

# A New Hierarchical Structure for Combining Different Versions of PSO

Mahdi Roshanzamir  
Iran University of Science  
and Technology  
Tehran, Iran

Nasser Mozayani  
Iran University of Science  
and Technology  
Tehran, Iran

Mohamad Roshanzamir  
Isfahan University of  
Technology  
Isfahan, Iran

## ABSTRACT

Particle swarm optimization is a population-based algorithm and used for optimization in a wide range of problems. In this article, a method that is called Hybrid Particle Swarm Optimization or HPSO is proposed. It is composed of some versions of particle swarm optimization algorithms, which have subgroups in their structures. They are DMS-PSO, PS2OS and MCP SO. In fact, a hierarchical structure is used to compose a new version of optimization algorithm and combine the results of other structures of PSO. Proposed structure has been tested on four unimodal and four multimodal test functions. Although the memory usage has no difference with other compared versions, it is much faster in many cases. Also the rank of fitness values, are good and suitable in all test functions. In addition, it is possible to execute it concurrently.

## General Terms

Optimization, Evolutionary Algorithms, Swarm Intelligence

## Keywords

Particle Swarm Optimization, Hierarchical Structure

## 1. INTRODUCTION

Kennedy and Eberhart [1] proposed the idea of Particle Swarm Optimization in 1995. They initially wanted to produce some form of computational intelligence by using social relationship that does not need individual ability. Their attempts led to a robust optimization algorithm called Particle Swarm Optimization or PSO algorithm. This algorithm has been inspired from social behavior of animals such as bird's flocking and fish movement. PSO is a population-based algorithm, and this population is consisted of some particles. Although these particles have not enough intelligence separately, they show an excellent intelligence as a whole by following their leader and using their own experience. The implementation of PSO is simple and a few parameters are needed to be valued. However, this algorithm has its own disadvantages such as low speed convergence and trapping in local optimum in some cases, so different versions of this algorithm is suggested. All of them try to improve different aspects of it and cover its weakness. In [2] a dynamic model is introduced that particles are divided into many small groups and each group executes PSO separately. The groups exchange some of their particles randomly to exchange information. It is used for preventing from trapping in local optimums. The same structure is used in [3, 4] however a local search that employs Quasi-Newton method is used in it. In [4] this structure is applied for large-scale optimization. Hanning Chen et al [5] designed hierarchical structure called hierarchical swarm optimization. They try to simulate natural hierarchical systems. Particles are divided into separate groups. Each group executes PSO independently and the best particles of each group are selected as their representative, and make higher level. In this level, the PSO is executed and best particle is selected as the global best. Ben Nui et al [6, 7] introduced a Master-Slave PSO. Particles are divided into some groups. One of them is master and others are slaves. Slaves groups execute PSO separately. In master group, particles' positions are updated using their own knowledge and slave groups' experiences. Yen and Daneshyari [8] propose a method to exchange information among multiple swarms in particle swarm optimization. This Method is developed to solve problems that have a high number of local optima. [9] proposes a multi-swarm algorithm based on fast particle swarm optimization for dynamic optimization problems. It employs a method to track multiple peaks by preventing overcrowding at a peak and a fast particle swarm optimization algorithm as a local search method. Chen [10] proposes a hierarchical particle swarm optimization. In the suggested algorithm, all particles are arranged in a regular tree structure and move up or down in the tree based on their fitness value. The velocity update of each particle depends on the position of each particle in the tree. A mutation operator is also added into the proposed approach. Bergh and Engelbrecht [11] present a cooperative behavior to improve the performance of the standard algorithm. This is done by using multiple swarms to optimize different components of the solution vector cooperatively. Most of these structures are using different groups in order to improve the quality of PSO.

There are also some surveys that discuss and compare different aspects of PSO [12, 13]. Author in [12] comprises a snapshot of particle swarming from the authors' perspective, including variations in the algorithm, current and ongoing research, applications and open problems. In [13] the history, various methods, and taxonomy of PSO are discussed and its different applications together with an analysis of these applications are evaluated. Sierra and Coello [14] did a comprehensive review of the various Multi-Objective Particle Swarm Optimizers. It includes a classification of the approaches, and the main features of each proposal. In the last part of it, they list some of the topics within this field that are considered as promising areas of future research.

In this research, standard PSO is described in section two. Section three introduces the proposed methodology. Section four and five explain experimental results and conclusions respectively.

## 2. REVIEW OF STANDARD PSO

PSO is inspired from nature and is based on iteration. It is like many other evolutionary algorithms such as Genetic Algorithms and Ant Colony Optimization, because all of them use an initial random population. Each component of population of the algorithm is called a particle. Each particle is initialized with a random velocity and position. These particles move in an n-dimensional space repeatedly. The dimension of the problem is equal to parameters of function that should be optimized. Each particle not only remembers the best position that has achieved so far but also the global best position among all other particles. By using these data, the particles set their movements in next iteration. In order to find the optimal solution, in each iteration both the velocity and position of each particle are updated using equations 1 and 2 respectively.

$$v^{new} = w \times v^{old} + \Gamma_1 \times r_1 \times (p^{personalBest} - p^{old}) + \Gamma_2 \times r_2 \times (p^{globalBest} - p^{old}) \quad (1)$$

$$p^{new} = p^{old} + v^{new} \quad (2)$$

Where  $v$  is velocity of particle,  $p$  is position of particle,  $r_1$  and  $r_2$  are random numbers uniformly distributed between [0, 1],  $\Gamma_1$  and  $\Gamma_2$  are learning rates,  $p^{personalBest}$  is the best position that a particle has found yet and  $p^{globalBest}$  is the best global position.  $w$  is inertia weight that causes the particle to continue to its way although it reaches the best position.

According to the equation 1, it is clear that the motion of a particle is influenced by three factors: (i) its best position that the particle has found so far, (ii) Best position that is founded so far by all particles and (iii) The previous path of the particle.

## 3. A HIERARCHICAL STRUCTURE FOR PSO

As it is mentioned, different versions of PSO have been suggested. Their main goal is improving PSO efficiency. In comparison, each version has better results in some cases and functions, and worse results in some others. The main goal is finding an algorithm, which has the best efficiency on all functions as fast as possible. One way to achieve this goal is finding a method to combine efficient algorithms in the best way. The obstacle to do this is the different structures of algorithms, and it is why their combination is not easy. In this research, three different versions of PSO is used and combined. They are selected because not only they have good efficiency but also they have nearly same structure. It makes their combination easier. They are DMS-PSO[2], PS2OS [5] and MCP SO [6]. Table 1 shows their average results of 20 times execution of algorithms on some test functions which are introduced in table 2. Also this structure can be used to combine other versions of PSO.

**Table 1. The mean of fitness values of 20 times execution on different functions in 30 dimensions**

| Type       | Function Name                  | Algorithm Name |             |             |             |
|------------|--------------------------------|----------------|-------------|-------------|-------------|
|            |                                | PSO            | PS2OS       | MCP SO      | DMS-PSO     |
| Unimodal   | <b>Step</b>                    | 40             | 10.2        | 0           | 0           |
|            | <b>Sum of Different Powers</b> | 6.68307E+16    | 61063177863 | 1.50713E+13 | 0           |
|            | <b>Rosenbrock</b>              | 2634.983       | 2051.209    | 57.12211    | 5.252174    |
|            | <b>Sphere</b>                  | 3920.4         | 871.2       | 6.92832E-88 | 0           |
| Multimodal | <b>Rastrigin</b>               | 136.1682       | 80.3533     | 73.57082    | 6.56673     |
|            | <b>Schwefel</b>                | 5755.476       | 3522.819    | 4394.075    | 5825.996    |
|            | <b>Griewangk</b>               | 27.15134       | 4.525284    | 0.00849     | 0.00234     |
|            | <b>Ackley</b>                  | 7.781164       | 1.041985    | 7.8948E-08  | 2.66454E-15 |

It is clear in table 1, that each version has better result on some functions. For example, DMS-PSO has the best result on almost all functions except Schwefel, which PS2OS is the best on this function. But Otherwise DMS-PSO is very slow and takes a long time to find optimum. Otherwise PS2OS is very faster and find solution with higher speed and lower time. We can combine their advantages using proposed structure.

In DMS-PSO, velocity update is happened in two steps. At 90% of iterations that form first step, particles are arranged in different groups and their movements are affected by the best position that a particle has ever found;  $P^{personalBest}$ , and the best position that is found by its neighborhood particles in its group;  $P^{localBest}$ . Then in 10% of iterations that form second step, particles movements are affected by the global best between all particles;  $P^{globalBest}$  instead of local bests. In the first step, a regroup operation is done in some predefined iterations in which some particles move from one subgroup to another one.

In PS2OS, particles are arranged in different groups and their movements are affected by the best position each particle has ever found, the best position its neighborhoods have found and global best position.

MCPSO divides particles into N groups. One group is master and others are slaves. Particle movement in slave groups are affected by the best position each particle found and the best position that found by the other particles in its group. In master group particle movement are affected by previous best position of the master swarm;  $p_i^m$ , best global position of the master swarm;  $p_g^m$  and previous best position of the slave swarms;  $p_i^s$ .

**Table 2: List of functions**

| Name                           | Definition   | Interval                     | Global Minimum | Optimal Point                 |
|--------------------------------|--|------------------------------|----------------|-------------------------------|
| <b>Step</b>                    | $f(x) = \sum_{i=1}^n [\text{floor}(x_i + 0.5)]^2$  | $-100 \leq x_i \leq 100$     | 0              | $x_i=0, i=1, \dots, n$        |
| <b>Sum of Different Powers</b> | $f(x) = \sum_{i=1}^n  x_i ^{i+1}$  | $-1 \leq x_i \leq 1$         | 0              | $x_i=0, i=1, \dots, n$        |
| <b>Rosenbrock</b>              | $f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$   | $-2.048 \leq x_i \leq 2.048$ | 0              | $x_i=0, i=1, \dots, n$        |
| <b>Sphere</b>                  | $f(x) = \sum_{i=1}^n x_i^2$  | $-5.12 \leq x_i \leq 5.12$   | 0              | $x_i=1, i=1, \dots, n$        |
| <b>Rastrigin</b>               | $f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$  | $-5.12 \leq x_i \leq 5.12$   | 0              | $x_i=0, i=1, \dots, n$        |
| <b>Schwefel</b>                | $f(x) = \sum_{i=1}^n [-x_i \sin(\sqrt{ x_i })] + 418.9829n$  | $-500 \leq x_i \leq 500$     | 0              | $x_i=420.9687, i=1, \dots, n$ |
| <b>Griewangk</b>               | $f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$                              | $-600 \leq x_i \leq 600$     | 0              | $x_i=0, i=1, \dots, n$        |
| <b>Ackley</b>                  | $f(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$ | $-30 \leq x_i \leq 30$       | 0              | $x_i=0, i=1, \dots, n$        |

In the proposed method that is a hybrid algorithm and is called HPSO, we use a hierarchical structure and particles are divided into three groups. Each group has some subgroups. Groups have equal number of particles. The first group executes DMS-PSO, the second one executes PS2OS, and the third one executes MCPSO. These algorithms have been chosen because they have good efficiency to some extent and also same structure in creating subgroups. Creating subgroups in all of these algorithms is illustrated in figure 1. In DMS-PSO and PS2OS, particles are affected by local best particles, which are members of predefined subgroups. In MCPSO, both master and slave groups are considered as subgroups. Figure 2 shows the Pseudo Code of this algorithm. They can be executed on separate processor. After each iteration, information is exchanged between the groups. This information is the best position that groups have been founded so far. Each algorithm found its own global best. Then the best one among these three bests is chosen as a global best and all other algorithms use it from now on. If the best positions of each group are named  $gBest_{DMSPSO}$ ,  $gBest_{PS2OS}$  and  $gBest_{MCPSO}$ , then:

$$gBest_{ALL} = Best(gBest_{DMSPSO}, gBest_{PS2OS}, gBest_{MCPSO}) \tag{3}$$

The Best(.) function chooses the best one. Update equations in DMS-PSO, PS2OS and MCPSO are changed respectively as equations 4, 5 and 6.

$$v^{new} = w \times v^{old} + \Gamma_1 \times r_1 \times (p^{personalBest} - p^{old}) + \Gamma_2 \times r_2 \times (gBest_{ALL} - p^{old}) \quad (4)$$

$$v^{new} = \chi(v^{old} + \Gamma_1 \times r_1 \times (p^{personalBest} - p^{old}) + \Gamma_2 \times r_2 \times (p^{localBest} - p^{old}) + \Gamma_3 \times r_3 \times (gBest_{ALL} - p^{old})) \quad (5)$$

$$v^{new} = w^* \times v^{old} + \Gamma_1 \times r_1 \times (p_i^M - p^{old}) + \Gamma_2 \times r_2 \times (p_i^S - p^{old}) + \Gamma_3 \times r_3 \times (gBest_{ALL} - p^{old}) \quad (6)$$

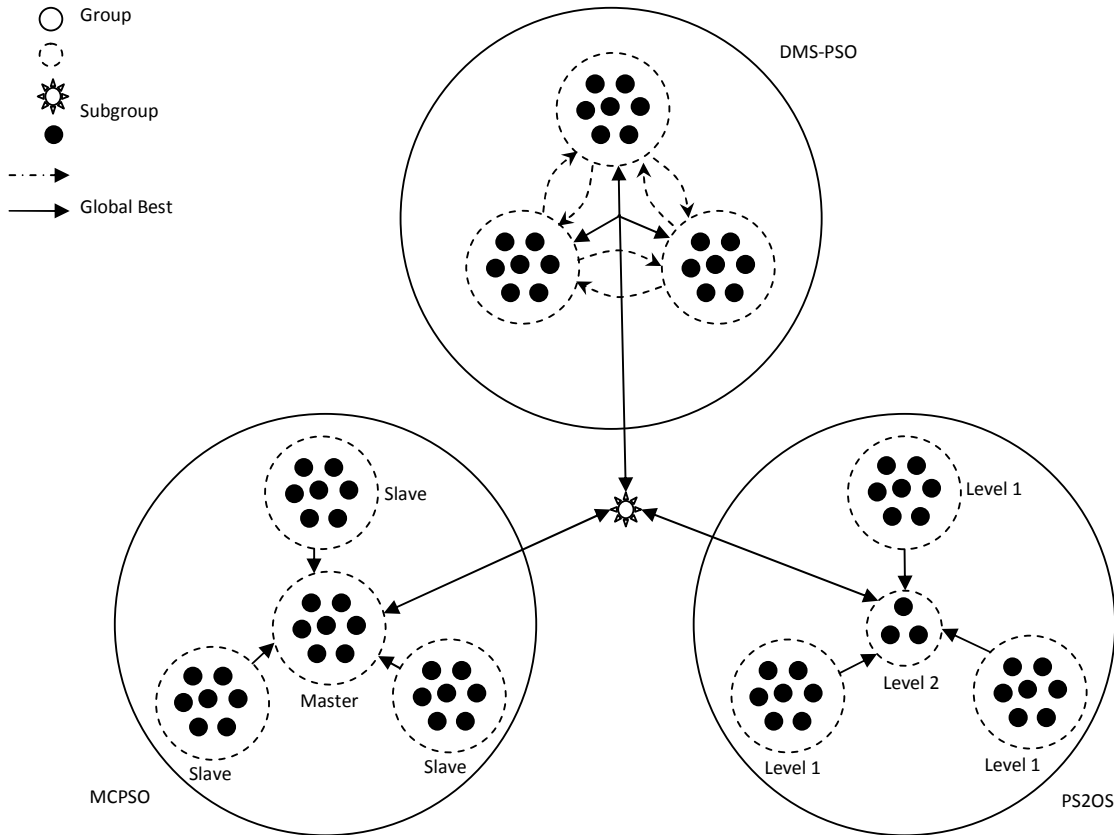


Fig 1: HPSO structure

*Algorithm HPSO*  
 Divide population into 3 groups randomly, with equal population in each group  
 Initialize particles position and velocity  
 Set globalBest as the best position of particles  
 Repeat  
     Do PS2OS, MCPSO and DMS-PSO in parallel using equations 4 to 6  
     When all process finished update globalBest using equation 3  
 Until a termination condition is met

Fig 2: Pseudo-code for the HPSO algorithm

#### 4. EXPERIMENTAL RESULTS

HPSO is tested on four unimodal and four multimodal test functions. These functions are shown in table 2. Table 3 shows parameters values that are needed for different algorithms. In DMS-PSO algorithm, regrouping happens after ten iterations. Population size of particles, the number of iterations, group number and dimension in all algorithms are 100, 10000, 5 and 30 respectively. Each algorithm is executed 20 times and the mean of fitness value are shown in table 4. The best values, standard deviations and execution time are shown in table 5, 6 and 7 respectively. Also figure 3 shows the evolutionary process of proposed structure and other algorithms. In this figure, horizontal axis shows number of iterations and vertical axis shows logarithm of fitness. As it is obvious from table 4 and figures 3 that our algorithm rank is not worse than second in all test functions and in some of them such as Rosenbrock and

Griewangk is the best. However, as it is clear in table 7 the most remarkable point about the results is its time cost. Although the number of particles is constant, execution time is often shorter than other algorithm except standard PSO. We do not consider standard PSO because of its very low efficiency. In Sum of Different Powers and Rosenbrock functions, HPSO algorithm has second and third rank. However, in other functions it is the fastest algorithm. In Rastrigin, Schwefel and Ackley functions, execution speed is about two times faster than others are.

**Table 3: Different algorithms parameters**

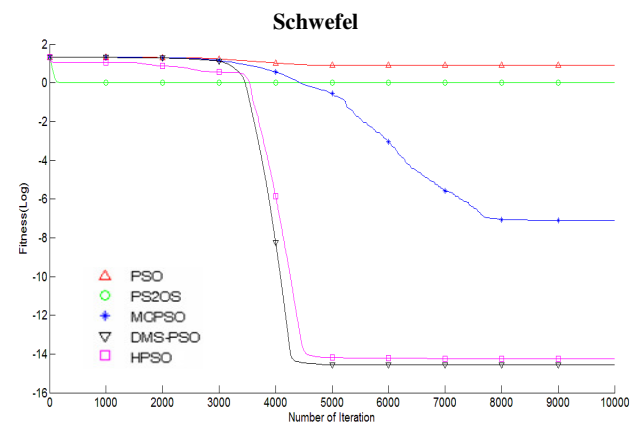
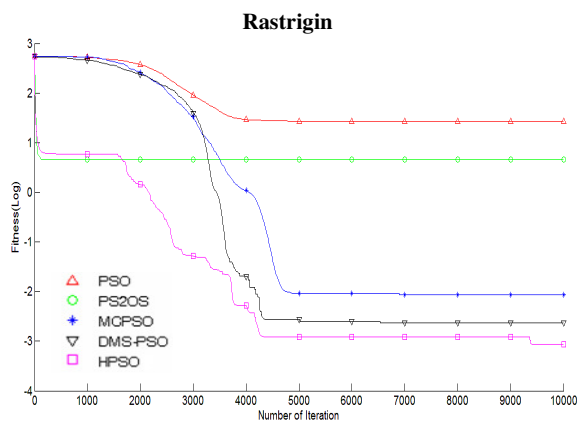
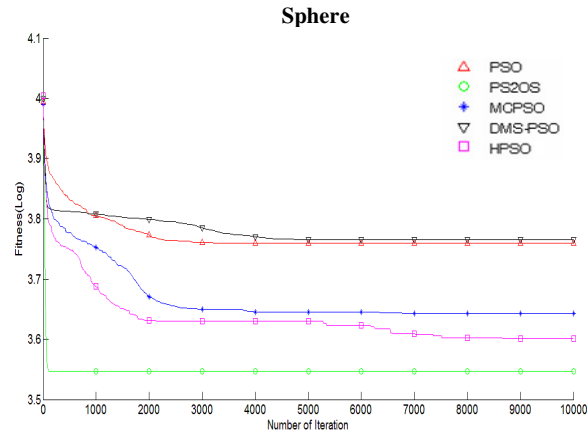
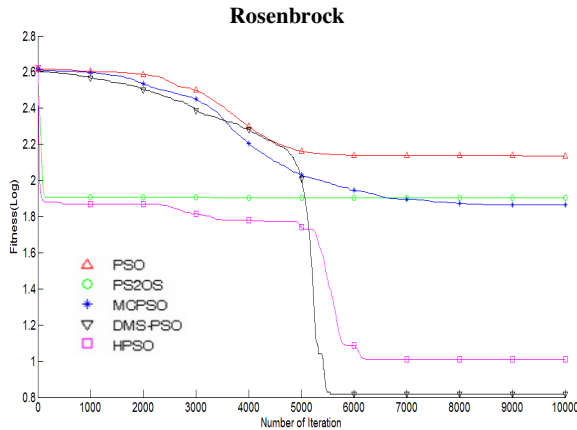
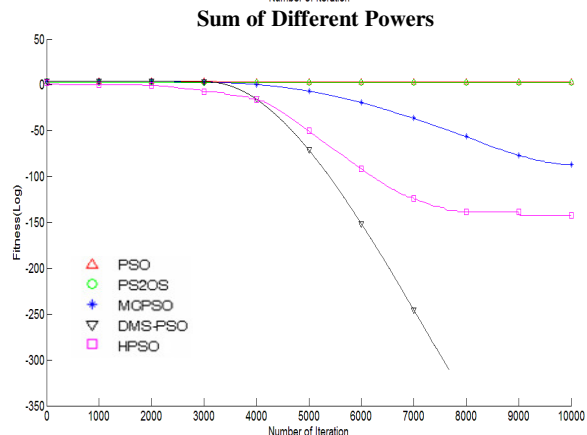
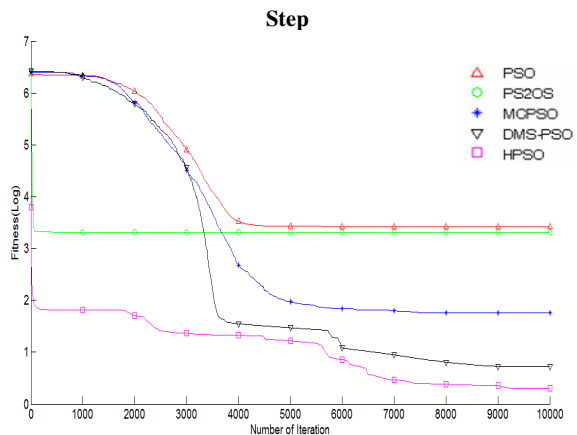
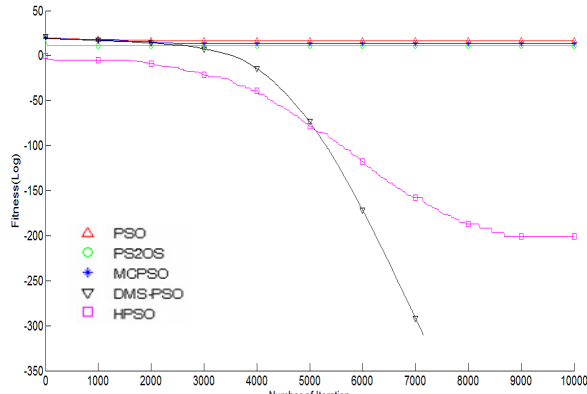
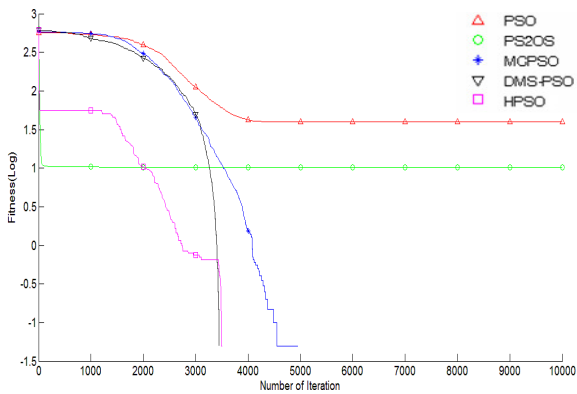
|                | $\Gamma_1$ | $\Gamma_1$ | $\Gamma_1$ | $\chi$ |
|----------------|------------|------------|------------|--------|
| <b>PS2OS</b>   | ---        | ---        | ---        | 0.4693 |
| <b>MCPSO</b>   | 2.05       | 2.05       | 0.5        | ---    |
| <b>DMS-PSO</b> | 1.3667     | 1.3667     | 1.3667     | ---    |

**Table 4: Fitness value averages**

| Type       | Function Name                  | Algorithm Name |             |             |             |             |
|------------|--------------------------------|----------------|-------------|-------------|-------------|-------------|
|            |                                | PSO            | PS2OS       | MCPSO       | DMS-PSO     | HPSO        |
| Unimodal   | <b>Step</b>                    | 40             | 10.2        | 0           | 0           | 0           |
|            | <b>Sum of Different Powers</b> | 6.68307E+16    | 61063177863 | 1.50713E+13 | 0           | 6.0666E-202 |
|            | <b>Rosenbrock</b>              | 2634.983       | 2051.209    | 57.12211    | 5.252174    | 1.994196    |
|            | <b>Sphere</b>                  | 3920.4         | 871.2       | 6.92832E-88 | 0           | 5.57791E-74 |
| Multimodal | <b>Rastrigin</b>               | 136.1682       | 80.3533     | 73.57082    | 6.56673     | 10.24808    |
|            | <b>Schwefel</b>                | 5755.476       | 3522.819    | 4394.075    | 5825.996    | 3994.704    |
|            | <b>Griewangk</b>               | 27.15134       | 4.525284    | 0.00849     | 0.00234     | 0.000863    |
|            | <b>Ackley</b>                  | 7.781164       | 1.041985    | 7.8948E-08  | 2.66454E-15 | 5.86198E-15 |

**Table 5: Bets fitness values**

| Type       | Function Name                  | Algorithm Name |             |             |             |             |
|------------|--------------------------------|----------------|-------------|-------------|-------------|-------------|
|            |                                | PSO            | PS2OS       | MCPSO       | DMS-PSO     | HPSO        |
| Unimodal   | <b>Step</b>                    | 0              | 0           | 0           | 0           | 0           |
|            | <b>Sum of Different Powers</b> | 1.1101E+11     | 0           | 11094.69    | 0           | 1.7509E-253 |
|            | <b>Rosenbrock</b>              | 3.074407       | 2.875E-25   | 0.806731    | 0.439654    | 0.001026    |
|            | <b>Sphere</b>                  | 1.41077E-56    | 0           | 3.17009E-92 | 0           | 3.0032E-156 |
| Multimodal | <b>Rastrigin</b>               | 63.91875       | 52.73256    | 9.984942    | 2.984877    | 4.974795    |
|            | <b>Schwefel</b>                | 4059.869       | 2576.789    | 3099.119    | 5596.943    | 23820.421   |
|            | <b>Griewangk</b>               | 0              | 0           | 0           | 0           | 0           |
|            | <b>Ackley</b>                  | 6.21725E-15    | 6.21725E-15 | 7.8948E-08  | 2.66454E-15 | 2.66454E-15 |



**Fig 3: Evolution of average fitness on test functions**

**Table 6: Standard deviation of fitness values**

| Type       | Function Name           | Algorithm Name |             |             |          |             |
|------------|-------------------------|----------------|-------------|-------------|----------|-------------|
|            |                         | PSO            | PS2OS       | MCPSO       | DMS-PSO  | HPSO        |
| Unimodal   | Step                    | 50.26247       | 30.71533    | 0           | 0        | 0           |
|            | Sum of Different Powers | 2.25161E+17    | 2.23092E+11 | 3.6763E+13  | 0        | 0           |
|            | Rosenbrock              | 4380.078       | 4084.7441   | 54.4867     | 2.777684 | 2.568101    |
|            | Sphere                  | 3128.413191    | 1787.665    | 1.79984E-87 | 0        | 1.2786E-142 |
| Multimodal | Rastrigin               | 43.91864       | 21.79923    | 35.44007    | 2.026243 | 3.114714    |
|            | Schwefel                | 727.9518       | 742.4373    | 689.716     | 201.5004 | 812.225     |
|            | Griewangk               | 42.52906       | 20.22434    | 0.008932    | 0.005562 | 0.002685    |
|            | Ackley                  | 8.093677935    | 3.1937318   | 2.28128E-07 | 0        | 2.27631E-15 |

**Table 7: Mean of 20 times execution of algorithms in second**

| Type       | Function Name           | Algorithm Name |          |          |          |          |
|------------|-------------------------|----------------|----------|----------|----------|----------|
|            |                         | PSO            | PS2OS    | MCPSO    | DMS-PSO  | HPSO     |
| Unimodal   | Step                    | 33.96467       | 951.3122 | 931.9807 | 994.7493 | 680.4106 |
|            | Sum of Different Powers | 63.88169       | 2153.533 | 2096.571 | 2759.359 | 2469.466 |
|            | Rosenbrock              | 45.84556       | 2187.975 | 2144.549 | 2469.64  | 2488.982 |
|            | Sphere                  | 34.80072       | 1179.87  | 1120.856 | 1252.537 | 781.094  |
| Multimodal | Rastrigin               | 42.7889        | 1475.199 | 1545.962 | 1678.039 | 719.9123 |
|            | Schwefel                | 56.3996        | 1470.374 | 1395.198 | 1583.201 | 775.41   |
|            | Griewangk               | 50.42329       | 2434.045 | 2360.87  | 2618.147 | 2203.812 |
|            | Ackley                  | 42.39741       | 1213.938 | 1223.791 | 1305.369 | 784.7638 |

## 5. CONCLUSIONS

In this paper, a hybrid PSO that is called HPSO has been proposed. It is composed of three version of PSO i.e. DMS-PSO, PS2OS and MCPSO. In this algorithm, a hierarchical structure is used .Population is divided into three groups and each group executes a different algorithm. Meanwhile, they use the best global best position that each algorithm has ever found. Then it is compared with its components algorithms. They are tested on four unimodal and four multimodal test functions. The performance of HPSO is better in some cases meanwhile its speed is often much better than the algorithms that composed it.

## 6. REFERENCES

- [1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. of the IEEE Int. Conf. Neural Networks (ICNN), Vol. 4, Nov. 1995, pp. 1942-1948.
- [2] J. J. Liang, P. N. Suganthan, "Dynamic Multi-Swarm Particle Swarm Optimizer", In: Proc. of IEEE Int. Swarm Intelligence Symposium, pp. 124-129, 2005.
- [3] J. J. Liang, P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer with local search," In IEEE Congress on Evolutionary Computation, pp. 522-528, 2005.
- [4] S. Z. Zhao, J. J. Liang, P. N. Suganthan, and M. F. Tasgetiren, "Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search for Large Scale Global Optimization," in Proc. of IEEE Congress on Evolutionary Computation, pp. 3845-3852, 2008.
- [5] H. Chen, Y. Zhu, K. Hu, and X. He, "Hierarchical swarm model: a new approach to optimization," Discrete Dynamics in Nature and Society, vol. 2010, Article ID 379649, 30 pages, 2010.
- [6] B. Niu, Y.L. Zhu, X.X. He, "Multi-population cooperative particle swarm optimization," in European Conference on Artificial Life, pp. 874-883, 2005.
- [7] B. Niu, Y. Zhu, X. He, and H. Wu, "MCPSO: a multi-swarm cooperative particle swarm optimizer," Applied Mathematics and Computation, vol. 185, no. 2, pp. 1050-1062, 2007.

- [8] G. G. Yen and M. Daneshyari, "Diversity-based information exchange among multiple swarms in particle swarm optimization," In: Proc. IEEE Congress on Evolutionary Computation, pp. 6150–6157, 2006.
- [9] C. Li and S. Yang, "Fast multi-swarm optimization for dynamic optimization problems", 4th International Conference on Natural Computation, p.624-628, 2008.
- [10] C.C. Chen, "Hierarchical particle swarm optimization for optimization problems," Tamkang Journal of Science and Engineering, Vol. 12, no. 3, Page(s). 289-298, 2009.
- [11] F. V. D. Bergh and A. Engelbrecht, "A cooperative approach to particle swarm optimization", IEEE Transactions on Evolutionary Computation , Vol. 8, No. 3, 2004.
- [12] R. Poli and J. Kennedy and T. Blackwell, "Particle swarm optimization An overview", Swarm Intelligence, 2007.
- [13] D. Sedighzadeh and E. Masehian, "Particle Swarm Optimization Methods, Taxonomy and Applications" International Journal of Computer Theory and Engineering, Vol. 1, No. 5, Page(s):486-502, December 2009.
- [14] M. R. Sierra, and C. C. Coello, "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the- Art", International Journal of Computational Intelligence Research, Vol. 2, No. 3, Page(s). 287–308, 2006.