A Particle Swarm with Selective Particle Regeneration for Multimodal Functions

CHI-YANG TSAI, I-WEI KAO
Department of Industrial Engineering and Management Department
Yuan Ze University University
No. 135 Yuan-Tung Road
Taoyuan, Taiwan, R.O.C.
iecytsai@saturn.yzu.edu.tw http://e-enterprise.iem.yzu.edu.tw/html/group_teacher_2.html

Abstract: - This paper proposes an improved particle swarm optimization (PSO). In order to increase the efficiency, suggestions on parameter settings is made and a mechanism is designed to prevent particles fall into the local optimal. To evaluate its effectiveness and efficiency, this approach is applied to multimodal function optimizing tasks. 16 benchmark functions were tested, and results were compared with those of PSO, HNMPSO and GA-PSO. It shows the proposed method is both robust and suitable for multimodal function optimization.

Key-Words: - Particle Swarm Optimization, Cognitive and Social Parameter, Selective Particle Regeneration, Mutation Operation, Multimodal functions,

1 Introduction

As many optimization problems become more complex, stronger and more robust optimization algorithms are always needed. The advancement of powerful computers and evolutionary algorithms render the capability of solving the global optimization of complex system [1]. Among the considered tools are Meta-heuristics who solve a problem that involves an empirical search or optimization method. Classical meta-heuristic algorithms include Particle swarm optimization (PSO), Genetic algorithm (GA), Simulation algorithm (SA) and Ant Colony Optimization (ACO) etc. are the classical heuristic algorithms. Among these, Particle swarm optimization (PSO) is a new and popular stochastic optimization technique which developed by Kennedy and Eberhart [2]. The development of this algorithm follows from the bird and fish flocks moving and finding goals and foods in nature. There is always a leader as well as the best performance of the particles in the entire population. The leader leads the group in moving. All members of the group follow the leader. As above, the mechanism of PSO simulates this kind of behavior to search the optimal solution.

Compared with GA, PSO has some attractive characteristics. They have memory, so knowledge of the best solutions is retained by all the particles. The process of basic GA includes selectivity, crossover and mutation. Mutation is an important step which prevents the chromosome from falling into local minima. Similarly, there are two positive constants in PSO velocity and location update formula. They are the parameter of cognition and social, respectively. These parameters determine the relative effect of the convergence efficiency and the capability of escape local optimal. In general, both of them are setting to the same number.

Usually PSO is considered because of the ease of implementation and effectiveness. It can solve continuous problem and obtain the good performance. So far, PSO has been successfully applied in various fields. There were still many researchers who developed improved PSO. Most of them improved the performance accuracy, robustness or efficiency. This study is another attempt to further improve the performance of PSO.

The rest of the paper is organized as follows: Literature survey is discussed in Section 2. We will introduce methodology in Section 3, includes original PSO and SRPSO. The setting of the conducted numerical experiment is explained in Section 4. This section also shows the experimental results. Finally, the conclusion and future work is presented in the last section.

2 Literature Survey

Particle swarm optimization algorithm was first described in 1995 by James Kennedy and Russell C. Eberhart [2]. Nowadays, PSO has been widely applied in many research areas and real-world engineering fields, such as, task assignment and scheduling [3] [4], odor source localization [5], power plants [6], phase balancing [7], data clustering [8], image process [9], demand forecast [10] [11] , identification [12] and layout design [13] [14].

One of the first applications of PSO to multimodal problems was performed in 1998 by Kennedy [15]. In the paper the problem is to locate the global optimum in a fitness landscape with multiple local optima. The results of several versions of PSO where compared with the results of GA, the result shows that performs of PSO was better. In 2003, Susana and Carlos [16] presented two hybrid particle swarm optimization algorithms that incorporate a mutation operator similar to the one used in evolutionary algorithm. This hybridization of PSO improved the performance when dealing with multimodal functions.

Zielinski and Laur [17] found the appropriate parameter combination for a multi-objective Particle Swarm Optimization algorithm from Design of Experiments and interaction effects of different parameters were discovered. A adaptive control was applied to the parameters which are incorporated in the update equations of PSO.

Park et al. [18] proposed an improved hybrid PSO (HPSO), which combines the conventional PSO framework with the crossover operation of genetic algorithm. By applying the crossover operation in PSO, it not only discourages premature convergence to local optimum but also explores and exploits the promising regions in the search space effectively. In addition, Hao et al. [19] presented a crossover step is added to the standard PSO. The crossover is between each particle's individual best position. After the crossover, the fitness of the individual best position is compared with that of the two offspring, and the best one is selected as the new individual best position. The crossover can help the particles jump out of the local optimization by sharing the others' information. The experiment on five benchmark functions shows that the modified PSO is more effective to find the global optimal solution than other methods.

Coelho [20] presented a Quantum-behaved PSO (QPSO) using chaotic mutation operator. The application of chaotic sequences based on chaotic zaslavskii map instead of random sequences in QPSO is a powerful strategy to diversify the QPSO population and improve the QPSO's performance in preventing premature convergence to local minima. Finally they applied QPSO to solve a well-studied continuous optimization problem of mechanical engineering design. Ling et al. [21] also proposed new hybrid particle swarm optimization which incorporates a wavelet-theory-based mutation operation. It applied the wavelet theory to enhance the PSO in exploring the solution space more effectively for a better solution. This method applied to solve a suite of benchmark test functions and three industrial applications.

Fan et al. [22] developed the hybrid Nelder–Mead (NM)–Particle Swarm Optimization algorithm based on the NM simplex search method and PSO. Wang et al. [23] also developed a hybrid technique based on particle swarm optimization algorithm combined with the nonlinear simplex search method(HNM-PSO). This approach is applied to multimodal function optimizing tasks and compared with NS-PSO and CPSO. Kao and Zahara [24] proposed hybrid method which combining two heuristic optimization techniques, genetic algorithms and particle swarm optimization, for the global optimization of multimodal functions. Denoted as GA-PSO, this hybrid technique incorporates concepts from GA and PSO and creates individuals in a new generation not only by crossover and mutation operations as found in GA but also by mechanisms of PSO. The results of various experimental studies using a suite of 17 multimodal test functions taken from the literature have demonstrated the superiority of the hybrid GA-PSO approach over the other four search techniques in terms of solution quality and convergence rates. The authors also compared the result of GA-PSO and NM-PSO. The outcomes proved GA-PSO better than NM-PSO.

In this study, the parameter setting of PSO will be discussed and modified thoroughly. Besides, a novel and powerful mechanism will be designed. This improved algorithm will be introduced in next section.

3 Methodology

3.1 Particle Swarm Optimization

In nature, the birds and fishes flock to reach their food and goal. There is a leader who leads the group in moving. All members of the group follow the leader. PSO simulates a commonly observed social behavior, where particles of swarm tend to follow the global best particle.

Particles continuously update their velocity and position, and try to find optimal solutions. The procedure of PSO is described as follows:

1) Initialization: In a population of potential solutions, the population size is problem-dependent and the most commonly used in literature is often

between 20 and 50, and each particle is assigned a randomized velocity.

2) Velocity and Location Update: The particle's velocity (V_{id}^{new}) and position (x_{id}^{old}) are updated as follows:

$$V_{id}^{new} = w \times V_{id}^{old} + c_1 \times rand \times (p_{id} - x_{id}^{old}) + c_2 \times rand \times (p_{gd} - x_{id}^{old}) \quad (1)$$

$$x_{id}^{new} = x_{id}^{old} + V_{id}^{new} \quad (2)$$

where $w = [0.5 + rand / 2]$ is an inertia weight and $rand$ a uniformly generated random number between 0 and 1. c_1 and c_2 are two positive constants between 0 and 2. They are the parameters of cognition and social, respectively. P_{id} is the best location in the neighbourhood of the particle and P_{gd} is the global best location of all particles. V_{id}^{old} and x_{id}^{old} are the particle's previous velocity and position, respectively. x_{id}^{new} is the new location of a particle and can be updated by equation (2), after V_{id}^{new} is computed by equation (1)

3) Evaluation and Location Update of P_{id} and P_{gd}: The fitness value of each particle can be computed by the objective function. If the new value of P_{id} or P_{gd} is better than the old ones, the values of P_{id} and P_{gd} will be updated.

4) Termination: Step (2) and step (3) are repeated until the termination conditions are met.

3.2 Cognitive and Social Parameter Setting

A particle's new velocity is based on P_{id} and P_{gd} with the respective weights of c_1 and c_2. Clearly, c_1 and c_2 determine the relative effect of social knowledge (P_{gd} - x_{id}^{old})and cognitive knowledge (P_{id} - x_{id}^{old}). Usually, the two parameters are assigned the same value. If this is the case, new position of a particle will tend to be located between P_{id} and P_{gd}. It may slow down the convergence of particles toward to the global best location and thus affect the efficiency of the search.

A setting of assigning a larger value to c_2 with respect to c_1 is suggested in the proposed algorithm. This setting put more weight on P_{gd} when determining the new velocity of a particle. Thus, the new location of the particle will be closer to the global best location and the convergence is accelerated.

In order to observe the parameter setting which affects particle's convergence efficiency, the effect of the balance setting (c_1=1.5, c_2=1.5) is compared with unbalance setting (c_1=0.5, c_2=2.5) for multimodal functions problem (ES). The fitness convergent behavior and particles distribution of the balance parameter setting PSO and the unbalance parameter setting PSO are illustrated in Fig. 1 and Fig. 2.

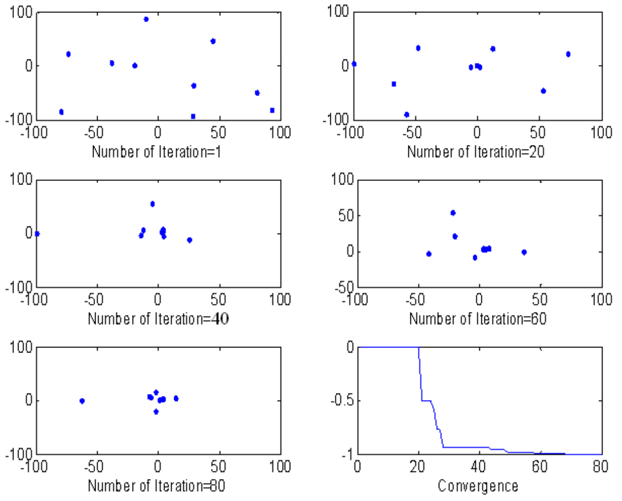

Fig. 1 Particles distribution and Fitness convergence behavior (Balance parameter setting of PSO)

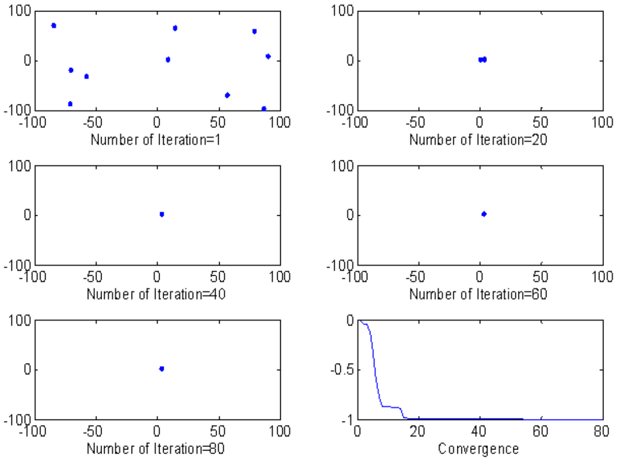

Fig. 2 Particles distribution and Fitness convergence behavior (Unbalance parameter setting of PSO)

The total number of iteration in the experiment is 80. The particles distribution is drawn when the iteration is 1, 20, 40, 60 and 80. The result of PSO is not completely converged until experiment end in Fig. 2. On the other hand, when the parameter setting is unbalance in PSO, the convergent efficiency increases and particle distribution decreases. The particles converge after about 20 iterations.

The result shows that the convergence is more effective when the parameter c_2 is set greater than c_1 in PSO. But this setting also increases the risk of particles fall into the local optimal. Therefore, the

"Selective Particle Regeneration" mechanism is designed in next section to prevent this situation happened.

3.3 Selective Regenerated Particle Swarm Optimization

In this paper, Selective Regeneration Particle Swarm Optimization (SRPSO) is proposed with two major modifications on PSO. Suggestion on setting of c_1 and c_2 is made and mechanism of regeneration of selective particles is designed. The procedure of SRPSO is illustrated Table 1.

3.4 Selective Particle Regeneration

The suggested parameter setting that c_2 is greater than c_1, may be able to improve the efficiency of convergence, but it also increases the risk of particles falling into local optimums. Therefore, a "Selective Particle Regeneration" mechanism is designed. It is a new operation in which is similar to the mutation mechanism in GA. Generally speaking, as a particle becomes closer to local optimal location, the possibility of this particle escaping from it decreases, especially with the suggested parameter setting. The "Selective Particle Regeneration" mechanism first computes the distance, in terms of fitness value, between a particle and global best particle (P_{gd}). For particles with distances to the global best particle smaller than a predetermined value, *f*, *d*% of these particles will be randomly selected and regenerated.

The purpose of particle regeneration is to help some of the particles that are close to the global best particle escape from local optimum if the current global best particle represents a local optimal solution. However, the current global best particle may still contain valuable knowledge that may lead to better solutions. Therefore, partial knowledge carried by the global best particle will be adopted when generating new locations of the selected particles. More specifically, when determining the value of a specific dimension for the new location of a particle, the value of the same dimension of the current global best location is adopted with a probability of c. With a probability of $(1- c)$, the value is randomly generated.

Finally, it is desired for these regenerated particles not to move toward the global best particle right away. Therefore, as opposed to setting c_2 to be greater than c_1 as suggested previously, c_1 is given a value larger than c_2 instead when determining the new velocities of the regenerated particles. By doing so, greater weight is assigned to cognitive knowledge. This setting, however, applies only to the determination of velocities for particles that are just regenerated. The setting of c_1 and c_2 remains as suggested in section 3.2.

Table 1 The procedure of SRPSO

```
1. Randomly population initialization
2. Fitness Evaluation
3. For each particle{
4.    If the particle is regenerated
5.       Setting 1.(Set c2 to be greater than c1)
6.    Else
7.       Setting 2.(Set c1 to be greater than c2)
8.    End
9.    Velocity and Location Update
10. }
11. For each particle{
12.    If the particle is close to Pgd
13.       Selective Particle Regeneration
14.          Distance Calculation
15.          Particle Selection
16.          Particle Regeneration
17.          Parameter Modification
18.    End
19. }
20. If the termination condition is met
21.    Stop
22. Else
23.    Go to line 2.
24. End
```

4 Experiment and Result

4.1 Experiment Setting

In order to evaluate the performance of the proposed algorithm, SRPSO and PSO are applied to solve continuous multimodal function problems. The SRPSO and PSO algorithm were coded in Matlab 2007a and the simulations were run on an AMD 1.7G CPU with memory capacity 1024 MB. Each test was performed 30 times for PSO and SRPSO. The termination condition is that the number of function evaluation is reached.

16 benchmark multimodal functions were selected for the experiment. The collection provides a balance of simple and difficult functions. These functions have been used in various particle swarm studies [25] [26]. Table 2 summarizes the characteristics of these benchmark functions. Functions were implemented at a lowest dimension

of 2 and at a highest dimension of 30. In all cases, there are minimization problems. E.G. is the Error Goal which is the search accuracy. If the fitness is lower than E.G., it will be considered and marked as “success”. X_{min} and X_{max} are the boundary of search space for each benchmark function.

Table 2 Parameters for each test function

Function	Branin	Easom	Shubert
Dimension	2	2	2
E.G.	10^{-6}	10^{-6}	10^{-6}
X_{min}	-5	-100	-10
X_{max}	15	100	10
Function	Zakharov	Hartmann	Rastrigin
Dimension	2, 10	3	10, 30
E.G.	10^{-6}, 10^{-3}	10^{-3}	10^{-6}, 10^{-3}
X_{min}	5	0	5.12
X_{max}	10	1	5.12
Function	Griewank	Sphere	Power
Dimension	10, 30	20, 30	20
E.G.	10^{-3}, 10^{-1}	10^{-2}, 10^{-1}	10^{-2}
X_{min}	-300	100	-1
X_{max}	600	100	1
Function	Rotated	Ackley	Schwefel
Dimension	20	30	30
E.G.	10^{-2}	10^{-1}	10^{-1}
X_{min}	-65.536	-30	-500
X_{max}	65.536	30	500

4.2 Experiment Setting

After thorough tests and experiments, we proposed the parameters setting for SRPSO in Table 3. N is the dimension of benchmark function. F.E. is the function evaluation. It is also determined by N which is $50 \times N \times$ Particle Size. The initial population is randomly generated. Both parameter c_1 and c_2 are 1.5 in PSO.

Table 3 Parameter setting

Parameter	c_1	c_2	f	d
Value	0.5	2.5	E.G.$\times$10	40%
Parameter	c	Partice Size	F. E.	
Value	30%	5 N	$50\times N\times$ Particle Size	

4.3 Experimental Result

The results for the all benchmark functions are provided in Table 4 to Table 6. Fig. 4 to Fig. 7 provide more insight into the convergence behaviors of PSO and SRPSO. Opt. is the optimal solution of each benchmark functions. If the value is smaller than 10^{-6}, we assume that is very close to zero. It will be replaces with “0”. We also provide the convergence figures for some benchmark functions. We have used various benchmark functions to compare SRPSO and PSO.

As can be seen clearly, SRPSO outperformed PSO in all benchmark functions. The average, worst, best values of SRPSO are pretty close to the optimal solutions. The relatively low standard deviation shows the robustness of SRPSO. For low-dimension or simple problems, the performances of SRPSO and PSO are not much different. But for high-dimensions or complex problems, the performances of SRPSO are obvious better than PSO. Fig. 3 to Fig. 6 present the convergence behaviors of function Branin, Rastrigin, Rotated hyper and Schwefel. Most of outcomes of PSO exhibit slower convergence. On the other hand, the convergence of SRPSO is faster and the solution are close to the global optimum. Especially in Rastrigin, SRPSO converges to the global optimum in about 200 iterations. PSO does not convergence until experiment termination which is 500 time iterations.

We applied SRPSO and PSO methods to calculate the success rate and convergence using CPU times and average number of function evaluation value which is shown in Table 7. For all benchmark function, SRPSO exhibits a significantly high success rate compared with PSO. CPU time and the number of function evaluation constitute the computation cost. Even if PSO needs less CPU time and Function Evaluation in Easom and Shubert, the results do not reach to global optimum in these two functions. In constrast, SRPSO successfully found them in both functions. Aside of Easom and Shubert, the outcomes of SRPSO are superior to PSO in every aspect. Especially in complex or high dimension function, SRPSO clearly reduces computation cost. Therefore, we conclude that SRPSO is more efficient, robust and accurate than PSO in multimodal function problem.

4.4 Comparison with other methods

SRPSO is compared with two Hybrid method in this section. The parameters of SRPSO are the same to Table 2. The number of function evaluation is equal to the references. The Hybrid Nelder Mead and Particle swarm optimization (HNMPSO) [23] and hybrid Genetic algorithms and Particle swarm optimization (GA-PSO) [24] were developed by Wang et al. and Kao et al. These two algorithms have been applied to solve continuous multimodal function. In order to compare SRPSO with HNMPSO and GA-PSO, we applied SRPSO to solve 9 and 17 test functions in these experiments. Experimental data obtained from these test functions are given in Table 8 and Table 9. Average error is the average of the errors between the best successful points found and the known global optimum. In Table 8, the Numbers of Function Evaluation (1) is set the same to the reference. SRSPO obtains the obviously superior performances to HNMPSO. In order for further comparison, we reduce the number of function evaluation in SRPSO (Numbers of Function Evaluation (2)) and also obtain the better outcomes. The results of GA-PSO and SRPSO are shown in Table 9. There are 17 benchmark functions, more than half outcomes of SRPSO are better than GA-PSO. We conclude that SRPSO approach remains quite competitive as compared to HNMPSO and GA-PSO.

Table 4 Fitness value for low dimension test function

Method		Branin (2)	Easom (2)	Shubert (2)	Zakharov (2)	Hartmann (3)
	Opt.	0.3978	-1	-186.7309	0	-3.8634
SRPSO	Avg.	**0.3978**	**-1**	**-186.7309**	**0**	**-3.8634**
	Worst	**0.3978**	**-1**	**-186.7309**	**0**	**-3.8634**
	Best	**0.3978**	**-1**	**-186.7309**	**0**	**-3.8634**
	Std	**0**	**0**	**0**	**0**	**0**
PSO	Avg.	0.5352	**-1**	-185.6815	**0**	-3.8617
	Worst	1.7596	**-1**	-182.5775	**0**	-3.8524
	Best	0.3979	**-1**	-186.7309	**0**	-3.8634
	Std	0.4302	**0**	1.3510	**0**	0.0034

Table 5 Fitness value for 10-dimension and 20-dimension test function

Method		Rastrigin (10)	Zakharov (10)	Griewank (10)	Sphere (20)	Power (20)	Rotated hyper(20)
	Opt.	0	0	0	0	0	0
SRPSO	Avg.	**0**	**0**	**0**	**0**	**0**	**0**
	Worst	**0**	**0**	**0**	**0**	**0**	**0**
	Best	**0**	**0**	**0**	**0**	**0**	**0**
	Std	**0**	**0**	**0**	**0**	**0**	**0**
PSO	Avg.	3.2733	**0**	**0**	**0**	**0**	0.0102
	Worst	5.9445	**0**	1.01e-5	**0**	**0**	0.0766
	Best	1.9899	**0**	**0**	**0**	**0**	7.0e-4
	Std	1.6553	**0**	**0**	**0**	**0**	0.0235

Table 6 Fitness value for 30-dimension test function

Method		Sphere (30)	Rastrigin (30)	Griewank (30)	Ackley (30)	Schwefel (30)
	Opt.	0	0	0	0	0
SRPSO	Avg.	**0**	**0**	**0**	**0**	4e-4
	Worst	**0**	**0**	**0**	**0**	4e-4
	Best	**0**	**0**	**0**	**0**	4e-4
	Std	**0**	**0**	**0**	**0**	**0**
PSO	Avg.	**0**	21.98	**0**	**0**	6.02e+3
	Worst	**0**	32.83	**0**	**0**	7.41e+3
	Best	**0**	11.93	**0**	**0**	4.31e+3
	Std	**0**	7.45	**0**	**0**	946.80

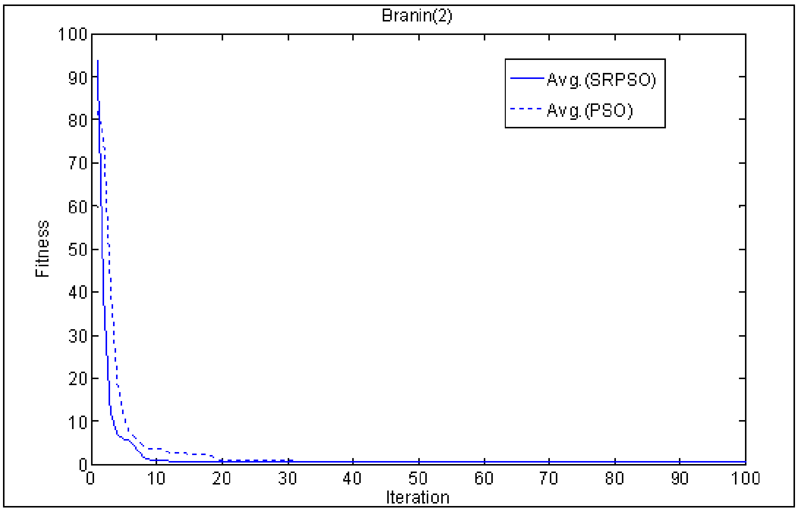

Fig. 3 Comparison of convergence behaviors in Branin for SRPSO and PSO

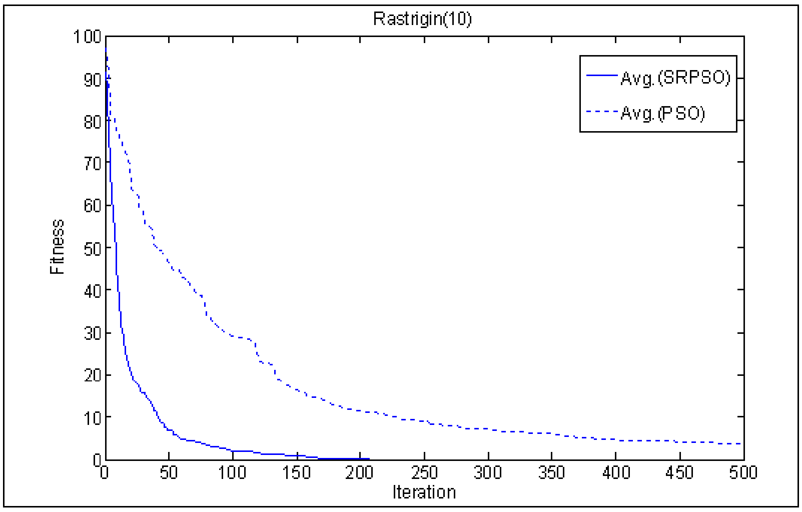

Fig. 4 Comparison of convergence behaviors in Rastrigin for SRPSO and PSO

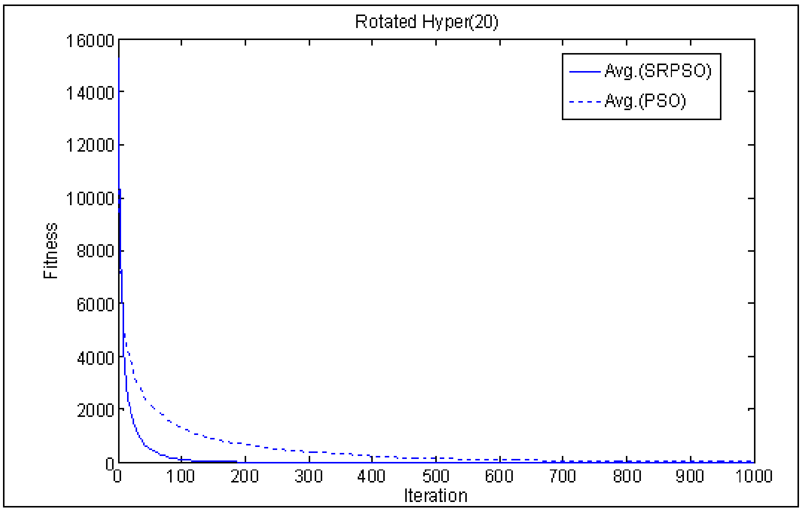

Fig. 6 Comparison of convergence behaviors in Rotated hyper for SRPSO and PSO

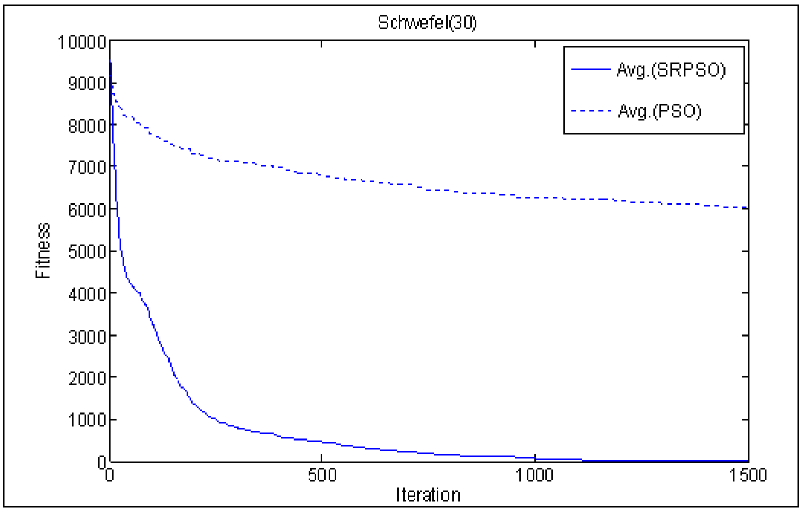

Fig. 7 Comparison of convergence behaviors in Schwefel for SRPSO and PSO

Table 7 The rate of success, CUP time and mean function evaluations for each test function

Test function	Success Rate		CPU Time		Function Evaluation	
	SRPSO	PSO	SRPSO	PSO	SRPSO	PSO
Branin(2)	**100%**	8%	**0.1993**	0.2524	**320**	456
Easom(2)	**100%**	16%	0.1966	**0.1054**	240	**156**
Shubert(2)	**100%**	28%	0.3521	**0.2374**	902	**525**
Zakharov(2)	**100%**	12%	**0.2370**	0.2485	**429**	511
Hartmann(3)	**100%**	**100%**	**0.3517**	0.4881	**915**	1005
Rastrigin(10)	**100%**	20%	**1.8109**	7.5093	**12830**	25452
Zakharov(10)	**100%**	52%	**1.0695**	3.560	**6338**	12826
Griewank(10)	**100%**	56%	**0.9091**	1.6760	**3465**	4055
Sphere (20)	**100%**	**100%**	**2.8095**	7.5478	**13373**	13437
Power(20)	**100%**	**100%**	**3.1918**	3.3049	**10522**	11782
Rotated hyper(20)	**100%**	90%	**27.1361**	81.9268	**45778**	90764
Sphere(30)	**100%**	**100%**	**6.0869**	11.3652	**32158**	56165
Rastrigin(30)	**100%**	6%	**17.0125**	34.0801	**86476**	124354
Griewank(30)	**100%**	**100%**	**9.2486**	17.1423	**27263**	45935
Ackley(30)	**100%**	**100%**	**11.1187**	18.5299	**36659**	55352
Schwefel(30)	**100%**	0%	**23.9948**	25.9907	**117762**	125638

Table 8 Results provided by SRPSO and HNM-PSO for 9 test functions

Test function	Numbers of Function Evaluation(1)	Average Fitness		Numbers of Function Evaluation(2)	Average Fitness
		SRPSO	HNMPSO		SRPSO
Branin(2)	1466	**0.39789**	**0.39789**	1000	**0.39789**
Easom(2)	2599	**-1**	**-1**	1600	**-1**
Shubert(2)	14727	**-186.7309**	**-186.73**	2000	**-186.7309**
Zakharov(2)	1089	**6.0915e-9**	4.567e-7	800	**2.5354e-8**
Hartmann(3)	3889	**-3.8634**	**-3.8634**	2000	**-3.8634**
Shekel(4)	8348	**-10.5364**	-8.0113	6000	**-10.5364**
Hartmann(6)	6135	**−3.32237**	-3.265	3000	**−3.32237**
Rastrigin(10)	5418	**8.3050**	9.7901	4000	**11.6010**
Zakharov(10)	22036	**3.14e-8**	9.1628e-7	10000	**8.33e-7**

Table 9 Results provided by SRPSO and GA-PSO for 17 test functions

Test function	Number of Function Evaluation	Average Error	
		SRPSO	GA-PSO
Branin RCOC(2)	8254	**8.6e-8**	9e-5
Easom(2)	809	1.2e-2	**3e-5**
Goldstein Price(2)	25706	**3.1e-12**	1.2e-4
Branin(2)	174	1e-2	**1e-5**
Shubert(2)	96211	**3.9e-13**	7e-5
Rosenbrock(2)	140894	**1e-10**	6.4e-4
Zakharov(2)	95	1e-2	**7e-5**
De Joung	206	**1.e-5**	4e-5
Hartmann(3)	2117	**2.5e-6**	2.0e-4
Shekel(5)	529344	**3.8e-6**	1.4e-4
Shekel(7)	56825	**2.2e-5**	1.5e-4
Shekel(10)	43314	**9e-5**	1.2e-4
Rosenbrock(5)	1358064	**4.8e-6**	1.3e-4
Zakharov (5)	398	1.2e-3	**1.3e-4**
Hartmann(6)	12568	**1.1e-4**	2.4e-4
Rosenbrock(10)	5319160	8e-4	**5e-5**
Zakharov(10)	872	2.5e-4	**0**

4 Conclusion and Future work

In this paper, an improved Particle Swarm Optimization algorithm, SRPSO, was proposed. There are two major modifications in this algorithm. In order to increase the efficiency, the "Cognitive and Social Parameter Setting" is modified in the first stage, after that, "Selective Particle Regeneration" was designed to prevent particles fall into the local optimal.

The SRPSO was thoroughly investigated and applied to solve continuous multimodal function optimization. 16 wide variety benchmark multimodal functions were selected for the experiment and compared to original PSO. The results include the average, worst, best, and standard deviation of fitness value and convergence behaviors. SRPSO is completely to improve global optimality of the solution attained. The outcome presents that SRPSO is better than PSO in every aspect. We also compared with other competitive methods which were developed by some researchers. The results lead us to allege that SRPSO is an efficient, accurate, and robust method for continuous multimodal function optimization problem.

Future work may focus on investigating the best parameter setting and reduce the parameter in SRPSO. Furthermore, SRPSO may be applied to the areas of engineering process control, data cluster, image process, data pattern and simulation and identification. Finally, SRPSO is a robust and accurate method for continuous problem, solving discrete problem would also be worth studying further.

Appendix

Branin (2 variables):

$$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7;$$

Easom (2 variables):

$$f(x) = -\cos(x_1)\cos(x_2)\exp(-((x_1 - \pi)^2 ;$$

Hartmann (3 variables):

$$f(x) = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2 \right];$$

Shubert (2 variables):

$$f(x) = \left\{ \sum_{j=1}^{5} j\cos[(j+1)x_1 + j] \right\} \times \left\{ \sum_{j=1}^{5} j\cos[(j+1)x_2 + j] \right\};$$

Zakharov (2, 10 variables):

$$f(x) = \left(\sum_{j=1}^{n} x_j^2 \right) + \left(\sum_{j=1}^{n} 0.5 j x_j \right)^2 + \left(\sum_{j=1}^{n} 0.5 j x_j \right)^4 ;$$

Rastrigin (R_n) (10, 30 variables):

$$f(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10);$$

Griewank (10, 30 variables):

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

Sphere (20, 30 variables):

$$f(x) = \sum_{i=1}^{n} x_i^2 \;;$$

Power (20 variables):

$$f(x) = \sum_{i=1}^{n} \left| x^{i+1} \right| ;$$

Rotated Hyper (20 variables):

$$f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_i^2 \right)^2 \;;$$

Ackley (30 variables):

$$f(x) = \text{-}20\exp(\text{-}0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n} \text{Cos}(2\pi x_i)) + 20 \cdot \text{e} \;;$$

Schwefel (30 variables):

$$f(x) = 418.9829 \cdot n + \sum_{i=1}^{n} x_i \,\text{Sin}(\sqrt{|x_i|}) \;;$$

References:

[1] S. K. S. Fan and E. Zahara, A hybrid simplex search and particle swarm optimization for unconstrained optimization, *European Journal of Operational Research*, Vol.108, No.2, 2007, pp. 527-548.

[2] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *Neural Networks, 1995 Proceedings., IEEE International Conference*, New Jersey, 1995, pp. 1942-1948.

[3] B. Liu, L. Wang, and Y. H. Jin, An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers, *Computers and Operations Research*, Vol.35, No.9, 2008, pp. 2791-2806.

[4] D.Y. Sha, and C. Y. Hsu, New particle swarm optimization for the open shop scheduling problem, *Computers and Operations Research*, Vol.35, No.10, 2008, pp. 3243-3261.

[5] W. Jatmiko, P. Mursanto, B. Kusumoputro, K. Sekiyama and T. Fukuda, Modified PSO algorithm based on flow of wind for order source localization problems in dynamic environments, *WSEAS Transactions on Systems*, Vol.7, No.2, 2008, pp. 106-113.

[6] L.D. Li, J. Zhou, Y. Yu and X. Li, Constrained power plants unit loading optimization using particle swarm optimization algorithm, *WSEAS Transactions on Information Science and Applications*, Vol.4, No.2, 2007, pp. 296-302.

[7] Y. Tuppadung and W. Kurutach, Particle swarm optimization application for phase balancing, *WSEAS Transactions on Systems*, Vol.5, No.8, 2006, pp. 1976-1981.

[8] Y. T. Kao, E. Zahara and I. W. Kao, A hybridized approach to data clustering, *Expert Systems with Applications*, Vol.34, No.3, 2008, pp. 1754-1762.

[9] E. Zahara, S. K. S. Fan and D. M. Tsai, Optimal multi-thresholding using a hybrid optimization approach, *Pattern Recognition Letters*, Vol.26, No.8, 2005, pp. 1082-1095.

[10] Ü. Alper, Improvement of energy demand forecasts using swarm intelligence: The case of Turkey with projections to 2025, *Energy Policy*, Vol.36, No.6, 2008, pp. 1937-1944.

[11] L. Gao, L. Yang, C. Zhou and Y. Hu, Category forecast application of neural network algorithm trained by particle swarm optimization, *Computer Integrated Manufacturing Systems*, Vol.12, No.3, 2006, pp. 465-469.

[12] F. A. Guerra and L. d. S. Coelho, Multi-step ahead nonlinear identification of Lorenz's chaotic system using radial basis neural network with learning by clustering and particle swarm optimization, *Solitons and Fractals*, Vol.35, No.5, 2008, pp. 967-979.

[13] S. Ȯnüt, U.R. Tuzkaya, and B. Doğaç, A particle swarm optimization algorithm for the multiple-level warehouse layout design problem, *Computers and Industrial Engineering*, Vol.54, No.4, 2008, pp. 783-799.

[14] Y. Zeng, C. A. Zhu, L. G. Shen, J. Y.Qi, Discrete optimization problem of machine layout based on swarm intelligence algorithm, *Computer Integrated Manufacturing Systems*, Vol.13, No.3, 2007, pp. 541-552.

[15] J. Kennedy and W. Spears, Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator, In *Proceedings of IEEE Congress on Evolutionary Computation (CEC 1998)*, 1998, pp. 74-77.

[16] C. E. Susana, A. C. C. Carlos, On the Use of Particle Swarm Optimization with Multimodal Functions, *Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003)*, 2003, pp. 1130-1136.

[17] K. Zielinski and R. Laur, Adaptive parameter setting for a multi-objective particle swarm

optimization algorithm, *2007 IEEE Congress on Evolutionary Computation*, *CEC 2007*, No. 4424856, 2007, pp. 3019-3026

[18] J. B. Park; Y. W. Jeong, J. R. Shin, K.Y. Lee and J. H. Kim, A Hybrid Particle Swarm Optimization Employing Crossover Operation for Economic Dispatch Problems with Valve-point Effects, *Intelligent Systems Applications to Power Systems*, *2007 ISAP 2007. International Conference* ,2007, pp.5-8.

[19] Z. F. Hao, Z. G. Wang and H. Huang, A Particle Swarm Optimization Algorithm with Crossover Operator, *2007 International Conference on Machine Learning and Cybernetics*, Vol.2, 2007, pp. 1036-1040.

[20] L. Coelho, A quantum particle swarm optimizer with chaotic mutation operator, *Chaos, Solitons & Fractals*, Vol.37, No.5, 2008, pp. 1409-1418.

[21] S. H. Ling, H. H. C. Iu, K. Y. Chan, H. K. Lam, B. C. W. Yeung, and F. H. Leung, Hybrid Particle Swarm Optimization With Wavelet Mutation and Its Industrial Applications**,** *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS*, Vol.38, No.3, 2008, pp. 743-763.

[22] S. K. S. Fan, Y. C. Liang and E. Zahara, Hybrid simplex search and particle swarm optimization for the global optimization of multimodal functions, *Engineering Optimization*, Vol.36, 2004, pp. 401-418.

[23] F. Wang, Y. Qiu and Y. Bai, A new hybrid NM method and particle swarm algorithm for multimodal function optimization" *Lecture notes in computer science 2005,*Vol.3646, 2005, pp.497-508.

[24] Y. T. Kao and E. Zahara, A hybrid genetic algorithm and particle swarm optimization for multimodal functions, *Applied Soft Computing*, Vol. 8, 2007, pp. 849-857.

[25] J. Kennedy, Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance, *Proceedings of IEEE Congr*ess *on EVolutionary Computation*, Washington, Vol.3, 1999, pp.1931-1938.

[26] J. Kennedy and R. Hlendes, Population structure and particle swarm performance, *Proceedings of the IEEE Congress on EVolutionary Computation*, Honolulu, 2002, pp.1671-1676.