2009 International Workshop on Nonlinear Circuits and Signal Processing
NCSP'09, Waikiki, Hawaii, March 1-3, 2009

Particle Swarm Optimization Containing Plural Swarms

Masaki Sugimoto, Taku Haraguchi, Haruna Matsushita and Yoshifumi Nishio

Tokushima University, Japan
Phone:+81-88-656-7470, Fax:+81-88-656-7471
{sugimoto, taku, haruna, nishio}@ee.tokushima-u.ac.jp

Abstract

In this study, we proposes a modified particle swarm optimization (PSO) called plural PSO (PPSO) for bust-out the local optimum solution. The feature of PPSO is that the swarm of PPSO is not one but plural. The plural swarms share an information of the best position in each swarm. Except the swarm including the best particle in whole swarms, all the particles are repositioned to escape the local optima. We investigate behaviors of PPSO and confirm its efficiency in multimodal functions.

1. Introduction

Particle swarm optimization (PSO) [1] is a popular optimization technique for the solution of object function and is an evolutionary algorithm to simulate the movement of flocks birds and the movement of a school fishes toward foods. Due to the simple concept, such as easy implementation and quick convergence, PSO has gained much attentions and wide applications in different fields in resent years. However, PSO has demonstrated a great performance for many problems yet its fast convergence often leads to premature convergence in local optima. The tradeoff between fast convergence and being trapped in local optima will be even more critical in multimodal functions having many local optima very close to each other. In order to escape from the local optima and avoid premature convergence, the search for global optimum should be diverse. Many researchers have improved the performance of PSO by enhancing its ability with a more diverse search [2]-[6]. Specifically, some researchers have proposed the PSO using multiple swarms, and exchange information among them [7], [8].

Meanwhile, in the real world, a large company reshuffles the personnel regularly. Furthermore, the company members work not as a group but as a organization consisting of some departments. Therefore, they can work more effective by exchanging informations among the departments in the company.

In this study, we propose a new PSO algorithm containing plural swarms (called PPSO). The proposed algorithm reflects the strategy of the firm for the business in the real world. The feature of PPSO is that its swarm is not one but plural. The swarms share an information of the best position in each swarm. Except the swarm including the best particle in whole swarms, all the particles are repositioned to escape the local optima. We confirm the effectiveness of the PPSO algorithm by numerical experiments.

2. Plural PSO (PPSO) Algorithm

The most important feature of PPSO is that PPSO has K swarms, i.e. plural particle groups. Each swarm is denoted as S_k $(k = 1, 2, \ldots, K)$ and consists of N/K particles. The position vector of each particle i and its velocity vector are represented by $\boldsymbol{X}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and $\boldsymbol{V}_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, respectively, where $(i = 1, 2, \ldots, N)$ and $x_{id} \in [x_{\min}, x_{\max}]$.

[PPSO1](Initialization) Let $t = 0$ and $t_R = 0$, i.e., t is the simulation step and t_R is the time step for the reposition. Randomly initialize the particle position $\boldsymbol{X}_i$ and its velocity $\boldsymbol{V}_i$ for all particles i and initialize $\boldsymbol{P}_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ with a copy of $\boldsymbol{X}_i$. Evaluate the objective function $f(\boldsymbol{X}_i)$ for all particle i and find $\boldsymbol{P}_g$ with the best function value among the all particles. Attach each particle i to any swarm S_k at random and find $\boldsymbol{P}_{s_k} = (p_{s_k 1}, p_{s_k 2}, \ldots, p_{s_k D})$ with the best function value among the particles belonging to the swarm S_k;

$$s_k = \arg\min_i \{f(\boldsymbol{X}_i)\}, \qquad i \in S_k. \tag{1}$$

[PPSO2] Evaluate the current cost $f(\boldsymbol{X}_i)$. Update the personal best position (called *pbest*) $\boldsymbol{P}_i$ for each particle i and the global best position *gbest* $\boldsymbol{P}_g$ among the all particles in all the swarms so far (called *gbest*). For each particle i belonging to each swarm S_k, update the swarm best position $\boldsymbol{P}_{s_k}$ with the best cost among the particles belonging to the swarm S_k so far (called *sbest*), if needed.

[PPSO3](Update) Update the velocity vector $\boldsymbol{V}_i$ and the position $\boldsymbol{X}_i$ of each particle i depending on its *pbest*, *gbest* and *sbest* according to

$$\begin{aligned} v_{id}(t+1) &= w v_{id}(t) + c_1 r_1 \{p_{id} - x_{id}(t)\} \\ &\quad + c_2 r_2 \{p_{s_k d} - x_{id}(t)\} + c_3 r_3 \left\{p_{g_d} - x_{id}(t)\right\}, \quad i \in S_k, \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1), \end{aligned} \tag{2}$$

where $d = 1, 2, \ldots, D$, r_1, r_2 and r_3 are three random variables distributed uniformly in $[0, 1]$, w is an inertia weight, and c_1, c_2, and c_3 are positive acceleration coefficients.

[PPSO4] If $t_R = T/K$, we perform [PPSO5], if not, we perform [PPSO6]. Thus, we perform [PPSO5] every time when T/K simulation steps are performed. T is the maximum number of the simulation.

[PPSO5](Reposition) Reposition the positions $\boldsymbol{X}_i$ of the particles belonging to all the S_k except the best swarm at random and reassign their velocities $\boldsymbol{V}_i$ at random. We reset $t_R = 0$.

[PPSO6] Let $t = t + 1$ and $t_R = t_R + 1$. Go back to [PPSO2], and repeat until $t = T$.

3. Numerical Experiments

In order to confirm the performance of PPSO algorithm, we carry out performed basic numerical experiments. The goal is to find the optimum (minimum) value of $f(x)$. Referring to [2], we use the following four benchmark functions.

1. Sphere function:

$$f_1(x) = \sum_{d=1}^{D-1} x_d^2, \quad (3)$$

where $x \in [-2.048, 2.047]^D$ and the optimum solution x^* are all $[0, 0, \ldots, 0]$.

2. Rosenbrock's function:

$$f_2(x) = \sum_{d=1}^{D-1} (100(x_d^2 - x_{d+1})^2 + (1 - x_d)^2), \quad (4)$$

where $x \in [-2.048, 2.047]^D$ and the optimum solution x^* are all $[1, 1, \ldots, 1]$.

3. Rastrigin's function and its optimum (minimum):

$$f_3(x) = \sum_{d=1}^{D} (x_d^2 - 5\cos(2\pi x_d) + 5), \quad (5)$$

where $x \in [-5.12, 5.12]^D$ and the optimum solution x^* are all $[0, 0, \ldots, 0]$.

4. Griewank's function:

$$f_4(x) = \sum_{d=1}^{D} \frac{x_d^2}{4000} + \prod_{d=1}^{D} \cos\left(\frac{x_d}{\sqrt{d}}\right) + 1, \quad (6)$$

where $x \in [-600, 600]^D$ and the optimum solution x^* are all $[0, 0, \ldots, 0]$.

The optimum function values $f(x^*)$ of all function are 0. f_1 and f_2 are unimodal functions, and f_3 and f_4 are multimodal functions with numerous local minima. All the functions have D variables. In this study, D is set to 30 and 100 to investigate the performances in various dimensions.

In order to evaluate the efficiency of PPSO and to investigate behaviors of PPSO, we compare the three algorithms that PSO1, PSO2 and PPSO. PSO1 is the standard PSO and PPSO is the proposed algorithm explained in Section 2. PSO2 is the standard PSO with only the reposition process of PPSO.

The population size N is set to 40 in PSO1 and PSO2. PPSO have $K = 4$ swarms, and each swarm contains 10 particles, i.e. $N = 40$. For PSO1, PSO2 and PPSO, the inertia weight is fixed as $w = 0.6$. For PSO1 and PSO2, the acceleration coefficients are set as $c_1 = c_2 = 1.8$. For PPSO, the parameters are set as $c_1 = 1.8, c_2 = 1.4$ and $c_3 = 0.4$. The timing of reposition is set as $t_R = 500$.

We carry out the simulation 30 times for all the optimization functions with 2000 generations, namely $T = 2000$. The

Table 1: Comparison results PSO1, PSO2 and PPSO on test functions with $D = 30$.

f	Methods	Mean	Minimum	Maximum
f_1	PSO1	7.92e-40	2.53e-43	7.61e-39
	PSO2	6.62e-29	4.26e-31	4.74e-28
	PPSO	**2.84e-54**	4.33e-56	2.84e-53
f_2	PSO1	25.08	3.64	76.45
	PSO2	**23.92**	0.34	90.84
	PPSO	24.94	3.37	80.68
f_3	PSO1	27.92	10.89	40.59
	PSO2	16.04	6.93	32.67
	PPSO	**12.67**	6.93	19.80
f_4	PSO1	1.15e-02	0	4.18e-02
	PSO2	8.20e-03	0	5.15e-02
	PPSO	**3.29e-04**	0	9.86e-03

Table 2: Comparison results PSO1, PSO2 and PPSO on test functions with $D = 100$.

f	Methods	Mean	Minimum	Maximum
f_1	PSO1	8.01e-05	1.66e-06	1.21e-03
	PSO2	**1.33e-05**	1.72e-06	4.16e-05
	PPSO	3.57e-04	1.13e-05	1.29e-03
f_2	PSO1	6.17e+04	1.54e+02	9.24e+05
	PSO2	3.42e+04	2.06e+02	8.42e+05
	PPSO	**2.87e+02**	1.61e+02	4.34e+02
f_3	PSO1	258.63	208.88	324.70
	PSO2	188.42	121.97	247.79
	PPSO	**123.62**	91.08	167.32
f_4	PSO1	3.53e-03	2.05e-08	2.22e-02
	PSO2	3.94e-03	4.08e-08	2.46e-02
	PPSO	**2.41e-03**	6.51e-07	4.64e-02

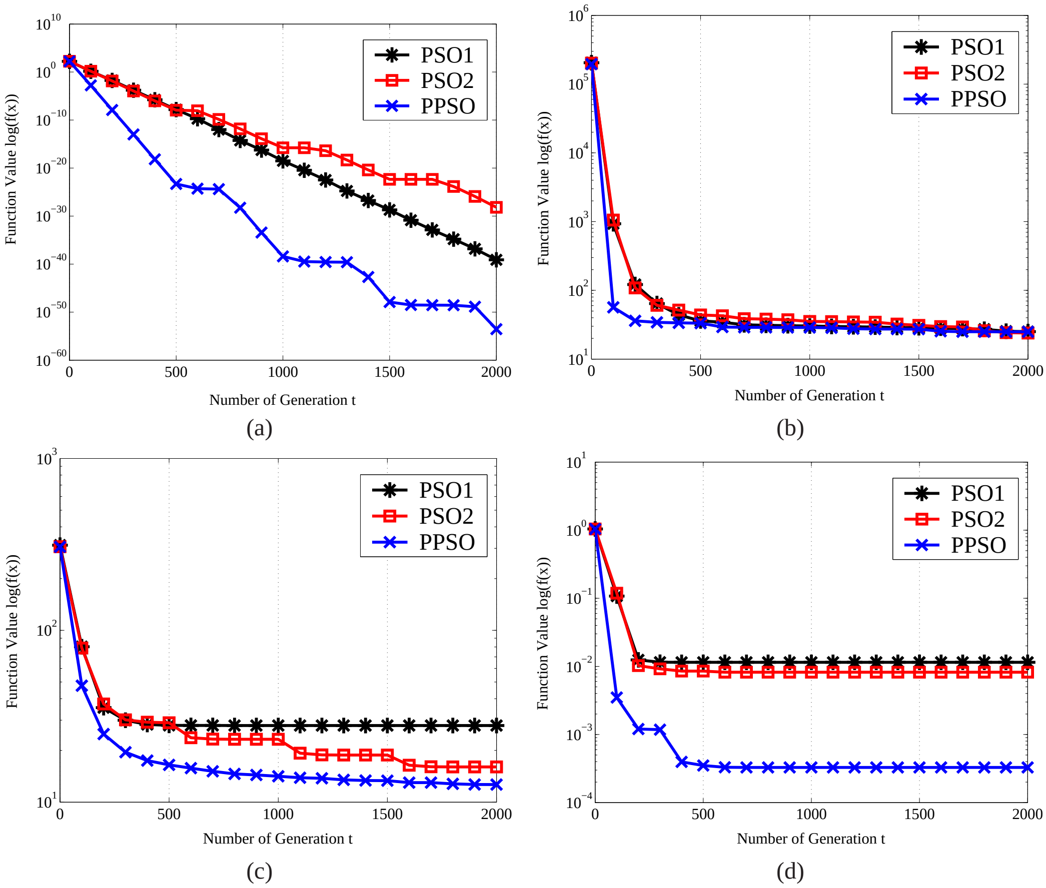

Figure 1: Mean *gbest* value of every generation for 30-dimensional four functions. (a) Sphere function. (b) Rosenbrock's function. (c) Rastrigin's function. (d) Griewank's function.

performances with minimum, maximum and mean function values on four functions over 30 runs with 30-dimension are listed in Table 1. We can see that the mean values of PPSO are the best in f_1, f_3 and f_4. However, in case of the performances on the test functions with $D = 100$ shown in Table 2, PPSO are the best in f_2, f_3 and f_4. From these results, we can say that PPSO is more effective for multimodal functions.

Figure 1 shows the mean *gbest* value of every generation over 30 runs for four test functions with 30 variables. The performances of PSO1 for the all functions do not achieve the good solution, instead, the proposed PPSO can obtain the best results 3 times. Meanwhile, the mean *gbest* values for 100 dimensional test functions are shown in Fig. 2. The performances of PPSO for the function f_1 as Figs. 1(a) is the worst values in the three algorithms. However, for the functions f_2, f_3 and f_4 as Figs. 2(b)–(d), PPSO can obtain the best results. In particular, PPSO greatly improves the performance from PSO1 on f_1, f_3 and f_4 with $D = 30$, and f_2 and f_3 with $D = 100$. Because PPSO has plural swarms and searches with sharing an information of the swarm best position *sbest*. Therefore, PPSO can find better position quickly in early stage of the optimization. On the other hand, we can see that the graphs of PSO2 and PPSO are waving by the effect of the reposition to search better position. However, because the performances of the proposed PPSO are better than PSO2, we can say that PPSO, which exchanges the information between some swarms, is more effective than PSO2 which is the standard PSO with only the reposition process. In other words, these effects avert the premature convergence and the particles of PPSO easily escape from the local optimum.

From these results, we can confirm that PPSO is more effective for multimodal function.

4. Conclusions

In this study, we have proposed the new PSO algorithm, Plural PSO (PPSO). PPSO has plural swarms, and all the swarms share the information of their the best position. We have investigated its behaviors with the simulation and have

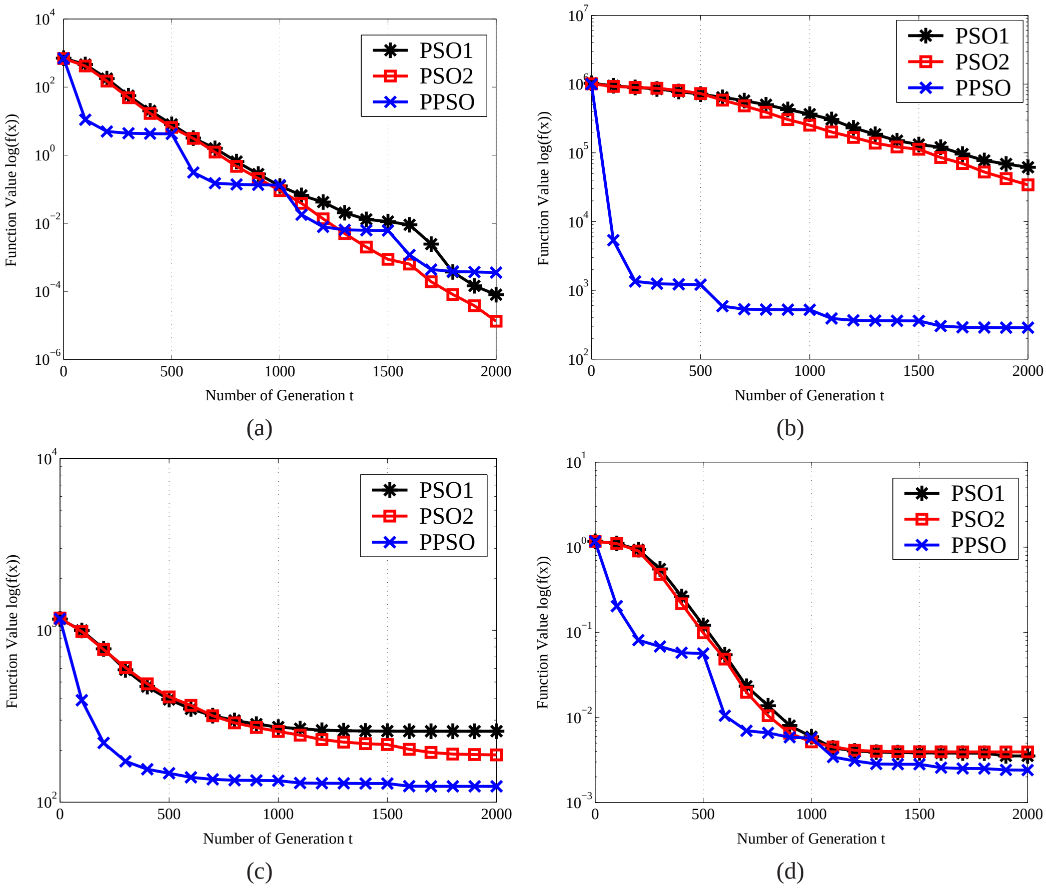

Figure 2: Mean *gbest* value of every generation for 100-dimensional four functions. (a) Sphere function. (b) Rosenbrock's function. (c) Rastrigin's function. (d) Griewank's function.

confirmed the efficiency of PPSO.

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization", in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, pp. 1942–1948, 1995.

[2] J. Kennedy and R. Medes, "Population structure and particle swarm performance" in *Proc. Congr. Evol. Comput.*, pp. 1671-1676, 2002.

[3] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients" in *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 240-255, 2004.

[4] F. V. d. Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization" in *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225-239, 2004.

[5] T. J. Richer and T. M. Blackwell, "The Levy Particle Swarm" in *Proc. Congr. Evol. Comput.*, pp. 3150-3157, 2006.

[6] E. Miyagawa and T. Saito, "Particle swarm optimizers with grow-and-reduce structure" in *IEEE Congress on Evolutionary Computation.*, pp. 3975-3980, 2008.

[7] G. Yen and M. Daneshyari, "Diversity-based Information Exchange among Multiple Swarm in Particle Swarm Optimization" in *IEEE Congress on Evolutionary Computation.*, pp. 6150-6157, 2006.

[8] M. Iwamatsu, "Multi-Species Particle Swarm Optimizer for Multimodal Function Optimization" in *IEICE Trans. Inf. & Syst.*, vol. E89-D, no. 3, 2006.