

Adaptive Adjustment of PSO Coefficients Taking the Notion from the Bee Behavior in Collecting Nectar

Abbas Fadavi¹, Karim Faez², Zeinab Famili³

¹ Department of Mechatronics, Science and Research Branch, Islamic Azad University, Semnan, Iran

² Department of Electrical Eng., Amirkabir University of Tech, Tehran, Iran

³ Adiban Higher Education Institute

Article Info

Article history:

Received Apr 16, 2016

Revised Jul 2, 2016

Accepted Jul 18, 2016

Keyword:

Adaptive setting

Global optimum

Particle swarm optimization

Standard search environment

The rate of convergence of particles

ABSTRACT

In particle swarm optimization, a set of particles move towards the global optimum point according to their experience and experience of other particles. Parameters such as particle rate, particle best experience, the best experience of all the particles and particle current position are used to determine the next position of each particle. Certain relationships received the input parameters and determined the next position of each particle. In this article, the relationships are accurately assessed and the amount of the effect of input parameters is horizontally set. To set coefficients adaptively, the notion is taken from bee behavior in collecting nectar. This method was implemented on software and examined in the standard search environments. The obtained results indicate the efficiency of this method in increasing the rate of convergence of particles towards the global optimum.

Copyright © 2016 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Abbas Fadavi,
Department of Mechatronics, Science and Research Branch,
Islamic Azad University,
Semnan, Iran.
Email: abbas_fadavi@yahoo.com

1. INTRODUCTION

The Particle Swarm Optimization algorithm [1]-[2] is composed of a set of particles. The aim of all the particles is approaching the optimum response and reducing error. The error of each particle is particle distance to response. Each particle can be a potential response. Each particle determines its future position by consulting with other particles and its experiences. The position of each particle is a result of its experiences and other particles' experiences. For example, we consider a person as a smart particle and the purpose as buying a suitable automobile. The person pays attention to two factors in buying a suitable automobile; First, his last experiences of buying an automobile and second, consulting with other people and asking their opinion about experiences of buying an automobile. The person, regarding his experiences and others' experiences in buying an automobile, selects his optimum automobile.

Figure 1 indicates how a hypothetical particle performs in the optimization algorithm of particle swarm optimization. The horizontal axis indicates the scope of search space and the vertical axis indicates the amount of error according to consistent function. As shown in Figure 1, there is a search space in which a particle tries to reach a global optimum. $x(t)$ is the position of a particle at the time t , $v(t)$ is the rate of a particle at the time t , $p_{best}(t)$ is the best experience of a particle to the time t and $g_{best}(t)$ is the best experience of all the particles to the time t . In PSO method, each particle tends to move towards its best experience and best experience of other particles. $p_{best} - x(t)$ is the distance of particle to its best experience and $g_{best} - x(t)$ is the distance of particle to the best experience of other particles. The rate $v(t+1)$ is the resultant of the two

components of $p_{best} - x(t)$ and $g_{best} - x(t)$. Based on these two components, experience gained during time and its experiences exchanged with other particles, particle x can move towards the optimum point.

Equation (1) indicates the calculation method of particle rate at the time $t+1$.

$$v(t+1) = w * v(t) + r_1 * c_1 * (p_{best} - x(t)) + r_2 * c_2 * (g_{best} - x(t)) \quad (1)$$

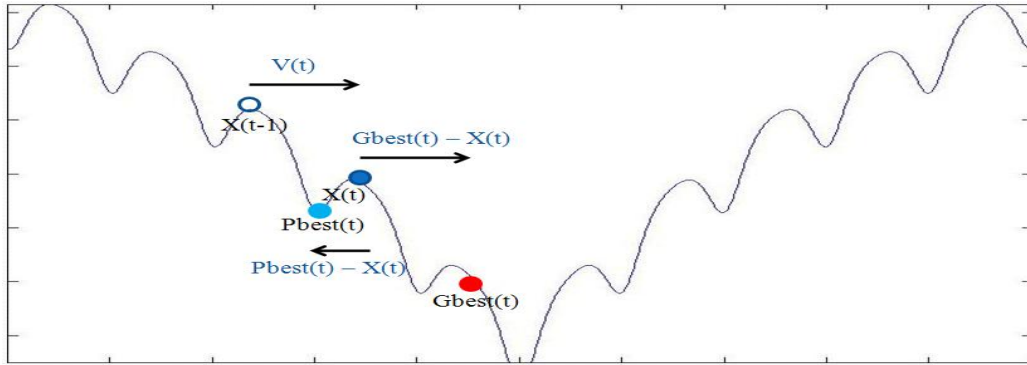


Figure 1. method for performance of a hypothetical particle in the optimization algorithm of particle swarm optimization

Components $v(t)$ is the rate of particle at the time t and coefficient w specifies the impact factor of $v(t)$ on $v(t+1)$. Component $p_{best} - x(t)$ is the distance of particle to its best experience and c_1 is the impact coefficient of this parameter on $v(t+1)$. Component $g_{best} - x(t)$ is the distance of particle to the best experience of other particles and c_2 is the impact coefficient of this parameter on $v(t+1)$.

Having the rate and current position, we can specify the next step $x(t+1)$ by equation (2). r_1 and r_2 are two random coefficients. These two coefficients are used to prevent particles' involvement in the local optima whose amount is between zero and one.

$$x(t+1) = x(t) + v(t+1) \quad (2)$$

1.1. Search space limitation

It is possible that the particles exit from the search space range while performing the algorithm. To alleviate this problem, Equation (3) is used.

$$x(t+1) = \begin{cases} x_{max} & \text{if } x(t+1) > x_{max} \\ x_{min} & \text{if } x(t+1) < x_{min} \\ x(t+1) & \text{else} \end{cases} \quad (3)$$

Where x_{min} is the minimum search space and x_{max} is the maximum search space. Equation (3) limits particle in the range of x_{min} and x_{max} .

1.2. Speed limit

Decreasing and increasing the particles' rate have a great influence on finding time for response in the optimization algorithm of PSO. If the rate of a particle is low, it must take more steps to reach where the response is. If the rate is high, the particle moves towards the response by taking larger steps and approaches the response area faster. If the amount of maximum rate is not limited, the particles become divergent and will be removed from the search space. For this reason, Equation (4) is used to limit the rate of each particle.

$$v(t+1) = \begin{cases} v_{max} & v(t+1) > v_{max} \\ v(t+1) & \text{else} \end{cases} \quad (4)$$

If $v(t+1)$ calculated by Equation (1) exceeds the allowed amount, it will be limited by equation (4).

1.3. Introduction of some of the proposed methods in PSO

SPSO [3] in this method, a certain amount is devoted to w which as time passes, reduces. In fact, it can be said that particles initially move, taking larger steps towards the area where the response is. Then, as time passes, the particles take smaller steps to be able to search more carefully. SAPSO [4] in neural networks, whenever one of the components has a good response, it will be encouraged but it will be punished in case of an unsuitable response. In this method, all the particles will be examined in each iteration. If p_{best} of each particle is not improved compared with its last time, that particle will be punished. Here, the weights of the particle will change.

DNSPSO [5] in the optimization algorithm of PSO, each particle pays attention to its best experience, all the other particles' best experience and its neighbors' best experience. TCPPO [6] in this method, PSO is composed of PSO Slave and Master. These two PSOs cooperate with each other to reach the optimum response. p_{best} in Slave PSO does not mean the best experience of the particle but is defined as the best experience of the particle and its neighbors. Master PSO uses the best experience of PSO Slave as well as its best experience and other particles' best experience.

PTPSO [7] materials occur in three phases of gas, liquid and solid. Gas molecules have the highest movement rate while solid molecules have the least. In this method, the notion of movement rate of molecules is used. Each particle has one of the material forms and moves according to the relevant formula of same material type. Particles change their phase based on various conditions. Particles move at different rates when conditions vary so that they can reach the optimum response rate.

Adaptive PSO [8] this method is the same as standard PSO but the only difference is that the amount of w is selected as adaptive. The equation of determining w is chosen in a way that PSO reduces the amount of w through finding the best g_{best} . This causes the particles initially moved by taking larger steps and then by finding better g_{best} , they look for the response by taking smaller steps.

RPSP [9] in this method, the parameter called $abest$ is used instead of g_{best} to position the particles. The amount of particles' p_{best} is examined and the better one is specified as the particle leader or best agent. Each particle moves towards the global optimum according to its best experience and the best position of agent. MPSO [10] in this method, there are four different equations in determining the position of the particle. In each attempt, each particle uses one of the equations randomly. Using different equations reduces the possibility of particles being involved in the local optima.

In the first section of this article, standard PSO will be examined completely after which some of the proposed methods will be discussed. In the second section, the offered method will be explained. Finally, in the third section, the results of performing the offered method and other proposed methods will be compared.

2. DESCRIPTION OF THE OFFERED METHOD

Equations (1) and (2) specify the position of a particle in PSO. These two equations can be combined and written as follows:

$$x(t+1) = x(t) + w * v(t) + r_1 c_1 (p_{best} - x(t)) + r_2 c_2 (g_{best} - x(t)) \quad (5)$$

To simplify the discussion, r_1 and r_2 are ignored and Equation (5) is written as Equation (6).

$$x(t+1) = x(t) + w * v(t) + c_1 (p_{best} - x(t)) + c_2 (g_{best} - x(t)) \quad (6)$$

According to Equation (2-1), we can write Equation (7).

$$v(t) = x(t) - x(t-1) \quad (7)$$

Then by combining (6) and (7), we can write

$$x(t+1) = x(t) + w * (x(t) - x(t-1)) + c_1 (p_{best} - x(t)) + c_2 (g_{best} - x(t)) \quad (8)$$

By simplifying Equation (8), we can write

$$x(t+1) = -w * x(t-1) + (1 + w - c_1 - c_2)x(t) + c_1 p_{best} + c_2 g_{best} \quad (9)$$

Equation (9) shows the four factors of $x(t-1)$, $x(t)$, p_{best} and g_{best} as the input and the amount $x(t+1)$ is calculated as the output. The amount of parameters' impact is determined by the coefficients w , c_1 and c_2 . For example, the impact of $x(t)$ is determined by the amount $1+w-c_1-c_2$.

2.1. New relation outline

This relation is posed by the notion that can receive the four factors of $x(t-1)$, $x(t)$, p_{best} and g_{best} as inputs. With the impact of each of the four coefficients is determined. Equation (10) is used to calculate $x(t+1)$.

$$x(t+1) = r_1 c_1 x(t) + r_2 c_2 x(t) + r_3 c_3 p_{best} + r_4 c_4 g_{best} \quad (10)$$

As is specified in Equation (10), four factors are used to determine the amount $x(t+1)$. The impact of each one of these factors is determined by the coefficients c_1 , c_2 , c_3 and c_4 . c_1 is the amount of impact of $x(t-1)$, c_2 the amount of impact of $x(t)$, c_3 the amount of impact of p_{best} and c_4 the amount of impact of g_{best} .

In the presented method of this article, we tried to control the amount of impact of each by a separate coefficient. In this method, the position of the best experience of the particle becomes prominent instead of the distance to the best experience of the particles. The coefficients r_1 , r_2 , r_3 and r_4 are random numbers in the range of zero and one used to prevent particles from falling in the local optima.

2.2. Determining the coefficients c_1 , c_2 , c_3 and c_4

Each one of the particles has its own specific coefficients. This enables PSO to regulate the amount of impact of parameters of the particles according to the conditions of the same particle. Each one of the four particles that has a greater amount will have a greater parameter impact amount relevant to it.

In this study, we applied the notion of bees' method of collecting nectar [11]-[14] in groups; bees perform a group of operations to collect nectar. We call those bees collecting nectar as working bees. After a period of time, nectar available in the garden will reduce and as such, a number of working bees turn into searching bees. The searching bees are obliged to be removed from the garden and enter into a new garden. Whenever one of the searching bees finds a new garden, it refers to other bees and gives the address to the rest of bees. The more nectar is available in the garden, the more working bees there will be. The lesser the nectar in the garden, the more searching bees there will be.

As we know p_{best} is the best experience of the particle. The response gained by p_{best} is better in some triestherefore, the amount of p_{best} varies. This indicates that the considered particle could find a better response. If the number of particles whose p_{best} has been optimized increases, more particles will find a better response. Here we consider each particle as a bee. We consider the percentage of the number of particles that optimizes their p_{best} in a try as nectar. At the start of algorithm, the PSO of all the particles was considered as the working particles. These particles search the optimum point according to their experiences and those of other particles.

If the percentage of the particles in each try that has not optimized their own p_{best} is not reduced from a certain amount, one of the particles varies randomly from a working particle to a searching particle. Whenever one of the particles selected as a searching particle finds a position better than p_{best} , all the searching particles turn into working particles and move towards the new optimum point. In fact, each particle can be placed in searching and working modes. Why do all the particles turn into searching particles when the amount of particles whose p_{best} is not optimized becomes lower than a certain amount?

There are two reasons why a percentage of the particles that has not optimized their p_{best} amount in each attempt are reduced from a certain amount.

1. The particles have approached the global optimum point and are finding the final response, taking smaller steps.
2. The particles are trapped in a local optimum by mistake. It is possible that particles are in the first case and it is not necessary to turn all the particles into searching particles.

In fact, the working and searching particles are controlled based on various conditions. It is a good condition that does not need to change when a large portion of particles is being optimized. However, if a large portion of particles is not being optimized, there must be a change in the general behavior of PSO particles.

2.3. Description of the working mode

Here it is assumed that the optimum point is minimum. The following points are implemented to determine the amount of coefficient of each working particle.

1. All the particles are inexperienced in $t=0$ and no particle is superior to another particle. For this reason, all the coefficients are equal in $t=0$.

2. Each amount of algorithm performance time is spent and the experience of particles increases. If a particle is placed in a proper position in the primary attempts randomly, it is possible to be determined as g_{best} . This particle loses its position over time, because other particles find better responses by their movement. In fact, it is shown that it is not valuable that a particle is g_{best} in the first moments. Over time, a particle cannot be g_{best} at random. For this reason, more attention should be paid to g_{best} over time. According to what was explained, by one iteration, the amount of c_4 increases by one unit.
3. Whenever $f(x(t-1)) > f(x(t))$, it indicates that the particle is probably moving in a right direction towards the response. Therefore, we increase c_2 by one unit. In the opposite case, the direction of movement is not probably suitable when the amount of c_1 increases by one unit.
4. Whenever a particle is selected as g_{best} , it shows a good experience of it. For this reason, this particle should pay more attention to its personal experiences. Therefore, whenever $f(g_{best}(t)) = f(x(t))$, the amount of c_3 increases by one unit.
5. In [15] we presented a method for recognizing the particles that are trapped in the local optimum. The standard parameter is defined as g_{best} in PSO. A parameter called g_{worst} is thus introduced in this study. The particle that has the worst efficiency function is known as g_{worst} and will thus be rearranged. This means its position varies randomly to move in another point until it probably joins the total active particles. If a particle is selected as g_{worst} , the amount of its next position will be selected randomly and all the coefficients of c_1 , c_2 , c_3 and c_4 are equal to one until the particle starts to move from the new point.

2.4. Searching mode

If a particle varies from the working mode to the searching mode, we determine the amount $x(t)$ as a random amount in the search space and consider $v(t)$ as zero. The searching particles use Equation (11) to determine their position.

$$x(t+1) = w * v(t) + c_1(p_{best} - x(t)) \quad (11)$$

As it is clear from Equation (11), the equation does not pay attention to the amount g_{best} and the particle moves independently regardless of the experience of other particles. This causes the particle to independently look for other responses in other points.

2.5. Using the random coefficients in combination

To prevent particles from being trapped in the local optimum, random coefficients are used in PSO. A combination of random coefficient has been used in previous studies [16]-[17]. The reports of these articles indicate better efficiency by combining random coefficients. In this study, we used a combination of random coefficient to increase efficiency. Therefore, the Equation (10) changes into the Equation (12)

$$x(t+1) = r_1(1 - r_2)c_1x(t) + (1 - r_1)r_2c_2x(t) + r_3(1 - r_4)c_3p_{best} + (1 - r_3)r_4c_4g_{best} \quad (12)$$

3. RESULTS AND DISCUSSION

In this section the results of the offered method is examined. The offered method is implemented on software that tests standard search environments [18]. Our aim is to present a new method so that it can reduce the amount of calculations. To examine the reduced calculations of the offered method, we implemented some different proposed methods in PSO methods on software. These methods have been implemented in the search environments and compared with the obtained results of the offered method.

To compare different PSO methods, we used the standard environments of Ackley, Griewank, Rastrigin. With the algorithms of SAPSO and DNPSO, the offered algorithms in this study are compared. In each performance, 10 different particles search the optimum response. We assumed $W=1$, $c_1=2$ and $c_2=2$. Dimensions of the search environment are considered as 1, 10 and 100. We illustrate the diagram of the performance time to the amount of the obtained consistent function in each algorithm. Each line of the diagram is the result of the average of 100 times of algorithm iteration in the definite environment.

As it is clear from Figures 2, 3 and Figure 4, the offered algorithms show better results compared with other methods and move towards the optimum response quickly. The efficiency function in the standard search environments makes us hopeful about the performance of the algorithm in the real environment.

The sum of the level below in every figure can be a suitable criterion for comparing the two methods. For example, in Figure 4, the sum of the level below DNPSO method equals 390. However, the sum of the level below in the offered method equals 80. This indicates that DNPSO method has better results compared with the offered method.

In this experiment, each method was performed 100 times and the figure of performance time to consistent function of each method was obtained. The sum of the below level of figures is shown as a measuring criterion of each method. Each method whose sum of the below level is lower could gain a faster response. Table 1 indicates the sum of the below level of different methods of different dimensions.

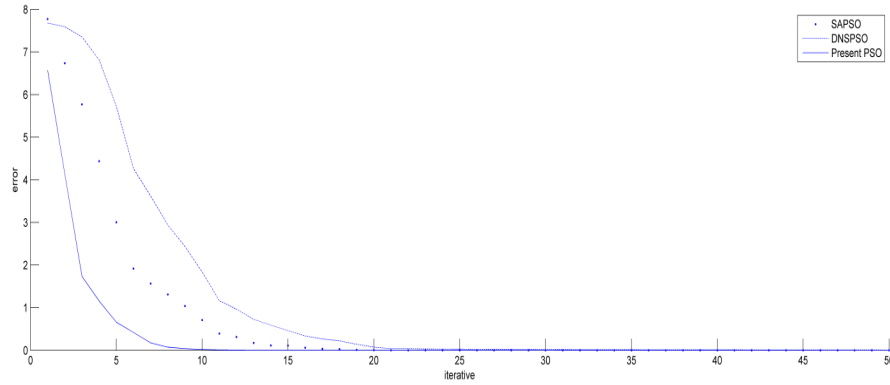


Figure 2. indicates the results of performances of different algorithms in Ackley environment of dimension 1

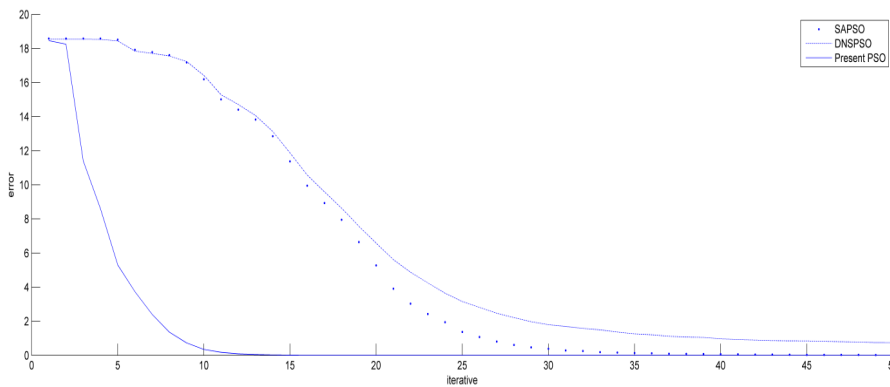


Figure 3. indicates the results of the performance of different algorithms in Ackley environments of dimension 10

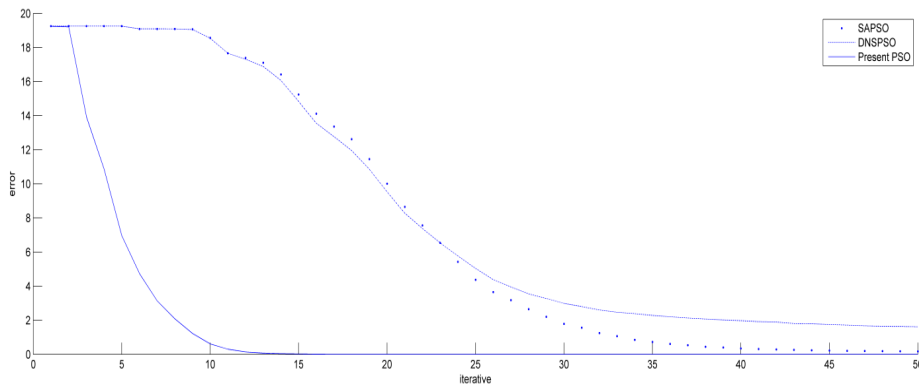


Figure 4. indicates the results of the performance of different algorithms in Ackley environments of dimension 100

Table 1. Sum of the belowlevel of different methods with dimensions 1, 10,100

	Ackley	Griewank	Rastrigin
	28	9	5
SAPSO	301	1175	1321
	397	18840	20063
	55	12	7
DNSPSO	368	1136	3418
	476	17308	53717
	18	3	0.6
Present PSO	72	351	651
	81	5507	11790

As it is show in Table 1, PSO shows a better result compared with other methods. For example, the sum of the below level of the figure in Ackley search environment of dimension 10 using DNSPSO method equals 368 but this amount equals 72 in the PSO presented.

REFERENCES

- [1] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Neural Networks, 1995. Proceedings, IEEE International Conference on*, vol. 4, pp. 1942-1948, 1995.
- [2] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on*, pp. 69-73, 1998.
- [3] Q. Li, *et al.*, "Optimization study on resource equilibrium with fixed time limit for a project based on SPSO algorithm," *Intelligent Information Technology Application Workshops, 2008 IITAW'08 International Symposium on*, pp. 70-73, 2008.
- [4] K. Yasuda and K. Yazawa, "Parameter self-adjusting strategy for Particle Swarm Optimization," *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, pp. 265-270, 2011.
- [5] H. Wang, *et al.*, "Diversity enhanced particle swarm optimization with neighborhood search," *Information Sciences*, vol. 223, pp. 119-135, 2013.
- [6] A. Afshar, *et al.*, "Honey-bee mating optimization (hbmo) algorithm for optimal reservoir operation," *Journal of the Franklin Institute*, vol. 344, pp. 452-462, 2007.
- [7] J. Ma, *et al.*, "Phase transition particle swarm optimization," *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pp. 2531-2538, 2014.
- [8] D. Wu and H. Gao, "Research of an adaptive particle swarm optimization on Engine Optimization Problem," *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2013 5th International Conference on*, vol. 1, pp. 42-45, 2013.
- [9] M. Anantathanavit and M. A. Munlin, "Radius particle swarm optimization," *Computer Science and Engineering Conference (ICSEC), 2013 International*, pp.126-130, 2013.
- [10] M. Pluhacek, *et al.*, "Investigation on the performance of Investigation on the Performance of a New Multiple Choice Strategy for PSO Algorithm in the task of Large Scale Optimization Problems," *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, 2013.
- [11] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Technical report-tr06, Erciyes university, engineering faculty, computer engineering department*, 2005.
- [12] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, pp. 108-132, 2009.
- [13] V. Nayak, *et al.*, "Implementation of Artificial Bee Colony Algorithm," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 1, pp. 112-120, 2012.
- [14] B. Akay and D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization," *Information Sciences*, vol. 192, pp. 120-142, 2012.
- [15] A. Fadavi and K. Faez, "The Effect of Rearrangement of the Most Incompatible Particle on Increase of Convergence Speed of PSO," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 3, pp. 238-245, 2013.
- [16] M. A. Arasomwan and A. O. Adewumi, "An adaptive velocity particle swarm optimization for high-dimensional function optimization," *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 2352-2359, 2013.
- [17] S. Sun and J. Li, "A two-swarm cooperative particle swarms optimization," *Swarm and Evolutionary Computation*, vol. 15, pp. 1-18, 2014.
- [18] K. Deep, *et al.*, "A new fine grained inertia weight Particle Swarm Optimization," *Information and Communication Technologies (WICT), 2011 World Congress on*, pp. 424-429, 2011.

BIOGRAPHIES OF AUTHORS

Abbas fadavi was born in Sari, Iran in 1978. He received the B.S. degree in electronic engineering from Azad university of Garmsar, Garmsar, Iran, in 2005 and the M.Sc. degree in Mechatronics from Science and Research Branch, Islamic Azad University Semnan, Semnan, Iran in 2012. His research interests include Image Processing, Pattern Recognition, Algorithm Optimization, and Neural Networks.



Karim Faez Was born in Semnan, Iran. He received his BSc. degree in Electrical Engineering from Tehran Polytechnic University as the first rank in June 1973, and his MSc. and Ph.D. degrees in Computer Science from University of California at Los Angeles (UCLA) in 1977 and 1980 respectively. Professor Faez was with Iran Telecommunication Research Center (1981-1983) before Joining Amirkabir University of Technology (Tehran Polytechnic) in Iran in March 1983, where he holds the rank of Professor in the Electrical Engineering Department. He was the founder of the Computer Engineering Department of Amirkabir University in 1989 and he has served as the first chairman during April 1989-Sept. 1992. Professor Faez was the chairman of planning committee for Computer Engineering and Computer Science of Ministry of Science, Research and Technology (during 1988-1996). His research interests are in Biometrics Recognition and authentication, Pattern Recognition, Image Processing, Neural Networks, Signal Processing, Farsi Handwritten Processing, Earthquake Signal Processing, Fault Tolerance System Design, Computer Networks, and Hardware Design. Dr. Faez coauthored a book in Logic Circuits published by Amirkabir University Press. He also coauthored a chapter in the book: *Recent Advances in Simulated Evolution and Learning*, Advances in Natural Computation, Vol. 2, Aug.2004, World Scientific. He published about 300 articles in the above area. He is a member of IEEE, IEICE, and ACM, a member of Editorial Committee of Journal of Iranian Association of Electrical and Electronics Engineers, and International Journal of Communication Engineering. Emails: kfaez@aut.ac.ir, kfaez@ieee.org, kfaez@m.ieice.org.



Zeinab Famili was born in Semnan, Iran in 1980. He received his B.Sc. degree in Electronic Engineering from Azad university of Garmsar, Garmsar, Iran, in 2005 and the M.Sc. degree in Electronic from Islamic Azad University Gazvin, Semnan, Iran in 2009. His research interests include Image Processing, Neural Networks. z_electron590@yahoo.com