Enhanced Comprehensive Learning Cooperative Particle Swarm Optimization with Fuzzy Inertia Weight (ECLCFPSO-IW)

Mojtaba Gholamian* , Mohammad Reza Meybodi

Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran

Abstract

So far various methods for optimization presented and one of most popular of them are optimization algorithms based on swarm intelligence and also one of most successful of them is Particle Swarm Optimization (PSO). Prior some efforts by applying fuzzy logic for improving defects of PSO such as trapping in local optimums and early convergence has been done. Moreover to overcome the problem of inefficiency of PSO algorithm in high-dimensional search space, some algorithms such as Cooperative PSO offered. Accordingly, in the present article, we intend, in order to develop and improve PSO algorithm take advantage of some optimization methods such as Cooperatives PSO, Comprehensive Learning PSO and fuzzy logic, while enjoying the benefits of some functions and procedures such as local search function and Coloning procedure, propose the Enhanced Comprehensive Learning Cooperative Particle Swarm Optimization with Fuzzy Inertia Weight (ECLCFPSO-IW) algorithm. By proposing this algorithm we try to improve mentioned deficiencies of PSO and get better performance in high dimensions.

Keywords: Particle Swarm Optimization, Cooperative PSO, Comprehensive Learning, Inertia Weight, Fuzzy Controller.

1. Introduction

The most common optimization methods are evolutionary algorithms that usually applied for solving difficult problems have not definite quick solution. So far many evolutionary algorithms are suggested for optimizing different problems that PSO is one of the most popular and the most efficient of them. Considering increasingly PSO's applications since its innovation until now, various versions and editions of it have been presented that besides enjoy its benefits try to improve its defect and weaknesses. In less than two decades, hundreds of articles have been published as a report on the application of PSO [1].

Moreover various problems with high complexity and high dimension environment exist and every day growing of these type problems continuing. Furthermore one of most important involvement of this kind of problems is overcoming their complexity and making more efficient existing algorithms in countering with them. Hence solutions for overcoming curse of dimension problem have presented that one of most famous is cooperative algorithms. On the other

* Corresponding author. Email: Mojtaba.Gholamian@qiau.ac.ir

hand objects related to the cooperative algorithm cause of possibility of utilizing benefits of parallel processing and parallel algorithms are paid attention. Consequently enhancement and increasing efficiency of this kind of algorithms like cooperative PSO could be helpful in solving some problems especially high dimension and complex problems.

By knowing the fact of setting up PSO's parameters have very significant impact on its efficiency, many efforts for setup parameters ideally, have been done. So in some articles such as [2], [3] and [4], researchers suggest linear decreasing of inertia weight from 0.9 to 0.4 while progressing of algorithm. In 2001, Shi and Eberhart, introduce adaptive Fuzzy PSO method[5]. Also for enhancement of PSO's performance in some articles e.g. [6],fuzzy logic is applied.

PSO's successes are wonderful. Less than two decades, hundreds of articles about applications of PSO have been published. PSO in many contexts such as finding optimums of functions, neural networks learning, Fuzzy systems control, clustering and classifications, biomedical, combinational optimization, control, design, distributed networks, electronics and electromagnetic, engines and motors, entertainment, faults, financial, graphics and visualization, image and video, antenna, modeling, prediction and forecasting, robotics, scheduling, security and military, sensor networks, signal processing, conclusions, power systems and plants and other problems witch Genetic Algorithm is successful, have good performance[1].

By knowing the fact of setting up PSO's parameters have very significant impact on its efficiency, many efforts for setup parameters ideally, have been done. So in some articles such as [2], [3] and [4], researchers suggest linear decreasing of inertia weight from 0.9 to 0.4 while progressing of algorithm. In 2001, Shi and Eberhart [5], introduce adaptive Fuzzy PSO method. Also for enhancement of PSO's performance in some articles e.g.[6],fuzzy logic is applied. One of main defects of optimization algorithms such as PSO is trapping in local minimums and this problem becomes more serious by increasing dimension of search space [7]. For countering this problem, so far revised models of PSO like cooperative PSO (CPSO) propose [8].

The paper is presented as follows: In the section 2 till VI component of proposed algorithm consist of PSO with fuzzy inertia weight, CPSO, Comprehensive Learning PSO (CLPSO) and applied methods for improving suggested algorithm described. In seventh section proposed algorithm presented and at the eighth section its evaluation has been done and the ninth section contains conclusion.

2. Fuzzy Particle Swarm Optimization

As we know in the original version of PSO, each particle faces with two mandatory moves, ones, attraction the best position so far particle achieved, and other attraction to the best position of particles group achieved. PSO include group of particles moving in multi dimension search space with real values of feasible solution problems. PSO can be easily implemented and have a low cost calculation. In another hand PSO in solving of many problems is efficient and in some cases, not involves with troubles of other evolutionary calculation techniques. Difficulty of PSO adjustment for achieving desired efficiency is one of its disadvantages and if we don't choose suitable parameters, it will be converged to local optimum. As this algorithm gradually converged to best solution found until now, and if this solution was local optimum, all particles will absorb into it and the standard PSO not prepare the solution to exit this local optimum. This is largest trouble of standard PSO that be inefficient for solving multimodal problems especially with large search space. Another standard PSO's trouble is early convergence in some problems. As mentioned, standard PSO algorithm trapped in local optimums and this problem becomes more serious in high dimension. For solving basic PSO, many solution such as combinational algorithms, have suggested. One sample of combine algorithm is FPSO (Fuzzy PSO) which is combining of Fuzzy logic and PSO. As shown in Fig.1one step before updating PSO, Fuzzy system determines parameters values for take apart with new defined values in updating. In this paper we use kind of FPSO, with a Fuzzy logic controller with an input and an output to PSO. The input parameter of fuzzy system controller is number of algorithm iteration and its output is inertia weight parameter. In this fuzzy system which its output is inertia weight, main idea of applied method is based on making balance between exploitation search and exploration search [9]. Sample of general fuzzy rules is as in (1).

[Rules]

$$\begin{aligned} &\text{If } Itr = L \text{ then } \omega = H \\ &\text{If } Itr = M \text{ then } \omega = M \\ &\text{If } Itr = H \text{ then } \omega = L \end{aligned} \tag{1}$$

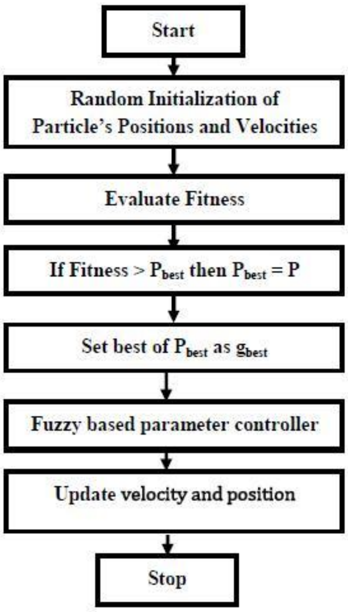

Fig. 1. Fuzzy Particle Swarm Optimization Diagram

3. Cooperative Fuzzy Particle Swarm Optimization

For overcome defects of PSO in high dimension search space, some algorithms such as Cooperative PSO (CPSO) presented. CPSO for counter problem of "Curse of Dimension" is used. In this algorithm swarm with high dimension is divided to swarms with smaller dimension and these swarms interchange information with each other for evaluation total value. In many cases this swarm with high dimension is divided to swarms with single dimension. Thus in cooperative method for solving a problem with D dimension instead of using swarm with S particle, we use D swarms with one dimension, each of them made of S particle. Global fitness function value is obtained from interpolation of all unique swarms Gbests and then combined fitness function is calculated. Important point is that only selecting best Gbest of each independent swarm for structuring combined vector of Gbests may be couldn't prepare best optimizing answer. Hence for cooperative PSO, evaluation of fitness has been done by introducing "Context Vector". We use abbreviation of CV for it. This vector implies cooperating between independent swarms. For solving a problem with D dimension, CV dimension also is D. Here, when for instance swarm of j is active, CV is configured by Gbest of D-1 swarms (which are considered as constant during evaluation of j's swarm) and the jth row of CV fill sequentially by each of jth swarm particles. Therefore CV is used for calculating combined fitness. So the answer of Pbest of ith particle and answer of jth swarm Gbest (shown by $x_j^{i,PBest}$ and x_j^{GBest}), defined by considering of CV's concept and not depend only performance of jth swarm. [8], [10].

4. Comprehensive Learning PSO

PSO Algorithm base on comprehensive learning is usually due to good performance on problems with complex multi-dimensional search space is known. Here, stagnation trouble that occurs because of early convergence could be controlled by this way that permits each particle define itself velocity (therefore itself position) according PBest of other particles. Consequently, this method helps to maintaining population diversity and subsequently solving early convergence trouble. Selecting particles that we consider for applying its PBest for updating velocity of given particle in the population on the following way:

Step 1: Produce a number in range [0,1], if this number is greater than P_c (that is defined as selection possibility) then particle uses itself PBest. Otherwise, particle uses another particle PBest (that will be selected by tournament selection method as is described in step 2 till 4) for updating itself velocity and position.

Step 2: leave the current particle, and select two particles randomly in population.

Step 3: Compare PBest value of these particles and select particles with higher quality.

Step 4: Particle its quality is better detected and selected for applying in current particle's velocity and position update. Thus velocity updating equation base on comprehensive learning is as (2):

$$v_j^i[t+1] = wv_j^i[t] + c_1 r_1\left(f_j^{i,PBest}[t] - x_j^i[t]\right) + c_2 r_2\left(x_j^{GBest}[t] - x_j^i[t]\right) \tag{2}$$

That f^i is PBest of particles which current particle should follows [11].

5. Coloning Procedure

When searching procedure not progresses for consequent iterations (or have negligible changes in improving value of fitness function) colonong procedure is activated. The act of procedure is as following: At the end of each generation progress status is checking. If the result has not any improvement rather than previous generation, this unsatisfied condition is counting by incremental counter. If this number counter received defined number i.e. for multiple sequent generation no improvement in global optimum gained, then by using elitism method some or percent of worst particles replaced with best particles. This procedure is efficient to prevent from slowing and stagnation of the search process. By using of this procedure, we impart exploitation search method advantage for achieve our aim, in addition we should be care by using correct and suitable percent or number of replaced particles and also appropriate value for counter prevent to infect the abuse of incorrect usage of exploitation search i.e. destroying population diversity and early convergence.

6. Local Search Function

Considering Coloning procedure (with elitism approach) we aimed change in stagnation status, in this section by using local search with exploration search approach and usage of mutation operator and balancing between exploitation and exploration and maintenance of population diversity, we try to find better answers. This function acting as following: when the condition of modifying the found global optimum so far prepared, by calling this function run exploration search around optimum point. If the modification condition met, some mutated versions of CV (best agent of each population) produce by defined mutation operator as (3):

$$New\_Position_i = Current\_Position + rand \tag{3}$$

In this equation $New\_Position_i$ is ith new produced position by mutation operator, Current_Position is the current CV position and rand is a vector that its arguments are produced randomly with normal distribution. Best produced position will be replaced with current CV position by elitism approach. Also this operator has a significant effect on prevent slowing and stagnation of search process.

7. Proposed Algorithm ECLCFPSO-IW

In this article we would like to introduce the new combinational algorithm that besides of improving shortcomings of its base algorithm PSO, i.e. not trapping in local optimum and prevent early convergence even in high dimension problems, in comparison of other popular algorithms has a better performance about accuracy and either speed. After variables definition, initializing first generation of population and dividing population to sub-populations equal number of dimensions, at each iteration in the suggested algorithm that its pseudo code in Fig.2 is presented, fuzzy inference system initializes inertia coefficient and then algorithm enters comprehensive learning simulation section. In this section updating particles velocities and positions by comprehensive learning method occurs. Then according produced values of particle's PBest, the Context Vector that applied in cooperative method update and evaluate and in continue PBest and GBest of particles dimension revised. Finally after running all iterations of algorithm, best gained GBest returned as algorithm output. Thus using fuzzy inertia weight coefficient at each sub-population in updating velocity of particles is for utilizing its benefits. It is necessary to tell calling of Coloning procedure happens after no improving or negligible changes at global optimum value in determined consecutive iterations. For more improving this optimization strategy, at the stagnation status (when for some consecutive iteration there is no improving in CV value) local search function also activated that search around the CV for searching better answers. At Fig. 2 pseudo code of proposed algorithm is presented.

8. Evaluation of Proposed Algorithm

In most cases, the analytical methods for solving optimization problems are not applicable; so many approximate methods for solving these problems have been proposed to approximate the optimums. Many of these methods have problems such as convergence to a local optimum and the slow speed convergence. Some methods due to their complexity and large space state are very suitable problems for benchmarking the ability of optimization algorithms.

Based on the characteristics of these functions, they will be divided into several groups. One classification is based on the optimums number of functions. Functions that have only one optimum in the space of problem are called "Unimodal Function" and functions with more than one optimum are called "Multimodal Functions".

Algorithm: Pseudocode for ECLCFPSO-IW

```
Environment Variable Definition
Global & Local Variables Definition
Search Domain Variable Definition Base on Benchmark Function
For Each Particle
    Initialize Particle
End For
For Each Iteration
    Evaluate W Coefficient for Current Iteration by Fuzzy Inference System (FIS)
    For Each Dimension   //Separate Swarms to Number of Dimension for Using Cooperative Method
        If Coloning Condition=True
            Coloning Procedure
        End If   //End If Coloning Condition=True
        For Each Particle
            // Comprehensive Learning Section
            Generate Seed Randomly Between 0 and 1
            If Seed <= Threshold
                Selecting Two Particles (P1,P2) from Swarm Randomly
                Selecting Better Particle Between (P1,P2) as PS by Tournament Selection Method
                Update Velocity of Each Dimension of Particles by Using PS PBest Instead of Particle PBest
                Update Position of Each Dimension of Particles by Using PS PBest Instead of Particle PBest
            Else
                Update Velocity of Each Dimension of Particles
                Update Position of Each Dimension of Particles
            End If   // End of If Seed <= Threshold
            Examine&Evaluate Context Vector (CV) Of Cooperative Method
            Update Personal Best (PBest) of Each Dimension of Particles
            Update Global Best (GBest) of Each Dimension of Particles By Local Search Around GBest
        End For   //End of For Each Particle
        Update Context Vector (CV) of Cooperative Method
    End For   //End of for Each Dimension
End For   //End of for Each Iteration
Display Results
Return GBest
```

Fig. 2. Pseudo Code of Proposed Algorithm (ECLCFPSO-IW)

Multimodal functions are used for measurement escape ability from local optimums. In cases exploration process of algorithm perform weak search and couldn't search entire problem space, will be trapped in thelocal optimums. From most famous of these functions are Sphere, Rosenbrock, Ackley, Griewank and Rastriginwhitch all of them have global optimum with zero value.

Sphere and Rosenbrock are sample of Unimodal functions and Ackley, Griewank and Rastrgin are in group of Multimodal functions. For evaluating proposed algorithm we could check their performance on benchmark functions and then compare it with other evolutionary algorithm about accuracy and speed of founding optimums in fair condition. At Table I. number of particles and number of iteration at each running for each dimension is presented. In addition for decreasing effect of accidental values on represented results, we run the algorithm 20 times independently and then extract the results. In Fig. 2 till Fig. 7 the result of comparing performance of proposed algorithm with three other evolutionary algorithms, in terms of fair comparison is shown. The results say the proposed algorithm is successful in founding global optimum at viewpoint of answer quality and either speed of convergence rather than three known evolutionary algorithms for their high performance and widespread application include PSO, GA (Genetic Algorithm) and ICA (Imperial Competition Algorithm). It is remarkable all three used optimization algorithms for evaluations are improved and evolved version of them. At table II. and III. applied value of these algorithms in evaluations are shown.

Further at tables IV till VII the result of running proposed algorithm on five benchmark functions in 20 times independent running of algorithms on particles with 10, 20, 30 and 90 dimensions presented. Performance of proposed algorithm in these tables with their base algorithm i. e. PSO and CPSO in five categories and the yield results are shown in scientific notations.

Table 1

Particle Dimension and Population Applied for Evaluation of Proposed Algorithm

Dimension Type	*Particle Dimension*	*Iteration Number at Each Running*	*Number of Running Proposed Algorithm Simulation on Benchmark Functions*	*Particles Population*
Low Dimension	10	1000	20	40
	30	1000	20	80
High Dimension	70	1000	20	80
	90	1000	20	120

Table 2

Value of Genetic Algorithm Applied in Comparitions

Parameter	*Value*
Crossover %	0.8
Mutation %	0.1
Mutation Rate %	1
Selection Procedure	Roulette Wheel
Iteration	1000

Table 3

Value of ICA Algorithm Applied in Comparitions

Parameter	*Value*
Number of Empires/Imperialists	10
Assimilation Coefficient (β)	2
Revolution Probability	0.1
Revolution Rate	0.05
Selection Pressure (α)	1
Colonies Mean Cost Coefficient (ζ)	0.1
Iteration	1000

Benchmark Function is Sphere							
Dimension is D=90		Dimension is D=70		Dimension is D=30		Dimension is D=10	
Average of GBest	Algorithm	Average of GBest	Algorithm	Average of GBest	Algorithm	Average of GBest	Algorithm
1.44E-253	ECLCFPSO-IW	4.42E-251	ECLCFPSO-IW	7.05E-254	ECLCFPSO-IW	2.72E-21	ICA
1.95E-07	ICA	2.04E-08	ICA	3.23E-17	ICA	4.00E-16	ECLCFPSO-IW
6.88E+00	GA	3.07E+00	GA	1.80E-03	PSO	4.40E-11	PSO
2.44E+02	PSO	2.34E+02	PSO	6.41E-02	GA	2.92E-03	GA

Fig. 3. Comparition of Gbest average values with three evoloutionary algorithm on Sphere benchmark function at 10, 30, 70 and 90 dimensions

Benchmark Function is Rosenbrock							
Dimension is D=90		Dimension is D=70		Dimension is D=30		Dimension is D=10	
Average of GBest	Algorithm	Average of GBest	Algorithm	Average of GBest	Algorithm	Average of GBest	Algorithm
5.97E+01	ECLCFPSO-IW	4.04E+01	ECLCFPSO-IW	9.13E-01	ECLCFPSO-IW	5.56E-03	ECLCFPSO-IW
2.43E+02	ICA	1.71E+02	ICA	3.04E+01	ICA	2.61E+00	ICA
4.27E+02	GA	3.34E+02	GA	7.70E+01	GA	7.23E+00	PSO
1.62E+03	PSO	1.11E+03	PSO	9.34E+01	PSO	7.90E+00	GA

Fig. 4. Comparition of Gbest average values with three evoloutionary algorithm on Rosenbrock benchmark function at 10, 30, 70 and 90 dimensions

Benchmark Function is Ackley							
Dimension is D=90		Dimension is D=70		Dimension is D=30		Dimension is D=10	
Average of GBest	Algorithm	Average of GBest	Algorithm	Average of GBest	Algorithm	Average of GBest	Algorithm
6.80E-14	ECLCFPSO-IW	5.47E-14	ECLCFPSO-IW	5.54E-10	ICA	7.64E-15	ECLCFPSO-IW
2.61E-04	ICA	5.27E-05	ICA	1.79E-02	GA	2.07E-13	ICA
1.26E-01	GA	1.03E-01	GA	6.21E-02	ECLCFPSO-IW	5.93E-03	GA
3.43E+00	PSO	3.54E+00	PSO	4.14E+00	PSO	7.23E-01	PSO

Fig. 5. Comparition of Gbest average values with three evoloutionary algorithm on Ackley benchmark function at 10, 30, 70 and 90 dimensions

Benchmark Function is Griewank							
Dimension is D=90		Dimension is D=70		Dimension is D=30		Dimension is D=10	
Average of GBest	Algorithm	Average of GBest	Algorithm	Average of GBest	Algorithm	Average of GBest	Algorithm
3.72E-02	ICA	4.43E-02	ICA	4.94E-02	ICA	3.03E-02	ECLCFPSO-IW
6.62E-02	ECLCFPSO-IW	9.06E-02	ECLCFPSO-IW	5.14E-02	GA	8.93E-02	ICA
2.51E-01	GA	2.00E-01	GA	5.95E-02	ECLCFPSO-IW	9.05E-02	GA
3.43E+00	PSO	2.80E+00	PSO	3.17E-01	PSO	1.78E-01	PSO

Fig. 6. Comparition of Gbest average values with three evoloutionary algorithm on Griewank benchmark function at 10, 30, 70 and 90 dimensions

Benchmark Function is Rastrigin							
Dimension is D=90		Dimension is D=70		Dimension is D=30		Dimension is D=10	
Average of GBest	Algorithm	Average of GBest	Algorithm	Average of GBest	Algorithm	Average of GBest	Algorithm
0.00E+00	ECLCFPSO-IW	0.00E+00	ECLCFPSO-IW	0.00E+00	ECLCFPSO-IW	0.00E+00	ECLCFPSO-IW
2.66E+01	GA	1.99E+01	GA	3.35E-02	GA	2.94E-26	ICA
1.78E+02	PSO	1.17E+02	ICA	1.64E+00	ICA	1.88E-03	GA
2.08E+02	ICA	1.50E+02	PSO	3.78E+01	PSO	9.20E+00	PSO

Fig. 7. Comparition of Gbest average values with three evoloutionary algorithm on Rastrigin benchmark function at 10, 30, 70 and 90 dimensions

Table 4

Comparition Results of Proposed Algorithm Running 20 Times on Five Benchmark Function with 10 Dimension particles

Function	Standard	PSO	CPSO	ECLCFPSO-IW
Sphere	Best	7.32E-25	1.46E-71	4.84E-259
	Average	4.40E-11	2.60E-70	4.00E-16
	Standard Deviation	1.92E-10	4.25E-70	1.74E-15
	First Iteration Yield Global optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Rosenbrock	Best	1.23E-01	4.51E-06	3.05E-03
	Average	7.23E+00	1.23E+00	5.56E-03
	Standard Deviation	1.39E+01	2.36E+00	8.49E-04
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Ackley	Best	1.32E-12	4.44E-15	4.44E-15
	Average	7.23E-01	1.47E-14	7.64E-15
	Standard Deviation	8.67E-01	6.82E-15	2.22E-15
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Griewank	Best	2.46E-02	0.00E+00	0.00E+00
	Average	1.78E-01	8.27E-02	3.03E-02
	Standard Deviation	1.38E-01	1.11E-01	4.59E-02
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	5	12
Rastrigin	Best	3.98E+00	0.00E+00	0.00E+00
	Average	9.20E+00	0.00E+00	0.00E+00
	Standard Deviation	4.38E+00	0.00E+00	0.00E+00
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	418	288
	Number of Running Yield Global Optimum	0	20	20

As is shown in tables IV till VII in most cases proposed algorithm rather than its base algorithms have improvement. Quantity of improvement depends on benchmark function is variant. Especially in Sphere benchmark function that is member of Unimodal benchmark functions and also is a steady function (Sphere function cause of having one optimum, is a standard for measuring of speed convergence to optimum) and Rastrigin is the member of Multimodal benchmark functions and is the complex function with many local optimums, by using proposed algorithm is resulted significant improvement rather than other three benchmark function i.e. Rosenbrock, Ackley and Griewank.

It is possible many of algorithms during test on Rastrigin function trapped in local optimums, then algorithms with ability of entire search or suitable exploration, could find better answers. Thus significant success of proposed algorithm on this benchmark function could be confirmation on effectives pacification of this algorithm about exploration and entire search.

Table 5

Comparition Results of Proposed Algorithm Running 20 Times on Five

Function	Standard	PSO	CPSO	ECLCFPSO-IW
Sphere	Best	1.06E-04	1.44E-74	2.50E-268
	Average	1.80E-03	6.53E-74	7.05E-254
	Standard Deviation	1.54E-03	3.77E-74	0.00E+00
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Rosenbrock	Best	3.05E+01	2.93E-05	5.42E-01
	Average	9.34E+01	8.82E-01	9.13E-01
	Standard Deviation	3.40E+01	9.34E-01	1.68E-01
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Ackley	Best	2.59E+00	2.22E-14	1.51E-14
	Average	4.14E+00	4.10E-14	6.21E-02
	Standard Deviation	8.53E-01	1.58E-14	2.71E-01
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Griewank	Best	1.14E-01	0.00E+00	0.00E+00
	Average	3.17E-01	2.06E-02	5.95E-02
	Standard Deviation	1.20E-01	3.00E-02	2.13E-01
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	8	12
Rastrigin	Best	2.61E+01	0.00E+00	0.00E+00
	Average	3.78E+01	0.00E+00	0.00E+00
	Standard Deviation	9.18E+00	0.00E+00	0.00E+00
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	377	272
	Number of Running Yield Global Optimum	0	20	20

Benchmark Function with 30 Dimension particles

Table 6

Comparition Results of Proposed Algorithm Running 20 Times on Five

Function	Standard	PSO	CPSO	ECLCFPSO-IW
Sphere	Best	7.49E+01	5.59E-71	3.94E+01
	Average	2.34E+02	3.78E-70	4.04E+01
	Standard Deviation	5.92E+01	2.15E-70	5.89E-01
	First Iteration Yield Global optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Rosenbrock	Best	6.39E+02	5.58E-10	3.94E+01
	Average	1.11E+03	8.74E-01	4.04E+01
	Standard Deviation	2.85E+02	1.51E+00	5.89E-01
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Ackley	Best	2.81E+00	5.77E-14	4.35E-14
	Average	3.54E+00	7.37E-14	5.47E-14
	Standard Deviation	5.07E-01	1.36E-14	8.87E-15
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Griewank	Best	1.86E+00	3.33E-16	6.89E-259
	Average	2.80E+00	4.35E-02	4.42E-251
	Standard Deviation	5.39E-01	1.04E-01	0.00E+00
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Rastrigin	Best	9.70E+01	0.00E+00	0.00E+00
	Average	1.50E+02	0.00E+00	0.00E+00
	Standard Deviation	2.89E+01	0.00E+00	0.00E+00
	First Iteration Yield Global Optimum(zero)in all Running	Doesn't Exist	391	278
	Number of Running Yield Global Optimum	0	20	20

Benchmark Function with 70 Dimension particles

About Ackley and Griewank Multimodal functions, even if proposed algorithm in many cases rather than other compared algorithm could get better answers, but this gained improvement rather than resulted improvement of running proposed algorithm on Sphere and Rastrigin is lower. Ackley function is Multimodal functions with one global minimum optimum in very narrow valley and several local minimum optimums and considering its local minimums are not very deep then getting away from them could be done easily.

Table 7

Comparition Results of Proposed Algorithm Running 20 Times on Five Benchmark Function with 90 Dimension particles

Function	Standard	PSO	CPSO	ECLCFPSO-IW
Sphere	Best	1.70E+02	6.44E-72	1.21E-261
	Average	2.44E+02	1.44E-71	1.44E-253
	Standard Deviation	4.52E+01	6.36E-72	0.00E+00
	First Iteration Yield Global optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Rosenbrock	Best	9.76E+02	1.70E-06	5.83E+01
	Average	1.62E+03	6.06E-01	5.97E+01
	Standard Deviation	4.31E+02	1.39E+00	6.29E-01
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Ackley	Best	2.91E+00	6.84E-14	5.06E-14
	Average	3.43E+00	9.49E-14	6.80E-14
	Standard Deviation	3.17E-01	1.65E-14	1.34E-14
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Griewank	Best	2.91E+00	4.44E-16	4.44E-16
	Average	3.43E+00	2.82E-02	6.62E-02
	Standard Deviation	3.17E-01	4.63E-02	2.66E-01
	First Iteration Yield Global Optimum(zero) in all Running	Doesn't Exist	Doesn't Exist	Doesn't Exist
	Number of Running Yield Global Optimum	0	0	0
Rastrigin	Best	1.22E+02	0.00E+00	0.00E+00
	Average	1.78E+02	0.00E+00	0.00E+00
	Standard Deviation	2.72E+01	0.00E+00	0.00E+00
	First Iteration Yield Global Optimum(zero)in all Running	Doesn't Exist	379	262
	Number of Running Yield Global Optimum	0	20	20

9. Conclusion

The Proposed algorithm ECLCFPSO-IW is combination of algorithms consisting of Fuzzy Particle Swarm Optimization (FPSO), Cooperative Particle Swarm Optimization (CPSO), Comprehensive Learning PSO (CLPSO), local search function and Coloning procedure. There for in this paper we explain components of proposed algorithm in sections and then describe the method of combination of them for forming it. In the structure of proposed algorithm, we add concept of coefficient fuzzy inertia weight that used fuzzy inference system (FIS) for set up the inertia weight parameter adaptively which presented as FPSO with CLPSO for updating velocities and positions of particles to CPSO. By this method we interest advantages of FPSO such as not trapping in local optimums and escaping early convergence and either benefits of CPSO consist of overcoming problems with high dimension and possibility of applying parallel processing gains also advantages of CLPSO contain countering complex multi dimension problems and resistance of stagnation trouble, together.

Considering yielded result, we could inference performance of proposed algorithm in low and high dimensions are suitable and in most times has priority to other compared algorithm. This algorithm besides improving defects of its base algorithm i. e. PSO, rather than compared popular evolutionary algorithm have good performance in accuracy and searching speed of optimums and we could apply it in common usage fields of evolutionary algorithms especially for complex environment and high dimension.

References

[1] R. Poli, “An Analysis of Publications on Particle Swarm Optimization Applications”, *Artificial Evolution and Applications*, Vol. 1, pp. 1–10, 2008.

[2] D. P. Tian, N. Q. Li, “Fuzzy Particle Swarm Optimization Algorithm”, *International JointConference on Artificial Intelligence*, pp. 263-267, 2009.

[3] X. Meng, L. Jia, “A New Kind of PSO Convergent Fuzzy Particle Swarm Optimization and Performance Analysis”, *4th International Conference on Networked Computing and Advanced Information Management*, pp. 102- 107, 2008.

[4] A. M. Abdelbar, S. Abdelshahid and D. C. Wunsch, “Fuzzy PSO: A Generation of Particle Swarm Optimization”, *In Proceeding of International Joint Conference on Neural Networks*,pp. 1086–1091, 2005.

[5] Y. Shi and R. C. Eberhart, “Fuzzy Adaptive Particle Swarm Optimization”, *In Proceedings of Congress on Evolutionary Computation*, Vol. 1, pp. 101–106, 2001.

[6] M. H. Noroozibeyrami and M. R. Meybodi, “Improving Particle Swarm Optimization Using Fuzzy Logic”,

InProceedings of the Second Iranian Data Mining Conference, Amir Kabir University of Technology, Tehran, Iran, Sept. 21–22, 2008.

[7] I. Boussaid, J. Lepagnot and P. Siarry, “A Survey on Optimization Metaheuristics”,*Information Sciences,* Vol. 237, pp. 82–117, 2013.

[8] R. C. Eberhart and J. Kennedy, “A New Optimizer Using Particle Swarm Theory”, *In Proceedings of IEEE 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, 1995.

[9] S. Kumar, D. K. Chaturvedi, “Tuning of Particle Swarm Optimization Parameter Using Fuzzy Logic”, *International Conference on Communication Systems and Network Technologies*, 2011.

[10] M. A. Potter and K. A. de-Jong, “A Cooperative Coevolutionary Approach to Function Optimization”, *3rd Parallel Problem Solving from Nature Springer*, pp. 249–257, 1994.

[11] J. J. Liang, A. K. Qin, P. N. Suganthan and S. Baskar, “Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions”, *IEEE Transactions on Evolutionary Computation*, Vol. 10, pp. 935–952, 2006.