2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Adaptive Clubs-based Particle Swarm Optimization

Hassan M. Emara

Abstract – **This paper introduces a new dynamic neighborhood network for particle swarm optimization. In Club-based Particle Swarm Optimization (C-PSO) algorithm, each particle initially joins a default number of social groups (clubs). Each particle is affected by its own experience and the experience of the best performing member of the social groups it is a member of. In the proposed Adaptive membership C-PSO (AMC-PSO), a time varying default Membership is introduced. This modification enables the particles to explore the space based on their own experience in the first stage, and to intensify the connections of the social network in later stages to avoid premature convergence. This proposed dynamic neighborhood algorithm is compared with other PSO algorithms having both static and dynamic neighborhood topologies on a set of classic benchmark problems. The results showed superior performance for AMC-PSO regarding its ability to escape from local optima, while its speed of convergence is comparable to other algorithms.**

I. INTRODUCTION

PARTICLE Swarm Optimization (PSO) is a computational intelligence method for solving global optimization problems. It was originally proposed by J. Kennedy as an emulation of the behavior of birds swarms and fish school while searching for food. It was introduced as an optimization method in [1].

Compared to many other evolutionary computation (EC) techniques, PSO is inspired not by the evolutionary mechanism encountered in natural selection, but rather by the social behavior of flocking organisms.

A more broad perception of the swarm as a group of particles, whether birds, humans, or any socializing group of particles began to emerge. Club-based PSO (C-PSO) is proposed in [2]. In the basic C-PSO algorithm, a set of social groups (clubs) is created. Each particle can join one or more clubs. Each club can accommodate any number of particles. Particles' motion is affected by other particles that are members of the same clubs. The disadvantage of this technique is that its performance depends on the values of several constants that need to be tuned empirically for each problem.

In this paper we propose and study a new dynamic version of C-PSO, where the social network is forced to have a loose structure in the first stages. The club structure is modified during the optimization iterations to consolidate the social network and improve the convergence property in later stages. This modification aims to enhance the C-PSO performance and decrease its dependence on the empirical parameter values.

II. PARTICLE SWARM OPTIMIZATION

In their first paper [1], Kennedy and Eberhart introduce the concept of PSO. They propose that the motion of each bird in the swarm is guided by a social component and a particle self experience component. The behavior of the particles is described by (1) and (2).

$$v_{id}(t+1) = w \times v_{id}(t) + lrn_1 \times rand_1 \times \left(p_{id}(t) - x_{id}(t) \right) + lrn_2 \times rand_2 \times \left(p_{gd}(t) - x_{id}(t) \right) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2)$$

In (1), v_{id} (x_{id}) is the speed (position) of i^{th} particle in the d^{th} dimension. The first right hand side term corresponds to the inertia force. The momentum gain w is first introduced in [3]. The second term corresponds to the cognitive or personal experience component. It attracts the particle towards p_{id} which is the best position found by the i^{th} particle in the d^{th} dimension. The third term corresponds to the social influence of the neighbors on the particle. It attracts the particles to the best position found by its neighbors p_{gd}. These influences are controlled by learning weights lrn_1 and lrn_2. The terms $rand_1$ and $rand_2$ are uniformly distributed random variables in the range (0, 1). The convergence property of PSO is investigated in [4]-[7].

In [8], An adaptive momentum version is proposed. The momentum term w is linearly decreased as the iteration number increases. Similarly, the adaptation of the learning rates lrn_1 and lrn_2 is proposed in [9]. The algorithm starts with high lrn_1 and low lrn_2. These values are changed linearly with the iteration number until reaching a low lrn_1 and a high lrn_2 .This algorithm results in a swarm with "free" particles in the first phase "exploration phase", while attracting the particles to the best found zone in a later stage to achieve better exploitation. In [10], equation (1) is modified by adding an extra term that attracts each particle to the particle that maximizes the fitness to distance ratio.

In [11] the fully informed particles swarm (FIPS) is introduced. In this algorithm, each particle is attracted to all other particles within its neighborhood by forces of magnitudes that depend on the particle fitness.

There are several configurations for the neighborhood of a particle [12]-[13]. These configurations define which particles in the swarm affect a given particle, i.e. they define the social network of each particle. The social networks can be generally classified as static or dynamic.

978-1-4244-4524-0/09/$25.00 ©2009 AACC

A. Static social network

Static networks are the simplest version of the PSO social networks. If a particle '*a*' is considered in the neighborhood of a particle '*b*', this relation can be graphically indicated by an undirected edge connecting two vertices labeled '*a*' and '*b*' that represent the particles.

In the basic algorithm proposed in [1], each particle is affected by all other particles in the swarm. The social network of the PSO is represented by a fully connected graph (fig 1.a). The particles are attracted to a single point P_{gb} that represents the best point discovered by all members in the swarm so far. This may result in premature convergence.

To avoid such premature convergence, the algorithm is modified in order to use several attractors by assuming that each bird interacts only with a specific number of neighboring particles. This decreases the information change over the social network. In this case, equation (1) is modified to be:

$$v_{id}(t+1) = w.v_{id}(t) + lrn_1.rand_1.\left(p_{id}(t) - x_{id}(t)\right) + lrn_2.rand_2.\left(p_{lid}(t) - x_{id}(t)\right) \quad (3)$$

where p_{lid} denotes the d^{th} component of the best position found by the particles in neighborhood of particle *i*.

Fig. 1.b presents the ring social network proposed for the PSO. In this algorithm, the particles are arranged in an imaginary ring and each particle is connected to its immediately preceding and succeeding particles in this ring. The resulting PSO will be referred to as PSO-l where "l" stands for local [14].

In [11], the four clusters configuration (shown in fig. (1.c)) is investigated. In this case the swarm is subdivided into 4 nearly isolated groups. Particles within each group are fully connected. Sociologically, it represents 4 mostly isolated communities where few individuals have an acquaintance outside their group.

In [15], a hierarchical PSO (H-PSO) social structure is proposed . In this structure, the social graph is represented by a rooted tree as indicated in fig (1.d). Each particle is neighbored to itself and to its parent in the hierarchy. This structure has a small diameter.

The speed of information spread can be investigated by considering the graph radius and the number of links (edges). Large number of edges, high vertices degree and small radius reveal fast information exchange between the swarm. This may result in premature convergence. Small number of edges, low vertices degree, and large radius lead to better exploration and slower conversion. Table I summarizes the graphical social networks properties of the considered four PSO variants. For a swarm of 20 particles, the vertex degree of the mesh is 19 indicating intensive information exchanges between particles. The ring has a fixed degree of 2 indicating slow spread of information. The four-clusters has a vertex degree of 4 or 5 (depending on the particle position in the social network) indicating that the rate of information exchange is lower than that for a mesh and higher than that of a ring.

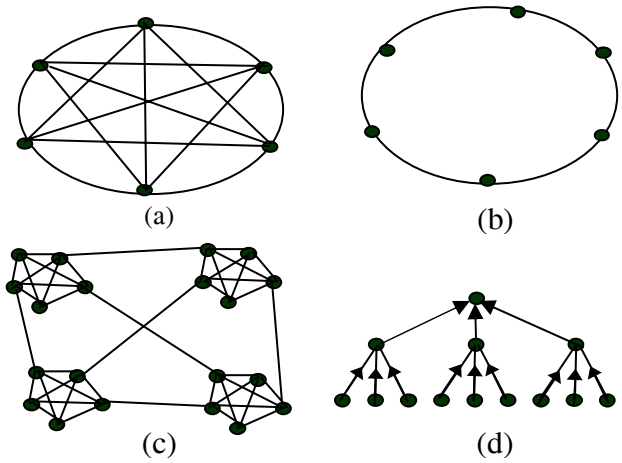

Fig. 1 PSO's static social networks topologies

B. Dynamic social network

Several researchers propose the use of dynamic social network having small number of edges initially and increasing the number of edges gradually. In [16], Suganthan proposes expanding the social network of the PSO from an initial simple graph where each particle is connected to itself to a fully connected graph. In [17], Mohais and Mendes use a randomly generated directed graph to represent the social network of FIPS. Two methods are presented for modifying the neighborhood structure. In the 'random edge migration', a random edge is disconnected from a vertex and connected to another neighbor at predefined interval. In the 'neighborhood restructuring', the social structure is re-initialized periodically after a specified number of iterations to preserve diversity.

In [15], an adaptive version of the H-PSO is proposed. In this version, the number of children for each parent (branching degree) is changed dynamically. Small branching degree increases the graph radius and is helpful in the exploration phase. Large branching degree helps swarm convergence as it reduces the graph radius, hence consolidating the effect of the best particle on all other particles in the swarm.

III. CLUB-BASED PSO (C-PSO)

Several algorithms that partition the swarm into several groups or clusters are proposed. In [18], Kennedy proposes to modify the basic swarm algorithm by applying clustering to identify different groups in the swarm. The centroid of each cluster is considered as the "stereotype" which the particles in that cluster would look at. Two algorithms are proposed. In the first one, the stereotype replaces the particle personal best, while in the second algorithm, it replaces the neighborhood best.

TABLE I
PSO SOCIAL NETWORKS PROPERTIES

Topology	Diameter	Vertex Degrees
Mesh	1	n-1
Ring	n/2	2
4 clusters	3	n/4 or (n/4)-1
Undirected tree	4	1, 4, or 3
Directed tree		In-degree:0 or 3 Out-Degree: 0 or 1

In [2], the social groups are called clubs following the social clubs, where peoples meet each other and possibly exchange their experience. These clubs are used to define the social network in the basic C-PSO algorithm. Each club can accommodate any number of particles. A particle in a particular club is neighbored by all particles members in the same club. Each particle can join one or more clubs. The membership degree of particle i (denoted by $m(i)$) is the number of clubs that this particle is a member of.

At every step, the particles' positions and speeds are updated following (2) and (3). The neighborhood of the i^{th} particle is the set of all particles included in the clubs that the i^{th} particle is a member of.

This form of C-PSO can be considered as a generalization of the static social networks presented in the previous section. For example, considering fig (1), we have:

- The fully connected network corresponds to a single club and a membership degree equal to 1 for all particles.
- The ring network of 6 particles correspond to 6 clubs each one having two members. The membership degree of all particles is equal to 2.

Similarly, it can be simply shown that the 4 clusters and the undirected H-PSO can be easily represented by C-PSO. It should be noted, however, that the H-PSO proposed in [15] is based on directed graphs, which is not supported by C-PSO.

A more sophisticated version of C-PSO is proposed in [19]. This version of C-PSO is denoted by JLC-PSO where 'JL' stand for the join and leave operations. The basic idea is presented in Algorithm A1 below. Simulation results show that the performance of JLC-PSO is better than C-PSO [19]. However, the main drawback of JLC-PSO is that its performance is very sensitive to the algorithm parameters (specially w and rr). These parameters need to be tuned for each specific optimization problem.

Algorithm A1: Join-Leave C-PSO

Initially:
- Each particle joins randomly a default number of clubs (= default membership degree dm).

Every iteration:
- The particles showing best performance in their clubs are enforced to leave a random club. Hence the number of particles attracted by the best particles is decreased, leaving more particles free to explore the space.
- Particles that show worst performance in their clubs are encouraged to learn from others by increasing their membership degree (joining more clubs).

Every rr (retention ratio) iterations:
- Particles that are not showing extreme performance are returned gradually to the default membership degree dm.

III. PROPOSED ADAPTIVE DEFAULT MEMBERSHIP CLUB BASED PSO (AMC-PSO)

In the basic C-PSO and in JLC-PSO, the default membership degree of the particles is kept constant. Particles with extreme performance (either best or worst) are allowed to deviate temporally from the default membership degree in JLC-PSO. However, they are returned back to the original membership degree as soon as they stop showing extreme performance.

In this paper, a new algorithm of C-PSO is proposed. In this algorithm, the value of the default membership dm degree is changed dynamically.

To study the effect of the default membership degree , let us consider a swarm of n birds, a default membership degree dm and c clubs. If a particle is not showing an extreme performance, then its membership degree $m(i)$ equals dm. If we assume that the swarm is consistent, then we have:

$$\sum_{i=1}^{n} m(i) = n.dm \tag{4}$$

where $m(i)$ is the membership degree of particle i.
The average number of members in each club is denoted by mc and is given by:

$$mc = \frac{n.dm}{c} \tag{5}$$

If $n.dm/c > 1$, the number of edges inside a club with mc members is given by $mc*(mc-1)/2$. Hence, the total number of edges in the swarm is approximately:

$$N\_edges = \frac{c.mc.(mc-1)}{2} \tag{6}$$

The average degree of the vertices is given by

$$vd = \frac{dm.(n.dm - c)}{c} \tag{7}$$

The diameter of the network cannot be easily computed due to the random nature of the particle distribution among the clubs. However, it should be noticed that if the default membership degree dm , the number of clubs c , and the number of particles n satisfy $dm.c < n$ $dm.n < c$, then the resulting social graph may not be connected. This means that, in this case, some particles are totally isolated from other swarm members and their experiences.

Algorithm A2: Proposed Adaptive C-PSO

```
begin
Initialize particles and clubs
dm =2;
while (termination condition = false) do
   evaluate particles fitness: f(x)
   update clubs' best particles list
   update particles speed and positions
            if (iteration mod join_leave_rate = 0)
            call procedure update membership level
            end-if
  if (iteration mod dm_update = 0)
     dm = dm + 1
  end-if
 iteration = iteration + 1
end-while

Procedure update membership level
for j = 1 to number of particles
   if (particle_j is best of neighbors_j) and
        (|membership_j| > min_membership)
      leave random club
   end-if
   if (particle_j is worst of neighbors_j) and
        (|membership_j| < max_membership)
      join random club
   end-if
   if (|membership_j| > dm) and (iteration mod rr = 0)
      leave random club
   end-if
   if (|membership_j| < dm) and (iteration mod rr = 0)
      join random club
   end-if
next j
end-procedure update membership level
```

It is clear that the default membership degree *dm* plays a great role in determining the C-PSO behavior. A small value of *dm* results in a low vertices degree. Hence, the social network is closer to the ring structure. If the value of *dm* becomes larger, the social network has a larger vertices degree, which is a characteristic of the mesh structure. Hence by varying *dm*, C-PSO can behave like PSO-l or PSO-g.

In the proposed adaptive C-PSO, the default membership degree is increased as the number of iteration increases. Hence the resulting C-PSO has the advantage of good exploration in the first stages as in PSO-l and good convergence in later stages of PSO-g. The proposed AMC-PSO algorithm is summarized in *Algorithm A2* above, where $\texttt{neighbors}_i$ is the set of particle i neighbors, $\texttt{membership}_i$, $|\texttt{membership}_i|$ are the set of clubs that particle i is a member of and the size of this set respectively, *join_leave_rate, dm_update* and *rr* are user defined parameters that controls the update intervals of the membership of the best and worst particles, the update of the default membership degree and the rate of returning of non-extreme (neither best nor worst) particles to the default membership degree *dm*.

Table II BENCHMARK FUNCTIONS AND ASSOCIATED PARAMETERS

Function	Dim.	Init. range	Objective
Sphere	30	$[-100; 100]^n$	0.01
Rosenbrock	30	$[-30; 30]^n$	100
Rastrigin	30	$[-5.12; 5.12]^n$	100
Griewank	10	[-600,600]	0.1
Schaffer's f6	2	$[-100; 100]^n$	1e-5
Achley	30	$[-32; 32]^n$	0.1

IV. EXPERIMENTS

In this section, the performance of the AMC_PSO is analyzed and compared to other PSO algorithms. Six well known benchmark problems presented in Table II are used for performance evaluation. The first two functions are simple unimodal functions. They test the ability of the optimizers to deal with smooth landscapes. The next four functions are multimodal functions.

Two metrics are used to compare the performance of different optimizers as in [15]. The first one is the ability to escape local minima. It is measured by the closeness of the achieved solution after a specific number of iteration to the global minimum. The second one is the convergence speed of the optimizer. It is measured by the required number of iterations to achieve a certain degree of closeness to the global optimum in the evaluation space.

Using these metrics on the six benchmark functions, we compare the performance of the AMC-PSO, C-PSO with join-leave , PSO-g ,PSO-l and four clusters algorithms.

For all simulation runs we use $lrn_1 = 1.494$, $lrn_2 = 1.494$, as in [15] and [4]. A swarm of 20 particles is used for all simulation runs.

In both JLC-PSO and AMC-PSO, 100 clubs are used. The retention ratio *rr* is set to 2. In JLC_PSO, the default membership degree is set to 10, whereas in AMC_PSO, the default membership degree is increased (as a linear function of the iteration number) from 2 to 80. When the default membership degree is set to 2, the number of edges and the average vertex degree of AMC-PSO is lower than that of a ring network. Hence, the particles are more independent in searching the space during first iterations.

The particles' positions and speeds are randomly initialized in the ranges shown in Table II depending on the benchmark problem used. The absolute speed values for particles are kept within the V_{max} limit for all dimensions during simulation. On the other hand, the particles' movements are not restricted by any boundaries, so particles may go beyond the initialization range and take any value.

Each simulation run is allowed to go for 20000 iterations, and each simulation has been repeated 50 times.

TABLE III
DISTANCES TO GLOBAL OPTIMA AFTER 20000 ITERATIONS

	PSO-l	PSO-g	4.Clusters	JLC-	AMC.PSO
Sphere	1.08e-47	1.51e-89	4.10e-42	**3.1e-180**	5.25e-159
Rosenb.	15.788	27.763	38.667	10.328	**9.9745**
Rastrigin	32.694	**18.287**	**19.163**	41.331	29.053
Griewank	3.86e-2	6.63e-2	4.76e-2	7.91e-2	**3.3e-2**
Schaf. f6	**0**	1.55e-3	5.83e-4	2.91e-3	**0**
Achley	6.71e-15	6.36e-15	6.43e-15	1.66e-1	**5.86e-15**

V. RESULTS

Each graph presented in this section represents the average of the 50 independent simulation runs for all optimizers unless otherwise stated.

A. Escaping Local Minima:

As shown in Fig. 2, for the Sphere problem, club-based PSO versions manage to finish closer to the unique minimum than the conventional PSO. JMC-PSO is the best, followed by AMC-PSO.

For the Rosenbrock problem presented in fig.3, AMC-PSO and JMC-PSO show very close performance at the final stages of the iterations. PSO-l and PSO-g follows them by a short distance, whereas the 4-clusters is the worst of all.

For multimodal problems, fig. 4 shows the optimizers' performance for the Rastrigin test problem. PSO-g is the best performer followed by the 4-clusters. AMC-PSO is the third followed by PSO-l. JMC-PSO converges prematurely and achieves the worst distance to global optimum.

As shown in fig. 5 and fig. 6, for both Griewank and Schaf. f6 problems, the performance of the optimizers is nearly the same. In both cases, the AMC-PSO is the closer to the global optimum followed by the PSO-l. The 4 clusters algorithm comes third, followed by PSO-g. In both cases, JMC-PSO is the worst performer.

For the Ackley problem, only JMC-PSO converges prematurely and hence is the worst performer. All other versions of PSO are so close at the end of the iterations. AMC-PSO is again the best performer. The distances to the global optima after 20000 iterations of the optimizers are shown in Table III.

B. Convergence Speed:

The second criterion to be considered is the convergence speed of the algorithms. It is measured by the number of iterations the algorithm takes to reach a certain degree of closeness to the global optimum. For the tested 6 benchmark problems, the closeness values are selected as indicated in Table II next to the problem names.

Table IV represents the average, the median, the maximum and the minimum of the closeness values. Data of successful runs were used to evaluate these values, so the sample number is not the same for all figures. The success rate of 50 independent simulation runs for the six optimizers is presented in the last column.

We can see from Table V that AMC-PSO is the only algorithm that achieves 100% success over all the six test problems. For the Sphere problem, all the algorithms achieve the desired closeness in every single run, though JLC-PSO comes ahead of them. For the unimodal Rosenbrock and the multimodal Rastrigen and Ackley problems, the JLC-PSO achieves the fastest approach to the selected closeness values, however the success rate is lower than that of the AMC-PSO which is the second best performer in the Ackley case and the 4th in the other 2 cases. The performance of the PSO-g precedes that of the 4 clusters and the PSO-l. The 4 clusters is the worst performer in the Rosenbrock and the Ackley cases while PSO-l is the worst performer for the Rastrigen problems.

In both Griewank and Schaffer's f6 problems, the AMC-PSO shows faster response compared to its response in the other problems. For example, it achieves the best Max values and it comes second in several indices.

C. Performance analysis of AMC-PSO in early stages:

As discussed in section IV, the performance of the AMC-PSO can be split into the exploration phase in the first stage (small *dm*), and the exploitation phase (high *dm*). If we consider for example, fig. 7, it is clear that in the first 1000 iterations, the best value found by the AMC-PSO is worse than that found by JLC-PSO and by PSO-g. The zoomed view of fig. 3 indicates the same performance of the optimizers in the first 800 iterations for the Rosenbrock problem. This may be interpreted by considering the properties of the JLC-PSO and PSO-g graphical networks. Based on (7), the average vertex degree for the JLC-PSO is 10, while the vertex degree of the PSO-g is 19. This relatively high vertex degree enables any particle located close to a local or global optimum to attract more particles to exploit this area. Hence, it is expected that both optimizers perform better in the first few iterations. In the first 800 iterations, the average membership *dm* degree of the AMC-PSO is increased from 3 to 6. For a value of *dm* less than or equal to 5, eq. (7) results in a negative or zero average vertex degree, indicating that the social network is 'loose', i.e. the particles are mainly guided by their own experience. Although this results in a poor performance in the first stages, it is clear from Table IV that this weak social behavior in the beginning results in an improved closeness to the global optimum

The above discussion demonstrates the ability of club based PSO to provide better response compared to conventional PSO. JLC-PSO provides the fastest convergence speed. The disadvantage of the JLC-PSO is that it may converge to local minimum as indicated by the lower success rates and high distance from global optimum in several problems. The advantages of the AMC-PSO can be summarized as follows:

- Providing the best closeness values for most test problems
- Providing the highest success ratio for the test problem

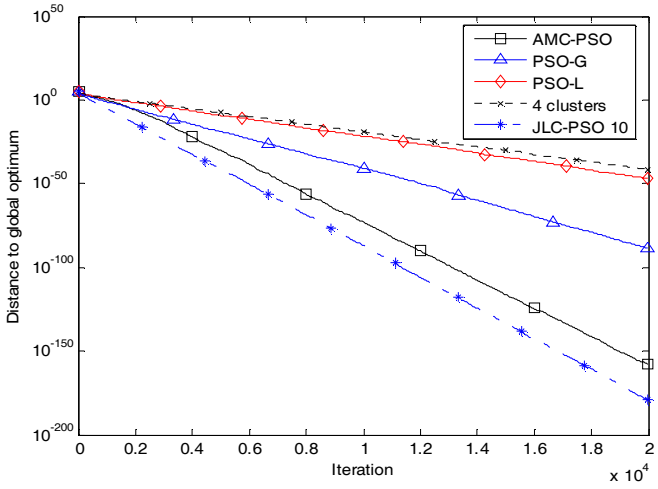

Fig.2. Sphere—Closeness to global optimum

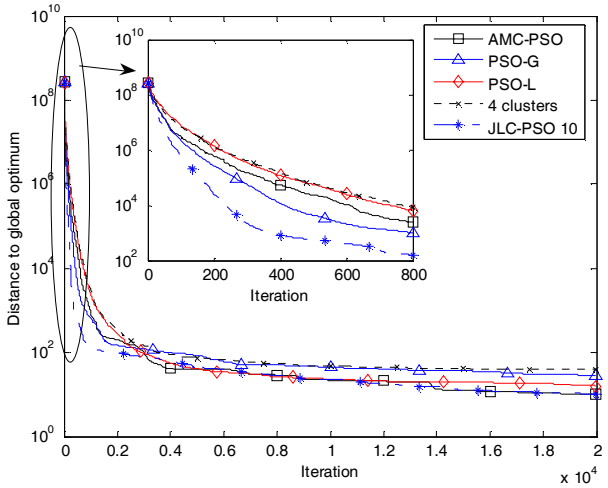

Fig. 3. Rosenbrock—Closeness to global optimum

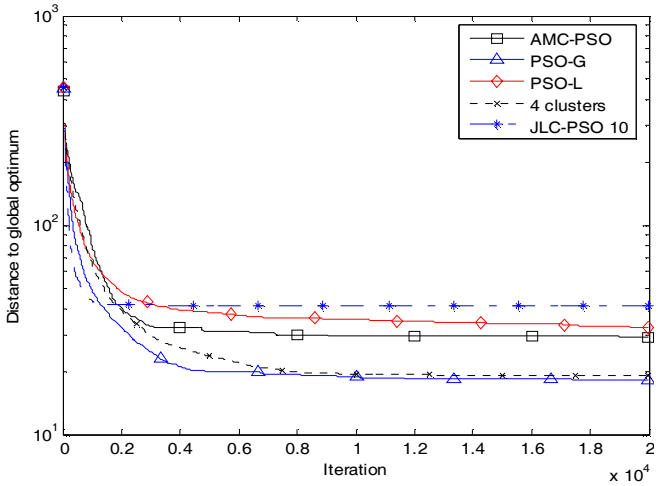

Fig. 4. Rastrigin—Closeness to global optimum

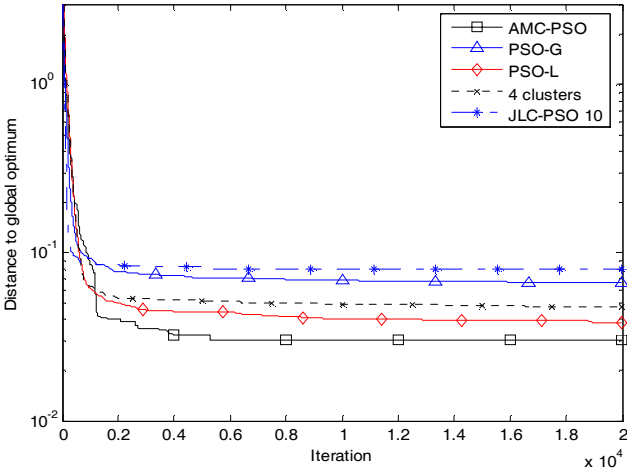

Fig. 5. Griewank —Closeness to global optimum

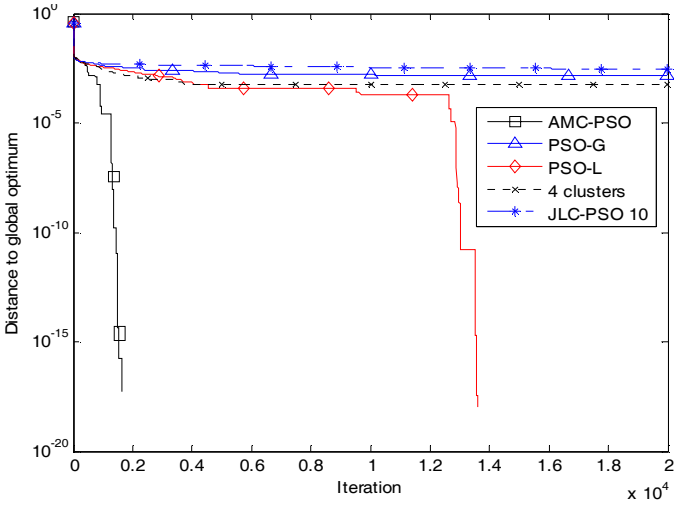

Fig. 6. Schaffer's f6—Closeness to global optimum

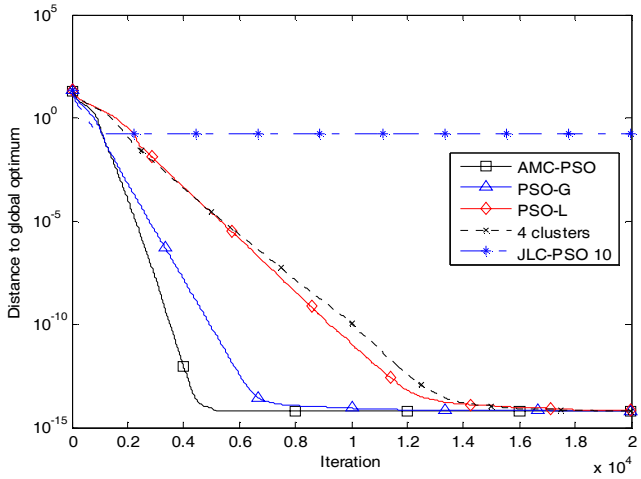

Fig. 7. Ackley—Closeness to global optimum

TABLE IV. NUMBER OF ITERATIONS NEEDED TO REACH A CERTAIN DEGREE OF CLOSENESS TO GLOBAL OPTIMUM FOR THE FIVE OPTIMIZERS. (BEST VALUES ARE BOLD FACED)

Algorithm	Avg.	Med.	Max.	Min.	Suc.%
	Sphere – (Closeness = 0.0001)				
PSO-l	2859	2873	3166	2231	**100**
PSO-g	1583	1571	1798	1313	**100**
4 CLUSTERS	3256	3261	3757	2750	**100**
JLC-PSO	**847**	**847**	**981**	**730**	**100**
AMC-PSO	1604	1610	1806	1376	**100**
	Rosenbrock – (Closeness = 100)				
PSO-l	2940	2768	5934	1548	**100**
PSO-g	2840	1565	16399	755	98
4 CLUSTERS	3470	2660	11738	1605	96
JLC-PSO	**1308**	**729**	**8651**	**390**	98
AMC-PSO	2997	1475	13786	914	**100**
	Rastrigin – (Closeness = 50)				
PSO-l	2831	1899	17196	737	94
PSO-g	1028	964	2338	354	**100**
4 CLUSTERS	1524	1396	4238	645	**100**
JLC-PSO	**551**	**469**	**1388**	**210**	78
AMC-PSO	1572	1507	2300	972	**100**
	Griewank – (Closeness = 0.1)				
PSO-l	1191	572	18893	284	96
PSO-g	916	471	4617	200	86
4 CLUSTERS	1001	585	13521	331	96
JLC-PSO	**579**	**216**	5069	**123**	72
AMC-PSO	710	587	**1217**	354	**100**
	Schaffer's f6 – (Closeness = 0.001)				
PSO-l	1505	636	12703	31	100
PSO-g	1323	593	10169	54	84
4 CLUSTERS	824	509	3737	93	94
JLC-PSO	2514	**292**	16222	**20**	70
AMC-PSO	**471**	424	**895**	169	**100**
	Ackley – (Closeness = 0.01)				
PSO-l	2825	2837	3549	2272	**100**
PSO-g	1454	1443	1835	1181	**100**
4 CLUSTERS	2754	2765	3232	2174	**100**
JLC-PSO	**853**	**808**	**1276**	**641**	88
AMC-PSO	1438	1440	1518	1350	**100**

VI. CONCLUSION

Particle swarm optimizers are very sensitive to the shape of their social network. Both PSO-g and PSO-l lack the ability of adapting their social network to the landscape of the problem they optimize.

The proposed AMC-PSO algorithm overcomes this problem. The dynamic social network of the optimizer shrinks the membership level of the superior particles to reduce their influence on other particles, while expanding the membership level for the worst particles to increase their chance in learning from better particles.

AMC-PSO versions achieves better results than PSO-l and PSO-g, either in escaping local optima or in convergence speed to global optima for almost all considered benchmark problems.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. on Neural Networks*, vol. 4, 1995, pp. 1942–1947.

[2] W. Elshamy, "Using Artificial Intelligence Models in System Identification" M.S. thesis, Dept. Elect. Power and Machines, Cairo Univ., Cairo , Egypt, 2007.

[3] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Proc. Evolutionary Programming VII*, pp. 591–600, 1998.

[4] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Inform. Process. Lett.*, vol. 85, pp. 317–325, 2003.

[5] H. M. Emara and H. A. Fattah, "Continuous swarm optimization technique with stability analysis," in *Proc. Amer. Control Conf .2004*, *ACC 2004*, vol. 3, pp. 2811–2817.

[6] M. Clerc and J. Kennedy, "The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.

[7] F. van den Bergh, "An analysis of particle swarm optimizers," Ph.D. thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa, 2002.

[8] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama, and Y. Nakanishi,"A particle swarm optimization for reactive power and voltage control considering voltage security assessment," *IEEE Trans. Power Syst.*, vol. 15, pp. 1232–1239, Nov. 2000.

[9] A. Ratnaweera, S. K. Halgamuge, , and H. C. Watson, " Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients," *IEEE Trans. on Evolutionary Comput.*, vol. 8, no.3, pp.240 – 255, Jun. 2004.

[10] K. Veeramachaneni, T. Peram, C. Mohan and L. A. Osadciw, "Optimization using particle swarms with near neighbor interactions," in *Proc. Genetic and Evolutionary Computation (GECCO 2003)*, pp. 110–121, 2003.

[11] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simple, maybe better," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 204–210, Jun. 2004.

[12] J. Kennedy, "Particle swarms: optimization based on socio cognition," in *Recent Development in Biologically Inspired Computing*, L. N. de Castro and F. J. Von Zuben, Ed. Idea Group, pp. 235–269, 2005.

[13] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," in *Proc. Cong. on Evolutionary Computation (CEC 1999)*, vol. 3, pp. 1931–1938, 1999.

[14] A. P. Engelbrecht, Computational Intelligence: An Introduction, *John Wiley & Sons, Ltd*, 2007.

[15] S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer and its adaptive variant," *IEEE Systems, Man and Cybernetics - Part B*, vol. 35, no. 6, pp. 1272–1282, Dec. 2005.

[16] P. N. Suganthan, "Particle swarm optimization with a neighborhood operator," in *Proc. Cong. on Evolutionary Computation (CEC 1999)*, vol. 3, pp. 1958–1962, 1999.

[17] A. S. Mohais, R. Mendes, C. Ward and C. Posthoff, "Neighborhood re-structuring in particle swarm optimization," in *Proc. Australian Conference on Artificial Intelligence,* vol. 3809, pp. 776–785, 2005.

[18] J. Kennedy, "Stereotyping: improving particle swarm performance with cluster analysis," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC '00)*, vol. 2, pp. 1507–1512, La Jolla, Calif, USA, July 2000.

[19] W. Elshamy, H. M. Emara, and A. Bahgat, "Clubs-based Particle Swarm Optimization," *IEEE Swarm Intelligence Symposium, 2007. SIS 2007*., pp.289-296, 1-5 April 2007.