PSOPB: A Two-population Particle Swarm Optimizer Mimicking Facultative Bio-parasitic Behavior

Quande Qin[1], Li Li[1], Rongjun Li[2], Ben Niu[1]
[1]*Department of Management Science, Shenzhen University,*
Shenzhen, Guangdong, 518060, P.R.China
E-mail:qinquande@gmail.com;llii318@163.com;drniuben@gmail.com
[2]*Depaatment of Decision Science, South China University of Technology,*
Guangzhou, Guangdong, 510640, P.R.China
E-mail:lirongjun1016@hotmail.com

Received 1 September 2010
Accepted 10 February 2011

Abstract

Inspired by the phenomenon of bio-parasitic behavior in natural ecosystem, this paper presents a novel particle swarm optimizer named PSOPB, in which particles are composed of the host and the parasite population. In the presented algorithm, the two populations mimic facultative bio-parasitic behaviour and exchange particles according to particles' fitness values sorted of each population in a certain number of iterations. The parasite mutation and the host immunity are also considered to tie it closer to bio-parasitic behaviour as well as improve the algorithm performance. In order to embody the law of "survival of the fittest" in biological evolution, the particles with poor fitness value in the host population are removed and replaced by the same numbers of the re-initialization particles to maintain constant population size. The experimental results of a set of 10 benchmark functions demonstrate the presented algorithm's efficacy.

Keywords: Particle swarm optimizer; Facultative bio-parasitic behaviour; PSOPB; Immune mechanism

1. Introduction

Particle swarm optimizer (PSO) is a kind of stochastic optimization algorithm, inspired by the sociological behaviors associated with birds flocking[1,2]. In the original version of PSO algorithm, the trajectory of each particle in the search range is adjusted by dynamically altering its velocity, according to two factors: each individual's best position ever found and its neighbors' best position ever found[2]. Comparing with other stochastic optimization algorithms, PSO is easy to implement and has been empirically shown to perform well on many optimization problems. For this reason, PSO has already been widely used in many areas, including power systems[3], artificial neural network training[4] and bandwidth prediction[5]. However, it has difficulties in keeping balance between exploration and exploitation[6]. Therefore it is prone to suffer from premature convergence when strongly multimodal problems are being optimized. In the past decades, numerous researches have provided some improved methods to overcome the drawback of trapping in the local optima. The most part of these improved methods can be summarized into the following categories: tuning the parameters[7], designing different population topologies[8], combining PSO with other evolutionary optimization operators[9] and adopting new learning strategies[10,11].

As PSO algorithm is derived from modeling sociological behaviors of animals, it is natural to incorporate other proper biological mechanisms into the canonical PSO, which may be a viable way to improve the algorithm's performance. So far, there have been some related papers concerning this topic. Silva constructed a predator-prey PSO model[12], in which

Published by Atlantis Press

particles are divided into two categories: predator and prey, the former are used to force the particles stagnated in the local optima to escape, whereas the latter are subject to predator's exclusion and gradually close to the global optimum. He provided a new type of PSO (PSOPC) according to passive congregation behavior in animals[13]. In PSOPC, message can be transferred among individuals of the whole swarm. In accordance with the bacterial chemotactic behavior, Niu presented an improved PSO[14], in which each particle is not only attracted by its personal best position and the group's best position, but also repulsed by the worst position of itself and the whole group. Gleaned from ideas from the co-evolution of the symbiotic species, Chen proposed a novel multi-species optimizer (PS^{2}O)[15]. From the existing literatures, the method of incorporating the biological mechanism into PSO has showed its efficacy, and will be a good way to improve the performance of PSO in the future.

An idea of incorporating bio-parasitic behavior into canonical PSO was addressed in this paper. The presented algorithm called PSOPB, in which particles are composed of two populations, to be specific, the host and the parasite population. The two populations mimic facultative bio-parasitic behavior of the host-parasites relationship in natural ecosystem. The host immune response and the parasite mutation are also formulated in PSOPB. The dynamics of the parasite population is the same as the canonical PSO, but the host population has a big difference. Experiments on some benchmark functions are conducted. The results demonstrate that our proposed PSO algorithm has a remarkable performance, especially for complicated problems. Hence, PSOPB may be a good alternative to deal with optimization problems, in particular the complicated.

The rest of this paper is organized as follows. Section 2 introduces the canonical PSO. In section 3, the bio-parasitic behavior and PSOPB framework are presented. We describe the experimental settings and results in section 4. The paper is concluded in section 5.

2. Canonical PSO

A swarm of particles are represented as potential solutions and the *i*th particle is associated with two factors, i.e., the velocity vector $V_i^t = [v_{i1}, v_{i2}, \mathbf{K}, v_{iD}]$ and the position vector $X_i^t = [x_{i1}^t, x_{i2}^t, \mathbf{K}, x_{iD}^t]$, where $i \in \{1, 2, \mathbf{K}, N\}$, N is the population size; D stands for the dimensions of the solution space and *t* denotes the current iteration number. $x_{id} \in [l_d, u_d]$, $d \in \{1, 2, \mathbf{K}, D\}$, where l_d and u_d are the lower and upper bounds of the *d*th dimension, respectively. The velocity and the position of each particle are initialized by random vectors within corresponding ranges. During the evolutionary process, the swarm is manipulated according to the following equations:

$$v_{id}^{t+1} = wv_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t) \qquad (1)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \qquad (2)$$

In Eq. (1), w, called inertia weight, is to assist with the balance between exploration and exploitation. r_1 and r_2 are random numbers, uniformly distributed with the interval $[0,1]$. c_1 and c_2 are acceleration coefficients . $P_i^t = [p_{i1}^t, p_{i2}^t, \mathbf{K}, p_{iD}^t]$ is the best previous position with best fitness value has been found so far by the *i*th particle; and $P_g^t = [p_{g1}^t, p_{g2}^t, \mathbf{K}, p_{gD}^t]$ is the best position discovered by the whole swarm. Generally, a maximum velocity on each dimension, $v_{\max,d}$, is specified to control excessive roaming of particles outside the user defined search space. If $|v_{id}|$ exceeds $v_{\max,d}$, then it is assigned to $sign(v_{id})v_{\max,d}$.

The value of w is linearly decreasing with the iterations as

$$w = (w_{start} - w_{end}) \times (t_{\max} - t) / t_{\max} + w_{end} \qquad (3)$$

where $t_{\max}$ is the predefined maximum number of allowable iterations; w_{start} and w_{end}, usually are set to 0.9 and 0.4[7], denote the initial and final values of the inertia weight, respectively. In the paper, this version of PSO is referred to standard PSO (SPSO).

Another important variant of PSO is proposed in form of constriction factor (CPSO), which is an alternative method for controlling the behavior of particles in the swarm[16]. In CPSO, the velocity is updated by the following equation:

$$v_{id}^{t+1} = l[v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_1 r_2 (p_{gd}^t - x_{id}^t)] \qquad (4)$$

where l is called constriction factor, given by:

$$l = 2 \Big/ \left| 2 - j - \sqrt{j^2 - 4j} \right| \qquad (5)$$

Usually the value of l is set to 0.729 with $j = c_1 + c_2 = 4.1$. c_1 and c_2 are usually both set to 2.05.

3. PSOPB

3.1 Bio-parasitic behavior

The term symbiosis, first proposed in 1879 by the German mycologist Anton de Bary[17], commonly describes close and often long-lasting physical relationships between different biological species. There are three different categories of symbiotic relationships: mutualism, commensalism and parasitism[18]. Mutualism describes a kind of symbiotic relationship that is actually beneficial to both species involved. Commensalism describes a relationship between two living organisms where one benefits and the other is not significantly harmed. Parasitism is a relationship in which one organism, known as the parasite, lives in or on another organism, known as the host, from which the parasite derives nourishment and the host is harmed[18]. Parasites are usually divided into two categories, facultative parasites and obligate parasites, in terms of their living style. Facultative parasites mostly live independent of a host but seldom hold the charge of a parasite. By contrast, obligate parasites are that cannot live elsewhere except on the living protoplasm of its host[19].

Many researches have shown that parasite evolved from free-living organisms. In the long co-evolutionary process, the free-living organisms gradually changed its physical shape or morphological structure to adapt the new living style and become parasites[19]. The hosts harmed increases their immunity, called acquired immunity which is in contrast to innate immunity, to resist the invasion behavior of the parasites. There are many cases demonstrated this phenomenon in nature ecosystem. Such as, vertebrates infected by parasites will produce a strong immune response in the case of subjecting to the second same parasitic infection[20]; Plants and inferior animals also can improve their immunity after being infected by parasites. The co-evolutionary process between parasites and hosts commonly reduce the "negative effect" of the parasitic harm behavior, or even parasitism evolves into mutualism[21].

3.2 PSOPB Framework

In the proposed PSO algorithm, particles are divided into two populations: the parasite population $(Swarm^P)$, in which the number of particles is denoted as N^P, and the host population $(Swarm^H)$, the size of which is denoted as N^H. We believe that facultative bio-parasitism relationship between two populations is suitable to be incorporated into the canonical PSO. If parasitic behavior occurs, $Swarm^P$ obtains nourishment from $Swarm^H$; Otherwise, $Swarm^P$ and $Swarm^H$ are evolved independently. The facultative host-parasite relationship is simulated as the exchange of particles in the two populations in a certain number of iterations, which we called reconstruction gap and denoted as k, in PSOPB. Therefore, $Swarm^P$ get some particles with good fitness from $Swarm^H$ and the same numbers of particles with bad fitness in it leave to $Swarm^H$. In this paper, we define the fitness values the larger the better, which means that when solving minimization problems, we will use the negative functions values as the fitness values.

During the host-parasite interactions, there are two kinds of immunity, innate immunity and acquired immunity, in $Swarm^H$. In our paper acquired immunity is modeled according to two ways: one is the numbers of exchanged particles decrease with increasing of iterations; the other is that particles in $Swarm^H$ learn from the global best particle in $Swarm^P$ to speed up convergence and improve its performance. In PSOPB, the numbers of exchanged particles is set to the rounded value of $m \times N^P$, and m is set to linearly decrease with the iterations as:

$$m = (m_{start} - m_{end}) \times (t_{\max} - t) / t_{\max} + m_{end} \qquad (6)$$

m_{start} and m_{end} denote the initial and final values of m, respectively. Conversely, $Swarm^H$ forfeits particles with good fitness values and get the bad particles from $Swarm^P$. Another acquired immune response embodied in the proposed algorithm is as follows: when the best particle's fitness value in $Swarm^H$ is worse than that in $Swarm^P$, particle i in $Swarm^H$ fly in accordance with three directions: P_i^H, P_g^H and P_g^P, where P_i^H is the best previous position of particle i in $Swarm^H$, P_g^H and P_g^P represent the particle with best fitness value in $Swarm^H$ and $Swarm^P$, respectively; Otherwise, each particle in $Swarm^H$ evolves according the canonical PSO. In this paper, PSOPB adopts the form of "constriction factor". Through the analysis above, the velocity of $Swarm^P$ is updated by the Eq.(4) and the velocity of $Swarm^H$ is set to update according to the following equations:

$$v_{id}^{t+1} = \begin{cases} l\,[v_{id}^t + c_{11} r_{11} (p_{id}^t - x_{id}^t) + \\ c_{12} r_{12} (p_{gd,H}^t - x_{id}^t) + c_{13} r_{13} (p_{gd,P}^t - x_{id}^t)] & fp_{g,H}^t < fp_{g,P}^t \\ l\,[v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + \\ c_2 r_2 (p_{gd,H}^t - x_{id}^t)] & fp_{g,H}^t \ge p_{g,P}^t \end{cases} \qquad (7)$$

where $c_{11}, c_{12}, c_{13}, c_1$ and c_2 are acceleration coefficients, $fp_{g,H}^t$ and $fp_{g,H}^t$ are the best particle's fitness value at tth iterations in $Swarm^H$ and $Swarm^P$, respectively, r_{11}, r_{12} and r_{13} are random numbers uniformly distributed with range $[0,1]$, the meaning of other parameters in Eq. (7) is the same as in Eq. (4).

$Swarm^H$ is harmed after bio-parasitic behavior. In order to show the law of "survival of the fittest" in biological evolution, some poorest particles, the number of which is set to the rounded value of $g \times N^H$, are removed and replaced by the re-initialization of the particles in order to maintain constant population size in $Swarm^H$.

In subsection 3.1, we have known that organisms from free-living to parasitic life style should change its physical form or morphological structure. This phenomenon was simulated as a mutation operator in our presented algorithm. When the best particle's fitness value in $Swarm^P$ is not improving with the increasing of a certain number of iterations, denoted as h . The velocity of one dimension of particle i uniformly randomly selected within $Swarm^P$ is randomly chosen to mutate. Only one dimension is chosen because the local optima particle is selected which is likely to have good structure of the global optimum, this should be protected. As every dimension has the same probability to be chosen, this mutation operation can be regarded to perform on every dimension in a statistical sense. The mutation operator performed in $Swarm^P$ can be mathematically represented as follows:

If $fp_{g,P}^t - fp_{g,P}^{t-h} = 0$

$$v_{id}^t = \begin{cases} 0.5 \times v_{\max,d} \times r_3, & r_4 < 0.5 \\ -0.5 \times v_{\max,d} \times r_3 & \text{otherwise.} \end{cases} \qquad (8)$$

End

where r_3 and r_4 .are random numbers uniformly distributed within $[0,1]$, $fgB_p{}^t$ is the fitness best particle's fitness value in $Swarm^P$ at the tth iterations. The complete pseudocode for PSOPB is as follows.

```
Begin
  Initialization;
    Set $t = 0$
    While (the termination conditions are not met)
        Do in parallel
        For each population ( $Swarm^H, Swarm^P$ )
        Evaluate the fitness value of each particle
        Update the velocity of each population
        $v_{id}^t = \min(\max(v_{id}^t, -v_{\max}), v_{\max})$
        Update the position of each population
        Update $p_i^t$ , $p_g^t$ of each population
        Calculate $fp_{g,H}^t$ and $fp_{g,P}^t$
      End Do in parallel
        If $\mathrm{mod}(t,k) = 0$ & $t \neq 0$
          Exchange particles;
          Remove particles in $Swarm^H$
          Re-initialize particles in $Swarm^H$
        End If
          If $fp_{g,P}^t - fp_{g,P}^{t-h} = 0$
            Velocity mutated in $Swarm^P$
          End If
      Set $t = t + 1$
    End While
End
```

4. Experiments and discussions

4.1 Benchmark functions

A set of 10 benchmark functions are conducted this experiments to evaluate the performance in terms of the optimum solution after a predefined number of iterations. In this paper, all benchmark functions are divided into two groups. The first group includes from function f_1 to function f_7 seven classical benchmark functions. In the first group, the distribution of local optima in some functions is regular and the variables are separable[11]. They can be solved by using divide-and-conquer methods, such as references [22]. The second group includes three rotated functions f_8, f_9 and f_{10}, in which the original variable x is rotated by left multiplying the orthogonal matrix M , i.e, $y = M \times x$.

Global optimum x^* , the corresponding fitness value $f(x^*)$, dimensions, search range, simplified as range, and initialization range of all benchmark functions are given in Table 1. Asymmetric initializations, in which the population is initialized only part of the search range, are used for all functions.

1. Schwefel Problem 1.2

$$f_1(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$$

2. Rosenbrock function

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

3. Rastrigin function

$$f_3(x) = \sum_{i=1}^{D} \left(x_i^2 - 10 \cos\left(2p\, x_i \right) + 10 \right)$$

4. Ackley function

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

5.Griewank function

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

6. Schwefel function

$$f_6(x) = 418.9829 \times D - \sum_{i=1}^{D}\left(x_i \sin\left(\sqrt{|x_i|}\right)\right)$$

7. Weierstrass function

$$f_7(x) = \sum_{i=1}^{D}\left(\sum_{k=0}^{20}\left[a^k \cos\left(2\pi b^k\left(x_i + 0.5\right)\right)\right]\right) - D\sum_{k=0}^{20}\left[a^k \cos\left(2\pi b^k \cdot 0.5\right)\right]$$

$$a = 0.5,\ b = 3,\ i = 1, 2, \cdots, D.$$

8. Rotated Rosenbrock function

$$f_8(x) = \sum_{i=1}^{D-1}\left(100\left(z_{i+1} - x_i\right)^2 + \left(z_i - 1\right)^2\right),$$

$$z = M * x,\ x = [x_1, x_2, \ldots, x_D]$$

9. Rotated Rastrigin function

$$f_9(x) = \sum_{i=1}^{D}\left(z_i^2 - 10\cos\left(2\pi z_i\right) + 10\right),$$

$$z = M * x,\ x = [x_1, x_2, \ldots, x_D]$$

10. Rotated Griewank function

$$f_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi z_i\right) + 20 + e$$

$$z = M * x,\ x = [x_1, x_2, \ldots, x_D]$$

Table 1 Global optimum, search ranges and initialization ranges of benchmark functions

Functions	Dimension	Range	Initialization range	x^*	$f(x^*)$
f_1	30	$(-100,100)^D$	$(50,100)^D$	$\{0\}^D$	0
f_2	30	$(-30,30)^D$	$(15,30)^D$	$\{1\}^D$	0
f_3	30	$(-5.12,5.12)^D$	$(2.56,5.12)^D$	$\{0\}^D$	0
f_4	30	$(-32,32)^D$	$(10,20)^D$	$\{0\}^D$	0
f_5	30	$(-600,600)^D$	$(300,600)^D$	$\{0\}^D$	0
f_6	30	$(-500,500)^D$	$(200,500)^D$	$\{420.97\}^D$	0
f_7	30	$(-0.5,0.5)^D$	$(0.2,0.5)^D$	$\{0\}^D$	0
f_8	30	$(-100,100)^D$	$(50,100)^D$	$\{0\}^D$	0
f_9	30	$(-30,30)^D$	$(15,30)^D$	$\{0\}^D$	0
f_{10}	30	$(-5.12,5.12)^D$	$(2.56,5.12)^D$	$\{0\}^D$	0

4.2 Parameters settings for the involved PSO algorithms

Experiments were conducted to observe the performance of the proposed PSOPB algorithm in comparison with SPSO, CPSO, PSOPC, FIPS(fully informed particle swarm)[23] and FDR-PSO(fitness-distance-ratio based particle swarm optimization)[24]. The population size for all algorithms was set to 80. As for PSOPB, $N^H = N^P = 40$ were employed. Dimensions D of all benchmark functions were set to 30. In many practical problems, there are bounds on the variables' range. In order to prevent particles from moving out the bounds, the method proposed in references [25] was used in this paper. The parameters used for SPSO, CPSO, PSOPC, FIPS and FDR-PSO were recommended in references [7,16,13,23,24]. The specific parameters setting for each algorithm are given in Table 2.

In order to investigate whether the performance of PSOPB is sensitive to the parameter k or not, we tested PSOPB with different value of k , to be specific, 30, 50, 100, 150 and 200, on function. f_1, f_2, f_3, f_4 and f_5 . The maximum number of allowable iterations is set at 3000.

The average and standard deviation of the optimum solution for 25 trials are presented in Table 3. From the results; it is easy to find that the value of k counts much for the performance of POSPB. When k was set to 100, all test function are obtained good results. So, in the following experiments, $k=100$ is used.

Table 2. Parameters settings for the involved PSO algorithms

Algotithms	Parameters settings	References
SPSO	$w: 0.9 \sim 0.4$, $c_1 = c_2 = 2$, $v_{\max,d} = 0.2\times \mathit{Range}$	[7]
CPSO	$l = 0.729$, $c_1 = c_2 = 2.05$, $v_{\max,d} = 0.5\times \mathit{Range}$	[16]
FDR-PSO	$w: 0.9 \sim 0.4$, $c_1 = c_2 = 1$, $c_3 = 2$, $v_{\max,d} = 0.2\times \mathit{Range}$	[24]
FIPS	$\sum c_i = 4.1$, $l = 0.729$, $v_{\max,d} = 0.2\times \mathit{Range}$	[23]
PSOPC	$w: 0.9 \sim 0.7$, $c_1 = c_2 = 0.5$, $c_3: 0.4 \sim 0.6$, $v_{\max,d} = 0.2\times \mathit{Range}$	[13]
PSOPB	$l = 0.729$, $c_1 = c_2 = 2.05$, $c_{11} = c_{12} = c_{13} = 1.367$, $g = 0.5$, $h = 50$, $m: 0.5 \sim 0.3$, $v_{\max,d} = 0.5\times \mathit{Range}$	

Table 3 PSOPB performance with various values of the reconstruction gap k

			PSOPB		
k	30	50	100	150	200
f_1	3.11e-010 (3.53e-010)	3.52e-010 (3.27e-010)	2.48e-010 (2.21e-010)	3.77e-009 (4.31e-009)	7.28e-009 (9.84e-009)
f_2	3.69 (3.05)	4.29 (2.38)	2.49 (2.65)	4.33 (4.85)	4.42 (4.54)
f_3	16.51 (8.48)	9.44 (5.37)	6.87 (1.58)	8.17 (2.39)	11.64 (3.28)
f_4	4.51e-012 (4.35e-012)	2.56e-013 (5.75e-013)	6.75e-015 (2.43e-015)	6.58e-015 (3.74e-015)	6.24e-015 (2.72e-015)
f_5	1.45e-002 (1.73e-002)	1.28e-002 (1.44e-002)	6.32e-003 (8.29e-003)	1.14e-002 (1.54e-002)	1.43e-002 (1.82e-002)

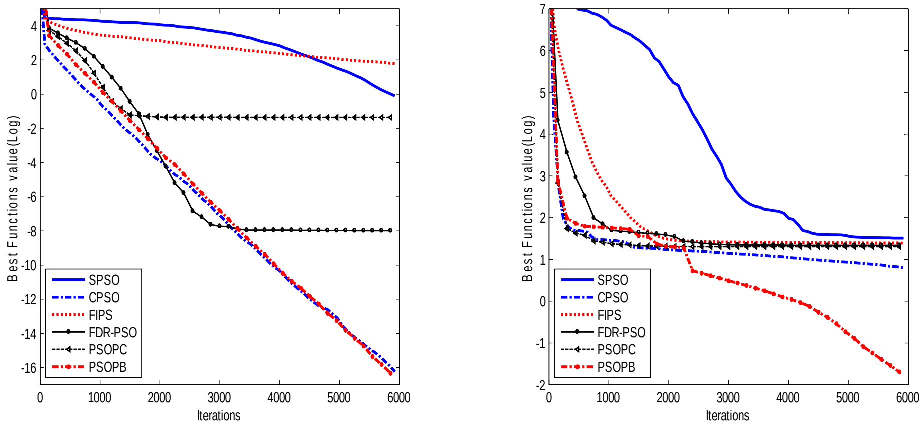
4.3 Comparisons with involved PSO algorithms

The experiment runs 30 trials independently for the involved PSO algorithms on a set of 10 benchmark functions and the maximum number of allowable iterations is set at 6000. The representative results including the best, worst, mean and standard deviation(Std) of these benchmark function values were presented in the Table 4, in which numbers in bold represented were the comparatively best values. The graphs of the variation of the best function values with the evolution of iterations are showed in Fig. 1.

Table 4 Search result comparisons of PSOs on all benchmark functions

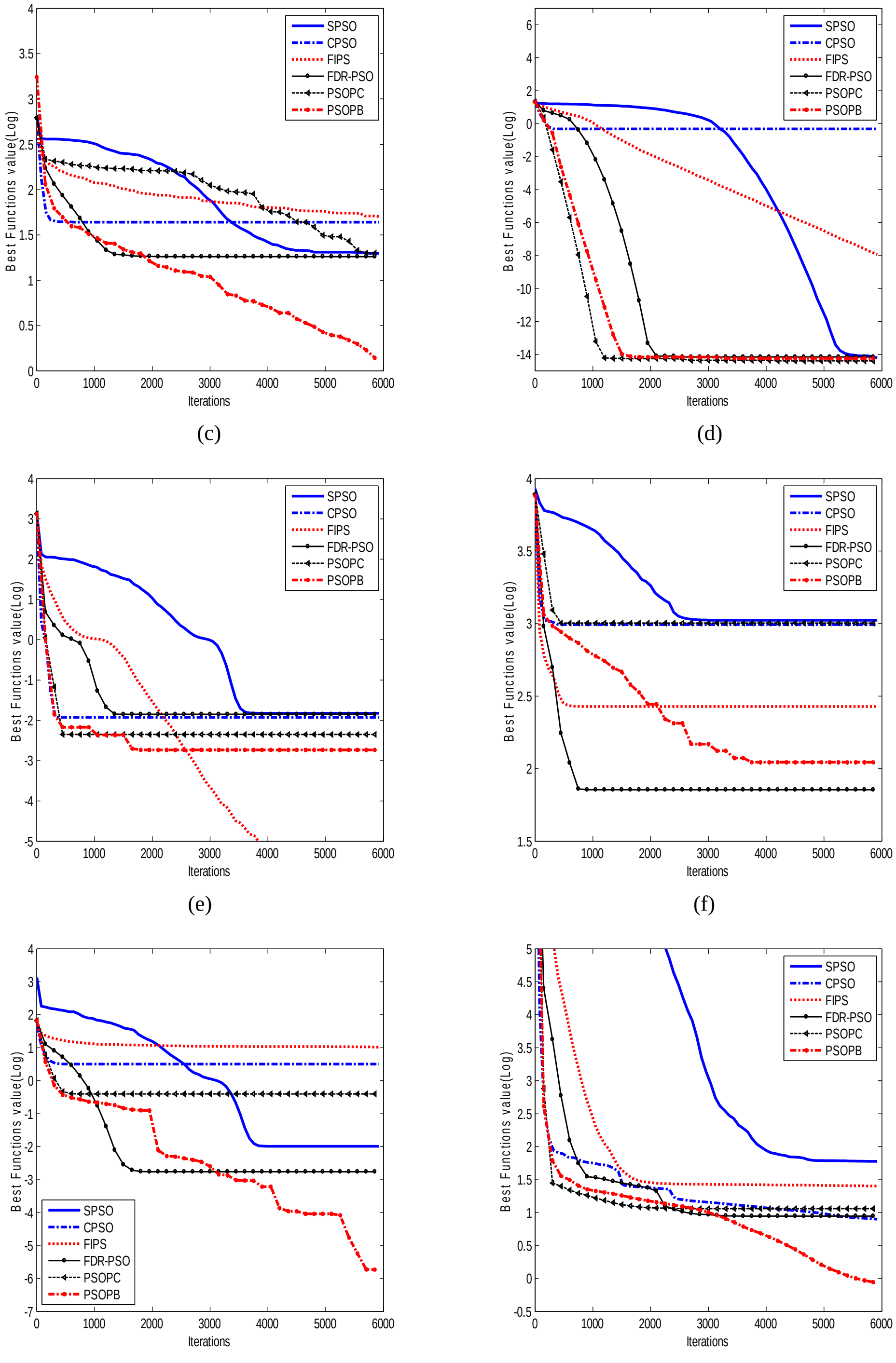
Function	Index	SPSO	CPSO	FIPS	FDR-PSO	PSOPC	PSOPB
f_1	Best	1.64e-001	3.05e-018	2.93e+001	4.78e-010	1.26e-004	**1.06e-019**
	Worst	1.51e+000	9.66e-017	8.11e+001	6.67e-008	2.49e-001	**6.09e-017**
	mean	5.90e-001	4.01e-017	5.96e+001	1.03e-008	4.37e-002	**1.10e-017**
	Std	4.45e-001	4.27e-017	1.73e+001	2.28e-008	8.37e-002	**1.84e-017**
f_2	Best	1.95e+000	2.53e-003	2.38e+001	2.77e-001	1.31e+001	**1.37e-004**
	Worst	7.84e+001	1.16e+001	2.49e+001	7.18e+001	7.06e+001	**6.13e-002**
	mean	3.07e+001	6.20e+000	2.46e+001	2.19e+001	2.01e+001	**1.47e-002**
	Std	2.61e+001	3.78e+000	3.55e-001	2.95e+001	1.78e+001	**2.07e-002**

f_3	Best	7.96e+000	3.38e+001	3.79e+001	7.96e+000	8.95e+000	**0**
	Worst	2.98e+001	6.96e+001	6.44e+001	2.49e+001	3.98e+001	**2.98e+000**
	mean	1.98e+001	4.37e+001	5.02e+001	1.83e+001	1.99e+001	**1.29e+000**
	Std	5.35e+000	1.14e+001	9.96e+000	6.04e+000	9.11e+000	**1.05e+000**
f_4	Best	**4.53e-017**	6.22e-015	4.84e-009	6.22e-015	2.32e-015	2.66e-015
	Worst	7.32e-014	1.78e+000	1.43e-008	**1.33e-015**	6.22e-015	6.22e-015
	mean	6.22e-015	4.77e-001	9.08e-009	7.11e-015	**4.09e-015**	5.44e-015
	Std	5.41e-015	7.74e-001	2.65e-009	2.51e-015	1.84e-015	**1.26e-015**
f_5	Best	**0**	**0**	7.51e-014	**0**	**0**	**0**
	Worst	6.38e-002	2.97e-002	**1.62e-007**	4.92e-002	1.48e-002	7.44e-003
	mean	1.49e-002	1.18e-002	**1.78e-008**	1.41e-002	4.45e-003	1.81e-003
	Std	2.21e-002	1.07e-002	**5.11e-008**	1.66e-002	5.91e-003	3.72e-003
f_6	Best	4.77e+002	6.51e+002	**3.82e-004**	**3.82e-004**	3.35e+002	**3.82e-004**
	Worst	2.02e+003	1.48e+003	6.47e+002	**3.35e+002**	1.44e+003	3.55e+002
	mean	1.06e+003	9.83e+002	2.71e+002	**7.15e+001**	1.01e+003	1.10e+002
	Std	5.24e+002	2.71e+002	2.68e+002	**1.19e+002**	3.56e+002	1.31e+002
f_7	Best	**0**	2.73e-001	7.29e+000	7.63e-004	3.34e-004	**0**
	Worst	2.46e-002	5.62e+000	1.38e+001	9.64e-005	1.60e+000	**2.82e-005**
	mean	1.03e-002	3.19e+000	1.05e+001	1.71e-003	3.99e-001	**1.88e-006**
	Std	9.83e-003	1.99e+000	2.09e+000	2.73e-003	5.42e-001	**7.29e-006**
f_8	Best	9.55e+000	9.36e-002	2.41e+001	1.66e+000	8.77e+000	**9.46e-002**
	Worst	2.66e+002	1.39e+001	2.75e+001	1.31e+001	1.40e+001	**1.34e+000**
	mean	5.96e+001	7.83e+000	2.52e+001	8.82e+000	1.15e+001	**8.45e-001**
	Std	6.98e+001	4.36e+000	9.33e-001	3.48e+000	1.61e+000	**7.43e-001**
f_9	Best	3.88e+001	4.38e+001	1.22e+002	3.08e+001	4.28e+001	**1.21e+001**
	Worst	1.05e+002	9.85e+001	1.67e+002	8.56e+001	8.06e+001	**4.03e+001**
	mean	7.82e+001	7.15e+001	1.44e+002	6.56e+001	5.90e+001	**3.01e+001**
	Std	1.95e+001	1.76e+001	1.54e+001	1.82e+001	1.78e+001	**1.13e+001**
f_{10}	Best	**0**	1.11e-016	4.84e-012	**0**	**0**	**0**
	Worst	3.69e-002	3.44e-002	**1.87e-006**	8.11e-002	4.17e-002	1.23e-003
	mean	1.62e-002	1.23e-002	**3.37e-007**	2.15e-002	1.23e-002	3.92e-003
	Std	1.40e-002	9.92e-003	**5.52e-007**	2.84e-002	1.32e-002	5.34e-003



(a)

(b)


Best Functions value(Log)
Iterations
SPSO
CPSO
FIPS
FDR-PSO
PSOPC
PSOPB
0
1000
2000
3000
4000
5000
6000

(c)

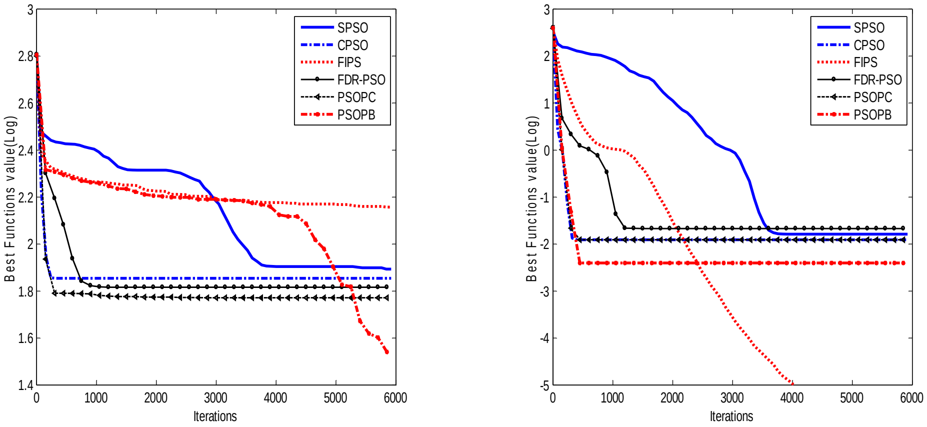
(d)

(e)

(f)

(g)

(h)



(i) (j)

Fig. 1. Variation of the best function values with the evolution of iterations (a) Schwefel problem 1.2. (b) Rosenbrock functions. (c) Rastrigin function. (d) Ackley function. (e) Griewank function. (f) Schwefel function. (g) Weierstrass function. (h) Rotated Rosenbrock function. (i) Rotated Rastrigin function. (j) Rotated Griewank function.

4.4 Results and discussions

From Table 4 and Fig. 1 we could observe that PSOPB obtained the remarkable search accuracy and fast convergence rate.

As for typical unimodal function f_1 and f_2, PSOPB performs better compared to the involved PSO algorithms except CPSO does marginally performance for function f_1. Especially for deceptive function f_2, in which global optimum is inside a long, narrow and parabolic shaped flat valley, PSOPB does not trap in local optima easily and sustainably search the global optimum. As the two populations in PSOPB exchange information periodically, PSOPB has a good ability to keep the balance of exploration and exploitation. It is expected to that PSOPB can avoid local optima and bring about improved performance on multimodal functions. Indeed, the experimental results given in the table 4 and Fig. 1 support this intuition. PSOPB does not trap in local optima in the entire search process and find the remarkable solutions than other PSO algorithms in optimizing function f_3 which is a complex multimodal problem with a large number of local optima. All PSO algorithms compared have a good performance of solving functions f_4, but PSOPB has a faster convergence rate except CPSO and lowest standard deviation values. As for function f_5 in which has a $\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)$ component causing linkage among variables and function f_6, the complexity of which is due to deep local optima is far from the global optimum, it is difficult to find the global optimum. The performance of PSOPB to solving them is junior to FIPS and PSOPC. PSOPB can find good solutions in terms of accurate rate than other PSO algorithms for function f_7. Functions f_8, f_9 and f_{10} are classical benchmark functions with coordinate rotation. Results for three rotated appears that all the PSO algorithms are affected by the coordination rotation; however, it is interesting to observe that PSOPB can still find good solutions. The robustness of the algorithms is tested based on standard deviation. The tabulated results show that PSOPB is a robust algorithm compared to the involved PSO variants for functions $f_1, f_2, f_3, f_4, f_7, f_8$ and f_9.

As the mechanism of information exchange in PSOPB, the population diversity of PSOPB can be maintained. Therefore, PSOPB have high probability to find promising solutions. From Table 4 and Fig. 1, it can be observed that PSOPB is able to improve solutions steadily for a long period without being trapped in local optima. PSOPB appears to exhibit the strongest search ability among the involved PSOs and we can concludes that PSOPB is a good choice for obtaining good mean results and lower standard deviation in optimizing many problems especially for complicated.

5. Conclusions and Prospects

Based on the deeper analysis of biological symbiotic relationship, the mechanism of facultative parasitic behaviour was incorporated into the canonical PSO to construct a novel two-population PSO called PSOPB, composed of the host and the parasite population. The two populations periodically exchange particles, which are determined by fitness values sorted in each population, in a certain number of iterations. After that, the host population was harmed and the parasite population got nourishments, that is, the parasite population obtained the particles with good fitness from the host population. In order to be closer to bio-parasitic behaviour, the host immune response and the parasite mutation were considered in the proposed PSO algorithm. The law of "survival of the fittest" in biological evolution was also demonstrated in the host population. A set of 10 benchmark functions have been tested PSOPB in comparison with SPSO, CPSO, PSOPC, FIPS and FDR-PSO. The experimental results showed that PSOPB had a remarkable performance. However, the proposed algorithm only simulated bio-parasitic behavior to some extent. In the future, we will study further and use our presented algorithm to optimize practical engineering optimization problems.

Acknowledgements

We would like to thank referees for their penetrating remarks and suggestions concerning the earlier version of this paper. This work is partly supported by Natural Science Foundation of China under grant NO.71071057 and Natural Science Foundation of Guangdong Province under grant No.S2011010001337.

References

1. J. Kennedy and R. C. Eberhart. Particle swarm optimization, in *proc* of *IEEE Conf on Neural Networks*，(Perth, Australia, 1995), pp. 1942-1948.
2. J. Kennedy and R. C. Eberhart. *Swarm Intelligence* (Morgan Kaufmann, San Mateo , CA, 2001).
3. J. B. Park, K. S. Lee, J. R. Shin and K. Y. Lee. A particle swarm optimization for economic dispatch with nonsmooth cost functions. *IEEE T. power sys.* 20, (2005)34–42.
4. R. Mendes, P. Cortez P, M. Rocha M and J. Neves. Particle Swarms for Feedforward Neural Network Training. In *proc of Int Joint Conf on Neural Networks*, (2002), pp.1895-1899.
5. L. Hu, X. L. Che and X. C Cheng. Bandwidth prediction based on nu-support vector regression and parallel hybrid particle swarm optimization, *Int. J. Comput. Int. Sys.* 3(1) (2010) 70–83.
6. P. J. Angeline. Evolutionary optimization versus particle swarm optimization and philosophy and performance difference. In *proc of 7th Annual Conference on Evolutionary Programming*, (San Diego, USA, 1998), pp.601-610.
7. Y. Shi and R. C. Eberhart. A modified particle swarm optimizer, in *proc of IEEE Congress on Evolutionary Computation*, (Piscataway, NJ, 1998), pp.69-73.
8. J. Kennedy and R. Mendes. Population structure and particle swarm performance, in *Proc. of the IEEE Congress of Evolutionary Computation*,(Settle,WA, 2002), pp.1671-1676.
9. P. J. Angeline. Using selection to improve particle swarm optimization. In *IEEE World Congress on Computational Intelligence*,(Anchorage, AK, 1998), pp. 84-89.
10. Y. X. Shen, G. Y Wang and C. T. Tao. Particle swarm optimization with novel processing strategy and its application, *Int. J. Comput. Int. Sys.* 4(1) (2011) 100–111.
11. J. J. Liang, A. K. Qin, and P. N. Suganthan and S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE T. Evolut. Comput*. 10(3) (2006) 281–295.
12. A. Silva, A. Neves and E. Costa. An empirical comparison of particle swarm and predator prey optimisation . *Lect. Notes Artif. Intel* 2464 (2002)103-110.
13. S. He, Q.H. Wu and J.Y Wen, et al. A particle swarm optimizer with passive congregation. *Biosystems* 78 (2004)135-147.
14. B. Niu B, Y. L. Zhu. An Improved Particle Swarm Optimization Based on Bacterial Chemotaxis. In *Proc of the 6th World Congress on Intelligent Control and Automation*. (Dalian, China, 2006), pp.3193-3197.
15. H. N. Chen and Y. L. Zhu. Optimization based on symbiotic multi-species coevolution. *Appl Math Comput.* 205(2008) 47-60.
16. M. Clerc and J. Kennedy. The particle swarm: explosion, stability, and convergence in multi dimensional complex space. *IEEE T. Evolut. Comput*. 6 (2002) 58-73.
17. V. Ahmadjian and S. *Paracer Symbiosis: An introduction to biological associations*, (Oxford University Press, Oxford, 2000).
18. A.E. Douglas. *Symbiotic interactions*, (Oxford University Press, Oxford, 1994).
19. M. J. Lajeunesse and M. R. Forbes. Host range and local parasite adaption. *P Roy Soc*. 269(1492) (2002) 703-710.
20. R.M. Anderson and R. M. May. Regulation and stability of host-parasite population interactions. *J of Animal Ecol* 47(1978)210-247.
21. W. X. Li , G. T. Wang. Regulation of parasites on host populations: A brief. *Acta Hydrobiologica Sinica* 26(5) (2002)550-554.
22. Z. Y. Yang, K. Tang and X. Yao. Large scale evolutionary optimization using cooperative coevolution, *Inform. Sciences*178(15) (2008)2985-2999.
23. R. Mendes R., J. Kennedy and J. Neves. The fully informed particle swarm: simpler, may be better, *IEEE T. Evolut. Comput*. 8 (2004)204-210.

24. T. Peram, K. Veeramachaneni K and C. K. Mohan. Fitness-distance-ratio based Particle Swarm Optimization. in *Proc of the Swarm Intelligence Symposium*, (Indianapolis, IN, 2003), pp. 174-181.
25. A. Carlisle and G. Dozier. An off-the-shelf PSO, in *Proc of the workshop on particle swarm optimization.* (Indianapolis, IN, 2001), pp. 1-6.