

A Pair-wise Bare Bones Particle Swarm Optimization Algorithm for Nonlinear Functions

Jia Guo¹ Yuji Sato²

¹ Graduate School of Computer and Information Science, Hosei University,
3-7-2 Kajinocho,
Koganei-shi, Tokyo 184-8584, Japan
E-mail: guojia@ieee.org

² Faculty of Computer and Information Sciences, Hosei University,
3-7-2 Kajinocho,
Koganei-shi, Tokyo 184-8584, Japan
E-mail: yuji@hosei.ac.jp

Abstract

Bare bones particle swarm optimization is a parameter-free swarm intelligence algorithm which is famous for easy applying. It has aroused wide concern of academic circle on its principles and applications in recent years. However, losing the diversity quickly still causes the premature convergence in the iteration process. Hence, a pair-wise bare bones particle swarm optimization algorithm is proposed in this paper to balance the exploration and exploitation. Moreover, a separate iteration strategy is used in pair-wise operator to enhance the diversity of the swarm. A pair of particles will be placed in different groups and will be applied with different evolutionary strategies. Also, to verify the performance of the proposed algorithm, a set of well-known nonlinear benchmark functions are used in the experiment. Furthermore, several evolutionary algorithms are also evaluated on the same functions as the control group. Finally, the experiment results and statistical analysis confirm the performance of PBBPSO with nonlinear functions.

Keywords: Bare bones, particle swarm optimization, pair-wise, diversity increasing

1. Introduction

Evolutionary algorithms (EA) are widely used in wireless networks¹, milling operations² and other different area. As an important branch of EA, swarm intelligence algorithms attracted many scholars by the virtue easy applying and high performance. At the beginning, particle swarm optimization (PSO) algorithm is first proposed by Kennedy and Eberhard³ in 1995. This population-based algorithm is inspired by birds flocking and fish schooling. Par-

ticles exchange information including position and velocity to calculate their evolve trend. Compared with other optimization algorithms, PSO has an advantage at convergence. Moreover, its high performance and easy applying have attracted the attentions from plenty researchers in both benchmark test and real-world engineering problems. To increase the exploring ability, plenty of researchers engaged in this field. For instance, Clerc⁴ used the message from a particle's neighbor to enhance the search ability. Trelea⁵ proposed a research about conver-

gence analysis and parameter selection of the PSO. Moreover, the particle swarm method is widely used in applications. For example, Eberhart⁶ proposed the research about the PSO in the engineering area.

In 2003, Kennedy⁷ proposed the Bare Bones Particle Swarm Optimization (BBPSO), which is a simple version of PSO. The standard BBPSO is originally formulated as a means of studying the particle distribution of PSO. It cancels the velocity and uses a Gaussian distribution to sample the searching space. Particles generate with a normal distributed random number around the mean of personal best position and global best position on each dimension. During the iteration process, the personal and global best position keep detecting and exploring in the search area. Moreover, parameter-free means the algorithm can easily adapt to different problems. Hence, both varies BBPSO and numbers of methods based on it are proposed for real world applications. Omran⁸ proposed a clustering method based on bare bones. The proposed algorithm finds the centroids of a user specified the number of clusters, where each cluster groups together similar patterns. Moreover, the application of the proposed clustering algorithm to the problem of unsupervised classification and segmentation of images is investigated.

However, the BBPSO still lacks at premature convergence and quickly losing diversity. To solve this problem, a pair-wise bare bones particle swarm optimization algorithm is proposed in this paper. Moreover, the rest of the paper is organized as follows: Section II gives a brief review of studies connected to PSO and BBPSO; Section III gives an introduction about the proposed algorithm; Section IV introduces the experiment to verify the performance of the proposed algorithm; Section V gives the conclusion of this paper.

2. Related work

2.1. Particle swarm optimization

Particles in particle swarm optimization (PSO) is an abstract conception. It is proposed for nonlinear functions. Particles in PSO change their velocity by gaining information from its personal best and group best position. The next position of a particle

is calculated by following equation:

$$\begin{aligned} v_{t+1}(i) &= w * v_t(i) + r_1 c_1 * (x_{best}(i) - x_t(i)) \\ &\quad + r_2 c_2 * (g_{best} - x_t(i)) \\ x_{t+1}(i) &= x_t(i) + c_3 v_{t+1} \end{aligned} \quad (1)$$

where $x_{best} = (x_{best}(1), x_{best}(2), \dots, x_{best}(n))$ is a matrix for recording the best position each element has ever reached; g_{best} is the best position that all element has ever reached; t is the iteration time; r_1 and r_2 are random number from 0 to 1; w , c_1 , c_2 and c_3 are the system parameter. And according to⁹, c_1 and c_2 are usually set as 2.05; c_3 is usually set as 0.7298.

To increase the performance of PSO, a fully informed particle swarm (FIPS) is proposed by Mendes¹⁰. Particles in FIPS are affected by all of their neighborhoods rather than they are only affected by best neighborhoods in PSO. The FIPS has been applied in multimodal optimization problem and is improved in 2006¹¹.

Liang proposes a comprehensive learning particle swarm optimizer (CLPSO), which uses a novel learning strategy whereby all other particles historical best information is used to update a particles velocity. The proposed strategy enables the diversity of the swarm to be preserved to discourage premature convergence¹².

2.2. Bare bones particle swarm optimization

Bare bones particle swarm optimization (BBPSO) is a simple version of PSO. A particle's next position is only calculated by its personal best position and swarm global best position. Along with the cancel of the velocity, BBPSO does not need any parameter anymore. Moreover, in order to speed up convergence, Kennedy⁷ introduce the interaction probability (IP) to BBPSO. Particles will iterate with following equation:

$$\begin{aligned} \mu &= \frac{p_i + g_{best}}{2} \\ \sigma &= |p_i - g_{best}| \\ x(t+1) &= \begin{cases} N(\mu, \sigma) & \text{if } (\omega > 0.5) \\ p_i & \text{else} \end{cases} \end{aligned} \quad (2)$$

where $P = (p_1, p_2, \dots, p_n)$ is the best position of each particle; g_{best} is the best position of the whole swarm; ω is a random number from 0 to 1. This means BBExp can save half of the time during the calculation. Moreover, according to ⁷, BBExp gives better results than some other versions of PSO in some benchmark functions.

To increase the accuracy during the optimization process, a bare bone particle swarm optimization with an integration of global and local learning strategies is proposed by Chen ¹³. Moreover, Blackwell formulates the dynamic update rule of particle swarm optimization. This rule is expressed as a second-order stochastic difference equation. Also, general relations are derived for search focus, search spread, and swarm stability at stagnation. The relations are introduced to standard PSO, its variant and BBPSO. Also, The simplicity of the Bare Bones swarm facilitates theoretical analysis, and a further no-collapse condition is derived, according to Blackwell's research ¹⁴.

In order to minimize the effects of the control parameters, in the research of Wang, a Gaussian bare-bones differential evolution (GBDE) and its modified version (MGBDE) are proposed ¹⁵. The original differential evolution (DE) is a population-based stochastic optimization algorithm, which has already shown the noteworthy performance on both real-world and benchmark optimization problems. However, the DE is affected by several significant parameters in the algorithm. To solve this problem, the proposed algorithm presents some eliminating and dynamically adjusting strategies.

Krohling introduces a jump strategy to bare bones particle swarm to solve the problem that BBPSO shows weak performance with functions with many local minima in high dimension area. The jump strategy is implemented based on the Gaussian or the Cauchy probability distribution. When there is no improvement of fitness value, the proposed will try to help the algorithm jump out of the current local wave. To verify the ability of the proposed algorithm, a set of well-known benchmark functions are used in the experiment. Simulation results show that the BBPSO with the jump strategy performs well in all functions investigated ¹⁶. More-

over, Chen insists that in the BBPSO, if a particle is restricted to move to a new position only when the new position is better than its original position, the particle then retains the best position it ever found. Based on this observation, all personal best particles are no longer required. Hence a revised BBPSO is proposed by Chen. The proposed algorithm tend to eliminate personal best particle. This strategy makes the use of memory more efficient utilization, especially when dealing with large scale problems or in microprocessor based applications ¹⁷.

Campos proposes a BBPSO with scale matrix adaptation (SMA-BBPSO). The proposed algorithm aims at solving the premature convergence problem of BBPSO with a strategy that selecting next position of a particle from a multivariate t -distribution with a rule for adaptation of its scale matrix. Also, to verify the searching ability of proposed algorithm, a set of well-known benchmark functions are used in the experiment. Moreover, a theoretical analysis is developed to explain how SMA-BBPSO works ¹⁸.

Richer Points out that many foragers and wandering animals follow the Lévy distribution of steps. The Lévy distribution is introduced to both PSO and BBPSO algorithm. From the Lévy distribution BBPSO, long step and short step are combined to explore the searching area. Moreover, in the comparative experiment, proposed strategies show powerful ability for searching the global best position ¹⁹.

Vafashoar points out that BBPSO is highly prone to premature convergence and stagnation. To solve this problem, cellular learning automata bare bones particle swarm optimization (CLA-BBPSO), a new multi swarm version of BBPSO is proposed in his research. Different with standard BBPSO, several probability distributions are used in the proposed algorithm. Hence the diversity of particles increase. Moreover, a new approach for adaptive determination of covariance matrices, which are used in the updating distributions, has been proposed in the research. As the result that the searching ability of CLA-BBPSO has been confirmed by experiments, to improve the convergence speed can be the main future work ²⁰.

However, most of the above researchers aim at introduce some new methods or new strategies to

BBPSO. The improvement of their results are exchanged by the calculation times. In addition, combines the BBPSO with inappropriate methods or harsh parameter may make the algorithm hard to apply. Hence, to balance the searching ability and the calculating speed, a simple and parameter free algorithm will be proposed in next section.

3. Proposal of PBBPSO

To increase the diversity of the searching swarm and avoid the premature convergence, the pair-wise bare bones particle swarm optimization (PBBPSO) is proposed in this section.

3.1. Initialization

Initialization is the first step of the PBBPSO. As a parameter-free algorithm, we only need to know some message from the original fitness function. Specifically, necessary information are listed below:

1. The number of particles N ;
2. The dimension of particles D ;
3. The fitness function F ;
4. The max iteration times T ;
5. The search range R .

Then all particles are randomly spread in the R . After that, the first personal best positions, $pbest$, is calculated by F . Moreover, the goal of the PBBPSO, global best position $gbest$ can be get. During this process, except the message about the original fitness function, PBBPSO doesn't need any parameters. This feature makes the proposed algorithm very easy to apply to different functions. Moreover, parameter-free saves a lot of time when dealing with real world problems.

3.2. Pair-wise strategy

In order to balance the exploration and exploitation abilities, a pair-wise strategy is introduced to BBPSO. The pair-wise strategy aims at slowing down the speed of swarm diversity losing. This is inspired by an old Chinese saying “never put all eggs in one basket.” The saying means we may lose all eggs when accidents happen the basket with all eggs. Applying this concept into optimization algorithms,

every particle moving to a same global best position may cause the whole swarm losing diversity very fast, and increasing the possibility of fall into local minimal. The pair-wise strategy is proposed to ameliorate this situation. By offering different strategies to different groups, pair-wise aims at slowing down the diversity losing speed of the whole swarm.

Before iteration, two particles are randomly selected from the swarm. The one with a better position, which means it has lower fitness value in a minimum problem, will be arranged at leader group (LG). Conversely, the other one will be placed in the follower group (FG). Then the LG and FG particles will be updated with their respective group rules. This strategy will repeat the same move until every particle is selected.

Specifically, LG particles iterate with the following equation:

$$\begin{aligned} u &= \frac{(pbest(i) + gbest)}{2} \\ l &= |pbest(i) - gbest| \\ x_t(i) &= N(u, l) \end{aligned} \quad (3)$$

where $pbest = (pbest(1), pbest(2), \dots, pbest(n))$ is a matrix used for recording the best position each element has ever reached; $gbest$ is the best position that all element has ever reached; $N(u, l)$ is a Gaussian distribution with mean u and standard deviation l . This equation is inherited from the standard BBPSO.

Conversely, particles in FG aim at supporting the LG. The next position of a FG particle is randomly selected from a Gaussian distribution with a mean $(LG(i) + FG(i))/2$ and a standard division $|LG(i) - FG(i)|$. Which means the FG evolves by following equation:

$$x_t(i) = N\left(\frac{LG(i) + FG(i)}{2}, |LG(i) - FG(i)|\right) \quad (4)$$

where $LG(i)$ is the particle in follow group, and $LG(i)$ is its leader. The LG particles moves toward to their leaders under this rule. More details are shown in Figure 1. According to Figure 1, FG particles have an opportunity to implement both local and global search. This strategy can enhance the diversity of the swarm, also can do a deeply search around

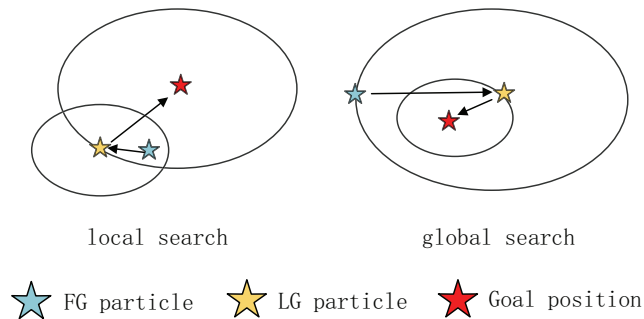


Figure 1: Possible situations of pair-wise

Algorithm 1 PBBPSO

Require: Max iteration time, T
Require: Dimension of the problem, D
Require: Fitness function, f
Require: Searching Range, R
Require: Particle swarm $X = x_1, x_2, \dots, x_n$
Require: Personal best position $Pbest = p_1, p_2, \dots, p_n$
Require: Global best position $Gbest$
Ensure: All particles are in R

```

1: while  $t < T$  do
2:   while  $X \neq \emptyset$  do
3:     Randomly select two particles,  $x_i$  and  $x_j$ , from  $X$ 
4:     if  $f(x_i) < f(x_j)$  then
5:       Update  $x_i$  with equation (3)
6:       Update  $x_j$  with equation (4)
7:     else
8:       Update  $x_i$  with equation (4)
9:       Update  $x_j$  with equation (3)
10:    end if
11:  end while
12:  Update  $Pbest$ 
13:  Update  $Gbest$ 
14:   $t = t + 1$ 
15: end while

```

Figure 2: Pseudo code of PBBPSO

the current global best position. Moreover, this process is all random, and no parameter is needed, which means the proposed system can easily apply to different functions. To show the working flow of PBBPSO, the flow chart of the algorithm is give in Figure 2.

4. Experiment

4.1. Benchmark functions

To verify the performance of PBBPSO, a comprehensive benchmark functions are chosen for the experiment. The combination of functions are considered in some early studies¹²²¹. The functions are divided into three groups according to their properties:

1. Group 1, ($f_1 - f_2$), unimodal functions;
2. Group 2, ($f_3 - f_5$), multimodal functions without shifted or rotated operator ;
3. Group 3, ($f_6 - f_7$), shifted or rotated multimodal functions.

All of the 7 functions are minimum value problems; the summarize of the test functions are shown in Table 1 and more details are presented blow:

a. Sphere function

$$f_1(x) = \sum_{i=1}^D x_i^2 \quad (5)$$

b. Rosenbrock function

$$f_2(x) = \sum_{i=1}^{D-1} \left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right) \quad (6)$$

c. Rastrigin function

$$f_3(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (7)$$

d. Ackley function

$$f_4(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e \quad (8)$$

e. Griewank function

$$f_5(x) = \sum_{i=1}^D \left(\frac{x_i^2}{4000} \right) - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1 \quad (9)$$

f. Shifted Rastrigin function

$$f_6(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + shift \quad (10)$$

$$z = (x - o)$$

where $x = (x_1, x_2, \dots, x_D)$ is one particle; $o = (o_1, o_2, \dots, o_D)$ is the shifted global optimum and the *shift* is the global best value;

g. Shifted Rotated Rastrigin function

$$f_7(x) = \sum_{i=1}^D \left(\frac{z_i^2}{4000} \right) - \prod_{i=1}^D \cos \left(\frac{z_i}{\sqrt{i}} \right) + 1 + shift \quad (11)$$

$$z = (x - o) * M$$

where $x = (x_1, x_2, \dots, x_D)$ is one particle; $o = (o_1, o_2, \dots, o_D)$ is the shifted global optimum; the *shift* is the global best value and M is the linear transformation matrix. According to²¹, the condition number of M is set as 2.

4.2. Results and discussion

To minimize the impact of accidental factors, the empirical error is calculated from 30 runs on each function with 1500 times iteration. The empirical error is defined as $|g_{best} - Minimum|$, where *gbest* is the final global best value given by an algorithm and *Minimum* is the theoretical optimal solution of

Table 1: Experiment Functions

Function	Search Range	Minimum	Reference
f_1 = SPHERE FUNCTION	(100,100)	0	1218
f_2 = ROSENBROCK FUNCTION	(-2.048,2.048)	0	1218
f_3 = RASTRIGIN FUNCTION	(-5.12, 5.12)	0	1218
f_4 = ACKLEY FUNCTION	(-32.768,32.768)	0	1218
f_5 = GRIEWANK FUNCTION	(-600,600)	0	1218
f_6 = SHIFTED RASTRIGIN FUNCTION	(-5,5)	-330	1821
f_7 = SHIFTED ROTATED GRIEWANK FUNCTION	(-600,600)	-180	1821

the function. Also, the dimension of each function is set as 30, the number of particles is set as 30.

In Table 2 and 3, the mean and standard division of each experiment's empirical error is displayed. Best results of each team are shown in boldface.

In group 1, PBBPSO gives excellent results with the two unrotated unimodal functions. The $3.37\text{E-}15$ empirical error on f_1 has an obvious advantage over other algorithms. Because the 2dn rank algorithm, PSO only gives $1.13\text{E-}10$. However, although PBBPSO still keeps its lead on f_2 , the advantage decreases. The $2.85\text{E+}01$ empirical error gets the 1st rank in all algorithm. But comparing with FIPS, the 2nd rank algorithm, the BBPSO gets a lead of 33.6%. These results are direct evidence that PBBPSO has good performance on unimodal functions. This can be attributed to the pair-wise system. When the minimum point is in the area that between FG particles and its leader, the crossover search enhance the probability of reaching the best point.

In group 2, PBBPSO gets one best and two second rank in all three unrotated multimodal functions. In f_3 , PBBPSO is defeated by BBPSOwj, winning the second place. The $2.79\text{E+}01$ empirical error PBBPSO gives, is more than two times larger than BBPSOwj gives. However, the result of PBBPSO is 87.4% smaller than the third algorithm. In f_4 , PBBPSO beats all other algorithms. The second rank algorithm, BBPSOwj, gives a $1.54\text{E-}01$ empirical error while PBBPSO gives $4.47\text{E-}02$. In addition, it is worth to mention that PBBPSO give 29 times exact 0.0 and 1 time 1.3404 in 30 independent run on f_4 . This situation not only confirms the ability of PBBPSO but also discloses the shortage of it.

In f_5 , PBBPSO takes the third rank. The empirical error it gives is $8.70\text{E-}03$, which is more than four times larger than the first rank algorithm. From the experiment results of group 2, it can be speculated that BBPSO can give better results than other algorithms in the experiment on unrotated multimodal functions. Especially in f_4 , PBBPSO runs out of the local minimal 29 times and fall into it one time. Hence, to enhance the stability of pair-wise system could be one of the future work.

In group 3, PBBPSO gives very disappointing result with the shifted or rotated multimodal functions. In f_6 , PBBPSO only gets the third rank. The empirical error it gives, $4.97\text{E+}01$, is more than four times larger than first rank algorithm gives. Also, the PBBPSO's result is 30.1% larger than that of Lévy BBPSO. This means that PBBPSO meets some problems in running out of the local minimal. In addition, PBBPSO gets only third rank in the f_7 . The empirical error is two times larger than the first rank algorithm, BBPSOwj gives. Also, BBPSO's result is 11.8% worse than PSO's.

It can be seen that in all the test functions, PBBPSO gives better results than the original BBPSO. So it is reasonable to believe that the pair-wise system can increase the diversity of the swarm and improve the optimization ability. This can be attributed to the pair-wise system. The simple and easy system randomly select two particles from the swarm and forces them move in different ways. This system greatly increases the diversity of the swarm. Also, the experiment result of group 2 and group 3 confirmed the ability of PBBPSO with unrotated unimodal and multimodal functions. However, the results of PBBPSO are not as good as BBP-

Table 2: Comparisons of empirical error between PSO, BBPSO, BBPSOwj and PBBPSO

Function	PSO ⁴		BBPSO ⁷		BBPSOwj ¹⁴		PBBPSO	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
<i>f1</i>	1.13E-10	7.94E-11	2.48E-09	3.96E-09	4.34E-03	2.37E-02	3.37E-15	4.11E-15
<i>f2</i>	5.26E+01	4.20E+01	3.25E+01	1.41E+01	6.50E+01	4.22E+01	2.85E+01	1.12E+01
<i>f3</i>	8.36E+01	1.98E+01	4.96E+01	1.27E+01	1.11E+01	3.45E+00	2.79E+01	6.36E+00
<i>f4</i>	1.90E+01	3.59E+00	2.71E-01	6.36E-01	1.54E-01	3.55E-01	4.47E-02	2.45E-01
<i>f5</i>	2.10E-03	4.16E-03	1.35E-02	1.86E-03	3.05E-02	2.84E-02	8.70E-03	8.50E-03
<i>f6</i>	6.70E+01	1.54E+01	6.87E+01	1.49E+01	1.02E+01	2.86E+00	4.97E+01	1.24E+01
<i>f7</i>	3.21E-02	1.06E-01	3.11E-01	2.58E-01	1.69E-02	1.94E-02	3.59E-02	2.97E-02

Table 3: Comparisons of empirical error between FIPS, Lévy BBPSO, PBBPSO

Function	FIPS ¹⁰¹¹		Lévy BBPSO ¹⁹		PBBPSO	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
<i>f1</i>	1.03E-04	6.95E-05	2.23E-10	4.59E-10	3.37E-15	4.11E-15
<i>f2</i>	4.29E+01	4.01E+01	6.01E+01	8.55E+01	2.85E+01	1.12E+01
<i>f3</i>	7.78E+01	1.63E+01	3.19E+01	1.83E+01	2.79E+01	6.36E+00
<i>f4</i>	6.86E-01	1.04E+00	1.27E+00	8.70E-01	4.47E-02	5.46E-01
<i>f5</i>	5.53E-02	8.15E-02	6.12E-01	7.96E-03	8.70E-03	8.50E-03
<i>f6</i>	1.26E+02	3.19E+01	3.82E+01	6.18E+00	4.97E+01	1.24E+01
<i>f7</i>	1.15E-01	1.00E-01	7.63E-01	1.35E-01	3.59E-02	2.97E-02

SOWj when facing the shifted or rotated functions. This is excusable because each particle in PBBPSO uses only one calculation time. And no topologies or judgment is needed during the iteration process. Hence the PBBPSO is the fastest algorithm in the experiment. This means the PBBPSO can keep the calculating speed while some new method is introduced to it. Moreover, the parameter-free makes the proposed algorithm very easy to adapt different application. So increase the ability on searching shifted or rotated functions and studies about real word applying should be a promising future work.

At last, to make the conclusion easy to understand, a ranking comparison is proposed to describe the searching ability of all algorithms in the experiment. Each function provides a rank value from 1 to 7 to all algorithms. The algorithm presents best result will get 1 and presents worst will get 7. The mean ranking value from all functions is calculated in the Table 4.

Table 4. Average ranking

Algorithm	Average Ranking
PBBPSO	1.86
BBPSOwj	3.00
BBPSO	3.57
PSO	3.57
Lévy BBPSO	4.28
FIPS	4.71

5. Conclusion

A pair-wise bare bones particle swarm optimization (PBBPSO), which can solve both unimodal and multimodal problems, is proposed in this paper. The PBBPSO keeps advantages from BBPSO such as simplicity and easy applying. Also, the algorithm inherits the searching concept from the original algorithm. In order to keep the diversity of searching swarm and avoid premature convergence, the pair-wise strategy is used in the proposed algorithm. A pair of particles will be selected from the swarm ev-

ery time. The one that has better position will be placed in the leader group and its partner will be in the follower group. Different iteration strategies are used in different groups to slow down the speed of diversity losing.

To verify the performance of PBBPSO, a set of well-known benchmark functions are used in the experiment. Furthermore, several variants of BBPSO and some other evolutionary algorithms are also evaluated on the same functions as the control group. Finally, the experiment result and statistical analysis confirm the performance of PBBPSO with nonlinear functions.

References

1. I. Amdouni, P. Minet, and C. Adjih, "OSERENA: a Coloring Algorithm Optimized for Dense Wireless Networks," *International Journal of Networked and Distributed Computing*, vol. 1, no. 1, pp. 9–24, 2013.
2. L. An, P. Yang, H. Zhang, and M. Chen, "Multi-Objective Optimization for Milling Operations using Genetic Algorithms under Various Constraints," *International Journal of Networked and Distributed Computing*, vol. 2, no. 2, pp. 108–114, 2014.
3. J. Kennedy and R. Eberhart, "Particle swarm optimization," *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948 vol.4, 1995.
4. M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
5. I. C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.
6. R. Eberhart and Yuhui Shi, "Particle swarm optimization: developments, applications and resources," *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, vol. 1, pp. 81–86, 2001.
7. J. Kennedy, "Bare bones particle swarms," in *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*. IEEE, 2003, pp. 80–87.
8. M. Omran and S. Al-Sharhan, "Barebones particle swarm methods for unsupervised image classification," in *2007 IEEE Congress on Evolutionary Computation*. IEEE, 2007, pp. 3247–3252.
9. D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," *Proceedings of the 2007 IEEE Swarm Intelligence Symposium, SIS 2007*, no. Sis, pp. 120–127, 2007.
10. R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.
11. J. Kennedy and R. Mendes, "Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 36, no. 4, pp. 515–519, 2006.
12. J. J. Liang, A. K. Qin, S. Member, P. N. Suganthan, S. Member, and S. Baskar, "Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
13. C. H. Chen, "Bare bone particle swarm optimization with integration of global and local learning strategies," *Proceedings - International Conference on Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 692–698, 2011.
14. T. Blackwell, "A study of collapse in bare bones particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 354–372, 2012.
15. H. Wang, S. Rahnamayan, H. Sun, and M. G. H. Omran, "Gaussian bare-bones differential evolution," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 634–647, 2013.
16. R. A. Krohling and E. Mendel, "Bare bones particle swarm optimization with Gaussian or cauchy jumps," *2009 IEEE Congress on Evolutionary Computation, CEC 2009*, pp. 3285–3291, 2009.
17. C. H. Chen, "A revised bare bone particle swarm optimizer and its variant," *iFUZZY 2013 - 2013 International Conference on Fuzzy Theory and Its Applications*, pp. 488–493, 2013.
18. M. Campos, R. A. Krohling, and I. Enriquez, "Bare bones particle swarm optimization with scale matrix adaptation," *IEEE Transactions on Cybernetics*, vol. 44, no. 9, pp. 1567–1578, 2014.
19. T. J. Richer, "The Lévy Particle Swarm," *IEEE Congress on Evolutionary Computation*, pp. 808–815, 2006.
20. R. Vafashoar and M. R. Meybodi, "Multi swarm bare bones particle swarm optimization with distribution adaption," *Applied Soft Computing*, vol. 47, pp. 534–552, 2016.
21. P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. Chen, A. Auger, and S. Tiwari, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization," *Report*, no. May, pp. 1–50, 2005.