2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Particle Swarm Optimization Containing Plural Kinds of Swarms

Masaki Sugimoto, Taku Haraguchi, Haruna Matsushita and Yoshifumi Nishio

Department of Electrical and Electronic Engineering, Tokushima University,
2-1 Minami-Josanjima, Tokushima 770-8506, Japan
Email: {sugimoto, taku, haruna, nishio}@ee.tokushima-u.ac.jp

Abstract— This study proposes a new Particle Swarm Optimization (PSO) algorithm containing plural swarms whose particles have different features. Each particle of the proposed PSO belongs to one of plural swarms, which have different characteristics, and the particle periodically changes the swarm that it belongs to. We confirm the effectiveness of the proposed PSO for both unimodal and multimodal functions with various dimensions.

1. Introduction

Particle Swarm Optimization (PSO) [1] is a popular optimization technique for solving objective functions and PSO is an evolutionary algorithm to simulate the movement of flocks of birds toward foods. Due to its simple concept, easy implementation and quick convergence, PSO has attracted attentions and has been widely applied to different fields in recent years. Furthermore, PSO has demonstrated great performances for many problems. However, quick convergence often leads to local optimum problem. It is important for multimodal functions with a lot of local optima to compromise between the quick convergence and being trapped in a local optimum. In order to escape from such local optima and to avoid the premature convergence, the search for a global optimum should be diverse. Many researchers have improved the performance of PSO by enhancing its ability with more diverse search [2]-[6]. In particular, some researchers have proposed the PSO method using multiple swarms whose particles exchange their information among them [7][8].

On the other hand, in real world, a company performs its business in an effective way. Each member of a company belongs to a department which has a determined role. Some departments cooperate each other to achieve a common goal. Assignment to a department is decided by a member's ability and/or outcome.

In our past study, we have proposed a modified PSO algorithm using plural swarms [9]. This algorithm contained plural swarms whose features were the same, and the swarms shared the information of the best position among them. Furthermore, all the particles were repositioned periodically. However, the effectiveness of using plural swarms has not been clearly understood.

In this study, we propose an improved PSO algorithm using plural swarms; PSO containing plural swarms whose particles have different features (called PPSO). The proposed algorithm reflects the concept of a company in the real world. The proposed PPSO does not have the reposition process and each swarm has different features, unlike our past algorithm. The important features of PPSO are that each particle of PPSO belongs to one of plural swarms, which have different characteristics, and are periodically reconstituted. At every generation, each particle is ranked by its cost among all the particles in all the swarms. Furthermore, the particles periodically change the swarm that it belongs to, according to its total ranking.

We explain the algorithm of PPSO in detail in Section 2. In Section 3, we perform basic numerical experiments by using four algorithm methods of PSOs including PPSO. Furthermore, we confirm the efficiency of PPSO for high dimensional functions.

2. PSO containing plural swarms whose particles have different features (PPSO)

Each particle of PSO has two information; position and velocity. 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 $(d = 1, 2, \ldots, D)$, $(i = 1, 2, ..., N)$ and $x_{id} \in [x_{\min}, x_{\max}]$.

The important features of PPSO are that each particle of PPSO belongs to one of plural swarms, which have different characteristics, and each particle periodically changes the swarm that it belongs to. Each swarm is denoted as S_k $(k = 1, 2, \ldots, K)$ and has N/K particles. The position vector of each particle i and its velocity vector are updated depending on its personal best position, the best position among all the particles and the best position among the particles belonging to the swarm. At every generation, each particle is ranked by its cost $f(\boldsymbol{X}_i)$ among all the particles in all the swarms. At every T_c generation, the particle changes the swarm, that it belongs to, according to its total ranking R_i.

[PPSO1] (Initialization) Let a generation step $t = 0$ and $t_c = 0$, i.e., t_c is the time step for the reconstitution of the swarms. Randomly initialize the particle position $\boldsymbol{X}_i$ and its velocity $\boldsymbol{V}_i$ for all particles i and initialize $\boldsymbol{P}_i = (p_{i1}, p_{i2}, ..., p_{iD})$ with a copy of $\boldsymbol{X}_i$. Evaluate the objective function $f(\boldsymbol{X}_i)$ for each 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.

[PPSO2] Evaluate the current cost $f(\boldsymbol{X}_i)$. Update the per-

sonal best position *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.

[PPSO3] Let $\boldsymbol{P}s_k = (p_{s_k1}, p_{s_k2}, \cdots, p_{s_kD})$ represents the swarm best position with the best cost among the particles belonging to the swarm S_k so far (called *sbest*). Update $\boldsymbol{P}s_k$ for each swarm S_k, if needed.

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

[PPSO4] Assign each particle i a rank $r = 1, \cdots, N$ depending on its cost $f(\boldsymbol{X}_i)$. The best rank of 1 indicates the particle with the smallest function value, and the worst rank of N indicates the particle with the largest function value. r_i denotes the sum of the rank of particle i so far; $r_i^{\text{new}} = r_i^{\text{old}} + r_i$.

[PPSO5] Updated $\boldsymbol{V}_i$ and $\boldsymbol{X}_i$ of each particle i depending on its *pbest*, its swarm best *sbest* and *gbest*;

$$\begin{aligned} v_{id}(t+1) =& w_k v_{id} + c_1 r_1 \{p_{id} - x_{id}(t)\} \\ &+ c_2 r_2 \{p_{s_kd} - x_{id}(t)\} + c_3 r_3 \{p_{gd} - x_{id}(t)\}, \\ x_{id}(t+1) =& x_{id}(t) + v_{id}(t+1), \end{aligned} \tag{2}$$

where r_1, r_2 and r_3 are random variables distributed uniformly on $[0, 1]$, and c_1, c_2 and c_3 are positive acceleration coefficients. w_k is an inertia weight of the each swarm S_k. In other words, each swarms has different features depending on w_k, and the particles belonging to different swarms behave differently.

[PPSO6] If $t_c = T_c$, perform [PPSO7]. If not, perform [PPSO10]. Thus, perform [PPSO7] every T_c generation. T_c is a fixed parameter and is the number of generations to reconstitute the swarms.

[PPSO7] Rank each particle i as $R = 1, \cdots, N$ according to its r_i. R_i denotes the total rank of the particle i. $R_i = 1$ is the best total rank and indicates the particle i with the smallest r_i.

[PPSO8] Reconstitute the respective swarms according to the total rank R_i of each particle i. Each swarm k contains $\frac{N}{K}$ particles, and the best swarm S_1 consists of the particles with high total rank R_i from 1 to $\frac{N}{K}$. In other words, the each swarm S_k includes the particles with the total rank R_i from $\left(\frac{(k-1)N}{K} + 1\right)$ to $\frac{kN}{K}$.

[PPSO9] Initialize the past total rank for all the particles, namely reset $R_i = 0$ for all the particles, and reset $t_c = 0$.

[PPSO10] Let $t = t+1$ and $t_c = t_c+1$. Go back to [PPSO2], and repeat until $t = T$.

3. Numerical Experiments

In order to confirm the performance of PPSO algorithm, we have performed basic numerical experiments. The problem is finding the optimum (minimum) value of $f(x)$ in the algorithm. We use the following four bench marks [2].

1. Sphere function:

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

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

2. Rosenbrock's function:

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

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):

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

where $x \in [-5.12, 5.12]^D$ and the optimum solution x^* are all $[0, 0, \ldots, 0]$. We consider that an almost optimum value is obtained if the algorithm attains the criterion $f_1(x) = 100$. This criterion is based on [2]. The range of initialization is $-5.12 \le x_i \le 5.12$.

4. Griewank's function:

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

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 functions 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 four algorithms;PSO1, PSO2, PPSO-R and PPSO. PSO1 is the standard PSO and PPSO is the proposed algorithm explained in Section 2. In PSO2, the inertia weight w is not the fixed value unlike the PSO1 and monotonically decreases with the generations as

$$w(t) = w_{\max} - \frac{w_{\max} - w_{\min}}{T} \times t, \tag{7}$$

where $w_{\max}$ and $w_{\min}$ are the maximum and minimum value of $w(t)$, respectively. In order to investigate the effect of omitting [PPSO6] to [PPSO9] in Section. 2, we compare PPSO with PPSO-R algorithm which does not have the ranking evaluation and the reconstitute process of the swarms. The inertia weight w of PPSO-R is the same

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

f		PSO1	PSO2	PPSO-R	PPSO
f_1	Avg.	**9.04e-61**	2.19e-33	8.24e-25	1.13e-29
	Min.	7.38e-66	1.16e-46	6.37e-39	8.56e-35
	Max.	1.45e-59	4.42e-32	2.24e-23	6.19e-29
f_2	Avg.	**20.34**	30.16	29.00	21.57
	Min.	0.06	2.13	15.08	0.06
	Max.	73.78	87.73	76.94	76.51
f_3	Avg.	31.05	22.44	15.75	**14.95**
	Min.	13.86	15.84	6.93	5.94
	Max.	46.53	30.69	28.81	22.77
f_4	Avg.	8.60e-03	1.14e-02	2.37e-03	**1.31e-03**
	Min.	0	0	0	0
	Max.	5.62e-02	6.87e-02	2.16e-02	1.48e-02

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

f		PSO1	PSO2	PPSO-R	PPSO
f_1	Avg.	1.08e-06	2.20e-01	3.48	**9.92e-10**
	Min.	2.18e-11	4.76e-04	2.08e-01	1.88e-15
	Max.	3.00e-05	1.01	18.41	1.28e-08
f_2	Avg.	17035.82	747.08	5478.69	**179.83**
	Min.	98.53	367.54	514.89	77.57
	Max.	503344.7	1958.46	14010.7	311.71
f_3	Avg.	254.68	307.93	208.83	**138.48**
	Min.	197.00	200.48	152.66	95.26
	Max.	324.70	401.12	314.86	190.07
f_4	Avg.	7.47e-03	5.29e-03	5.78e-02	**2.71e-03**
	Min.	6.92e-12	1.15e-04	1.67e-02	6.66e-16
	Max.	2.22e-02	2.76e-02	1.81e-01	2.22e-02

as PPSO, which has different value by each swarm; however, all the particles keep on the constitution of the initial swarm.

The population size N is set to 60 in PSO1 and PSO2. PPSO-R and PPSO have $K = 6$ swarms, and each swarm contains 10 particles, i.e. $N = 60$. For PSO1, the inertia weight is fixed as $w = 0.6$ which is a mean value of all the inertia weights used in PPSO-R and PPSO. For PSO2, w is decreased with time according to Eq. (7) where $w_{\max} = 0.9$ and $w_{\min} = 0.4$. For PSO1 and PSO2, the acceleration coefficients are set as $c_1 = c_2 = 1.8$. Because PPSO-R and PPSO consist of the plural swarms whose features are different, we use difference inertia weight on each swarm. For PPSO-R and PPSO, the parameters are set as $c_1 = 1.8, c_2 = 1.4, c_3 = 0.4, w_1 = 0.9, w_2 = 0.8, w_3 = 0.7, w_4 = 0.6, w_5 = 0.5$ and $w_6 = 0.4$. The timing of reconstruction is set as $T_c = 100$.

We carry out the simulation 30 times for all the optimization functions with 3000 generations, namely $T = 3000$. The performances with minimum, maximum and mean function values on four functions with 30-dimension are listed in Table 1. We can see that the mean values of PPSO are the best in only f_3 and f_4 which are the two multimodal functions. However, in case of the performances on the test functions with $D = 100$ shown in Table 2, PPSO can obtain the best mean values for all the test functions. In particular, PPSO greatly improves the performance from PSO1 on f_1, f_2 and f_3 with $D = 100$. From these results, we can say that PPSO is more effective for higher dimensional functions.

Figure 1 shows the mean *gbest* values of every generation over 30 runs for four test functions with 30 variables. For the unimodal functions f_1 and f_2 as Figs. 1(a) and (b), PSO1, which is the standard PSO, can obtain the best results. However, the performances of PSO1 for the multimodal functions f_1 and f_2 as Figs. 1(c) and (d) are bad values, instead, the proposed PPSO can obtain the best results. Meanwhile, the mean *gbest* values for 100 dimensional test functions are shown in Fig. 2. The performances of PPSO are the best values in all the four test functions, and PSO1 cannot obtain good results for both the unimodal functions and the multimodal functions. Because it is difficult for PSO1 to find the optimum solution on the high-dimensional functions, PSO1 are easily trapped in the local optima and prematurely converge.

Let us consider PPSO-R and PPSO which have plural swarms and search with sharing the information of the swarm best position *sbest*. The particles belonging to the different swarms have the different inertia weight w_k, namely, their behaviors are different. The search range of PPSO-R and PPSO are wider than PSO1 because there are some kinds of particles in PPSO-R and PPSO and they are updated depending on its *pbest*, *sbest* and *gbest*. However, we can see that PPSO-R cannot find the optimum solution. Then, because PPSO-R does not have the reconstitute process of swarms, the attracting force of PPSO-R to the global best position *gbest* is weak even if the particles belong to the swarm whose *sbest* is far from the optimum solution. On the other hand, in PPSO, the particles moving around the area, where is very far from the optimum solution, are attracted toward *gbest* by the reconstitute process which is to redefine the role of respective particles. In other words, the particles of PPSO can search the solution at the wide region and in depth.

Meanwhile, PSO2 also uses the different inertia weight in the simulation. However, the inertia weights of all the particles simultaneously decrease with generation, and PSO2 cannot achieve the good results.

From these results, we can say that PPSO is the effective algorithm, which has the plural kinds of particles and whose swarm situation is periodically reconstituted, for the complex problems such as the multimodal and the high-dimension.

4. Conclusions

In this study, we have proposed a new Particle Swarm Optimization algorithm (PPSO) containing the plural swarms whose particles had different features. The im-

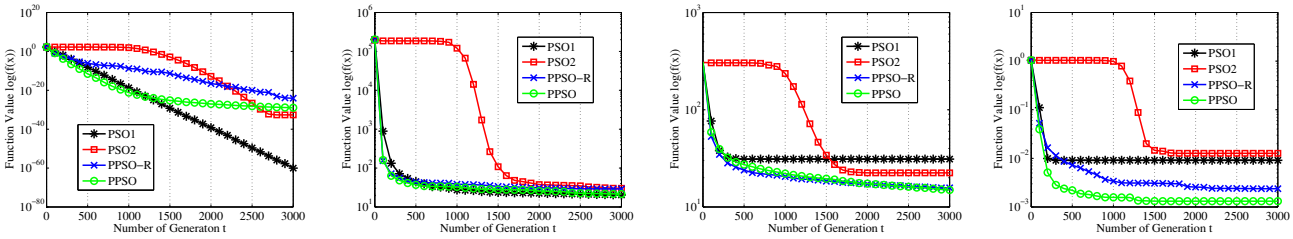

Figure 1: Mean *gbest* value of every generation for 30-dimensional four functions. (a) Sphere function. (b) Rosenbrock's function. (c) Rastrigin's function. (d) Griewank's function.

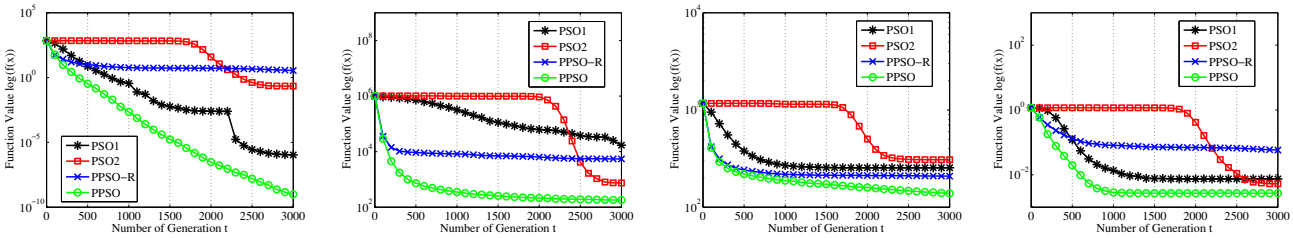

Figure 2: Mean *gbest* value of every generation for 100-dimensional four functions. (a) Sphere function. (b) Rosenbrock's function. (c) Rastrigin's function. (d) Griewank's function.

portant features of PPSO were that each particle of PPSO belonged to one of plural swarms, which had different characteristics, and periodically changed the swarm that it belongs to. We have performed the basic numerical experiments for various dimensions by using four algorithm methods of PSOs including PPSO. We have confirmed that PPSO could obtain the effective results especially for the complex problems such as the multimodal function and high-dimension.

References

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

[2] J. Kennedy and R. Medes, "Population structure and particle swarm performance," *Proc. IEEE Congress on Evolutionary Computation*, vol. 2, 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," *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[4] F.V.d. Bergh and A.P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[5] T.J. Richer and T.M. Blackwell, "The Levy particle swarm," *Proc. IEEE Congress on Evolutionary Computation*, pp. 808–815, 2006.

[6] E. Miyagawa and T. Saito, "Particle swarm optimizers with grow-and-reduce structure," *Proc. IEEE Congress on Evolutionary Computation*, pp. 3974–3979, 2008.

[7] G. Yen and M. Daneshyari, "Diversity-based information exchange among multiple swarm in particle swarm optimization," *Proc. IEEE Congress on Evolutionary Computation*, pp. 1686–1693, 2006.

[8] M. Iwamatsu, "Multi-species particle swarm optimizer for multimodal function optimization," *IEICE Trans. on Information and Systems*, vol. E89-D, no. 3, pp. 1181–1187, 2006.

[9] M. Sugimoto, T. Haraguchi, H. Matsushita and Y. Nishio, "Particle swarm optimization containing plural swarms," *Proc. RISP Int. Workshop on Nonlinear Circuits and Signal Processing*, pp. 419–412, 2009.