Journal of the Korean Institute of Industrial Engineers
Vol. 37, No. 3, pp. 163-170, September 2011.

A Modified Particle Swarm Optimization Algorithm : Information Diffusion PSO

Junhyuk Park[1] · Byung-In Kim[2†]

[1]Institute of Information Technology, Inc., Magnolia, Texas, USA
[2]Department of Industrial and Management Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Kyungbuk, 790-784, Korea

새로운 위상 기반의 Particle Swarm Optimization 알고리즘 : 정보파급 PSO

박준혁[1] · 김병인[2]

[1]미국 Institute of Information Technology사 / [2]포항공과대학교 산업경영공학과

This paper proposes a modified version of Particle Swarm Optimization (PSO) called Information Diffusion PSO (ID-PSO). In PSO algorithms, premature convergence of particles could be prevented by defining proper population topology. In this paper, we propose a variant of PSO algorithm using a new population topology. We draw inspiration from the theory of information diffusion which models the transmission of information or a rumor as one-to-one interactions between people. In ID-PSO, a particle interacts with only one particle at each iteration and they share their personal best solutions and recognized best solutions. Each particle recognizes the best solution that it has experienced or has learned from another particle as the recognized best. Computational experiments on the benchmark functions show the effectiveness of the proposed algorithm compared with the existing methods which use different population topologies.

Keywords: Metaheuristics, Particle Swarm Optimization, Continuous Nonlinear Optimization, Population Topology

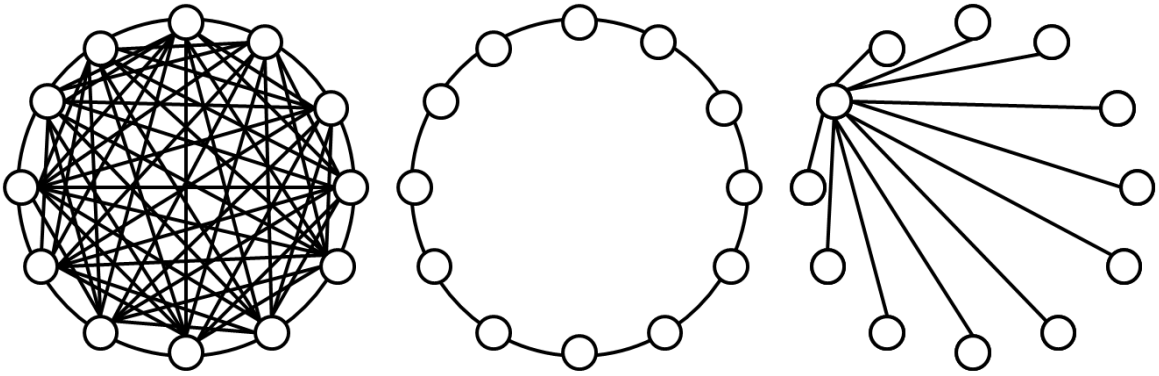
1. Introduction

This paper introduces a new variant of Particle Swarm Optimization (PSO) algorithm which is inspired by the theory of information diffusion (Thompson, 1979; Kawachi *et al*., 2008) and calls it Information Diffusion PSO (ID-PSO). The original PSO algorithm is an iterative and stochastic approach which was first introduced by Kennedy and Eberhart (1995) for continuous nonlinear optimization. PSO mimics the swarming behaviors of birds and fish. The population consists of different solutions called *particles*. Each particle remembers the best solution, called the personal best, which it has experienced. Particles also share their information with the other members of population. An individual particle is influenced by social network which is called *population topology*. Among different population topologies, *gbest* topology is most widely used. In the *gbest* topology, each particle is influenced by the best solution found by the entire population. The *gbest* topology usually converges fast and can enhance intensified search near the *gbest*. However, the *gbest* topology has a weakness for premature convergence which leaves large parts of the search space unexplored.

Many researches considered other population topologies

† Corresponding author : Professor Byung-In Kim, Department of Industrial and Management Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Kyungbuk, 790-784, Korea, Fax : +82-54-279-2870, E-mail : bkim@postech.ac.kr
Received May 5, 2011; Accepted June 7, 2011.

besides *gbest*. Kennedy (1999) was the first research which investigated the effect of population topologies. New population topologies such as *Circles*, *Wheels*, and *Random edges* are introduced in the study. Those topologies are shown in <Figure 1>. In the figure, each node represents a particle, and an arc between two nodes represents that these nodes are directly communicate with each other. The study showed that the population topology significantly affects the performance of PSO. Kennedy and Mendes (2002) systematically investigated the effects of various population topologies on the PSO performance and compared several special-structured graphs. Mendes (2004) compared various population topologies while



(a) *gbest* (b) *circle* (c) *wheel*

Figure 1. The population topologies of *gbest*, circles, and wheels

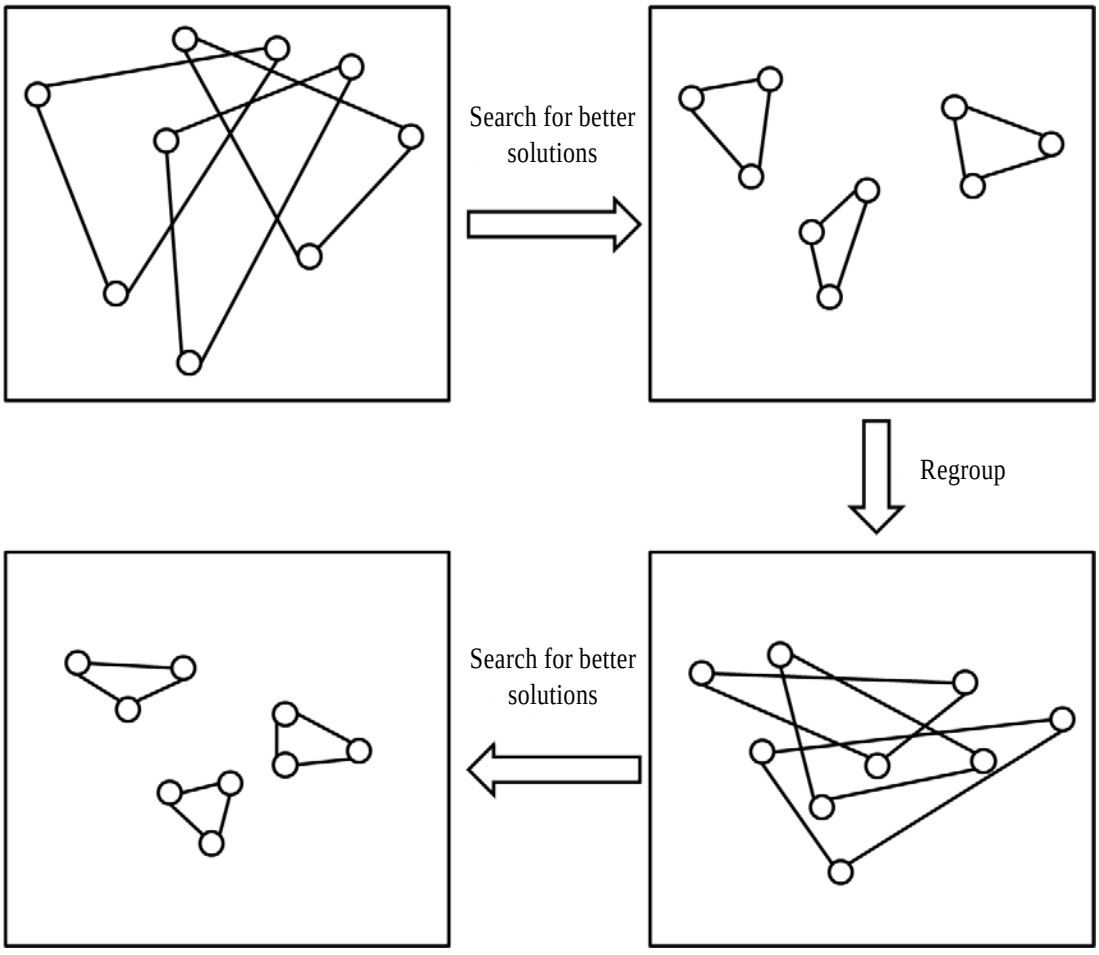


Figure 2. DMS-PSO Algorithm(m = 3)

considering various graph statistics such as degree, distance, and clustering. In the study, the best results were achieved when the average degree of topology was between 3 and 5. Note that Kennedy (1999) and Mendes (2004) did not allow isolated subpopulations. Løvbjerg *et al*. (2001) introduced the concept of subpopulations. In their study, they divided the entire population into same-sized subpopulations. They also introduced a recombination mechanism called breeding for interaction between subpopulations. Liang and Suganthan (2005) extended Løvbjerg *et al*. (2001) by dynamically regrouping subpopulations. The concept of their algorithm is illustrated in <Figure 2>. In this example, there are three subpopulations and each subpopulation consists of three particles. The entire population is divided into three subpopulations and each swarm searches for better solutions. These subpopulations are randomly regrouped every *R* generations and this process is repeated until a certain stopping criterion is met. Experimental results showed that their algorithm is better than the other PSO variants.

In this paper, we propose a new population topology which is originated from the theory of information diffusion. The theory of information diffusion has been studied by sociologists, psychologist, and mathematicians (Thompson, 1979; Kawachi *et al*., 2008). In information diffusion theory, the transmission of information or a rumor is usually modeled as a one-to-one interaction between two individuals. Likewise, ID-PSO is based on one-to-one interactions between particles. At each iteration of the algorithm, particles are matched in pairs to share their experience or knowledge (solutions). After sharing information, each particle updates its recognized best solution if the other particle has a better recognized solution. Note that the particle's personal best solution is the best solution it has actually experienced and is not influenced by solutions from other particles. Unlike the *gbest* topology, when a new global solution is found, it is not announced to the entire population. In ID-PSO, we separate the real global best and the recognized best. The real global best is a solution which has the best fitness value in the entire population. Contrary to the real global best, the recognized best is the best solution among the solutions that a particular particle has actually experienced or learned from other particles.

In the *gbest* topology, the global best influences the entire population and could limit search space. However, in ID-PSO, the influence of the global best is weakened. As a result, particles can make a more extensive search for uncharted regions. However, ID-PSO could have poor convergence because the *real* global best is unknown to the population. To cope with this problem, we introduce a set called *informed_ particles*. The *informed_ particles* is a subset of the population whose size changes as iteration proceeds. The particles in this set use the *real* global best in the update function, making the function equivalent to that in the *gbest* topology. <Figure 3> shows the population topology of ID-PSO. The entire particles are matched two by two. The members of *informed_ particles* are connected to an *oracle* which informs the real global best to those particles. And the set *informed_ particles* dynamically adjusts its members. Computational experiments on the benchmark functions show the performance of the proposed algorithm compared with the existing methods.

The remainder of this paper is organized as follows. Section 2 reviews several PSO variants. Section 3 provides a detailed explanation of ID-PSO algorithm. Section 4 shows and compares ID-PSO with existing PSO variants. Finally, Section 5 provides concluding remarks and future research directions.

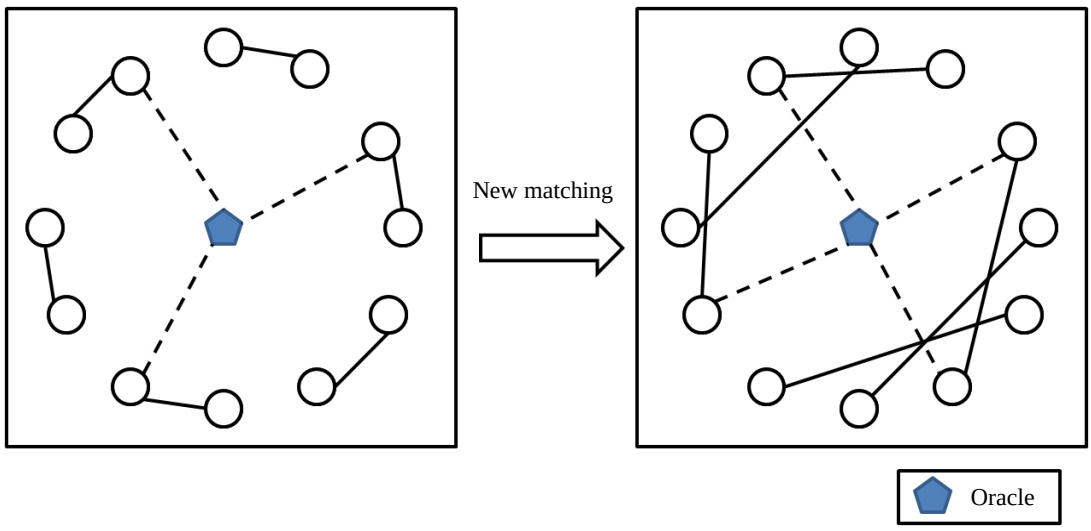


Figure 3. The population topology of ID-PSO

2. PSO and its variants

2.1 Constricted Particle Swarm Optimization

This section introduces Constricted Particle Swarm Optimization (CPSO) which uses the *gbest* topology. CPSO was proposed by Clerc and Kennedy (2002) and ensures the convergence of the search procedures and has a better solution quality than the original PSO. Assume an optimization problem which has r decision variables. Let n be the population size (i.e., number of particles). Each particle i (i = 1, ⋯, n) has a r-dimensional position vector $P_i = (p_{i1}, p_{i2}, \cdots, p_{ir})$ and a velocity vector $V_i = (v_{i1}, v_{i2}, \cdots, v_{ir})$. The fitness of each particle is evaluated by a cost function fitness(P_i). Each particle keeps its personal best $pbest_i$ and among these personal best, the global best *gbest* is identified. Each particle i updates its velocity and position for dimension j using following equations :

$$v_{ij} \leftarrow \chi(v_{ij} + \varphi_1\, rand_1(pbest_{ij} - p_{ij}) + \varphi_2\, rand_2(gbest_j - p_{ij})), \quad (1)$$

$$\chi = \frac{2}{\left|2 - (\varphi_1 + \varphi_2) - \sqrt{(\varphi_1 + \varphi_2)^2 - 4(\varphi_1 + \varphi_2)}\right|}, \quad (2)$$

and

$$p_{ij} \leftarrow p_{ij} + v_{ij}, \quad (3)$$

where, φ_1 and φ_2 are the cognitive coefficients, $rand_1$ and $rand_2$ are random numbers drawn from a uniform distribution U(0, 1), and χ is the constriction coefficient. Clerc and Kennedy (2002) recommended setting φ_1 and φ_2 to 2.05, so $\varphi_1 + \varphi_2$ is 4.1 and χ is 0.7298.

2.2 Fully Informed Particle Swarm (FIPS)

In a classical PSO algorithm, each individual is influenced by the best particle of its neighborhood. In the Fully Informed Particle Swarm (FIPS) model, which was proposed by Mendes *et al*. (2004), social influence comes from all the particles in the neighborhood. In the FIPS model, the velocity of each particle is calculated as follows :

$$v_{ij} \leftarrow \chi(v_{ij} + \varphi(p_{mj} - p_{ij}), \quad (4)$$

and

$$\varphi_k \in U\left[0, \frac{\varphi_{\max}}{|\Omega_i|}\right], \quad (5)$$

$$\varphi = \sum_{k \in \Omega_i} \varphi_k, \quad (6)$$

$$\text{and } p_{mj} = \frac{\sum_{k \in \Omega_i} \varphi_k pbest_{kj}}{\varphi}, \quad (7)$$

where, Ω_i is the set of neighbors of i and $\varphi_{\max}$ = 4.1. And $pbest_k$ is the best solution found by particle k. Note that the particle i's best solution ($pbest_i$) is not included in (4). In the experiments of Mendes *et al*. (2004), $pbest_i$ was included in the formula but the influence of including $pbest_i$ was not significant. χ means the same with above (2)

2.3 Dynamic Multi-Swarm Particle Swarm Optimizer (DMS-PSO)

Dynamic Multi-Swarm Particle Swarm Optimizer (DMS-PSO), which was proposed by Liang and Suganthan (2005), changes swarms dynamically. DMS-PSO is a kind of parallel searching algorithm which uses multiple swarms that have relatively small population size. In DMS-PSO, the entire population is divided into subpopulations with size m and these subpopulations are randomly regrouped every R generations. Once a subpopulation is determined, information is exchanged within the subpopulation. When velocity is calculated, the *gbest* topology is used. The study of Liang and Suganthan (2005) found that most of the high quality solutions come from the parameter setting at m = 3 and R = 5, i.e., each sub population consists of three particles and these subpopulations are randomly regrouped every 5 generations.

3. ID-PSO

3.1 Description of ID-PSO Algorithm

There is a main difference between CPSO and ID- PSO. In ID-PSO, $gbest_i$ represents the *recognized* best solution of particle i. We assume that the global best is unknown to the particles. The velocity of particle i for dimention j can be calculated by

$$v_{ij} \leftarrow \chi(v_{ij} + \varphi_1\, rand_1(pbest_{ij} - p_{ij}) + \varphi_2\, rand_2(gbest_j - p_{ij})). \quad (8)$$

Except for the subscript i next to *gbest*, Equation (1) and (8) are identical. The procedure of ID-PSO algorithm is presented in <Figure 4>. In the initialization step, the initial position of each particle is randomly determined and the initial velocity vector is set to (0, ⋯, 0). For each iteration, all particles are matched one by one, and the algorithm

1. Initialize.
   1.1 Randomly Generate position vector $P_i = (p_{i1}, p_{i2},\cdots, p_{ir})$ and set velocity vector $V_i = (v_{i1}, v_{i2},\cdots, v_{ir})$ to (0, 0,⋯, 0), $1 \le i \le n$ for n particles.
   1.2 Set $pbest_i$ and $gbest_i$ to P_i.
2. Repeat until the stopping criterion is met.
   2.1 Match all particles into pairs.
   2.2 For two matched particles i and j, update $gbest_i$ to $gbest_j$ if $fitness(gbest_i) > fitness(gbest_j)$. Otherwise, update $gbest_j$ to $gbest_i$.
   2.3 For each particle i, update velocity vector V_i by
   $v_{ij} \leftarrow \chi(v_{ij} + \phi_1\, rand_1(pbest_{ij} - p_{ij}) + \phi_2\, rand_2(gbest_{ij} - p_{ij}))$, $\forall j = 1,\cdots, r$
   2.4 For each particle i, update particle's position by
   $p_{ij} \leftarrow p_{ij} + v_{ij}$, $\forall j = 1, \cdots, r$
   2.5 Update the set *informed_particles*. If $pbest_i$ has not been improved for t_1 iterations, then particle i is added to the set *informed_particles*.
   2.6 If the *real* global best has not been improved for t_2 iterations, bottom 50% particles are randomly initialized while the best half of population is kept.

Figure 4. The procedure of the proposed algorithm

prevents two matched particles from matching again for the next Θ iterations. In our implementation, Θ is set to 15. Note that there could be no feasible matching of particles if Θ is set to too large. After a feasible matching is obtained, a particle i exchanges information with its partner j. If particle j has a better $gbest_j$ than $gbest_i$, particle i updates its recognized best $gbest_i$. Otherwise, $gbest_j$ is updated with $gbest_i$. After information is exchanged, each particle calculates its new velocity vector and updates its position vector. The algorithm updates the set *informed_ particles* throughout the process. If the personal best solution of particle i has not been improved for t_1 iterations, then particle i is added to the set *informed_ particles*. Any member of *informed_ particles* is removed from the set as soon as its personal best is improved.

If the *real* global best has not been improved for t_2 iterations, bottom 50% particles are randomly initialized while the best half of population is kept. In our implementation t_1 and t_2 are set to 30 and 200 respectively. Note that these parameter values were determined through preliminary experiments.

4. Experimental results

This Section evaluates the performance of ID-PSO and compares the results with following three algorithms :

- CPSO with the *gbest* topology,
- FIPS model with average degree between 3 and 4,
- DMS-PSO with $m = 3$ and $R = 5$.

The size of entire population is set to 20 for CPSO and FIPS. For DMS-PSO and ID-PSO, we use 30 particles. The experiment was conducted on 5 benchmark functions with dimensions 10 and 30. These functions are listed below :

- Sphere Function

$$f(x) = \sum_{i=1}^{D} x_i^2,$$

  where $x_i \in$[-5.12, 5.12]

- Rosenbrock's Function

$$f(x) = \sum_{i=1}^{D-1} \left[100(x_i^2 - x_{i+1}) + (x_i - 1)^2\right],$$

  where $xi \in$[-2.048, 2.048]

- Ackely's Function

$$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sqrt{\sum_{i=1}^{D}\cos(2\pi x_i)}\right) + 20 + e,$$

  where $x_i \in$[-32.768, 32.768]

- Griewank's Function

$$f(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

  where $x_i \in$[-600, 600]

• Rastrigin's Function

$f(x) = \sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i) + 10 \right)$,

where $x_i \in [-5.12, 5.12]$

We programmed the four algorithms in C++ language and ran the experiments on a Pentium IV 3.0GHz with 1GB RAM. Since the global optimum values of these functions are zero, we define the stopping criterion as

$$f < 10^{-8}, \tag{9}$$

where, f denotes the objective function value obtained by an algorithm. We also set the number of particles to 20 and the maximum number of function evaluation (NEF) to 10,000 $\times D$ where D is a dimensionality of a benchmark function.

<Table 1> presents the test results of ID-PSO and compares them with CPSO, FIPS and DMS-PSO. The first and second columns show the number of dimensions and the names of functions. Each algorithm is repeated for each function for 25 times as suggested by previous literature (Liang and Suganthan, 2005). The numbers in the remaining columns show the rate of successful runs out of 25 repetitions. If an algorithm finds a solution that satisfies the stopping criterion within the maximum NEF, we call this run *a successful run*. And the numbers in the parentheses is the number of successful runs out of 25 repetitions. The results show that DMS-PSO shows the highest overall success rate and ID-PSO is slightly worse than DMS-PSO. For two test functions, ID-PSO obtains more successful runs than other algorithms. For three functions, ID- PSO shows the same success rate with DMS-PSO. For Griewank(30), DMS-PSO obtains more successful runs than ID-PSO. One possible reason why DMS-PSO performs better for Griewank function is that the subpopulation of size m performs local search independently for R generations. Note that Localtelli (2003) showed that *Multistart algorithms*, which sample uniform random points over feasible region and perform independent local search from them, work well for the Greiwank function.

Second experiment compares the average objective function values obtained by each algorithm. In this experiment, we do not use the stopping criterion, i.e., all the function are evaluated 10,000×D times. <Table 2> summarizes the results. The average objective function values which are calculated to four decimal places are presented in <Table 2>. The numbers in the parentheses is the standard deviation. Compared to DMS-PSO, ID-PSO shows better performance for 6 functions on average. DMS-PSO performed better than ID-PSO on Rosenbrock's function only. It is worth noting that DMS-PSO outperforms other algorithms not only in average objective function values but also in standard deviation. This result implies that DMS-PSO consistently generates good solutions for Rosenbrock's function. For Rastrigin's function, ID-PSO consistently generates good solutions. Note that previous literature such as Kennedy(1999) also concluded that the effect of population topologies is largely dependent on the functions.

5. Conclusions

We have proposed a modified PSO called ID-PSO which uses a new population topology. The basic idea employed in ID-PSO is reducing the influence of the global best solution by adopting the theory of information diffusion which models

Table 1. The number of successful runs

Number of variables	Function Name	ID-PSO	CPSO	DMS-PSO	FIPS
10	Sphere(10)	**100%(25)**	92% (23)	**100%(25)**	**100%(25)**
10	Rosenbrock(10)	0% (0)	0% (0)	0% (0)	0% (0)
10	Ackley(10)	**100%(25)**	88% (22)	**100%(25)**	60% (15)
10	Griewank(10)	0% (0)	0% (0)	0% (0)	0% (0)
10	Rastrigin(10)	**4%(1)**	0% (0)	0% (0)	0% (0)
30	Sphere(30)	**100%(25)**	36% (9)	**100%(25)**	88% (22)
30	Rosenbrock(30)	0% (0)	0% (0)	0% (0)	0% (0)
30	Ackley(30)	**80%(20)**	0% (0)	72% (18)	0% (0)
30	Griewank(30)	20% (5)	4% (1)	**40%(10)**	8% (2)
30	Rastrigin(30)	0% (0)	0% (0)	0% (0)	0% (0)
Average Success Rate(Overall Success)		40.4% (101)	22% (55)	**41.2%(103)**	25.6% (64)

Table 2. The average objective function values

Number of variables	Function Name	ID-PSO	CPSO	DMS-PSO	FIPS
10	Sphere(10)	**0.0000 (0.000)**	**0.0000 (0.000)**	**0.0000 (0.000)**	**0.0000 (0.000)**
10	Rosenbrock(10)	2.0493 (2.5600)	40.7097 (199.3924)	**0.6929 (1.4793)**	32.8717 (108.3504)
10	Ackley(10)	**0.0000 (0.000)**	1.9319 (4.0317)	**0.0000 (0.000)**	0.6394 (1.6675)
10	Griewank(10)	**0.0552 (0.0225)**	1.0203 (4.4902)	0.0759 (0.0357)	0.8137 (0.6581)
10	Rastrigin(10)	**2.2434 (2.5088)**	13.5712 (7.7329)	7.4025 (3.1735)	35.6789 (9.8587)
30	Sphere(30)	**0.0000 (0.000)**	6.2915 (6.4100)	0.5243 (1.8146)	**0.0000 (0.000)**
30	Rosenbrock(30)	9325.0988 (20315.6063)	39180.2258 (41192.2252)	**18.2707 (59.7978)**	2395.3543 (10985.2994)
30	Ackley(30)	**0.7511 (2.3747)**	10.6339 (3.2773)	0.9804 (1.8383)	5.3765 (2.9050)
30	Griewank(30)	**0.0082 (0.0117)**	32.8931 (32.8443)	0.0120 (0.0126)	0.5386 (0.4959)
30	Rastrigin(30)	**47.5677 (24.9138)**	111.4934 (27.8326)	85.0162 (28.3894)	208.9554 (23.4295)

the transmission of information as one-to-one interactions between people. The strength of ID-PSO is its simplicity. Although the differences between ID-PSO and PSO variants such as CPSO and FIPS are relatively minor, our algorithm performs better. The performance of ID-PSO is also comparable to DMS-PSO which outperformed 7 recent PSO variants in the study of Liang and Suganthan (2005).

ID-PSO calls for further research in several aspects. While current ID-PSO selects two particles randomly for matching if they were not matched in last certain iterations, the performance of algorithm could be improved if different criteria are applied. For example, a particle may be matched with its closest particle. In addition, ID-PSO could be applied to other domains. Recently, PSO has been applied to various optimization problems such as bin packing (Liu *et al*., 2008), flow shop scheduling (Tseng and Liao), single machine scheduling (Anghinolfi and Paolucci, 2009) and binary classification (Unler and Murat, 2010). However, ID-PSO is designed for continuous variable functions and may not be suitable for discrete variable problems. Proper representation methods are needed for discrete variable problems.

References

Anghinolfi, D. and Paolucci, M. (2009), A new discrete particle swarm optimization approach for the single-machine total weighted tardiness scheduling problem with sequence-dependent setup times, *European Journal of Operational Research*, **193**(1), 73-85.

Clerc, M. and Kennedy, J. (2002), The particle swarm explosion, stability, and convergence in a multidimensional complex space, *IEEE Transaction on Evolutionary Computation*, **6**, 58-73.

Kawachi, K., Seki, M., Yoshida, H., Otake, Y., Warashina, K., and Ueda, H. (2008), A rumor transmission model with various contact interactions, *Journal of Theoretical Biology*, **253**(1), 55-60.

Kennedy, J. and Eberhart, R. C. (1995), Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks*, **IV**, 1942-1948.

Kennedy, J. (1999), Small worlds and mega-minds : effects of neighborhood topology on particle swarm performance, *Proceedings of IEEE Congress on Evolutionary Computation*.

Kennedy, J. and Mendes, R. (2002), Topological structure and particle swarm performance, *Proceedings of the Fourth Congress on Evolutionary Computation* (CEC-2002), Honolulu, Hawaii, 12-17.

Liang, J. J. and Suganthan, P. N. (2005), Dynamic multi-swarm particle swarm optimizer, *Proceedings of the IEEE International Swarm Intelligence Symposium*.

Liu, D. S., Tan, K. C., Huang, S. Y., and Goh, C. K. (2008), On solving multiobjective bin packing problems using evolutionary particle swarm optimization, *European Journal of Operational Research*, **190**(2), 357-382.

Locatelli, M. (2003), A note on the Griewank test function, *Journal of Global Optimization*, **25**, 169-174.

Løvbjerg, M., Rasmussen, T. K., and Krink, T. (2001), Hybrid particle swarm optimser with breeding and subpopulations, *Proceedings of the Genetic and Evolutionary Computation Conference*.

Mendes, R. (2004), Population topologies and their influence in particle swarm performance, PhD dissertation, University of Minho, Braga, Portugal.

Mendes, R., Kennedy, J., and Neves, J. (2004), The fully informed particle swarm : simpler, maybe better, *IEEE Transactions on Evolutionary Computation*, **8**(3), 204-210.

Thompson, M. (1979), Information diffusion in populations with immigration, *Information Sciences*, **17**, 113-130.

Tseng, C.-T. and Liao C.-J. (2008), A discrete particle swarm optimization for lot-streaming flowshop scheduling problem, *European Journal of Operational Research*, **191**(2), 360-373.

Unler, A. and Murat, A. (2010), A Discrete Particle Swarm Optimization Method for Feature Selection in Binary Classification Problems, *European Journal of Operational Research*, **206**(3), 528-539.