Proceedings of the 2005 International Conference on Simulation and Modeling
V. Kachitvichyanukul, U. Purintrapiban, P. Utayopas, eds.

A NON-HOMOGENOUS PARTICLE SWARM OPTIMIZATION WITH MULTIPLE SOCIAL STRUCTURES

Pisut Pongchairerks and Voratas Kachitvichyanukul

Industrial Engineering & Management
School of Advanced Technologies
Asian Institute of Technology
Pathumthani, 12120, THAILAND

ABSTRACT

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique. One of its most important parameters is the social structure of PSO. Each structure's performance depends on geography of the search problem. This paper introduces two new versions of particle swarm optimization algorithm. The first proposed version is GLN-PSO. It is based on a structure that is built by combining previously published structures. The second is GLNR-PSO, a non-homogenous PSO algorithm that is the same as the first but allow for some particles to have different parameters.

The two proposed algorithms were tested using the benchmark test functions previously published, namely Sphere, Rosenbrock, Rastrigin and Griewank functions. The results of the experiments indicated that the first proposed algorithm GLN-PSO outperformed standard PSO and FDR-PSO on all the test functions. The second proposed algorithm GLNR-PSO further outperformed GLN-PSO on the Sphere, Rosenbrock, and Griewank functions.

1 INTRODUCTION

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique by Kennedy and Eberhart (1995). The underlying motivation for the development of PSO algorithm was social behavior of bird flocking or fish schooling. The original PSO algorithm started with a population of K particles flying in the D-dimensional problem space. Each particle is consisted of a solution, called "position" of the particle, and a randomized set of D dimensions that is used for evolving a new position, called "velocity". The velocity of each particle is dynamically adjusted by its own flying experience and that of its neighbors. Let the position of the i^{th} particles be represented by $X_i = (x_{i1}, x_{i2}, x_{i3}, ..., x_{iD})$ and the velocity of the i^{th} particle be represented by $V_i = (v_{i1}, v_{i2}, v_{i3}, ..., v_{iD})$ where i = 1, 2, ..., K. To update a new velocity, each particle uses its own personal best previous position (pbest) and the best previous position that was found by all individuals in the population (gbest). The pbest of the i^{th} particle, which is the previous position with the best objective function value found by the i^{th} particle, is represented by $P_i = (p_{i1}, p_{i2}, p_{i3}, ..., p_{iD})$ for i = 1, 2, ..., K, and the gbest is represented by P_g , the previous position that obtains the best objective function value found by all K particles. At every iteration, each particle updates its new velocity by using pbest, gbest, and its old velocity. Then, it uses that new velocity to move to a new position. However, the original PSO is weak in its ability to converge. To solve that problem, Shi and Eberhart (1998) introduced an inertia weight parameter, represented by w, to reduce the effect of the old velocity on the update of new velocity. The social structure of the original PSO is called gbest. All particles interact with the particle that has P_i that gains the best objective function value. Historically, many researches introduced other social structures to improve performance of PSO such as Local Best (lbest), Near Neighbor Best (nbest), wheel topology, pyramid, etc. Each structure's performance is depended on geography of the search problem. On the other hand, there are some research works that focus on re-initializing position and velocity of each particle with the aim to improve overall performance such as Dissipative PSO that was introduced by Xie, Zhang, and Yang (2002a).

This paper presents a structure that is derived from combining some well-known structures, and also presents a non-homogenous PSO algorithm in the sense that each particle is not required to have the same parameters. Section 2 describes three well-known structures namely Global Best (gbest), Local Best (lbest), and Near Neighbor Best (nbest). Section 3 presents a modified social structure that is build by combining the three well-known structures. Section 4 describes a non-homogenous PSO that includes particles with added ability to reinitialize their positions and with V_{max} different from those of other particles in the population. The experiments and results are given in sections 5 and 6.

2 THE THREE SOCIAL STRUCTURES

The three social structures used in this paper consist of Global Best, Local Best, and Near Neighbor Best. A brief description is given below.

1. Global Best (gbest) - This social structure is the original structure of particle swarm. P_g is the P_i that gains the best objective function value from all K particles.
2. Local Best (lbest) - For this structure, each particle is affected by the best performance of its k immediate neighbors in the population. Neighbors in this structure are not related by distance, but related on indices of particles. The lbest of i^{th} particle is the best position that are found by k immediate neighbors of i^{th} particle, included itself, represented by $P_{l\,i} = (p_{l\,i1}, p_{l\,i2}, p_{l\,i3} \ldots p_{l\,iD})$ for i =1, 2, ..., K. The lbest will be same as gbest when k is set to equal to K. To use lbest, p_{gd} in velocity's formula is replaced by $p_{l\,id}$ for each i^{th} particle. This structure allows particles search on different spaces at the same time. However, the system might be slower than gbest in terms of convergence.
3. Near Neighbor Best (nbest) - This structure was presented in Veeramachaneni, Peram, Mohan, and Osadciw (2003) by combining nbest term and the three standard terms, i.e., inertia term, pbest, and gbest. Each particle interacts with a particle that is nearby and obtains a good objective function value. The nbest of i^{th} particle is represented by $P_{n\,i} = (p_{n\,i1}, p_{n\,i2}, p_{n\,i3}, \ldots, p_{n\,iD})$ for i =1, 2...K. The velocity of i^{th} particle in d^{th} dimension is updated by using P_n, with prior best position Pj, selected to maximize
$$FDR(j, i, d) = \frac{Fitness(X_i) - Fitness(Pj)}{|p_{jd} - x_{id}|}$$
for minimization problem.

3 A PARTICLE SWARM OPTIMIZATION WITH COMBINED GBEST, LBEST, AND NBEST SOCIAL STRUCTURES (GLN-PSO)

In this algorithm, movement of each particle is pulled by an inertia term, a cognitive term pbest, and three social terms: gbest, lbest, and nbest. A parameter to control inertia term is the inertia weight (w), and parameters to control cognitive and social terms are the acceleration constants. In the case that some of these terms are moving in the same direction, the particle is pulled faster toward that direction by multiple forces. On the other hand, these forces may be canceling each others, when directions of these terms are opposite. Formulae to update the new velocity and position are shown below.

$$\begin{aligned} v_{id}(t+1) = {} & w^* v_{id}(t) \\ & + c_p{}^*rand()^*(p_{id} - x_{id}(t)) \\ & + c_g{}^*rand()^*(p_{gd} - x_{id}(t)) \\ & + c_l{}^*rand()^*(p_{lid} - x_{id}(t)) \\ & + c_n{}^*rand()^*(p_{nid} - x_{id}(t)) \end{aligned} \quad (1)$$

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

where the definitions of variables are given in the nomenclature below.

Indices:
i = index of particle i = 1, 2, ..., K
d = dimension d = 1, 2, ..., D
t = iteration t = 1, 2, ..., T

Variables:
$x_{id}(t)$ = the current position of i^{th} particle at d^{th} dimension in t^{th} iteration
$v_{id}(t)$ = the current velocity of i^{th} particle at d^{th} dimension in t^{th} iteration
p_{id} = the best previous position at d^{th} dimension known by i^{th} particle
p_{gd} = the best previous position at d^{th} dimension known by all K particles
pl_{id} = the best previous position at d^{th} dimension known by neighbors of i^{th} particle
pn_{id} = the best previous position at d^{th} dimension known by near-neighbors of i^{th} particle

Parameters:
w = inertia weight
c_p = acceleration constant of cognitive term (pbest)
c_g = acceleration constant of gbest social term
c_l = acceleration constant of lbest social term
c_n = acceleration constant of nbest social term
rand () = random number in range [0, 1]

The GLN-PSO can be implemented as follows:

1. Initialize K particles as a population, generate the i^{th} particle with a random position $X_i \in [-X_{max}, X_{max}]$, $v_{id} = 0$ for the d^{th} dimension of velocity V_i, each i^{th} particle also has k immediate neighbors
2. Evaluate the objective function value for each i^{th} particle
3. Compare the objective function value of each i^{th} particle with its pbest. The current position is set to be pbest if the objective function value of current position is better than pbest. And, the current position is also set to be gbest if the objective function value of the current position is better than gbest.
4. Compare the objective function value of each i^{th} pbest with pbest of its k immediate neighbors. lbest of each i^{th} particle is the best pbest of its k immediate neighbors.

5. For each d^{th} dimension of the i^{th} particle, nbest of d^{th} dimension of the i^{th} particle is selected by the position that is maximizing FDR (j, i, d)
   when $FDR(j, i, d) = \dfrac{Fitness(X_j) - Fitness(P_j)}{|p_{jd} - x_{id}|}$
   for minimization
6. Update the velocity and position of each i^{th} particle depending on equations (1) and (2)
7. If $v_{id}(t+1)$ is less than $-V_{max}$ then $v_{id}(t+1) = -V_{max}$. And, if $v_{id}(t+1)$ is exceed than V_{max} then $v_{id}(t+1) = V_{max}$
8. If $x_{id}(t+1)$ is less than $-X_{max}$ then $x_{id}(t+1) = -X_{max}$ and $v_{id}(t+1) = 0$. And, if $x_{id}(t+1)$ is exceed than X_{max} then $x_{id}(t+1) = X_{max}$ and $v_{id}(t+1) = 0$.
9. Return to step 2 until the stop criteria is met.

The notation used in this paper is in the form GLN-PSO [c_p c_g c_l c_n]. The figures inside the bracket [...] are values of the acceleration constants. GLN-PSO [2200] represents the standard PSO that $c_p = c_g = 2$. GLN-PSO [1102] represents a FDR-PSO (1, 1, 2) that $c_p = c_g = 1$, and $c_n = 2$. GLN-PSO [1111] is a full version of GLN-PSO that all values of acceleration constants are equal to 1.

4 A NON-HOMOGENOUS PSO

This version is modified from GLN-PSO version by adding the ability to reinitialize position for some particles in the population. These extra particles are assigned ability to explore new search areas that might lead to better solutions while other particles are still following the rule of GLN-PSO. It is commonly accepted that initial position of the population has high influence on performance of PSO and the position re-initialization might improve overall performance. This section describes a non-homogenous PSO with different values of parameters for each particle.

The extra particles start to reinitialize their positions at the R^{th} iteration, after that they will reinitialize their position every r iterations. It means that these extra particles reinitialize their position at R^{th}, $(R+r)^{th}$, $(R+2r)^{th}$, $(R+3r)^{th}$...and stop at T^{th} iteration. V_{max} of the extra particles is much smaller than others to allow these particles to spend enough times to search in other areas before they are pulled back by social terms. Formula to update new velocity and position is shown below.

Formula to update new velocity and new position:

$$v_{id}(t+1) = w^*v_{id}(t) + c_p^*rand()^*(p_{id} - x_{id}(t)) + c_g^*rand()^*(p_{gd} - x_{id}(t)) + c_l^*rand()^*(p_{lid} - x_{id}(t)) + c_n^*rand()^*(p_{nid} - x_{id}(t)) \quad (3)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (4)$$

IF ($i \in G$ and $t >= R$ and $t \bmod r = 0$)
THEN $x_{id}(t+1) = Random(-X_{max}, X_{max})$ (5)
IF ($i \in G$ and $t >= R$ and $t \bmod r = 0$)
THEN $v_{id}(t+1) = 0$ (6)

where the additional index and parameters are defined as:

Indices:

G = set of particle's indices that have ability to reinitialize position

Parameters:

R = the first iteration that particles in G reinitialize their positions

r = number of iterations between position-reinitialization of particle in G

The non-homogeneous PSO, GLNR-PSO, can be implemented as follows:

1. Initialize K particles as a population, generate the i^{th} particle with a random position $X_i \in [-X_{max}, X_{max}]$, $v_{id} = 0$ for dth dimension of velocity V_i, each i^{th} particle also has k immediate neighbors
2. Assign particle's indices to set G
3. Evaluate the objective function value for each i^{th} particle
4. Compare the objective function value of each i^{th} particle with its pbest. The current position is set to be pbest if the objective function value of current position is better than pbest. Then, the current position is also set to be gbest if the objective function value of current position is better than gbest.
5. Compare the objective function value of each i^{th} pbest with pbest of its k immediate neighbors. lbest of each i^{th} particle is the best pbest of its k immediate neighbors.
6. For each d^{th} dimension of the i^{th} particle, nbest of d^{th} dimension of the i^{th} particle is selected by the position that maximizes FDR (j, i, d)
   when $FDR(j, i, d) = \dfrac{Fitness(X_j) - Fitness(P_j)}{|p_{jd} - x_{id}|}$
   for minimization
7. Update the velocity and position of each i^{th} particle depending on equations (3), (4), (5), and (6)
8. If $v_{id}(t+1)$ is less than $-V_{max}$ then $v_{id}(t+1) = -V_{max}$. And, if $v_{id}(t+1)$ is exceed than V_{max} then $v_{id}(t+1) = V_{max}$
9. If $x_{id}(t+1)$ is less than $-X_{max}$ then $x_{id}(t+1) = -X_{max}$ and $v_{id}(t+1) = 0$. And, if $x_{id}(t+1)$ is exceed than X_{max} then $x_{id}(t+1) = X_{max}$ and $v_{id}(t+1) = 0$.
10. Return to step 2 until the stop criteria is met.

This Non-homogenous PSO version is represented by GLNR-PSO [c_p c_g c_l c_n], and notations inside the bracket [...] are the values of the acceleration constants.

5 THE EXPERIMENTS

The comparison of performance in this paper includes the following PSO versions:

1. GLN-PSO [2200] represents the standard PSO
2. GLN-PSO [1102] represents a version of FDR-PSO
3. GLN-PSO [1111] represents a full version of GLN-PSO
4. GLNR-PSO [1111] represents a version of position reinitializing GLN-PSO

notations inside the bracket [...] are values of c_p, c_g, c_l, and c_n.

Test functions and their X_{max} values in this paper are based on Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization introduced by Eberhart and Shi (2000) as shown below:

1. Sphere ($X_{max} = 100$)
   $f = \Sigma^{D}_{d=1} x_d^2$
2. Rosenbrock ($X_{max} = 30$)
   $f = \Sigma^{D-1}_{d=1} (100(x_{d+1} - x_d^2)^2 + (x_d - 1)^2)$
3. Rastrigin ($X_{max} = 5.12$)
   $f = \Sigma^{D}_{d=1} (x_d^2 - 10\cos(2\pi x_d) + 10)$
4. Griewank ($X_{max} = 600$)
   $f = 1 + (\Sigma^{D}_{d=1} x_d^2 / 4000) - (\Pi^{D}_{d=1} \cos(x_d / \sqrt{d}))$

Parameters setting of the PSO for the experiments are as follows:

1. Number of experiments = 30
2. Population size (K) = 30
3. Neighborhood size (k) = 5
4. Number of dimension (D) = 30
5. Max. iteration (T) = 1000
6. w was set to 0.9 at the first iteration and reduced linearly to 0.4 at the 1000th iteration
7. $V_{max} = X_{max}$
   (Steps 8 to 11 are for GLNR [1111], only)
8. Indices of reinitializing particles G = {10th , 20th, 30th }
9. First iteration to reinitialize (R) = 500
10. iteration between reinitializing (r) = 10
11. $V'_{max} = V_{max} / 40$

6 RESULT AND DISCUSSION

The results from the experiments are summarized in Table 1 below. The entries in the table are the average of the minimum objective function values found within 1000 iterations by four benchmarking versions. GLN-PSO [2200] represents the standard PSO, GLN-PSO [1102] represents a version of FDR-PSO, GLN-PSO [1111] represents a full version of GLN-PSO, and GLNR-PSO [1111] represents a non-homogenous PSO that is introduced in this paper. Parameters of these versions are set as shown in section 5.

The result shows that GLNR-PSO [1111] gains the best performance in Sphere, Rosenbrock, and Griewank, while GLN-PSO [1111] gains the best performance in Rastrigin. And, GLN-PSO [1111] outperforms the standard PSO and the FDR-PSO in all four test functions.

Table 1: The average of the minimum objective function value of benchmarking versions

VERSION	FUNCTION			
	Sphere	**Rosen-brock**	**Rastrigin**	**Griewank**
GLN-PSO [2200]	2.07E-03	3.23E+02	6.28E+01	1.68E-02
GLN-PSO [1102]	1.22E-13	4.23E+01	7.16E+01	1.87E-02
GLN-PSO [1111]	3.80E-14	3.88E+01	**5.21E+01**	8.83E-03
GLNR-PSO [1111]	**2.23E-16**	**2.31E+01**	6.79E+01	**7.51E-03**

In the case of multi-modal functions, the GLN-PSO [1111] allows particles to search in different spaces before the system starts to converge. Each particle is pulled by two neighbor social terms that are lbest and nbest. Then, it has higher opportunity to find some good search spaces. Each particle searches space around gbest, lbest, and nbest when these terms are in difference positions. On the other hand, particle is faster to move when these forces are in the same direction. In the case of uni-modal functions, GLN-PSO [1111] also performs best. It might be because all the social terms are heading in same direction. The system converges to the optimum faster than other versions.

For GLNR-PSO [1111], this version is better than GLN-PSO [1111] in Sphere, Rosenbrock, and Griewank. This may be that the GLNR-PSO allowed only 10 percent of whole population to have the ability to reinitialize and the three special particles are allowed to explore some other regions without negative impact on information sharing and convergence's ability of the whole population. However, this version does not perform well for the Rastrigin function. It is because this function's characteristic has many deep local optimums and position reinitialization might reduce convergence's ability of overall system.

For GLN-PSO version, it is still not conclusive that all acceleration constants equal to 1 are the best set of parameters. Some researches may still be needed to find suitable values of acceleration constants. And, the result might be changed when some importance parameters such as V_{max}, X_{max}, and inertia weight are changed.

7 CONCLUSION

This paper introduces two new versions of particle swarm optimization algorithm. The first proposed version is GLN-PSO. It combines three social structure terms to the velocity's equation. The movement of each particle is forced by five terms that include inertia, cognitive, global social, local social, and near neighbor social terms. The result of the experiments indicated that a full version of GLN-PSO outperforms standard PSO and FDR-PSO for all the test functions. The second proposed version is the non-homogeneous version called GLNR-PSO. It is modified from GLN-PSO by allowing some particles to reinitialize theirs position. These modified particles have much smaller V_{max} than other particles in the population and are able to explore other spaces before they are pulled back by social terms. This version is an example of a non-homogenous particle swarm optimization algorithm in the sense that some particles may have different parameters from the population. According to experimental results, performance of this version is better than GLN-PSO in Sphere, Rosenbrock, and Griewank functions.

REFERENCES

Eberhart, R. C. and Y. Shi. 2000. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. *Proceedings of the 2000 Congress on Evolutionary Computation*. Piscataway, NJ, IEEE Service Center.

Kennedy, J. 1999. Small Worlds and Mega-Minds: Effect of Neighborhood Topology on Particle Swarm Performance. *Proceedings of the 1999 Congress of Evolutionary Computation.* IEEE Press.

Kennedy, J. and R. C. Eberhart. 1995. Particle Swarm Optimization. *IEEE International Conference on Neural Network*. Perth, Australia.

Lovbjerg, M., Rasmussen, T. K., and T. Krink. 2001. Hybrid Particle Swarm Optimiser with Breeding and Subpopulations. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001).* Morgan Kaufmann.

Mendes, R., Kennedy, J., and J. Neves. 2003. Watch thy Neighbor or How the Swarm can Learn from its Environment. *IEEE Swarm Intelligence Symposium*. Indianapolis, IN, USA.

Shi, Y. and R. C. Eberhart. 1998. A Modified Particle Swarm Optimizer. *IEEE International Conference on Evolutionary Computation*. Anchorage, Alaska.

Veeramachaneni, K., Peram, T., Mohan, C., and L. A. Osadciw. 2003. Optimization Using Particle Swarms with Near Neighbor Interactions. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003)*. Chicago, IL, USA.

Xie, X. F., Zhang, W. J., and Z. L. Yang. 2002a. A Dissipative Particle Swarm Optimization. *IEEE Congress on Evolutionary Computation*. Honolulu, Hawaii, USA.

Xie, X. F., Zhang, W. J., and Z. L. Yang. 2002b. Hybrid Particle Swarm Optimizer with Mass Extinction. *International Conf. on Communication, Circuits and Systems (ICCCAS).* Chengdu, China.

AUTHOR BIOGRAPHIES

PISUT PONGCHAIRERKS is a doctoral student in Industrial Engineering & Management, School of Advanced Technologies, Asian Institute of Technology, Thailand. He received his B.Eng and M.Eng in Industrial Engineering from Kasetsart University. His research interests include scheduling, metaheuristics for optimization, performance measurement, and network flow problems. His email address is <Pisut.Pongchairerks@ait.ac.th>

VORATAS KACHITVICHYANUKUL is an Associate Professor in the Industrial Engineering & Management field of study, School of Advanced Technologies, Asian Institute of Technology, Thailand. He received a Ph. D. from the School of Industrial Engineering at Purdue University in 1982. He has extensive experiences in simulation modeling of manufacturing systems. He had worked for FORTUNE 500 Companies such as Compaq Computer Corporation and Motorola Incorporated. He had also worked for SEMATECH as technical coordinator of the future factory program. His teaching and research interests include planning and scheduling, high performance computing and applied operations research with special emphasis on industrial systems. His email address is <voratas@ait.ac.th>