Fuzzy optimum PSO: PSO with optimized fuzzy controllers

Omid Mokhlessi[1], Seyed Hamid Zahiri[1], Nasser Mehrshad[1]
1- Department of Electrical Engineering, Birjand University, Birjand, Iran.
Email: Omidmokhlessi@yahoo.Com (Corresponding author)
Email: Shzahiri@yahoo.Com
Email: N.mehrshad@gmail.Com

Received: March 2012 Revised: July 2012 Accepted: July 2012

ABSTRACT:
Since the search process of the particle swarm optimization (PSO) technique is non-linear and very complicated, it is hard if not impossible, to mathematically model the search process and dynamically adjust the PSO parameters. Thus, already some fuzzy systems proposed to control the important structural parameters of basic PSO. However, in those researches no effort were reported for optimizing the structural parameters of the designed fuzzy controller. In this paper, a new algorithm called Fuzzy Optimum PSO (FOPSO) has been introduced. FOPSO utilizes two optimized fuzzy systems for optimal controlling the main parameters of basic PSO. Extensive experimental results on many benchmark functions with different dimensions show that the powerfulness and effectiveness of the proposed FOPSO outperforms other versions of PSO.

KEYWORDS: Optimized Fuzzy Controller, Particle Swarm Optimization, Fuzzy Logic.

1. INTRODUCTION

PSO algorithm is a global intelligence technique which was discussed and developed by Kennedy and Eberhart in 1995. The global intelligence of social behavior of a society's individuals is as a corresponding collaboration to achieve the final goal. This approach is more effective than individuals separately struggle to achieve the final goal. PSO may be assumed as a society, including an organized exhibition of particles, which have simple structure and interact and collaborate interchangeably. The structure of PSO algorithm like the other similar global intelligence algorithms is extracted from neutral`s cooperation. In PSO, the particles stream in investigation space so that their displacement in investigation space is subject to their own experience and knowledge or those of their neighbors.

In PSO, the position of other particles of the society affects the way an individual particle investigates.

The modeling result of this social behavior is an investigation process which leads the particles to successful areas. Utilizing the gained knowledge, the particles of the society learn to move towards the area of their best neighbors.

In this paper, a compound algorithm of fuzzy and PSO has been used so that the problems concerned with the standard algorithm have been noticeably dealt and have caused an increase in convergence as well.

2. OPTIMIZATION ALGORITHM OF THE PARTICLES OF THE SOCIETY

PSO is asserted to a family of algorithms, which can be applied to discover the optimum responses of various functions. These functions can be the problems defined either numerically or quality. Many scholars have researched about PSOs introduced by Kennedy and Eberhart in recent years. Out of them, we can mention Mahfout and Lie_Abraham whose findings have been used in this paper. [1][2][3]

Standard PSO should be pointed at once in order to consider the influence of new factors based on new fuzzy controllers in this algorithm.

2.1. Standard PSO

What referred to as basic PSO, is that particles move through investigation space, and each particle maintains its best individual experience such as location vector and velocity vector, as well as the whole particle communities intrinsically providing and maintaining the best group experience as a whole.

What referred to as the velocity vector in this algorithm, is actually the output of three components, including the velocity in previous step, the best individual experience and the best group experience, which have been updated and yield the next speed [1][8].

The first value is the best experience that the particle itself has ever acquired, which is denoted by P-best. The second value is the best experience gained among all particles in the neighborhood, which is

denoted by G-best. In some versions of PSO, the particle chooses parts of population from the neighborhood nearby and shares its trend only with them. In this case, L-best is used instead of G-best [4].

In general, the PSO algorithm and the principle relations for speed update and particle location influence can be shown as follows.

While (not termination)
Update Local Bests ()
Update Global Bests ()
For each particle
R_1=Uniform random number
R_2=Uniform random number

$$V_{ij} = W.V_{ij} + C_1.R_1(X_{L-best} - X_{ij}) + C_2.R_2(X_{G-best} - X_{ij}) \quad (1)$$

$$X_{ij} = X_{ij} + V_{ij} \quad (2)$$

End for
End While

In order to prevent the algorithm divergence, the final value of speed has also been confined to an interval as $[V_{min}, V_{max}]$.

The algorithm termination condition is as convergence achievement to a specified limit. The achievement to the number of iterations is already defined.

3. STANDARD PSO PARAMETERS AND THE NECESSITY FOR IMPROVEMENT APPLICATION

3.1. Acceleration factor, α

Regarding Mahfout statements in his paper, in method which he refers to as AWPSO, the influence of a parameter called acceleration factor or α has been discussed. In fact, this parameter incorporates the factor of a kind of a virtual acceleration into the main PSO equation. Thus, with the influence of this parameter, the following change will apply to the speed update relevant equation.

$$V_{ij}(t+1) = W.V_{ij}(t) + \alpha.[C_1.R_1(X_{L-best} - X_{ij}) + C_2.R_2(X_{G-best} - X_{ij})] \quad (3)$$

α defines as an acceleration factor as follows.

$$\alpha = \alpha_0 - \frac{t}{T} \quad (4)$$

In this relation, t is the number of current iterations, and T is the number of overall iterations.

α_0 is a random number between 0.5 and 1. Through the experiments performed iteratively, better convergence can be achieved by replacing this value as follows, which can be caused more in effect through the final stages of convergence [3].

3.2. The personal parameters C_1 and group parameters C_2

As seen in main equation of standard PSO, the values C_1 and C_2 have been used for speed improvement. These two parameters together with random vectors R_1 and R_2 specify the personal and social parameters influence on random investigation. These parameters could be introduced as particles reliance parameters. The factor C_1 determines how much a particle is dependant on its experiences, whereas the parameter C_2 determines how much how much a particle relies on its neighbors experiences.

In general, the tendency of each member towards local and global optimum points is determined by these two factors. Various experiences offer the number 2 for both of these parameters, which by moving toward the global optimum point the influence of these factors gradually decreases and convergence of algorithm is fulfilled [1].

In fact, our point of view in this paper is the control of these two parameters by using one of the fuzzy controllers so as to provide the variations of these parameters with use of a smooth and appropriate fuzzy logic.

3.3. Inertia weight parameter, W

Of very important parameters in searching location and PSO investigation ability, which is multiplied by previous stage speed within the speed update equation, is W or the inertia factor. What Liu-Abraham had previously dealt with in their paper was the standstill and statics influence of lazy particles and the loss of dynamics to search other particles, which all together lead to termination of a huge category of some particles neighborhood experiences. The point of view of improving PSO investigation which is mentioned in this paper is the use of a speed threshold to provide the particles with more acceleration and increase their stimulation within neighborhood of particles in such case that the particles are slower than this threshold. They introduced this PSO as TPSO or PSO with turbulence capability. [2]

4. RELEVENT FUZZY SYSTEMS

4.1. A preface over fuzzy systems

Fuzzy logic is a relatively novel technology which is implemented more appropriately and more comfortably in comparison with complicated mathematical methods and those requiring advanced mathematical possibilities. This logic was presented by Professor Zadeh for the first time in 1960.

The Fuzzy logic provides a general concept for description and measurement. Most fuzzy logic systems encode human reasoning into a program to make decisions or control a system. Fuzzy logic compromises fussy sets, which are a way of

representing non-statistical uncertainty and approximate reasoning, which includes the operations used to make inferences in fuzzy logic. Fuzzy rule-based systems have been successfully applied to various engineering problems [9].

In this section the basic concepts and definitions of fuzzy systems are alluded to.

4.1.1. Membership Functions

Fuzzy logic-based systems are expressed using simple conversational linguistic variables and applying personal and linguistic knowledge. Each fuzzy system has several rules which declare the relationship between the inputs and outputs generation using conditional rules. Each linguistic phrase in a fuzzy system represents a membership function [6].

Unlike traditional two-valued logic, in fuzzy logic, fuzzy set membership occurs for a fuzzy variable by degree over the range [0,1]. This is represented by a membership function. The function can be linear or nonlinear. Commonly used are the left-trapezoidal, right-trapezoidal, triangle, Gaussian, and sigmoid functions, as shown in fig. 2. Definitions of these membership functions as used in this chapter are as follows.

a) Left-trapezoidal membership function:

$$LTrap - MF(x) = \begin{cases} 1 & if\ x < a \\ \dfrac{b-x}{b-a} & if\ a \le x \le b \\ 0 & if\ x > b \end{cases} \quad (5)$$

b) Right-trapezoidal membership function:

$$LTrap - MF(x) = \begin{cases} 0 & if\ x < a \\ 2\dfrac{x-a}{b-a} & if\ a \le x \le b \\ 2\dfrac{b-x}{b-a} & if\ \dfrac{b+a}{2} \le x \le b \\ 0 & if\ x > b \end{cases} \quad (6)$$

c) Triangle membership function:

$$Triangle - MF(x) = \begin{cases} 0 & if\ x < a \\ 2\dfrac{x-a}{b-a} & if\ a \le x \le \dfrac{b+a}{2} \\ 2\dfrac{b-x}{b-a} & if\ \dfrac{b+a}{2} \le x \le b \\ 0 & if\ x > b \end{cases} \quad (7)$$

d) Gaussian membership function:

$$Gaussian - MF(x) = e^{-0.5y2} \quad (8)$$

$$where\ \ y = \frac{8(x-a)}{b-a} - 4$$

e) Sigmoid membership function:

$$Sig - MF(x) = \frac{1}{1-e^{(-y+6)}} \quad (9)$$

$$where\ \ y = \frac{12(x-a)}{b-a}$$

f) Reverse-sigmoid membership function:

$$Rsig - MF(x) = 1 - sig - MF(x) \quad (10)$$

From the definitions, it can be seen that each above mentioned membership function is determined by two values (the start-point is a, and the end-point is b). Theoretically, each fuzzy variable can have many fuzzy sets with each having its own membership function, but commonly used are three, five, seven, or nine fuzzy sets for each fuzzy variable. Fig. 1 shows a fuzzy variable with triangular membership functions.

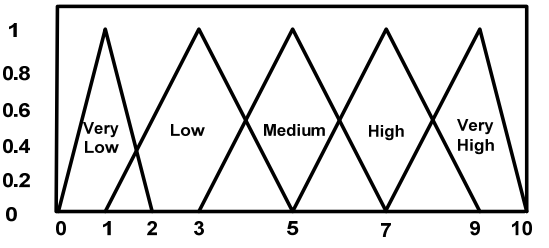


Fig. 1. A fuzzy variable with triangular membership functions

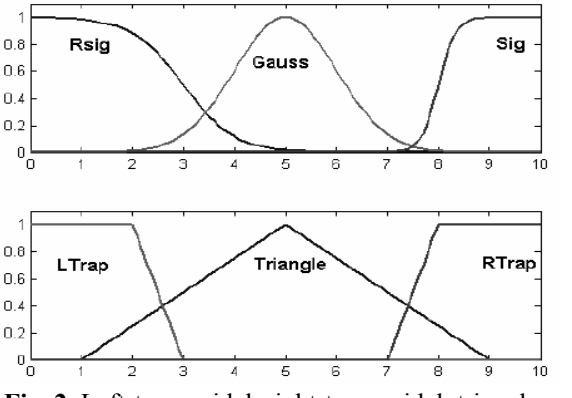


Fig. 2. Left-trapezoidal, right-trapezoidal, triangle, Gaussian, and sigmoid membership functions

4.1.2. Fuzzy Rules

Determination of the fuzzy membership values of the inputs is often called fuzzification. In these systems the process is performed by deductive reasoning in which simple fuzzy rules IF_THEN play the decisive role. There are two chief methods to deduction fuzzy rules in fuzzy systems. The first model is Mamdani type fuzzy deduction which was presented by Ibrahim Mamdani in 1975. The second type is the Takagi Sogno method presented in 1985. These two methods are the same in many aspects [6][9].

However the main difference between these two methods is their outputs. The Sogno's output is member of functions which may be linear or constant, while the membership functions outputs are fuzzy in Mamdani deduction. In this paper we have applied the Mamdani deduction to both of the relevant fuzzy systems design.

[5][6]

The general form of a Mamdani-type fuzzy rule in a fuzzy system is:

IF x_1 is A_1 AND x_2 is A_2, …, x_n is A_n THEN y_1 is C_1,…, AND y_k is C_k (11)

where each y_i is the consequent (output) variable whose value is inferred, each x_i is an antecedent (input) variable and each Ai and C_i is a fuzzy represented by a membership function. The antecedents are combined by AND fuzzy operator. AND's antecedents are usually calculated by T-norm. Other fuzzy operators are defined (e.g. OR, Aggregation operator, and Implication operator) [6]. In our application, a fuzzy system is utilized as a fuzzy parameters controller. In our fuzzy systems the most utilized operator for controlling parameters is AND operator. All the fuzzy rules in a fuzzy system are fired in parallel.

The fuzzy system works as follows:

1. Determine the fuzzy membership values activated by the inputs.
2. Determine which rules are fired in the rule set.
3. Combine the membership values for each activated rule using the AND operator.
4. Trace rule activation membership values back through the appropriate output fuzzy membership functions.
5. Utilize defuzzification to determine the value for each output variable.
6. Make decision according to the output values.

Each input may activate one or more fuzzy sets of that input variable according to the definitions of the fuzzy membership functions. Only the rules with at least one antecedent set activated are said to be fired by the inputs. The AND operator is typically used to combine the membership values for each fired rule to generate the membership values for the fuzzy sets of output variables in the consequent part of the rule. Since there may be several rules fired in the rule sets, for some fuzzy sets of the output variables there may be different membership values obtained from different fired rules. There are many ways to combine these values. One commonly used way is to use the OR operator, that is to take the maximum value as the membership value of the fuzzy set. Next, a defuzzification method is used to produce a single scalar value for each output variable. A common way to do the defuzzification is called centroid method. Then according to the output values, some decisions can be made to solve the problem [5][6][9].

4.2. Proposed fuzzy systems inputs and outputs introduction

Of chief problems in standard PSO is to fall within local optimums trap and slow down the convergence rate. Various methods have been proposed to improve the PSO. But the thing that should be in mind about gaining appropriate fuzzy outputs for C_2, C_1 and W is that these values must be chosen in such a way that at least the value gained for $V_{ij}(t+1)$ is meaningful [2]. In the other words, inappropriate values for $V_{ij}(t+1)$ can lead to the i_{th} member kickoff from the investigation space. This occurs because the i_{th} member location coordinates in step t+1 is calculated by sum of the two vectors V_{ij} and X_{ij}. The proposed optimized fuzzy algorithm is referred to as FOPSO. This algorithm makes use of two separate fuzzy systems.

The first system outputs are actually the factors C_1 and C_2 for which two inputs including α or acceleration factor and D_k or the parameter in charge of the particles normalized scatter to control these two outputs, which is calculated as below.

$$D_k = (D_t - d_{i\min}) / (d_{i\max} - d_{i\min}) \qquad i = 1,2,...N_{pop} \tag{12}$$

$$D_t = \sum_{i}^{N_{pop}} (d_i / N_{pop}) \qquad i = 1,2,...N_{pop} \tag{13}$$

$$d_i = | X_{Gbest\ i} - X_{Pbest\ i} | \tag{14}$$

In equations above, N_{pop} is the number of whole population, d_i is the difference of the best individual and group location for i_{th} particle and D_t is the average of whole population d_i's.

The second system inputs are the maximum of fitness values for p_{best} of each particle based on own experience in each iteration or F_{best} and the second one is the normalized value of fitness function or Fitt which is described in equation below. F_{max} and F_{min} are maximum and minimum of fitness values in each iteration. W is the output of our second fuzzy system which is controlled by these inputs.

$$Fitt = (F_{best} - F_{\min}) / (F_{\max} - F_{\min}) \tag{15}$$

5. THE PROPOSED FUZZY SYSTEMS INFLUENCE ON PSO

The two presented fuzzy systems control the relevant outputs by taking the inputs previously discussed. These outputs influence the PSO algorithm behavior. Out of them, we can mention the influence on local investigation, global investigation, convergence rate and finding an optimum based on less iteration steps. For example in the second fuzzy system, if the α value is high (considering [0.5,1.5]) it means that we are within the final steps and the investigation precision must increase; also if the D_k value is low, it means that the particles group is located at an inappropriate point. When this is the case, the most logical conditions are that the social parameter value increase and the personal parameter value decrease. Thus with similar reasoning we propose the two intended independent fuzzy systems together with 9 fuzzy IF-THEN rules as follows.

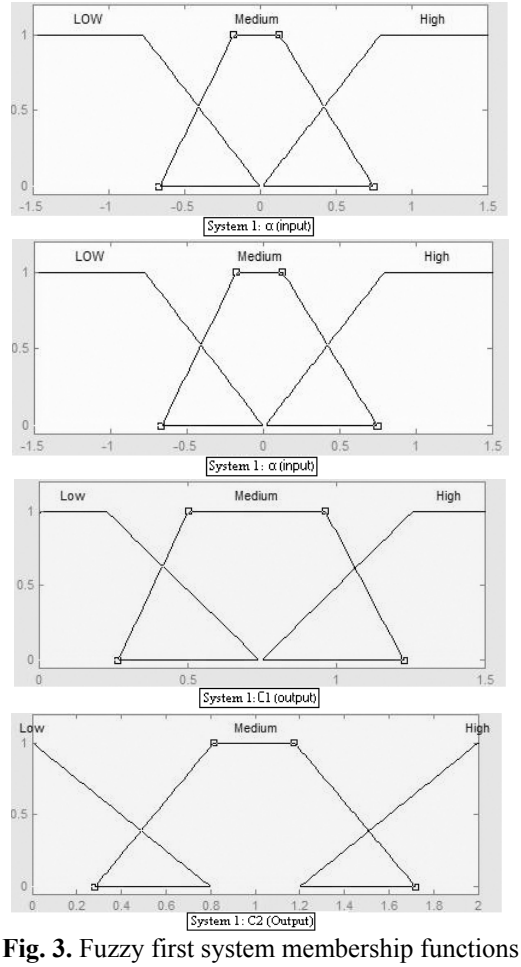


Fig. 3. Fuzzy first system membership functions

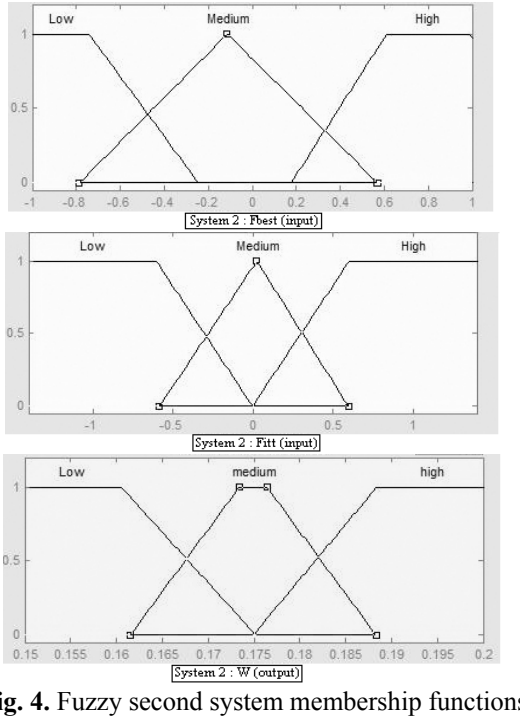


Fig. 4. Fuzzy second system membership functions

Table 1. Fuzzy Rules for first system

If (a is Low) and (D_k is Low) then (C_1 is Low) and (C_2 is Low)
If (a is Medium) and (D_k is Low) then (C_1 is Medium) and (C_2 is Medium)
If (a is High) and (D_k is Low) then (C_1 is Low) and (C_2 is High)
If (a is Low) and (D_k is Medium) then (C_1 is Medium) and (C_2 is Low)
If (a is Medium) and (D_k is Medium) then (C_1 is Medium) and (C_2 is Medium)
If (a is High) and (D_k is Medium) then (C_1 is High) and (C_2 is Medium)
If (a is Low) and (D_k is High) then (C_1 is High) and (C_2 is Low)
If (a is Medium) and (D_k is High) then (C_1 is Medium) and (C_2 is Medium)
If (a is High) and (D_k is High) then (C_1 is Low) and (C_2 is High)

Table 2. Fuzzy Rules for second system

If (f_{best} is Low) and (fitt is Low) then (w is Medium)
If (f_{best} is Medium) and (fitt is Low) then (w is Low)
If (f_{best} is High) and (fitt is Low) then (w is Low)
If (f_{best} is Low) and (fitt is Medium) then (w is High)
If (f_{best} is Medium) and (fitt is Medium) then (w is Medium)
If (f_{best} is High) and (fitt is Medium) then (w is Low)
If (f_{best} is Low) and (fitt is High) then (w is High)
If (f_{best} is Medium) and (fitt is High) then (w is Medium)
If (f_{best} is High) and (fitt is High) then (w is Low)

6. EXPERIMENTS

To compare FOPSO with standard PSO and AWPSO, we make use of a well-known suite of functions, which have appeared in many papers in the literature. For a fair competition, 30 particles in 20 dimension and the other of settings for all algorithms is shown in Table 3 in below:

Each algorithm was tested with 13 numerical functions shown in below. Each algorithm (for each benchmark) was repeated 10 times with different random seeds. The average fitness values of the best solutions, the best solution and worst solution were recorded and were shown in Table 5 below.

Some of these function, for example Rosenbrock have a single minimum, while the other functions are highly multimodal with multiple local minima. It is also useful for us to validate the algorithms without knowing the optimal value. Some of the functions have the sum of their variables, some of them have the product (multiplying) and some of them have dimensional effect. Each algorithm for different dimensions of the same benchmark function has similar solution so that we have represented this turnover as table 1 in below. [10]

Table 3. The comparison of algorithm `s parameters

Parameter	FOPSO	PSO	AWPSO	TPSO
Number of iterations	500	500	500	500
search-space [X_{min},X_{max}]	[-10.10]	[-10.10]	[-10.10]	[-10.10]
Range of velocity (Vmax)	4.0	4.0	4.0	4.0
w	Fuzzy	1.0	1.0	1.0
C1	Fuzzy	2.0	2.0	2.0
C2	Fuzzy	2.0	2.0	2.0
Neighborhood topology	global best	global best	global best	global best

Numerical benchmark functions

Sphere Function :

$$F_1(x)=\sum_{i=1}^{n}X_i^2$$

$[-5.12,5.12]^n$

Booth Function :

$$F_2(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$$

$[-10,10]^2$

Ackley Function :

$$F_3(x)=20+e-20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}}-e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}$$

$[-32 , 32]^n$

Dixon and price Function :

$$F_4(x)=(x_i-1)^2+\sum_{i=0}^{n}i(2x_i^2-x_{i-1})^2$$

$[-10 , 10]^n$

Goldstein and Price Function:

$$F_5(x)=(1+(x_1+x_2+1)^2(19-14x_1+13x_1^2-14x_2+6x_1x_2+3x_2^2))*(30+(2x_1-3x_2)^2(18-32x_1+12x_1^2-48x_2-36x_1x_2+27x_2^2))$$

$[-2 , 2]^2$

Griewank Function :

$$F_6(x)=\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$$

$[-600 ,600]^n$

Levy function :

$$F_7(x)=\frac{\pi}{n}[10\sin^2(\pi x_1)+\sum_{i=1}^{n-1}((y_i-1)^2(1+10\sin^2(\pi y_{i+1})))+(x_n-1)^2]$$

$$y_i=1+\frac{1}{4}(x_i-1)$$

$[-10 ,10]^n$

Powell Function :

$$F_8(x)=\sum_{i=1}^{\frac{n}{4}}(x_{4i-3}+10x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2+(x_{4i-2}-x_{4i-1})^4+10(x_{4i-3}-x_{4i})^4$$

$[-4 , 5]^n$

Rastrigin Function :

$$F_9(x)=\sum_{i=1}^{n}(X_i^2-10\cos(2\pi x_i)+10)$$

$[-5.12,5.12]^n$

Rosenbrock Function :

$$F_{10}(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$$

$[-2.04,2.04]^n$

Sum squares function :

$$F_{11}(x)=\sum_{i=1}^{n}ix_i^2$$

$[-10 , 10]^n$

Trid Function :

$$F_{12}(x)=\sum_{i=1}^{n}(x_i-1)^2-\sum_{i=2}^{n}x_ix_{i-1}$$

$[-10 , 10]^n$

Zakharov function

$$F_{13}(x)=\sum_{i}^{n}x_i^2+(\sum_{i}^{n}\frac{1}{2}ix_i)^2+(\sum_{i}^{n}\frac{1}{2}ix_i)^4$$

$[-5 , 10]^n$

Table 4. Results of each functions

Functions		PSO	AWPSO	FOPSO
Sphere	Best	3.69	$1.12*10^{+1}$	$1.91*10^{-1}$
	Worst	$7.2\times10^{+2}$	$6.57\times10^{+1}$	8.88×10^{-3}
	Average	$1.69\times10^{+2}$	$3.24\times10^{+1}$	1.62×10^{-2}
Booth	Best	3.81×10^{-28}	0	0
	Worst	9.94×10^{-26}	0	0
	Average	2.88×10^{-25}	0	0
Ackley	Best	1.67	3.65	1.17
	Worst	3.83	8.86	2.69
	Average	2.35	6.37	2.08
Dixon and price	Best	2.24	$1.42\times10^{+2}$	7.60×10^{-1}
	Worst	$1.67\times10^{+3}$	$4.06\times10^{+3}$	6.98
	Average	$1.79\times10^{+2}$	$1.11\times10^{+3}$	3.66
Goldstein and Price	Best	3	3	3
	Worst	3	3	3
	Average	3	3	3
Griewank	Best	1.32×10^{-2}	4.98×10^{-1}	5.63×10^{-4}
	Worst	3.25×10^{-1}	8.76×10^{-1}	1.34×10^{-2}
	Average	7.95×10^{-2}	7.42×10^{-1}	6.42×10^{-3}
Levy	Best	1.73	13.23	1.96×10^{-1}
	Worst	$1.34\times10^{+1}$	23.47	2.35
	Average	5.34	18.18	6.25×10^{-1}
Powell	Best	2.80	$1.02\times10^{+1}$	6.64×10^{-1}
	Worst	$3.42\times10^{+3}$	$6.98\times10^{+1}$	3.98
	Average	$4.20\times10^{+3}$	$2.94\times10^{+1}$	1.96
Rastrigin	Best	$5.87\times10^{+1}$	$4.17\times10^{+1}$	$1.95\times10^{+1}$
	Worst	$1.08\times10^{+2}$	$1.84\times10^{+2}$	$4.45\times10^{+1}$
	Average	$8.98\times10^{+1}$	$8.95\times10^{+1}$	$3.04\times10^{+1}$
Rosenbrock	Best	$1.62\times10^{+1}$	$2.76\times10^{+1}$	$1.19\times10^{+1}$
	Worst	$1.12\times10^{+5}$	$6.38\times10^{+2}$	$7.94\times10^{+1}$
	Average	$1.35\times10^{+3}$	$2.99\times10^{+2}$	$3.46\times10^{+1}$
Sum squares	Best	2.42×10^{-3}	9.39×10^{-3}	2.82×10^{-1}
	Worst	11.68	5.70	3.94
	Average	4.28	1.64	1.50
Trid	Best	-740.23	-533.28	-163.45
	Worst	-170.89	-101.45	-156.58
	Average	-480.78	-267.27	-160.74
Zakharov	Best	22.53	5.58	1.97×10^{-2}
	Worst	314.97	87.85	1.29
	Average	$1.09\times10^{+2}$	$3.42\times10^{+1}$	5.88×10^{-1}

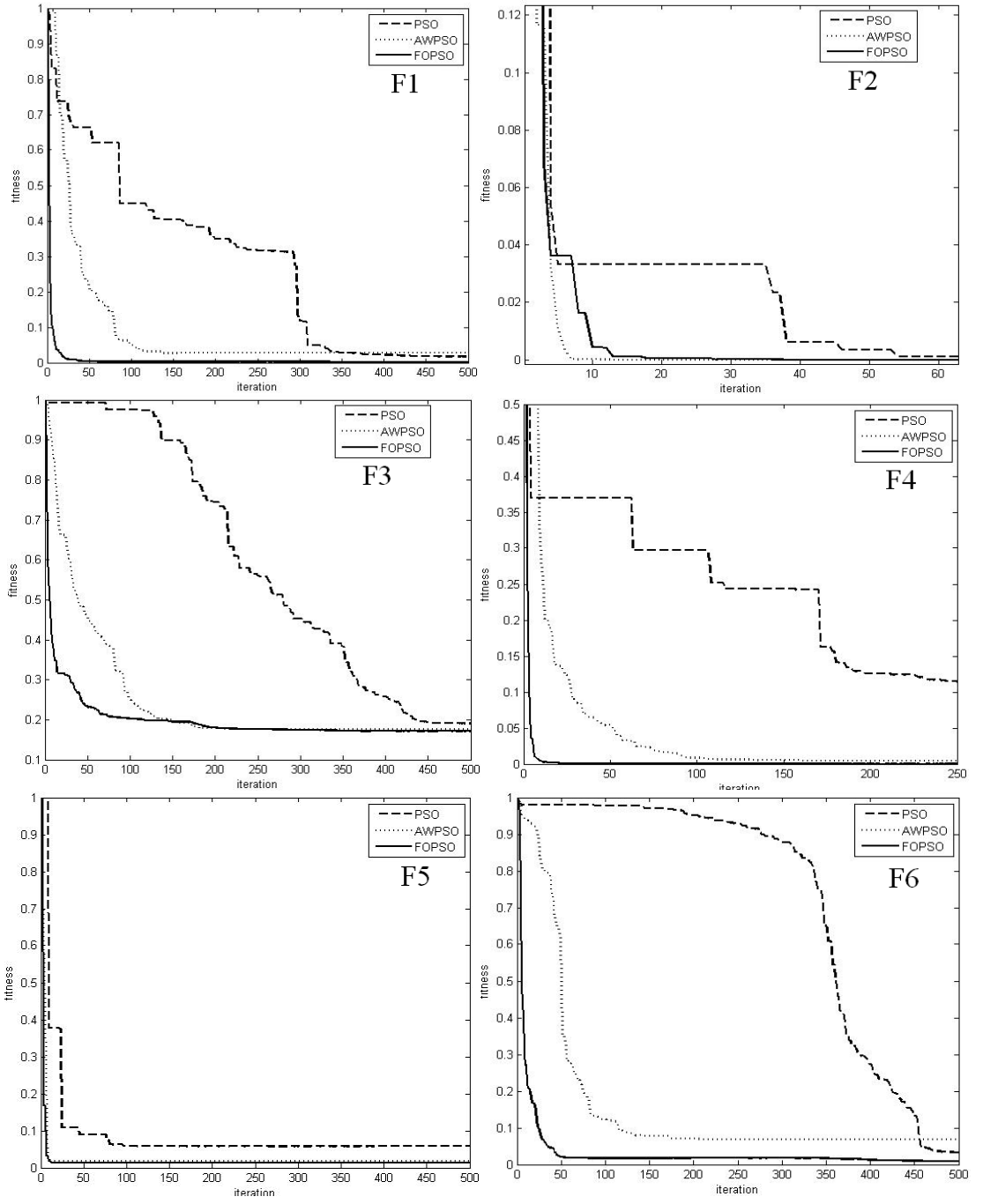
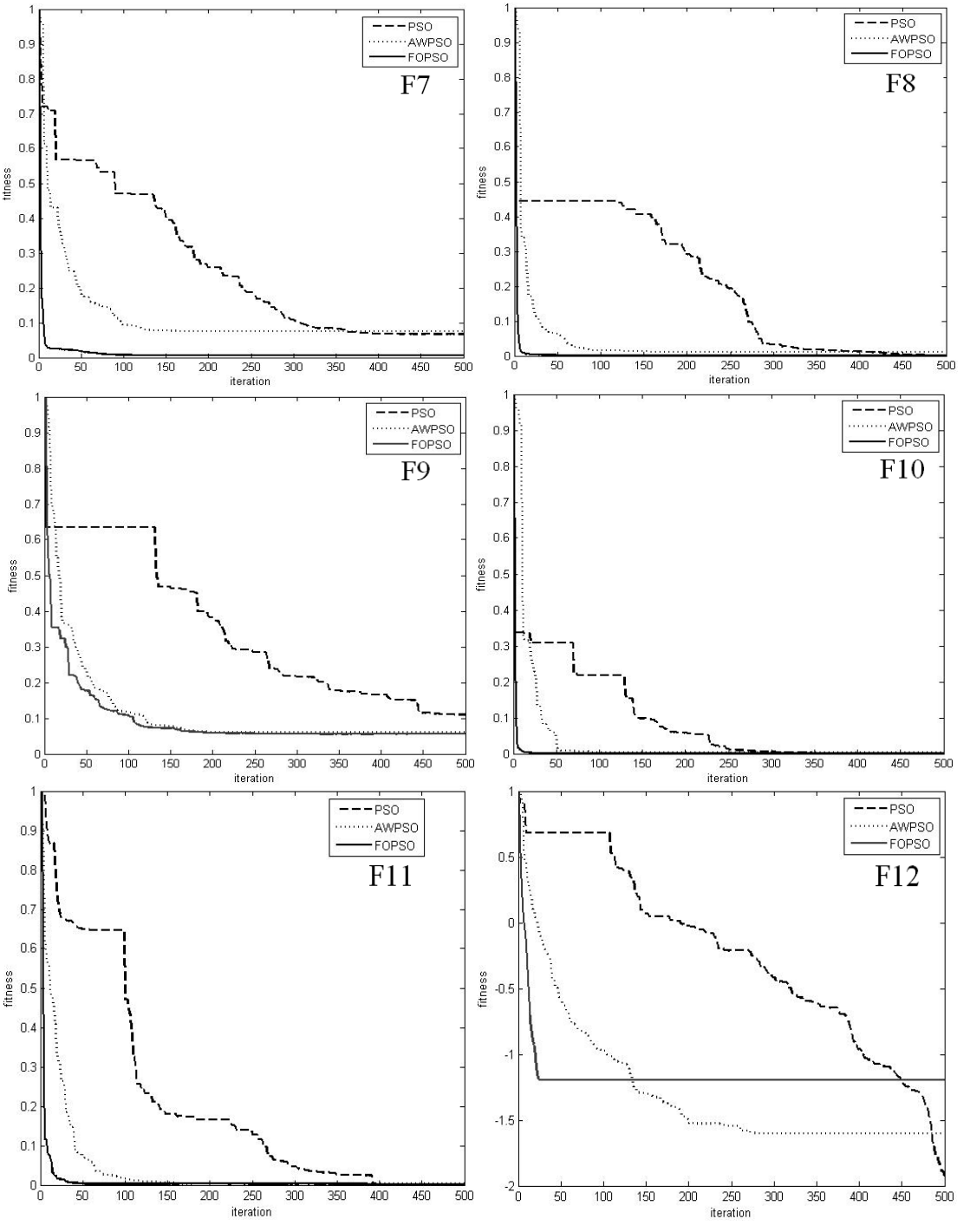
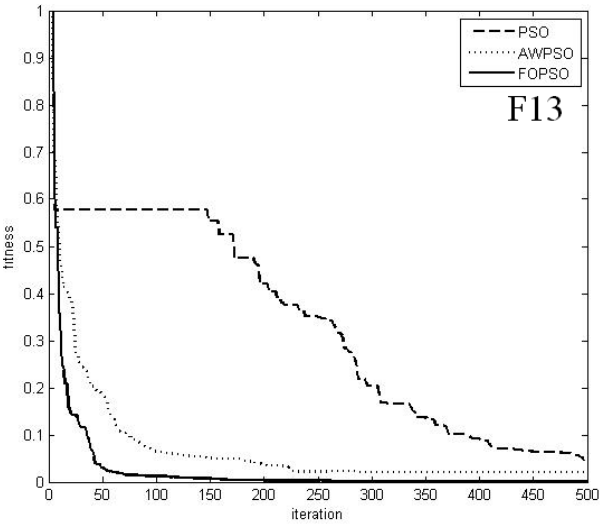


Fig 5. Comparing convergence the function optimization problems



Continue of Fig 5. Comparing convergence the function optimization problems



Continue of Fig 5. Comparing convergence the function optimization problems

7. CONCLUSIONS

In this paper, using an appropriate fuzzy logic, and control parameters are initialized such as weight, inertia and acceleration parameters. According to obtained Results in this study, the proposed algorithm shows good behavior in mathematical using. This algorithm has been successful in identifying a local optimum and it can avoid the jam in local optimum. Although the computational cost should be considered according to the applications, But in future, Using the indirect calculation of parameters using fuzzy logic can be use for other matters of computing applications with pso algorithm.

REFERENCES

[1] J.Kennedy, R. Eberhart, **"Particle swarm optimization"**., *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1995.

[2] H.Liu , A.Abraham. "**Fuzzy adaptive turbulent particle swarm optimization"** *IEEE*, 2005, pp. 39-47

[3] Mahfouf , M.Minyou-Chen , D. A. Linkens. **"Adaptive weighted particle swarm optimization (awpso) of mechanical properties of alloy steels"**. *8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, Birmingham (UK),2004.

[4] X. Hu , Y. Shi and R. Eberhart, **"Recent advances in particle swarm optimization"**, *IEEE*, 2004

[5] M. Abdelbar, S. Abdelshahid , Donald C. Wunsch II . **"Fuzzy PSO: a generalization of particle swarm optimization"**. *Proceedings of International Joint Conference on Neural Networks*, Montreal, Canada, July 31 - August 4, 2005.

[6] Y. Shi, R.C. Eberhart, **"Fuzzy adaptive particle swarm optimization"**, *Proc. of IEEE Congress on Evolutionary Computation*, vol. 3, pp. 1945-1950, (2001).

[7] M. Clerc ,J. Kennedy, **"The particle swarm: explosion, stability, and convergence in a multidimensional complex space"**, *IEEE Transaction Evolutionary Computation* 6. 2002, pp. 58–73.

[8] N.Jin , Yahya Rahmat-Samii. "**user's manual of ucla-pso algorithm (matlab version)"**, http://www.ee.ucla.edu/antlab .April 2007.

[9] Y. Shi, R.C. Eberhart, **"Fuzzy adaptive particle swarm optimization"**, *Proc. of IEEE Congress on Evolutionary Computation*, vol. 3, pp. 1945-1950, (2001).

[10] A. P. Engelbrecht, **"Foundamentals of Computational Swarm Intelligence"**, *John Wiley & Sons Ltd*, (2005).