Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 61 (2013) 16655-16661

ISSN: 2229-712X

# A new solutions for continuous optimization functions by using bacterial foraging optimization and particle swarm optimization algorithms

Farhad Soleimanian Gharehchopogh[1,*], Isa Maleki[2] and Behnam Zebardast[3]

[1]Department of Computer Engineering, Hacettepe University, Ankara, Turkey.

[2]Department of Computer Engineering, Dehdasht Branch, Islamic Azad University, Dehdasht, Iran.

[3]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, West Azerbaijan, Iran.

**ARTICLE INFO**

**Article history:**

Received: 15 May 2013;

Received in revised form: 24 July 2013;

Accepted: 31 July 2013;

**Keywords**

Optimization Algorithms,
BFOA,
PSO,
Continuous Functions.

**ABSTRACT**

Nowadays, optimization algorithm has good efficiency in solving complicated optimization problems. The main capabilities of theses algorithms are the ability to search in very big spaces in short time, escape from local optimal points, low computational costs and easy implementation. Therefore, one of the popular methods for solve complicated optimization is to use optimization method based on swarm intelligence. In this paper, we discuss about Bacterial Foraging Optimization Algorithm (BFOA) and Particle Swarm Optimization (PSO) in continuous optimization area and use them in minimizing the number of continuous functions. The obtained results indicated that PSO had better efficiency to solve continuous optimization problems.



## 1.Introduction

Nowadays, for solve many of complicated optimization problems due to time consuming and highly complicated features of classic methods than optimization methods based on swarm intelligence. Algorithms based on swarm intelligence are among of optimization algorithm which has considerable success in solving continuous and discrete optimizations. Techniques based on swarm intelligence have many applications. In [1, 2, 3, and 4], the researchers have been used swarm intelligence techniques to solve dynamic travelling salesman problem. Swarm intelligence techniques is shown good efficiency in solving complicated optimization problems by using very detailed mathematical models. It is introduced several optimization methods which can be noted to the most important ones such as PSO [5], BFOA [6], genetic algorithm [7], ant colony optimization algorithm [8] and so on. PSO [5] is one of the meta-heuristic algorithms which are usually used in solving various engineering problems. It is established to search in continuous spaces. In this algorithm, particles continuous the efficiency of the problem solutions continuously and provide a position in which they have the best success as the answer.

BFOA is one of the optimization algorithms to solve problems in continuous environments and is used in solving different problems such as job shop scheduling problem [9], digital images processing [10] and text summarizing [11]. The bases of BFOA is taken from bacterial behavior and acted based on random search, population and behavior-oriented attitudes. It has characteristics such as high convergence speed, flexibility and error tolerance which make it acceptable to solve problems of continuous optimization. The base of BFOA is according to functions which are taken from nature. The bacteria can find places which have more food and is verified by bacteria individual or colony search. In this algorithm, the bacteria have random behavior and there is no direct relation among them. Their search is based on indirect relation and is achieved by using direct and probable environmental move. Direct and probable move let the bacteria to find the shortest route which lead to food. These two features provide necessary flexibility to solve each optimization problem in continuous environments. The bacteria uses probable move in their routes which can lead to selects a route which might have more food. So, a simple change of move resulted in finding the shortest route to food source. It is widely used population algorithms based on swarm search for optimization. The major group of these algorithms is created from physical processes or creatures behavior. These algorithms are inspired from creatures swarm behaviors and mainly insects in solving complicated problems. In another words, it is used group behavior as a powerful tool to solve sophisticated problems. The continuous functions of mathematics are those which can't be solved by analytical methods. So, for accurate solve of this problems, it can be used different optimization methods. In these continuous functions, the main goal is to find maximum and minimum points.

By developing different types of optimization methods and intelligent algorithms, we face considerable changes in solving complicated optimization problems. Intelligent algorithm application is not only limited to solve mathematics complicated equation and problems but also include the wide range of optimization problems in different fields of engineering sciences. The swarm intelligent algorithm are considered important by many researchers in the recent years. In [12], the researchers analyzed BFOA. They indicated BFOA efficiency by solving a few mathematics functions. The goal of this function is to find minimum function in multi-dimension spaces. They also review the effect of the number of bacteria to solve functions and state that determining the accurate value of parameters cause BFOA has considerable effect in optimization problems. Researchers [13] discuss about solving continuous

Tele:
E-mail addresses: bonab.farhad@gmail.com

mathematical function by using Dynamic PSO and Simple PSO. They review the particles behavior in environments 10 and 2 dimensions to indicate the efficiency of two algorithms. They discuss about Dynamic PSO and Simple PSO algorithm parameters to analyze mathematics functions and their results indicate that Dynamic PSO algorithm has better efficiency to solve continuous problems and create the closer answer to optimal.

In this paper, it is used methods based on swarm intelligence namely PSO and BFOAs to Solve Continuous Optimization Functions (SCOFs). We organized the general structure of this paper as below: in the section 2, we introduce BFOA; in the section 3, PSO is introduced; in the section 4, BFOA and PSO are reviewed and assessed. In the section 5, BFOA and PSO are discussed and finally in the section 6, we come to make conclusion.

**2. Bacteria Foraging Optimization Algorithm**

BFOA is one of the optimization methods based on swarm intelligence which firstly introduced by Passino to SCOF in 2002 [6, 14]. Although, BFOA has similarities with other evolutionary ones but using distance and move to improve food searching operation makes it different from others [15]. In BFOA, it is modeled from the real bacteria behavior to find the shortest route to food source. As a bacterium finds a better route to food, it will attract the other bacteria and they will reach to the source faster. So, swarm operation resulted in group movement of bacteria to food source.

BFOAs are among optimization algorithms which inspired from bacteria movement rule to find food and search agents are a set of bacteria. These agents indicate simple behaviors under certain conditions and affect each other locally.

BFOA is inspired from the available natural bacteria such as E.Coli, has tools called Flagella, (to move bacteria to surround environment) [16]. The bacteria which have Flagella can search their surround environment in two methods. In most Flagella bacteria, as Flagella moves toward counter clockwise, the bacteria move forward but as it moves clockwise, the bacteria begin tumbling in place. In the BFOA, algorithm is called bacteria randomly movement as tumbling and bacteria movement in current rout is called swimming. Also to bacteria move in toward find food ''Run'' is called. As shown in fig (1), the bacteria inside the circle, indicates them in tumbling, and the others move indicate the moments that the bacteria move forward (''Run'').

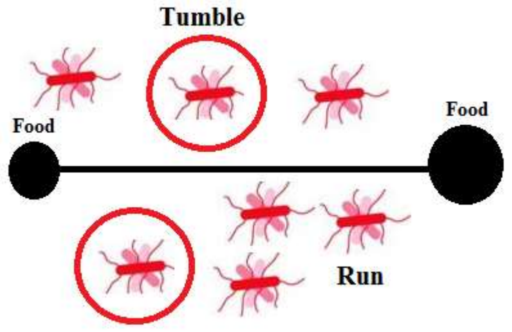


**Fig (1): Bacteria Movement**

In the cases that there is no absorbent to attract the bacterium, it moves and tumbles for a few seconds and then begins to move randomly. There are absorbent sensors on bacteria surface. If there is an absorbent in the environment, it will connect to the surface of these sensors and stimulates the bacteria to move. These stimulations must be continuous to make the bacteria moving. If the bacterium moves in a direction which the absorbent density decreased, it begins to far from the absorbent and tumbles. Then, it begins to move randomly and if there is absorbent, it continues moving. So, the combination of running and random moving cause the bacterium can look for food randomly in different directions more optimal and effective. Generally, to do so, if a bacterium feels that it closes to the stronger food source, it turns the Flagella toward counter clockwise and moves forward but if it wants to move far from the existing food source and looks for better one, it turns Flagella clockwise. So, the bacteria movement direction is a combination of running and random movement which resulted in placing each bacterium in a place with maximum food source. The implementation and designing process of BFOA has three consecutive stages and each one has different steps. In each step, it must be performed activities to implement that step.

These three stage include 1- Chemotaxis; 2- Reproduction; 3- Elimination and Dispersal

**3.1. Chemotaxis**

To the directional movement of bacteria randomly and in a certain direction to find food Chemotaxis is defined. If the movement is done randomly and toward an optimal place, it continues, otherwise, it will stop. Chemotaxis is performed in certain frequencies and by all bacteria. So, it has considerable effect in finding food sources by bacteria and moving in a certain direction causes that algorithm can reach to the better answer. Chemotaxis is defined by the following Equation (1):

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i).\Delta(i)}} \qquad (1)$$

In Equation (1), Δ (i) is a vector in [-1, 1] range and C (i) is the coefficient of Chemotaxis step size which is usually considered equal to small number. Determining the exact value of this parameter had considerable effect in bacteria movement to find food sources due to the type of problem.

**3.2. Reproduction**

Reproduction is one of the obvious characteristics of bacteria to reach the optimal answer. It performs in healthy and fast bacteria. In reproduction stage, the half of the bacteria who can't find good food will remove and the half remains which include healthy ones, will divide in two bacteria. It causes that the bacteria population become constant. It means that the number of bacteria becomes doubled repeatedly and their population increase considerable. The reproduction agent task is to select the numbers of previous generation healthy bacteria and place them in the next generation bacteria set at the aim at creating balance among searching process in environment. By doing so, the numbers of bacteria in each generation will be twice in the next one. The reproduction process is defined as Equation (2):

$$J^i_{health} = \sum_{j=1}^{NC+1} J(i,j,k,l) \qquad (2)$$

**3.3. Elimination and Dispersal**

In BFOA, the bacteria population is gradually changed by food consuming and /or other cases. These changes can be cases such as death or dispersal. Although, it might be resulted in losing movement stage toward food at first, but can be effective, too. As bacteria dispersal might be placed in places close to the good food sources, so, it has either positive or negative effects. In each stage that elimination and dispersal is performed, each bacterium with less efficiency will be dispersed or eliminated with $P_{ed}$ possibility. So, to keep the number of bacteria constant, if a bacterium is removed, the new one would be placed in the search space randomly. It prevents from the bacteria to be trapped in the local optimal point.

BFOA is frequently used Chemotaxis, reproduction and elimination and dispersal operators to produce new chain of bacteria. It increases the accuracy scale of bacteria to reach the close answer to optimal. Finally, if the considered provision to stop algorithm performance is achieved, it would be stopped and the best achieved answer would be introduced as the answer of optimization problem.

## 3. Particle Swarm Optimization

PSO is one of the swarm intelligence algorithms which performed based on population and random search. PSO algorithm was firstly introduced by Kennedy and Eberhart in 1995 and inspired from the birds' social behavior live in small and big groups [5]. PSO is a simulation of birds' social behavior in group who look for food in the environment. None of them know about the food source but they know in each stage that how far it is. Accordingly, the best approach to find food is to follow the closest bird to food source.

PSO is similar to evolutionary algorithm of population algorithm in which few particles formed a swarm (population). These particles are the solution of function volunteer or a problem [17]. The available population in the problem space begins to move and try to find the optimal solution in the search space based on their both individual and collective experiences. PSO algorithm as an optimization algorithm provides a search based on population in which each particle change its position over time. In PSO algorithm, the particles begin to move in a multi dimension search space of problem possible solution. In this space, it is defined an assessment criteria and the quality assessment of problem solutions is performed via it. Any change in the particles mode in a group is affected by its individual experiences and/or the neighbors' knowledge and a particle search behavior is also affected by the others in the group. This simple behavior leads to find optimal points of search space. So, in PSO algorithm, each particle properly informs the others as soon as find an optimal position. Then, each particle decides to follow the others based on the achieved values of cost function with a certain possibility. The search in problem space is done by using the previous knowledge of particles. It causes that all particles don't closes to each other excessively and can handle continuous optimization problems effectively.

In PSO algorithm, the group members are firstly created in the problem space randomly. And, the search is started to find the optimal answer. In the general structure, the search of each individual follows the others which have the most optimal fitness function. However, it doesn't forget individual experiences and follow the status in which it has the best fitness function. So, in each algorithm repetition, each member changes the next position according to two values. The first value is the best position which the person has so far (pbest) and the other value is the position which is achieved by the whole group so far and in fact the best pbest is in the general population (gbest). In conceptual meaning, pbest for each individual is considered in his/her biologic memory. gbest is the general knowledge of population. As the individuals' change their position based on gbest, in fact, they try o reach their knowledge level to the population knowledge level. From conceptual aspect, the best particle group related all group particles to each other. Determining the next position for each particle is done by Equations (3) and (4).

$$v_{i+1} = w.v_i + c_1.r_1.(P_{best_i} - x_i) + c_2.r_2.(g_{best_i} - x_i), \quad (3)$$

$$x_{i+1} = x_i + v_{i+1} \quad (4)$$

In Equation (3), $c_1$ and $c_2$ are learning parameters. Rand () is a function to produce random numbers in [0, 1] range. $X_i$ is the current position and $v_i$ the individuals' speed rate. W is a controlling parameter which controls the effect of the current speed ($v_i$) on the next speed rate and create a balance state between algorithm capability in searching as locally and generally. Consequently, we reach to the answer in the shorter time as an average. Parameter w is defined as follow to get optimal function of algorithm in the search space [18, 19]:

$$w = w_{Max} - \frac{((w_{Max} - w_{Min}) \times Iter)}{Iter_{Max}} \quad (5)$$

In Equation (5), $Iter_{max}$ indicates the maximum of algorithm repetition number and Parameter Iter Counter out the repetitions to find the optimal answer. In Equation (5), $W_{Min}$ and $W_{Max}$ parameters are the primary value and the final value of inertia weight during algorithm performance, respectively. The value of inertia weight is changed linearly from 0.9 to 0.4 during algorithm performance. The big and small values of W resulted in general and local search, respectively. In order to make a balance among the searches, it is necessary to decrease the inertia weight during algorithm performance, monotonously. So, by decreasing the value of W, the search process is performed local and around the optimal answer.

## 4. SCOFs by Using BFOA and PSO Algorithm

In continuous functions, due to multi-variability and multi-dimensional spaces, using optimization classical methods are generally inefficient and computationally expensive. In another words, applying classic optimization methods for continuous functions resulted in local optimal answer in most cases. A possible way to solve these complicated optimization problems are to use meta-heuristic algorithms which have higher possibility of success to reach general optimal answer. As meta-heuristic algorithms can search for very big spaces in almost short time at the aim at finding general optimal answer, effectively, so we used BFO and PSO algorithms to assess continuous functions efficiency.

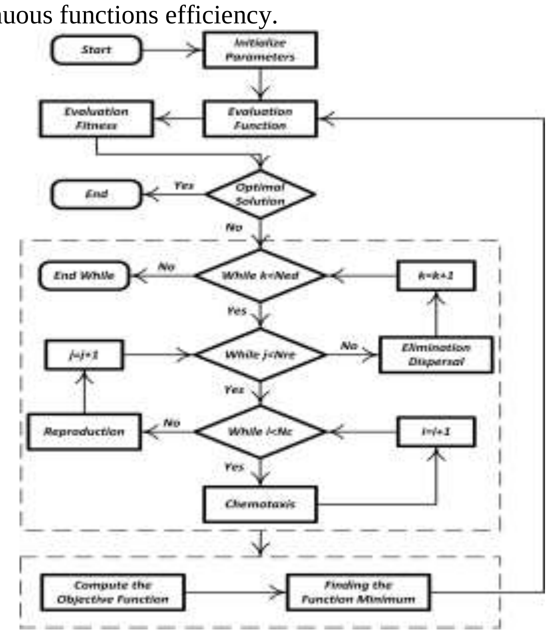


**Fig (2): The Flowchart of BFOA to Solve Continuous Functions**

### 4.1. BFOA for SCOFs

BFOA is a meta-heuristic method to find general optimal answer in problems with big spaces. This algorithm acts well as the goal of optimization is to find an acceptable answer (and not necessarily generally optimal). In fact, it calculates the problems at the aim at finding optimal answer (either maximization and /or minimization). In Fig (2), the flow chart of BFOA is shown to SCOFs.

In each repetition of algorithm, the bacteria begin to find new answers by random searching around the achieved answers in previous repetitions. It is clear that all these answers don't necessarily optimal. To do so, the achieved answers are calculated by the bacteria, so, the close answer to optimal achieved for SCOF. The quasi code of BFOA to SCOFs include below stages:

*1. Initialize Parameters*
*2. Do*
*3. Evaluation of Continuous Functions*
*4. Evaluation Fitness*
*5. BFO Algorithm*
***Elimination-dispersal***
***Reproduction***
***Chemotaxis***
*6. Finding Best Position*
***Search domain of Function***
***Best Solution Found***
*7. Updating Position*
*While* ***(checking termination criterion)***

**Fig (3): The Quasi Code of BFOA to SCOFs**

### 4.2. PSO for SCOFs

PSO is an optimization technique which mainly used to SCOFs. In Fig (4), the flow chart of PSO is shown to SCOFs.

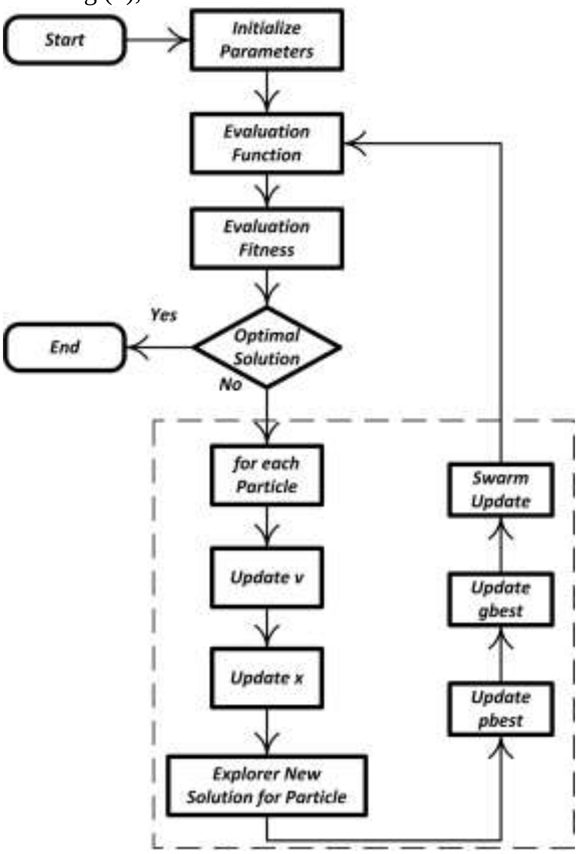


**Fig (4): the Flowchart of PSO to SCOFs**

The quasi code of PSO to SCOFs include below stages:

*1. Initialize Parameters*
*2. Do*
*3. Evaluation of Continuous Functions*
*4. Evaluation Fitness*
*5. Evaluate Fitness of Individual Particles*
*6. Update pbest*
*For each* ***particle i with position $x_i$*** *do*
*If* ***($x_i$ is better than pbest)*** *then*
***pbest = $x_i$***
*End if*
*End for*
*7. Update gbest*
*For each* ***particle i with position $x_i$*** *do*
*If* ***($x_i$ is better than gbest)*** *then*
***gbest = $x_i$***
*End if*
*End for*
*8. Renew the Particle*
*For each* ***particle*** *do*
***Calculate particle velocity according to equation (3)***
***Update particle position according equation (4)***
*End for*
*While* ***(a stop criteria maximum iteration)***

**Fig (5): The Quasi Code of PSO to SCOFs**

## 5. Evaluation and Results

In solving widespread problems of mathematics field, the swarm intelligence method is used to find functional maximum and minimum of multi variable at the aim at optimization. In an optimization problem, there are bounds which dominate the problem. The lack of attention to these bounds usually provides the possibility of inefficient answer. In multi- variable problems, the dominated bounds can be defined improper value of variables in a chain of available member in range. So, in complicated mathematical problems, it can't be selected the desirable values of variables. Rather, these variables must have detailed values. In this problem, when the number of variables is more than 2 or 3, determining the number of bounds will become complicated and we can't determine the detailed value. As a result, using swarm intelligence method cause that the problems are frequently and with different values of parameters solved and then the results are compared to reach an acceptable solution. In order to assess BFO and PSO algorithms, we come to solve several continuous and multi dimensional functions. The functions we consider to do experiments are as follow [20]:

- Two-dimensional Rastrigin function

$$f_1(x) = 20 + \sum_{i=1}^{2} [x_i^2 - 10\cos(2\pi x_i)]$$

- Two-dimensional Rosenbrok function

$$f_2(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$$

- Three-dimensional Rastrigin function

$$f_3(x) = 30 + \sum_{i=1}^{3} [x_i^2 - 10\cos(2\pi x_i)]$$

Rastrigin function (even in two variables) is considered as one of the difficult issues in continuous optimization field and its solving is almost more difficult than two variables Rosenbrok function. Rastrigin function has a lot of general maximum and minimum points which make it interesting to use as test function to assess efficiency of different optimization algorithms.

In PSO and BFO algorithms, there are several parameters which affect algorithms performance and function. In Table (1), Parameter S is the number of bacteria as well as Parameter Nc which indicates the number of Chemotaxis stages. Parameter Ns determines the number of random steps in each stage of chemotaxis. Parameter $N_{re}$ is also determines the number of reproduction steps. Parameter Ned is the number of elimination and dispersal steps. Each available bacterium in population will be removed by $P_{ed}$ possibility and Parameter C (i) is the size of chemotaxis step in (i) Bacterium. In PSO algorithm, Parameter P determines the number of particles. Parameter W is also indicates the inertia weight to balance the particle speed rate. The value of parameters C1 and C2 help to learn particles to find optimal points. The primary population is also considered in both algorithms equal to 100.

**Table (1): Parameters Values**

| BFOA | |
|---|---|
| Parameter Name | Value |
| S | 100 |
| Nc | 20 |
| Ns | 5 |
| Nre | 8 |
| Ned | 4 |
| Ped | 0.25 |
| C(i) | 0.05 |
| PSO | |
| Parameter Name | Value |
| P | 100 |
| $W_{Max}$ | 0.4 |
| $W_{Min}$ | 0.9 |
| C1 | 1.5 |
| C2 | 1.5 |

The obtained results are taken from simulation after performing 100 times repetitions. To prevent changes range of variables to be limited in a certain range and also to be entered the searched neighboring to the out of allowed changes range of variables, the search space keeps constant along algorithm function process. Determining the search space range causes that algorithms has better capability to find more exact position.

According to the results in Table (2), PSO is suitable than BFOA to solve problems in which local minimum points are surrounded by maximum points with wide range. In Table (2), Parameter D is the multi-dimension of the search space in which we consider the value of this parameter for two-dimension functions of $f_1$ and $f_2$ and three-dimension functions of $f_3$. As it can be seen in Table (2), BFOA acted almost well and can reach to the optimal value for f2 function.

**Table (2): Finding the Optimal Answer**

| ***Function*** | ***Range of Search Space*** | ***BFOA*** | ***PSO*** |
|---|---|---|---|
| $f_1$ | $[-20 , 20]^D$ | ***0.0001*** | ***0.0000*** |
| $f_2$ | $[-20 , 20]^D$ | ***0.0000*** | ***0.0000*** |
| $f_3$ | $[-20 , 20]^D$ | ***6.1968*** | ***0.0000*** |

The exact calculation of the neighbor states in SCOFs by using BFOA is almost more complicated and difficult. Although, there is a search possibility in PSO algorithm to find better answer after particles convergence to an optimal route. It is also possible to run away from local optimal points and to find the exact position of general optimal answer. As it can be seen in Table (2), PSO algorithm can get a general optimal point with high accuracy by testing more points in the search space. So, it can be concluded that PSO algorithm had better convergence than BFOA to general optimization in big spaces range.

Table (3) is shown the variables optimal value after the algorithm ends up. The results of Table (3) indicate that PSO algorithm efficiency is more accurate in finding closer points to optimal. So, PSO algorithm with very-high possibility can reach to the general optimal point.

**Table (3): Finding Optimal Position**

| Function | Optimum Position | |
|---|---|---|
| | BFOA | PSO |
| $f_1$ | [-0.0001 -0.0005] | [0.0000 0.0000] |
| $f_2$ | [1.0002 1.0004] | [1.0000 1.0000] |
| $f_3$ | [1.0031 1.0111 0.0073] | [0.0000 0.0000 0.0000] |

In order to indicate PSO and BFOA well, we consider the search space range bigger in Table (4). The other reason is that at one side the continuous functions has a lot of local optimal points and at the other side the value of cost function in these points are too close.

**Fig (4): the Effect of Search Space in Finding Optimal Answer**

| Function | Range of Search Space | BFOA | PSO |
|---|---|---|---|
| $f_1$ | $[-50 , 50]^D$ | 0.0051 | 0.0000 |
| $f_2$ | $[-50 , 50]^D$ | 0.0001 | 0.0000 |
| $f_3$ | $[-50 , 50]^D$ | 15.1075 | 0.0000 |

As it can be seen in Table (4), BFOA convergence is slower in compared with PSO algorithm. The important thing is that the efficiency of BFOA decreases by increasing the search space. So, it can be concluded that PSO algorithm has better capability to find general optimal answer by increasing the search space. Table (5) indicates the variables optimal value in big spaces after the algorithms end up. As indicated in Table (5), PSO algorithm could find the optimal value of variables better and the continuous functions become closer to the optimal answer.

**Table (5): the Effect of Search Space in Finding Optimal Position**

| Function | Optimum Position | |
|---|---|---|
| | BFOA | PSO |
| $f_1$ | [0.0021 0.0029] | [0.0000 0.0000] |
| $f_2$ | [1.0004 1.0007] | [1.0000 1.0000] |
| $f_3$ | [1.9775 1.0072 -0.0019] | [0.0000 0.0000 0.0000] |

PSO algorithm keeps the balance among local and general search in a proper method to implement search process. As a result, by increasing the dimension of PSO algorithm optimization problem, it can be searched the problem space, effectively and reach better results than BFOA. In another word, in PSO algorithm, the neighboring space around each particle is searched better.

**6. Discussion**

PSO algorithm is one of the most successful and efficient algorithms which is crated to solve complicated optimization problems so far. As most researchers believe that PSO algorithm is the best algorithm to SCOFs. It can be noted to advantages such as easy implementation, low parameters, and quick convergence to reach close answer to optimal. The main problem of it is increasing rate of particles speed which causes it can't get the optimal position accurately and effectively. So, the high speed rate of particles has negative effect on optimization algorithm of particle group. To prevent the effect of this algorithm high speed rate and find new spaces, it must be created balance among local and general searches by inertia weight parameter. As the inertia weight is decreased during the algorithm operation monotonously. So, the algorithm is convergent to the optimal points by general movement of the particles toward the proper space.

BFOA has better efficiency in high dimension functions and those which have many local minimum functions. The bacteria movements to far from improper spaces, causes that the algorithm properly acts about the problems with very big dimensions and those have improper primary distribution and indicate suitable results. So, the success rate of BFOA is almost good in solving optimization problems. In BFOA, the primary population of bacteria in each stage of chemotaxis makes itself far from improper spaces and then moves to proper ones. And finally, in these spaces, reproduction agent tries to close optimal points. Although, in PSO in which there is no need to perform any operation on particles, the particles only search for group knowledge and learning about the problem space. One of the other known problems in BFOA is that the number of almost optimal ones has dominated on bacteria population rapidly and causes that the algorithm convergent to a local optimal point. As the bacteria population converge to a local optimal point (early convergence), the algorithm ability to continue search to find better solution is removed.

BFOA problem in solving complicated mathematical problems is that it isn't capable of changing a status from an optimal point to the other. Generally, BFOA is an approximation algorithm. In it, even if the best solution found, the search process might be stopped. In this way, optimal solution can be far from solutions in which the algorithm passes. But, PSO algorithm provides an opportunity to escape from local optimizations and mutated to the other parts of search space as placed in particles local optimization. Then, they begin to search optimal answer and repeat it as reach to the general optimal one.

PSO algorithm is considered almost simple in compared with BFOA. As it used only controlling parameters, so, it can be determined the proper value of these parameters in practice by test and error. PSO implementation algorithm is also simple and has high chance to find general optimal answer if it uses the proper value of these parameters. Although, BFOA has more parameters in compared with PSO algorithm and there is already no proper value of parameters.

**7. Conclusion and Future Works**

In this paper, we discuss about two major swarm intelligence algorithms namely BFOA and PSO to solve SCOFs. As you can see, BFOA uses three basic functions of Chemotaxis, reproduction and elimination and dispersal to find optimal answer. And finally, the best found answer is considered as the final answer of SCOF. Due to the obtained results from two algorithms, we find out that PSO algorithm is more capable than BFOA in flexibility and convergence speed rate to reach optimal answer. We hope to use other meta-heuristics to find better solution for SCOFs in future.

**References**

[1] F.S.Gharehchopogh, I.Maleki, S.R.Khaze, "*A New Optimization Method for Dynamic Travelling Salesman Problem with Hybrid Ant Colony Optimization Algorithm and Particle Swarm Optimization*", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol. 2, Issue 2, pp. 352-358, February 2013.

[2] F.S.Gharehchopogh, I.Maleki, S.R.Khaze, "*A New Approach in Dynamic Traveling Salesman Solution: A Hybrid of Ant Colony Optimization and Descending Gradients*", International Journal of Managing Public Sector Information and Communication Technologies (IJMPICT), Vol. 3, No. 2, pp. 1-9, December 2012.

[3] I.Maleki, S.R.Khaze, F.S.Gharehchopogh, "*A New Solution for Dynamic Travelling Salesman Problem with Hybrid Ant Colony Optimization Algorithm and Chaos Theory*", International Journal of Advanced Research in Computer Science (IJARCS), Vol. 3, No. 7, pp. 39-44, Nov-Dec 2012.

[4] F.S.Gharehchopogh, I. Maleki, M. Farahmandian, "*New Approach for Solving Dynamic Traveling Salesman Problem with Hybrid Genetic Algorithms and Ant Colony Optimization*", International Journal of Computer Applications (IJCA), Vol **.**53, No.1, pp. 39-44, September 2012.

[5] J. Kennedy, R. C. Eberhart, "*Particle Swarm Optimization*", In Proceedings of the IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.

[6] K.M. Passino, "*Biomimicry of bacterial foraging for distributed optimization and control*", Control Systems Magazine, IEEE, Vol. 22, pp. 52-67, 2002.

[7] J. Holland, "*Adaptation in Natural and Artificial Systems*", University of Michigan, Michigan, USA, 1975.

[8] M.Dorigo, V.Maniezzo, A.Colorni, "*Ant system: optimization by a colony of cooperating agents*", IEEE Trans. on Systems, Man and Cybernetics, Part B, Vol.26, No.1, pp.29-41, 1996.

[9] S. Narendhar, T. Amudha, "*A Hybrid Bacterial Foraging Algorithm for Solving Job Shop Scheduling Problems*", International Journal of Programming Languages and Applications (IJPLA), Vol. 2, No. 4, pp. 1-11, October 2012.

[10] E. B.George, M.Karnan, "*MR Brain Image Segmentation using Bacteria Foraging Optimization Algorithm*", International Journal of Engineering and Technology (IJET), Vol. 4 No. 5, pp. 295-301, Oct-Nov 2012.

[11] M. D.Nikoo, A.Faraahi, S. M.Hashemi, S. H.Erfani, "*A Method for Text Summarization by Bacterial Foraging Optimization Algorithm*", International Journal of Computer Science Issues, Vol. 9, Issue 4, No. 1, pp.36-40, July 2012.

[12] L. Kaur, M. P.Joshi, "*Analysis of Chemotaxis in Bacterial Foraging Optimization Algorithm*", International Journal of Computer Applications, Vol. 46, No. 4, pp. 18-23, May 2012.

[13] H. S. Urade, R. Patel, "*Performance Evaluation of Dynamic Particle Swarm Optimization*", International Journal of Computer Science and Network (IJCSN), Vol. 1, Issue 1, February 2012.

[14] K. M.Passino, "Biomimicry of Bacterial foraging for Distributed Optimization", University Press, Princeton, New Jersey, 2001.

[15] H. Chen, Y. Zhu, K. Hu, "*Cooperative Bacterial Foraging Optimization*", Discrete Dynamics in Nature and Society, Hindawi Publishing Corporation, 2009.

[16] Y. Liu, K. M.Passino, "*Bio mimicry of Social Foraging Bacteria for Distributed Optimization: Models, Principles, and Emergent Behaviors*", JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 115, No. 3, pp. 603-628, December 2002.

[17] Y. Kumar, D. Kumar, "*Parametric Analysis of Nature Inspired Optimization Techniques*", International Journal of Computer Applications (IJCA), Vol. 32, No.3, pp. 42-49, October 2011.

[18] Y.Shi, R.C.Eberhart, "*A modified particle swarm optimizer*", In Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 69-73, Anchorage, AK, 1998.

[19] Y.Shi, R.C.Eberhart, “Parameter selection in particle swarm optimization”, In Proceedings of Evolutionary Computation VII (EP98), Springer-Verlag, pp. 591-600, San Diego, California, USA, 1998.

[20] M. Molga, C. Smutnicki, “*Test functions for optimization needs*”, 2005.