig.1. *Flowchart of FOA*

Then, there is control on population of forest trees and those should be omitted and also those that should being an introduction for forming in way of global seeding. If it wants to finish the process, algorithm will be finished and if it wants to continue, it returns to second stage that it seeding on the trees with age 0.

In stage of global seeding percentage of population have been chosen for being for from forest. Global seeding adds a number of potential solutions for forest that this problem is done for removing of local way. Now, forest trees have categorized based on their massive analogy and the tree that has the most massive analogy and the tree is chosen that its age be 0 that this action is done, for avoiding from trees oldness (because local way causes the increase of trees age and this thing is for the best tree). These stages will be continuing until the time of transmission criterion occurrence. In following, the stages of forest algorithm are dictated with details.

2.1 Primary trees:

In forest optimization algorithm a potential solution for each tree is be considered. Each tree shows the number of variables. In addition, each section includes the age of related tree. Age of each tree is be considered 0 for new generated trees that this problem is same with the result of local and global seeding. After finished of each stage from seeding age of all trees except the new-generated trees rises to 1. This age raising is known as controlling mechanism in number of forest trees. Fig.2. Shows a tree as after the form of N_{var} that V_{is} shows the number of variables and “age” part shows the age of related tree.

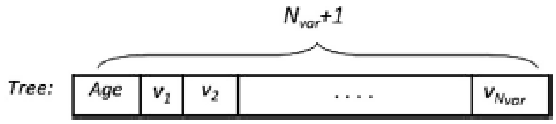


Fig.2. *A solution representation of FOA*

A tree also can be considered as a criterion of length in form of 1*(Nvar +1), that N_{var} after problem and the section of age is age of tree. This maximum value is defined the age of a tree as a parameter and called it as "life time". This parameter defined as the start of algorithm.

When age of a tree reaches to this stage, that tree will be omitted and adds to candidate population. If we choose a lot of trees for this parameter, each section from algorithm only raises the age of trees that age of trees teaches to oldness and the forest will be full of old trees that don't have any role in local seeding. In contrast, if we put a low value for trees those will be old so early and will omit in same start of competition. So this parameter should provide a suitable chance for local seeking.

2.2 Trees local seeding:

In nature, when trees seeding process starts, some of seeds fall on the land and just near the trees and after a few time, they switch into young trees. Now, competition between adjacent trees starts and those have better growing that have sunlight and better position, will be winner in the competition trees. Local seeding gives the possibility of simulating of this process in nature. This operator acts on the trees with age 0 and adds some adjacent trees to each tree in forest. Fig.3 shows the age of trees with the text of inside trees, that for each new- generated tree is considered 0, then local seeding is done. Age of all trees except new-generated trees raises to 1. The increasing of trees age like a controlling operative acts on the forest trees, that if a tree be in first of way, its age is be considered 0 and it will be possible to add the adjacent trees but if it doesn't occur, trees with age increasing that will have in each stage of algorithm, will die after some time.

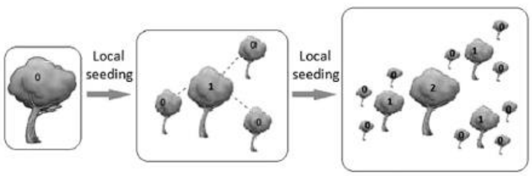


Fig.3. An example of local seeding on one tree for 2 iterations

Some of seeds that fall on the land near the trees and after some time, they change into young trees and have been considered as a parameter in algorithm and called "local seeding change (modifications)" that it has showed the amount of this parameter in fig.3, as collecting the local seeding performance on the trees with age 0, will generate three new trees that this parameter should be considered by the dimension of related problem. In first section of algorithm all of trees, their age is 0. Then local seeding is done on all of trees and for each tree with age 0, some trees add to forest. In next sections, some of trees those add will be decreased by local seeding performer. Because the age of trees will be more than 0 that doesn't any role in local seeding. Local seeding performer simulates the local seeking for this algorithm. Fig.4 shows a model from local seeding performer for real things in 4-dimensions continuous space. LSC value is divided to two categories include [–r, r] that both of these random generated values are in period of (-dx, dx). dx is a low value and less than variable upper limit that this way is used in procedure of seeking in internal boundary, and causes local seeking.

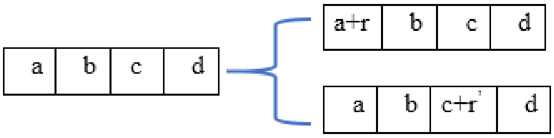


Fig.4. *An example of local seeding for continuous search space*

In order to performance of this operator, one variable is chosen with random. This procedure is repeated for each tree with age 0. Numerical value of it has been showed in fig.5 for instance. Value of LSC in fig.5 is 1 and dx is be considered 1. As a conclusion the value of a variable with a value that has been generated randomly will be raise that this value is [-1,1]. Now, new tree with age 0 will add to the forest.

Fig.5. *A numerical example of local* seeding

On one tree, *LSC*=1, r'=0.4 $\in [-\Delta x, \Delta x] = [-1,1]$ About the added values, they may be less or more than the related variables to upper limits. In order to avoiding from occurrence of these positions values that are less than minimum limits and also the values that are more than maximum limits will join to the limits. Local seeding performer adds some trees to the forest.

So, there should be some limitation for a numbers of trees that this control is done in next stage of suggested algorithm.

2.3 Population limiting:

Number of trees should be faced with limiting that this aims with avoiding from over spreading out from population of jungle. There are 2 parameters that cause this limitations life time and area limiting. At the first, the trees that faced with "life time" remove from forest and will define as candidate population. The second limiting that is "area limiting" categories the trees based on the analogy values. If number of trees be over of average, the extra trees will be removing and add to candidate population. Forest limiting will create another limiting that called area limiting. In some experiments in this article the "population limiting" is be considered with same initial trees. So, after this stage, number of trees will be same with number of initial trees. After forest population limiting, global seeding stage will be done on a percentage of candidate population that will be mention to it in next stages.

2.4 Forest global seeding

There are different trees in the forest and different animals and birds feed on the seeds and feed from the fruits of trees. So, the seeds in whole of forest have been scattered. As a set, habitat of trees will be more extended and also the natural processes like wind and water flow helps to spreading the trees in the forest and guaranties the variety of them in the forest. Global seeding process helps to making isotope for spreading the tree seeds. Global seeding operator performs on some percent of candidate population. This is another parameter from algorithm that first should be defined and called transfer rate.

Global seeding operative form as following:

First, a tree has chosen from the related population. Then, some variables of each tree, has chosen randomly. At this time, value of each variable exchange with generated value that it will be done in relation to variable period. This way is be considered in study gendered space and is not special for a specific area.

As a set, a tree with age 0 adds to forest. This operative is doing a general research in the problem space. The number of variables will be changes in this process in term of value, that this forms another parameter from algorithm that called global seeding changes (GSC). Fig.6 shows a model of performance of global seeding operative in continuous space for a tree. In this example, the value of GSC is be

considered with 2. Two variables that has chosen randomly, will be exchange with their values in two new-generated variable (-r, +r) in the related variable periods.
Numerical instance has shown for global seeding operative in fig.7, while GSC parameter value is 2 and it's same for related periods with the values of [-5, 5]. As a set, the value of 2 has been chosen randomly from variables

2.5 Update of best tree

In this stage, after categorizing of trees based on their analogy a value the tree has chosen as best tree that has the most value of analogy. Then, for avoiding from olding of best tree that define as result of local seeding, the age of best tree has considered with 0. In this way, it will be possible that best tree has been optimized in its situation that it will be done by local seeding operative, because as it mentioned, the local seeding will be performed on trees with age of 0.

2.6 Step conditions

Like other evolutionary algorithms, three kinds of stop condition can be considered:
-Number of defined performances
- Not changing in value of best tree analogy for several phases.
- Getting to specified level of validity

1. Pair making discrete of forest optimization algorithm:

Kennedy and eberhart in 1997 have introduced kind of PSO[15] that it can be seeked in a pair space. Domain of factual single- value can change into a pair domain simply. We limit sigmoid of speed by a sum in [0, 1].
This became an idea that the evolutionary algorithms that have been introduced in continuous space with using from an idea, they present binary version. From this algorithm, we can say: evolutionary algorithm, firefly, bee colony that have used from this idea for their binary version [16-21]. In refer [14], forest optimization algorithm had been introduced for solution of continuous problems. But, it's necessary that in many of optimization problems, seeking has done in discrete space. So, it's necessary that it introduce for optimization algorithms of a binary version.in discrete space, explorer ingredients move in a space of 0 and 1. Consider a special cloud. That coordination of each of its sides has specified with 0 and 1. Each of sides of this cloud is degree 3 of problem response. For seeking of this space, it's necessary that seeds move in layers of sides.
In discrete space, each situation ingredient has been shown in each dimension with 0 and 1. Movement of ingredient in each dimension means the value changing of it will be from 0 to 1 or from 1 to 0.
Idea of this is that new situation (X_{new}) in FOA local seeding stage, in each dimension is be considered as a possibility sum, and based on that ingredient, it will be change with a possibility. In fact, in binary version instead of that X_{new} shows the movement of ingredient. It shows the possibility of being 0 or 1. Since that X_{new} in binary algorithm should be defined as possibility sum, so it should be limited to [0, 1], for this we use from the equation 1.

$$S = \frac{1}{1+e^{-X_{NextHabitat}}} \qquad (1)$$

After calculating the above possibility sum, seed moves in each dimension with equation 2. According to this equation, seed changes its situation with a possibility that if speed of seed in a dimension is be more, possibility of seed movement is be more. Rand shows a randomly number with steady distribution in [0, 1].

$$if \quad S \le rand \quad Then \quad X_{NextHabitat} = 0$$
$$if \quad S \succ rand \quad Then \quad X_{NextHabitat} = 1 \qquad (2)$$

Pseudo code for the algorithm is changed as follows in fig.6:

Input: life time, LSG, GSC, transfer rate, area limit

Output: near optimize solution for objective function f(x)

1. Initialize forest with random trees
   1.1 each tree is a (D+1) dimensional vector x, x=(age, x_1, x_2, …, x_D) for a D-dimensional problem
   1.2 the age of each tree is initially zero
2. while stop condition is not satisfied do

2.1 perform local seeding on trees with age 0

- for i=1: 'LSC"
  -randomly choose a variable of the selected tree
  - Add a small amount dx, -dx …. To the randomly selected variable
- increase the age of all trees by 1 expect for new generated tree in this stage
- Using sigmoid function on new generated tree as following:

$$S = \frac{1}{1+e^{-X_{NextHabitat}}}$$

- Then generate random number $\in [0,1]$

$$if \quad S \le rand \quad Then \quad X_{NextHabitat} = 0$$
$$if \quad S \succ rand \quad Then \quad X_{NextHabitat} = 1$$

2.2 population limiting

- Remove the trees with age bigger than 'life time" parameter and add them to the candidate population
- Sort trees according to their fitness value
- Remove the extra trees that exceed the 'area limit" parameter from the end of forest and add them to the candidate population

2.3 global seeding

- Choose "transfer rate" percent of the candidate population
- For each selected tree
  - Choose "GSC" variables of the selected tree randomly
  - Change the value of each variable with other randomly generated value in the variable's range and add a new tree with age 0 to the forest

2.4 update the best so far tree

- Sort trees according to their fitness value
- Set the age of best tree to 0

3. Return the best tree as the result

Fig.6 Pseudo code of BFOA

2. Investigation of suggested algorithm usage and experiments results:

For evaluating of BFOA algorithm, we reported its conclusions on a set of available benchmark function in [18] that the features of experimented functions has been shown in table 1.1, m shows functions dimension that it is 5.

Table 1: benchmark function information that used in order to evaluating of BFOA algorithm

Test function	Type	Range
$F_1(X) = \sum_{i=1}^{m} x_i^2$	Unimodal	$[-100,100]^m$
$F_2(X) = \sum_{i=1}^{m} \lvert x_i \rvert + \prod_{i=1}^{m} \lvert x_i \rvert$	Unimodal	$[-10,10]^m$

$F_3(X) = \sum_{i=1}^{m}(\sum_{j=1}^{i} x_j)^2$	Unimodal	$[-100,100]^m$
$F_4(X) = \max_i \{\lvert x_i \rvert, 1 \le i \le m\}$	Unimodal	$[-100,100]^m$
$F_5(X) = \sum_{i=1}^{m-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	Unimodal	$[-30,30]^m$
$F_6(X) = \sum_{i=1}^{m}(\left[x_i + 0.5\right])^2$	Unimodal	$[-100,100]^m$
$F_7(X) = \sum_{i=1}^{m} i x_i^4 + random[0,1)$	Unimodal	$[-1.28,1.28]^m$
$F_8(X) = \sum_{i=1}^{m} -x_i \sin(\sqrt{\lvert x_i \rvert})$	Multimodal	$[-500,500]^m$
$F_9(X) = \sum_{i=1}^{m}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	Multimodal	$[-5.12,5.12]^m$
$F_{10}(X) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{m} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{m}\cos(2\pi x_i)) + 20 + e$	Multimodal	$[-32,32]^m$
$F_{11}(X) = \frac{1}{4000}\sum_{i=1}^{m} x_i^2 - \prod_{i=1}^{m}\cos(\frac{x_i}{\sqrt{i}}) + 1$	Multimodal	$[-600,600]^m$
$F_{12}(X) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{m}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + \ldots$ $\ldots + (x_m - 1)^2[1 + \sin^2(2\pi x_m)]\} + \sum_{i=1}^{m} u(x_i, 5, 100, 4)$	Multimodal	$[-50,50]^m$

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$$

7 is first sample of (F_1, f2) from unimodal kind, for this, set, convergence rate of algorithm is more important than optimized final result. F_8 to f_{12} is from multimodal kind that has local optimization and algorithm should be able to find the optimized or semi-optimized response and shouldn't stick in local optimization. All of functions that had been shown in table 11 are minimal optimization problems. f $_{opt}$ minimal value for this functions is 0 except, f_8 function that has minimum value of -418.9829.X_{opt} optimized situation for optimized functions is in $[0]^m$ except the functions f_5 and f_{13} that X_{opt} in $[1]^m$ and for f_8 function is in $[420.96]^m$.

For presenting introduced algorithm significance, we compared the algorithm on these functions and by BGA, BPSO, BGSA and BCOA algorithms. For three compared algorithm, the population rate is 50 and BCOA algorithm starts with 5 cuckoo and number of them raises to 20 because of being variable of cuckoo's number in COA, seeding limit [2, 4] has been used. The dimension of functions is 5. The following table 2 shows the parameters used in the algorithm BFOA

Table 2. *The parameters used in the BFOA*

Maximum iteration	500
Max Runs	30
dimension	5
area_limit	10
Life_time	15
Transfer_rate	10

Following table also shows the used parameters in BFOA algorithm. Average conclusions of best answer in 30 runs, median and standard of best answer has been calculated and presented in table 3.

Table 3: comparison of BFOA algorithm conclusions with BGSA, BGA, BPSO and BCOA algorithms.

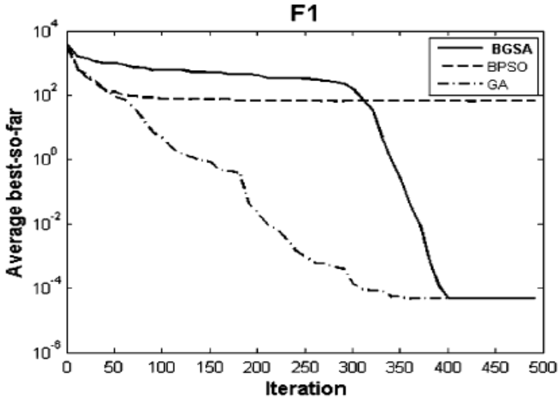
		BFOA	**BCOA**	**BGSA**	**BGA**	**BPSO**
F1	ABSF	0	0	4.65×10^{-5}	4.65×10^{-5}	64.62
	STDV	0	0	0	0	73.16
	MBSF	0	0	4.65×10^{-5}	4.65×10^{-5}	68.83
F2	ABSF	0	0	0.0016	0.0015	1.0823
	STDV	0	0	1.49×10^{-7}	5.22×10^{-8}	0.1087
	MBSF	0	0	0.0015	0.0015	1.0404
F3	ABSF	0	0	26.29	649.73	56.16

	STDV	0	0	$4.64\times10^{+3}$	$3.65\times10^{+5}$	$1.02\times10^{+3}$
	MBSF	0	0	0.6855	408.53	51.37
F4	ABSF	1	0	1.28	0.9186	4.39
	STDV	0	0	15.56	0.7583	2.07
	MBSF	1	0	0.0214	0.705	4.2787
F5	ABSF	133	0	3.8456	406.26	647.24
	STDV	1.7604e+2	0	2.0474	$4.68\times10^{+5}$	$2.87\times10^{+4}$
	MBSF	157.5000	0	3.9854	13.39	465.41
F6	ABSF	1.25	1.25	1.7240	1.9035	45.11
	STDV	0	0	0.0386	0.0375	166.39
	MBSF	1.25	1.25	1.7503	1.9712	47.73
F7	ABSF	0.000132	0.000188	0.0025	0.0032	0.0416
	STDV	0.000161	0	2.04×10^{-6}	7.54×10^{-6}	0.0011
	MBSF	0.000214	0.000314	0.0024	0.0025	0.0412
F8	ABSF	-2092.5763	-2091.8	-2083.1	-1034.7	-2009.1
	STDV	1.3597	2.5367	596.97	$3.16\times10^{+4}$	930.3
	MBSF	-2088.4	-2087.7	-2081.2	-1012.6	-2001
F9	ABSF	0	0	4.96	5.96	9.3
	STDV	0	0	4.58	11.78	4.75
	MBSF	0	0	5.73	6.07	9.98
F10	ABSF	8.8818×10^{-16}	8.8818×10^{-16}	0.004	8.82	5
	STDV	0	0	0	30	1.25
	MBSF	8.8818×10^{-16}	0.0882	0.004	10.11	4.75
F11	ABSF	0.1361	0	0.0409	0.1716	1.28
	STDV	0.0761	0	0.0004	0.0078	0.0922
	MBSF	0.1442	0	0.034	0.1511	1.37
F12	ABSF	0.0996	0.0569	0.7734	0.1835	0.9359
	STDV	0.1099	0.0986	3.39	0.0454	1.62
	MBSF	0.0489	0.0741	0.2037	0.0689	0.5694

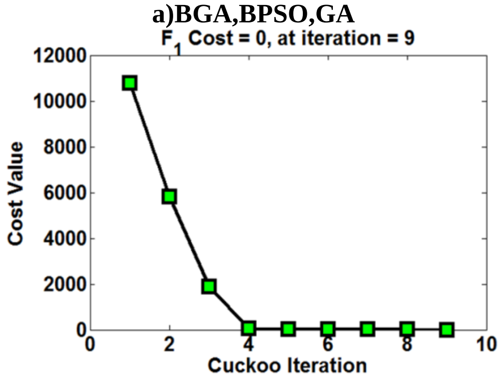
As it has been shown from table 3, in functions of $f_1,f_2,f_3,f_6,f_9,f_{10}$ gained conclusions from BFOA algorithm shows that the same answers with BCOA algorithm that is one of strongest recent algorithms, and in comparison of it with three other algorithms gained conclusions from above functions, BFOA algorithm had been a better function.

In f_4, BFOA algorithm is worse than GA and BCOA and is better than BGSA and BPSO. In f_5 and f_{11} also it is worse than BGSA and BCOA and is better than BPSO and GA and finally in functions of f_2,f_8,f_{12} it gained the near answer to BCOA and it had been operationed a better function in compare with other algorithms.

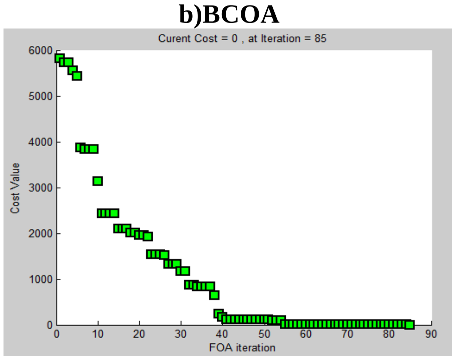
As it has been seen, BFOA algorithm is more efficient than three algorithms (GA, BPSO, BGSA), and it can compatible with BCOA algorithm. Dominance of forest continuous algorithm , it has been seen in speed and vision in access in regard to overall optimization even in pair form for discrete optimizations following form in fig.7 and fig.8 shows the BFOA convergence rate for f_1,f_7. This shows that BFOA Convergence is faster than BGA, BPSO, BGSA and it will be in overall optimized point in same time.



a)BGA,BPSO,GA

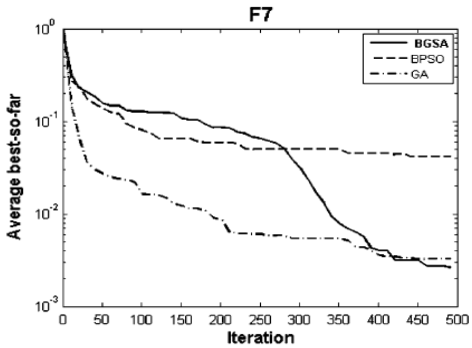


b)BCOA

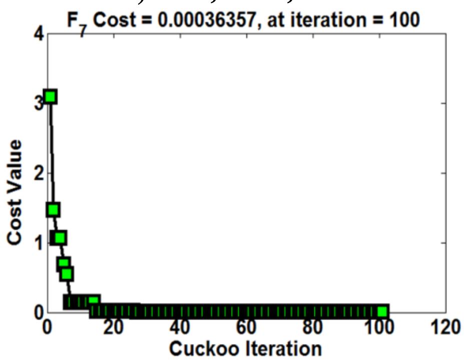


c)BFOA

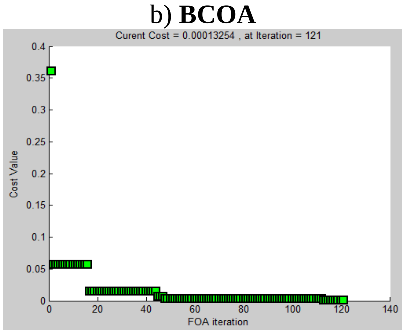
Fig.7. comparison of BFOA Convergence rate with BGA, BPSO, BGSA and BCOA in f_1



a) **BGA,BPSO,GA**



b) **BCOA**



c) **BFOA**

Fig.8. comparison of BFOA Convergence rate with BGA, BPSO, BGSA and BCOA in f_7

1. Conclusions

In this paper, forest optimization algorithm had posed for binary problems optimization. According to high Convergence and high visibility of forest algorithm in access to overall optimized point in optimization problems it had been expected that with using of this algorithm, with high pace and visibility it will be able to solve the discrete problems. Used way in this paper, is sigmoid function. Experiments conclusions in standard functions optimization in compare with BCOA, BGSA, BPSO, BGA algorithms, shows the algorithm suitable effectiveness. Suggested algorithm by global and local seeds operatives, is doing Exploration and exploitation in problems solution. In continue, we can use from suggested algorithms in this paper for solution of discrete optimization problems.

References:

B. Babu, R. Angira, A differential evolution approach for global optimization of MINLP problems, in: Proceedings of 4th Asia-Pacific conference on simulated evolution and learning, Citeseer, 2002, pp. 880-884.

C. Deng, B. Zhao, Y. Yang, A. Deng, Novel binary differential evolution algorithm for discrete optimization, in: Natural Computation, 2009. ICNC'09. Fifth International Conference on, IEEE, 2009, pp. 346-349.

C. Veenhuis, Binary invasive weed optimization, in: Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on, IEEE, 2010, pp. 449-454

E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: Evolutionary computation, 2007. CEC 2007. IEEE Congress on, IEEE, 2007, pp. 4661-4667.

E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, BGSA: binary gravitational search algorithm, Natural Computing, 9 (2010) 727-745.

G. Pampara, A. Engelbrecht, Binary artificial bee colony optimization, in: Swarm Intelligence (SIS), 2011 IEEE Symposium on, IEEE, 2011, pp. 1-8.

J. Holland, Adoption in Natural and Artificial Systems., University of Michigan, Ann Arbor, (1975).

J. Kennedy and R. C. Eberhart, Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks.Piscataway, NJ,, (1995) 1942-1948.

J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, in: Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on, IEEE, 1997, pp. 4104-4108.

M. Dorigo, G.D. Caro, L.M. Gambardella, Ant algorithms for discrete optimization, Artificial life, 5 (1999) 137-172.

M. Ghaemi, M.-R. Feizi-Derakhshi, Forest optimization algorithm, Expert Systems with Applications, 41 (2014) 6676-6687.

M. Mitchell, An introduction to genetic algorithms, MIT press, 1998.

M. Yazdani, F. Jolai, Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm, Journal of Computational Design and Engineering, (2015).

R. Rajabioun, Cuckoo optimization algorithm, Applied soft computing, 11 (2011) 5508-5518.

R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, Journal of global optimization, 11 (1997) 341-359.

S. Mahmoudi, S. Lotfi, Modified cuckoo optimization algorithm (MCOA) to solve graph coloring problem, Applied Soft Computing, 33 (2015) 48-64.

S. Palit, S.N. Sinha, M.A. Molla, A. Khanra, M. Kule, A cryptanalytic attack on the knapsack cryptosystem using binary firefly algorithm, in: Computer and Communication Technology (ICCCT), 2011 2nd International Conference on, IEEE, 2011, pp. 428-432.

sh. mahmoudi, R, rajabioun, sh. lotfi, Binary cuckoo optimization algorithm, First National Conference on Advances in computer science and information retrieval approaches, 2013.

X.-S. Yang, Firefly algorithm, stochastic test functions and design optimisation, International Journal of Bio-Inspired Computation, 2 (2010) 78-84.

X.-S. Yang, Nature-inspired metaheuristic algorithms, Luniver press, 2010.

X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, in: Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, IEEE, 2009, pp. 210-214.