Available online at www.sciencedirect.com

SCIENCE DIRECT®

BioSystems 78 (2004) 135–147

BioSystems

www.elsevier.com/locate/biosystems

A particle swarm optimizer with passive congregation

S. He[a], Q.H. Wu[a,*], J.Y. Wen[a], J.R. Saunders[b], R.C. Paton[c]

[a] *Department of Electrical Engineering and Electronics, The University of Liverpool, Liverpool L69 3GJ, UK*
[b] *School of Biological Sciences, The University of Liverpool, Liverpool L69 3BX, UK*
[c] *Department of Computer Science, The University of Liverpool, Liverpool L69 3BX, UK*

Received 19 August 2004; received in revised form 25 August 2004; accepted 25 August 2004

Abstract

This paper presents a particle swarm optimizer (PSO) with passive congregation to improve the performance of standard PSO (SPSO). Passive congregation is an important biological force preserving swarm integrity. By introducing passive congregation to PSO, information can be transferred among individuals of the swarm. A particle swarm optimizer with passive congregation (PSOPC) is tested with a set of 10 benchmark functions with 30 dimensions and compared to a global version of SPSO (GSPSO), a local version of SPSO (LSPSO), and PSO with a constriction factor (CPSO), respectively. Experimental results indicate that the PSO with passive congregation improves the search performance on the benchmark functions significantly.

Keywords: Particle swarm optimizer; Passive congregation; Optimization

1. Introduction

The particle swarm optimizer (PSO) is a population-based algorithm that was invented by Kennedy and Eberhart (1995), which was inspired by the social behavior of animals such as fish schooling and bird flocking. Similar to other population-based algorithms, such as evolutionary algorithms, PSO can solve a variety of difficult optimization problems but has shown a faster convergence rate than other evolutionary algorithms on some problems (Kennedy and Eberhart, 2001). Another advantage of PSO is that it has very few parameters to adjust, which makes it particularly easy to implement.

Angeline (1998) pointed out that although PSO may outperform other evolutionary algorithms in the early iterations, its performance may not be competitive as the number of generations is increased. Recently, several investigations have been undertaken to improve the performance of standard PSO (SPSO). Løbjerg et al. (2001) presented a hybrid PSO model with breeding and subpopulations. Kennedy and Mendes (2002) investigated the the impacts of population structures to the search performance of SPSO. Other investigations on improving PSO's performance were undertaken using cluster analysis (Kennedy, 2000) and fuzzy adaptive inertia weight (Shi and Eberhart, 2001).

* Corresponding author. Tel.: +44 151 7944535; fax: +44 151 7944540.
E-mail address: q.h.wu@liv.ac.uk (Q.H. Wu).

0303-2647/$ – see front matter
doi:10.1016/j.biosystems.2004.08.003

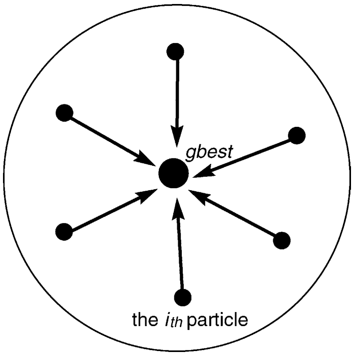

Fig. 1. Interaction between particles and the best particle *gbest*.

The foundation of PSO is based on the hypothesis that social sharing of information among conspecifics offers an evolutionary advantage (Kennedy and Eberhart, 1995). The SPSO model is based on the following two factors (Kennedy and Eberhart, 1995):

(1) The autobiographical memory, which remembers the best previous position of each individual (P_i) in the swarm;
(2) The publicized knowledge, which is the best solution (P_g) found currently by the population.

Therefore, the sharing of information among conspecifics is achieved by employing the publicly available information P_g, shown in Fig. 1. There is no information sharing among individuals except that P_g broadcasts the information to the other individuals. Therefore, the population may lose diversity and is more likely to confine the search around local minima if committed too early in the search to the global best found so far.

Biologists have proposed four types of biological mechanisms that allow animals to aggregate into groups: passive aggregation, active aggregation, passive congregation, and social congregation (Parrish and Hamner, 1997). There are different information sharing mechanisms inside these forces. We found that the passive congregation model is suitable to be incorporated in the SPSO model. Inspired by this research, we propose a hybrid model of PSO with passive congregation.

Section 2 introduces the SPSO. A PSO algorithm with passive congregation is presented in Section 3. In Section 4, we describe the test functions, experimental settings, and the experimental results. The discussions are given in Section 5. The paper is concluded in Section 6.

2. Standard particle swarm optimizer

PSO is a population-based optimization algorithm. The population of PSO is called a *swarm* and each individual in the population of PSO is called a *particle*. The ith particle at iteration k has the following two attributes:

(1) A current position in an N-dimensional search space $X_i^k = (x_1^k, \ldots, x_n^k, \ldots, x_N^k)$, where $x_n^k \in [l_n, u_n]$, $1 \le n \le N$, l_n and u_n is lower and upper bound for the nth dimension, respectively.
(2) A current velocity V_i^k, $V_i^k = (v_1^k, \ldots, v_n^k, \ldots, v_N^k)$, which is bounded by a maximum velocity $V_{\max}^k = (v_{\max,1}^k, \ldots, v_{\max,n}^k, \ldots, v_{\max,N}^k)$ and a minimum velocity $V_{\min}^k = (v_{\min,1}^k, \ldots, v_{\min,n}^k, \ldots, v_{\min,N}^k)$.

In each iteration of PSO, the swarm is updated by the following equations (Kennedy and Eberhart, 1995):

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \quad (1)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (2)$$

where P_i is the best previous position of the ith particle (also known as *pbest*). According to the different definitions of P_g, there are two different versions of PSO. If P_g is the best position among all the particles in the swarm (also known as *gbest*), such a version is called the global version. If P_g is taken from some smaller number of adjacent particles of the population (also known as *lbest*), such a version is called the local version. P_i and P_g are given by the following equations, respectively:

$$P_i = \begin{cases} P_i & : \quad f(X_i) \ge P_i \\ X_i & : \quad f(X_i) < P_i \end{cases} \quad (3)$$

$$P_g \in \{P_0, P_1, \ldots, P_m\} | f(P_g) = \min(f(P_0), f(P_1), \ldots, f(P_m)) \quad (4)$$

where f is the objective function, $m \le M$ and M is the total number of particles, r_1 and r_2 are elements from two uniform random sequences in the range (0, 1): $r_1 \sim U(0, 1)$; $r_2 \sim U(0, 1)$, and ω is an inertia weight (Shi and Eberhart, 1997), which is initialized typically in the range of [0,1]. A larger inertia weight facilitates global exploration and a smaller inertia weight tends to facilitate local exploration to fine-tune the current search area (Shi and Eberhart, 1998). The variables c_1 and c_2 are acceleration constants (Eberhart and Shi, 2001), which control how far a particle will move in a single iteration.

Another important variant of standard PSO is the constriction factor approach PSO (CPSO), which was proposed by Clerc and Kennedy (2002). The velocity of CPSO is updated by the following equation:

$$V_i^{k+1} = \chi(V_i^k + c_1 r_1(P_i^k - X_i^k) + c_2 r_2(P_g^k - X_i^k)) \tag{5}$$

where χ is called a constriction factor, given by:

$$\chi = \frac{2}{|2 - \varphi + \sqrt{\varphi^2 - 4\varphi}|} \quad \text{where}$$
$$\varphi = c_1 + c_2,\ \varphi > 4 \tag{6}$$

The CPSO ensures the convergence of the search procedures and can generate higher-quality solutions than the standard PSO with inertia weight on some studied problems (Eberhart and Shi, 2000).

3. Particle swarm optimizer with passive congregation

The PSO algorithm is inspired by social behaviors such as spatial order, more specially, aggregation such as bird flocking, fish schooling, or swarming of insects. Each of these cases has stable spatio-temporal integrities of the group of organisms: the group moves persistently as a whole without losing the shape and density.

For each of these groups, different biological forces are essential for preserving the group's integrity. Parrish and Hamner (1997) proposed mathematical models of the spatial structure of animal groups to show how animals organize themselves. In these models, aggregation sometimes refers to a grouping of the organisms by non-social, external, physical forces. There are two types of aggregation: passive aggregation and active aggregation. Passive aggregation is a passive grouping by physical processes. One example of passive aggregation is the dense aggregation of plankton in open water, in which the plankton are not attracted actively to the aggregation but are transported passively there via physical forces such as water currents. Active aggregation is a grouping by attractive resource, such as food or space, with each member of the group recruited to a specific location actively. Congregation, which is different from aggregation, is a grouping by social forces, that is the source of attraction is the group itself. Congregation can be classified into passive congregation and social congregation. Passive congregation is an attraction of an individual to other group members but where there is no display of social behavior. Social congregations usually happen in a group where the members are related (sometimes highly related). A variety of inter-individual behaviors are displayed in social congregations, necessitating active information transfer (Parrish and Hamner, 1997). For example, ants use antennal contacts to transfer information about individual identity or location of resources (Gordon et al., 1993).

From the definitions above, the third part of Eq. (1): $c_2 r_2(P_g^k - X_i^k)$ can be classified as either active aggregation or passive congregation. But since P_g is the best solution the swarm has found so far, which can be regarded as the place with most food, we argue that it is better to classify $c_2 r_2(P_g^k - X_i^k)$ as active aggregation.

It has been discovered that in spatially well-defined congregations, such as fish schools, individuals may have low fidelity to the group because the congregations may be composed of individuals with little to no genetic relation to each other (Hilborn, 1991). Schooling fish are generally considered a "selfish herd" (Hamilton, 1971), in that each individual attempts to take the sweeping generalization advantage from group living, independent of the fates of neighbors (Pitcher and Parrish, 1993). In these congregations, information may be transferred passively rather than actively (Magurran and Higham, 1988). Such asocial types of congregations can be referred to as passive congregation. Because PSO is inspired by fish schooling, it is therefore natural to ask if a passive congregation model can be employed to increase the performance of SPSO. Here, we do not consider other models such as passive aggregation, because PSO is not aggregated passively via physical processes. And social congregation usually happens when group fidelity is high, such that the chance of each individual meeting any of the others is high (Alexander, 1974). Social congregations frequently display a division of labor. In a social insect colony, such as an ant colony, large tasks are accomplished collectively by groups of specialized individuals, which is more efficient than performing sequentially by unspecialized individuals (Bonabeau et al., 1999). The concept of labor division can be employed by data clustering, sorting (Deneubourg et al., 1991) and data analysis (Lumer and Faieta, 1994).

Group members in an aggregation can react without direct detection of an incoming signals from the environment, because they can get necessary information from their neighbors (Parrish and Hamner, 1997). Individuals need to monitor both environment and their immediate surroundings, such as the bearing and speed of their neighbors (Parrish and Hamner, 1997)). Therefore, each individual in an aggregation has a multitude of potential information from other group members that may minimize the chance of missed detection and incorrect interpretations (Parrish and Hamner, 1997). Such information transfer can be employed in the model of passive congregation. Inspired by this result, and to keep the model simple and uniform with SPSO, we propose a hybrid PSO with passive congregation:

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) + c_3 r_3 (R_i^k - X_i^k) \quad (7)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (8)$$

where R_i is a particle selected randomly from the swarm, c_3 the passive congregation coefficient, and

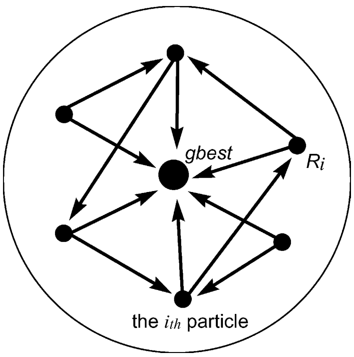

Fig. 2. Interactions of particles with passive congregation.

Table 1
Pseudocode for the PSOPC algorithm

Set $k := 0$;
Randomly initialize positions and velocities of all particles;
WHILE (the termination conditions are not met)
FOR (each particle i in the swarm)
Calculate fitness: Calculate the fitness value of current particle: $f(X_i)$
Update *pbest*: Compare the fitness value of *pbest* with $f(X_i)$. If $f(X_i)$ is better than the fitness value of *pbest*, then set *pbest* to the current position X_i;
Update *gbest*: Find the global best position of the swarm. If $f(X_i)$ is better than the fitness value of *gbest*, then *gbest* is set to the position of the current particle X_i;
Update R_i: Randomly select a particle from the swarm as R_i;
Update velocities: Calculate velocities V_i using Eq. (7). If $V_i > V_{\max}$ then $V_i = V_{\max}$. If $V_i < V_{\min}$ then $V_i = V_{\min}$;
Update positions: Calculate positions X_i using Eq. (8);
END FOR
Set $k := k + 1$;
END WHILE

r_3 is a uniform random sequence in the range (0,1): $r_3 \sim U(0, 1)$. The interactions between individuals of PSOPC are shown in Fig. 2. The pseudocode for PSOPC is listed in Table 1.

4. Experimental studies

4.1. Test functions

In our experimental studies, a set of 10 benchmark functions was employed to evaluate the PSOPC algorithm in comparison with others.

Sphere model:

$$f_1(x) = \sum_{i=1}^{30} x_i^2$$

Schwefel's Problem 1.2:

$$f_2(x) = \sum_{i=1}^{30} \left(\sum_{j=1}^{i} x_j \right)^2$$

Schwefel's Problem 2.21:

$$f_3(x) = \max_i \{ |x_i|, 1 \le i \le 30 \}$$

Generalized Rosenbrock's function:

$$f_4(x) = \sum_{i=1}^{29} (100(x_{i+1} - x_i^2)^2 + (x_i - 1))^2$$

Generalized Schwefel's Problem 2.26:

$$f_5(x) = -\sum_{i=1}^{30} (x_i \sin(\sqrt{|x_i|}))$$

Generalized Rastrigin's function:

$$f_6(x) = \sum_{i=1}^{30} (x_i^2 - 10\cos(2\pi x_i) + 10)^2$$

Ackley's function:

$$f_7(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30} x_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{30}\cos 2\pi x_i\right) + 20 + e$$

Generalized Griewank function:

$$f_8(x) = \frac{1}{4000}\sum_{i=1}^{30}(x_i - 100)^2 - \prod_{i=1}^{30}\cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$$

Generalized Penalized Functions:

$$f_9 = \frac{\pi}{30}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{29}(y_i - 1)^2 \times [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{30} u(x_i, 10, 100, 4) \tag{9}$$

and

$$f_{10} = 0.1\left\{\sin^2(\pi y_1) + \sum_{i=1}^{29}(y_i - 1)^2 \times [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{30} u(x_i, 10, 100, 4) \tag{10}$$

where

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$$

$$y_i = 1 + \tfrac{1}{4}(x_i + 1)$$

The above benchmark functions were tested widely by Yao et al. (1999), Chellapilla (1998), Fogel (1991). They can be grouped as unimodal (function f_1–f_4) and multimodal functions (function f_5–f_{10}) where the number of local minima increases exponentially with the problem dimension. The dimension of each function n, feasible solution space, and $f_{\min}$ are listed in Table 2.

4.2. Experimental setting

To evaluate the performance of the proposed PSOPC, three variants of standard PSO were used for comparisons: global version of standard PSO (GSPSO), local version of standard PSO (LSPSO), and

Table 2
Basic characters of the test functions

Function	n	Feasible solution space	$f_{\min}$
f_1	30	$[-100, 100]^n$	0
f_2	30	$[-100, 100]^n$	0
f_3	30	$[-100, 100]^n$	0
f_4	30	$[-30, 30]^n$	0
f_5	30	$[-500, 500]^n$	−12569.5
f_6	30	$[-5.12, 5.12]^n$	0
f_7	30	$[-32, 32]^n$	0
f_8	30	$[-600, 600]^n$	0
f_9	30	$[-50, 50]^n$	0
f_{10}	30	$[-50, 50]^n$	0

constriction factor version of PSO (CPSO). The parameters used for these three standard PSO were recommended from Kennedy and Eberhart (2001), Clerc and Kennedy (2002), Eberhart and Shi (2001), Shi and Eberhart (1998), Kennedy and Mendes (2002), or hand selected.

The population size of all algorithms used in our experiments was set at 100. The maximum velocity $V_{\max}$ and minimum velocity $V_{\min}$ for GSPSO, LSPSO and CPSO were set at half value of the upper bound and lower bound, respectively. $V_{\max}$ and $V_{\min}$ for PSOPC was set to the upper bound and lower bound, respectively. The acceleration constants c_1 and c_2 for GSPSO and LSPSO were both 2.0 (Kennedy and Eberhart, 2001). For CPSO, a setting of $c_1 = c_2 = 2.05$ was adopted (Clerc and Kennedy, 2002). The acceleration constants $c_1 = c_2 = 0.5$ were used in PSOPC.

The inertia weight ω is critical for the convergence behavior of GSPSO and LSPSO. A suitable value for the inertia weight ω usually provides a balance between global and local exploration abilities and consequently results in a better optimum solution. Initially, the inertia weight was constant. However, experimental results indicated that it is better to initially set the inertia to a large value in order to promote global exploration of the search space and decrease it to get more refined solutions (Eberhart and Shi, 2001). Therefore, a decaying inertia weight starting at 0.9 and ending at 0.4 following Shi and Eberhart (1998), was used for GSPSO and LSPSO. The inertia weight for PSOPC started at 0.9 and ended at 0.7. For CPSO, the constriction factor was calculated with Eq. (6), that is $\chi = 0.73$. The neighborhood size of LSPSO was set to be 2 (Kennedy and Mendes, 2002).

The newly introduced passive congregation coefficient c_3 is important for the search performance of PSOPC. Experiments were executed to select a proper value of c_3. Four benchmark functions: f_1 (Sphere function), f_5 (Rosenbrock function), f_9 (Rastrigin function), and f_{11} (Griewank function) were tested with different values of c_3. The average test results obtained from 25 runs are listed in Table 4. When $c_3 = 0.6$, PSOPC generated good results on functions f_1 and f_6. For functions f_4 and f_8, the best results were generated at the point $c_3 = 0.8$. With $c_3 \geq 0.9$, the search performance of PSOPC on function f_1, f_4, and f_8 is deteriorated (Table 3.). For functions f_6, c_3 should be equal or smaller than 0.6 otherwise PSOPC will not converge in 2000 generations. Therefore, a

Table 3
Average fitness values of functions f_1, f_4, f_6 and f_8 with different c_3

c_3	Function			
	f_1	f_4	f_6	f_8
0.0	5.6×10^{-12}	69.65	29.02	1.2×10^{-2}
0.1	1.4×10^{-8}	56.39	23.33	7.1×10^{-3}
0.2	9.6×10^{-11}	49.95	12.04	1.3×10^{-2}
0.3	6.5×10^{-14}	31.58	6.87	5.3×10^{-3}
0.4	4.2×10^{-18}	30.21	4.15	7.4×10^{-3}
0.5	1.1×10^{-20}	35.83	3.45	3.9×10^{-3}
0.6	5.8×10^{-28}	33.70	2.95	4.1×10^{-3}
0.7	2.7×10^{-21}	34.53	167.86	3.3×10^{-3}
0.8	4.6×10^{-6}	29.70	231.14	9.2×10^{-4}
0.9	4510.26	1.4×10^{6}	273.78	29.72
1.0	20336.59	1.4×10^{7}	300.20	129.17

Table 4
Average fitness value of Rastrigin (f_9) function with different linearly increasing c_3

$c_{3\,\max}$	$c_{3\,\min}$						
	0.1	0.2	0.3	0.4	0.5	0.6	0.7
0.1	23.33	NA	NA	NA	NA	NA	NA
0.2	22.54	12.04	NA	NA	NA	NA	NA
0.3	22.33	11.49	6.87	NA	NA	NA	NA
0.4	20.60	10.35	5.43	4.15	NA	NA	NA
0.5	21.04	11.75	5.23	4.24	3.45	NA	NA
0.6	17.71	8.91	4.43	2.66	2.79	2.95	NA
0.7	17.51	6.52	6.87	3.72	107.72	156.69	167.86
0.8	31.55	212.03	219.24	230.91	233.38	232.10	229.56
0.9	148.70	257.99	264.46	267.02	269.83	274.41	270.42
1.0	202.57	296.50	295.15	301.24	301.71	301.25	302.82

generic c_3 for all functions should be equal or smaller than 0.6.

It is our interest to investigate whether PSOPC with a linear increasing c_3 generates better results on the benchmark functions than PSOPC with a fixed value of c_3. Therefore, f_9 (Rastrigin function) was selected and tested with different ranges of linearly increasing c_3. The results are tabulated in Table 4. The best result was generated by PSOPC with a linearly increasing passive congregation coefficient c_3, which started at 0.4 and ended at 0.6.

The parameters setting for all algorithms are summarized in Table 5.

All experiments were repeated for 50 runs. A fixed number of maximum generations 2000 was applied to all algorithms.

4.3. Experimental results and comparison

The experimental results (i.e., the mean and the standard deviations of the function values found in 50 runs) for each algorithm on each test function are listed in Table 6. To measure the statistical significance of our experimental results between PSOPC and other three standard PSO variants, a set of two-tailed tests was adopted. The results are listed in Table 7. The critical value with 49 degrees of freedom at $\alpha = 0.05$ is 2.0, which means if $|t| > 2.0$ the difference between two means is statistically significant.

From Table 6, PSOPC outperformed the other three standard PSO algorithms significantly for most of the benchmark functions. The two exceptions are f_2 and

Table 5
Parameter setting

	PSOPC	GSPSO	LSPSO	CPSO
Population size	100	100	100	100
Neighborhood size	100	100	2	100
ω	[0.7, 0.9]	[0.4, 0.9]	[0.4, 0.9]	NA
χ	NA	NA	NA	0.73
c_1	0.5	2.0	2.0	2.05
c_2	0.5	2.0	2.0	2.05
c_3	[0.4, 0.6]	NA	NA	NA

Table 6
Comparison between PSOPC, GSPSO, LSPSO, and CPSO

Function	Mean function value (standard deviation)			
	PSOPC	GSPSO	LSPSO	CPSO
f_1	9.5×10^{-29} (5.9×10^{-28})	1.9×10^{-12} (3.7×10^{-12})	3.0×10^{-3} (5.3×10^{-3})	2.3×10^{-15} (4.9×10^{-15})
f_2	2.71 (2.79)	62.33 (34.69)	1805.03 (458.70)	2.29 (1.99)
f_3	7.5×10^{-3} (1.0×10^{-2})	1.2 (5.1×10^{-1})	11.37 (1.93)	6.1×10^{-2} (4.2×10^{-2})
f_4	32.44 (23.48)	52.83 (38.38)	347.92 (417.85)	39.70 (31.51)
f_5	−12267.77 (−451.86)	−10768.82 (−417.20)	−10928.65 (−1013.97)	−10443.47 (−618.94)
f_6	2.91 (1.65)	21.56 (5.12)	59.07 (10.00)	43.76 (12.13)
f_7	2.3×10^{-14} (2.2×10^{-14})	9.0×10^{-8} (9.3×10^{-8})	1.41 (8.6×10^{-1})	1.6×10^{-8} (1.8×10^{-8})
f_8	3.2×10^{-3} (5.6×10^{-3})	1.4×10^{-2} (1.6×10^{-2})	9.6×10^{-3} (1.0×10^{-2})	1.9×10^{-2} (2.0×10^{-2})
f_9	4.5×10^{-26} (3.2×10^{-25})	2.0×10^{-3} (1.5×10^{-2})	1.06 (9.2×10^{-1})	3.9×10^{-2} (8.6×10^{-2})
f_{10}	1.1×10^{-3} (3.3×10^{-3})	8.84×10^{-4} (3.0×10^{-3})	13.35 (21.57)	3.8×10^{-1} (1.8)

f_{10}. For function f_2, the result generated by PSOPC is better than those generated by GSPSO and LSPSO but slightly worse than the result of CPSO. For function f_{10}, GSPSO slightly outperformed PSOPC while the result of PSOPC is far better than LSPSO and CPSO. However, from Table 7, for functions f_2 and f_{10}, the results generated, respectively, by CPSO and GSPSO are not significantly better than PSOPC. For function f_9, the results obtained from PSOPC do not differ significantly from those generated by GSPSO. For functions f_4 and f_{10}, the differences between the results generated by PSOPC and CPSO are not statistically significant. It can be concluded that PSOPC significantly outperforms LSPSO on all the tested benchmark functions.

Table 7
Two-tailed test on PSOPC, GSPSO, LSPSO, and CPSO

Function	t		
	PSOPC–GSPSO	PSOPC–LSPSO	PSOPC–CPSO
f_1	−3.57	−3.90	−3.28
f_2	−12.46	−27.77	0.85
f_3	−16.50	−41.71	−8.65
f_4	−3.25	−5.29	−1.17
f_5	−15.76	−8.05	−17.87
f_6	−24.14	−40.23	−23.86
f_7	−6.82	−11.62	−6.42
f_8	−4.88	−3.80	−5.14
f_9	−1.00	−8.09	−3.24
f_{10}	0.37	−4.37	−1.44

The value of t with 49 degree of freedom is significant at $\alpha = 0.05$ by a two-tailed test and $t_{0.025} = 2.0$.

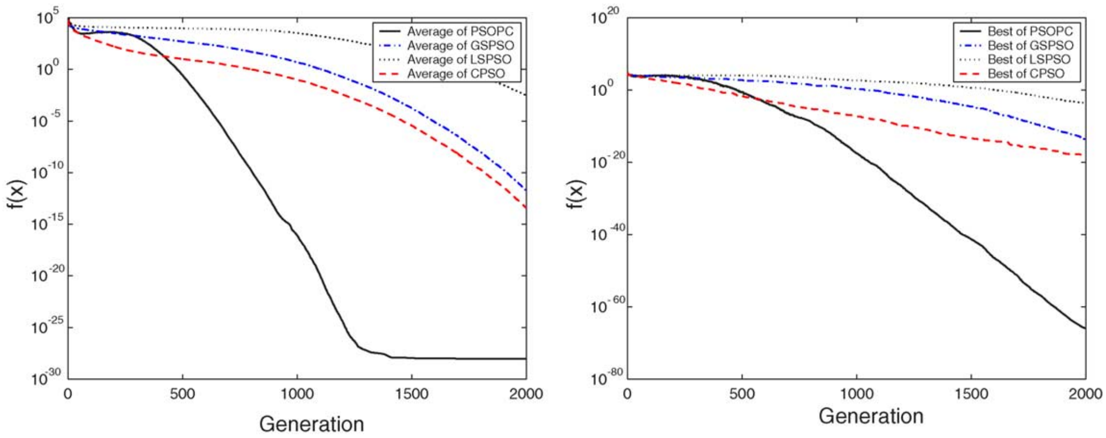

Fig. 3. f_1, Sphere function.

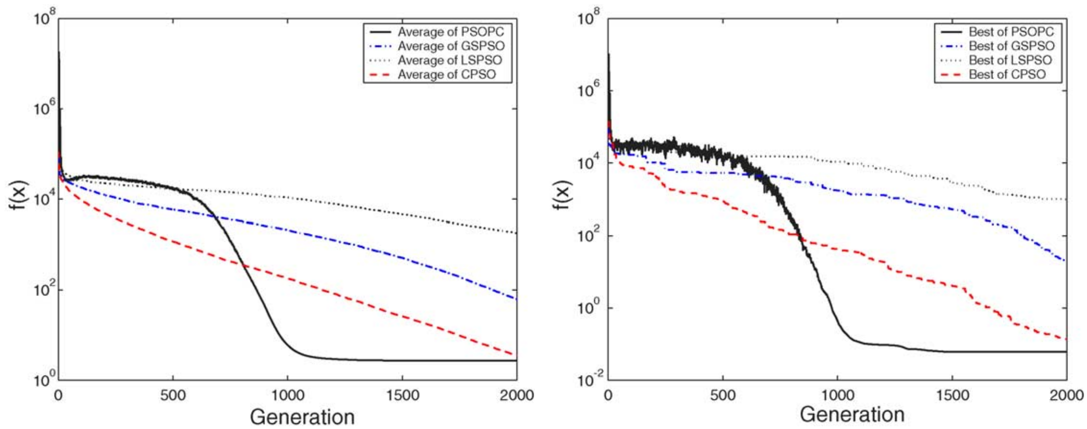

Fig. 4. f_2, Schwefel's Problem 1.2.

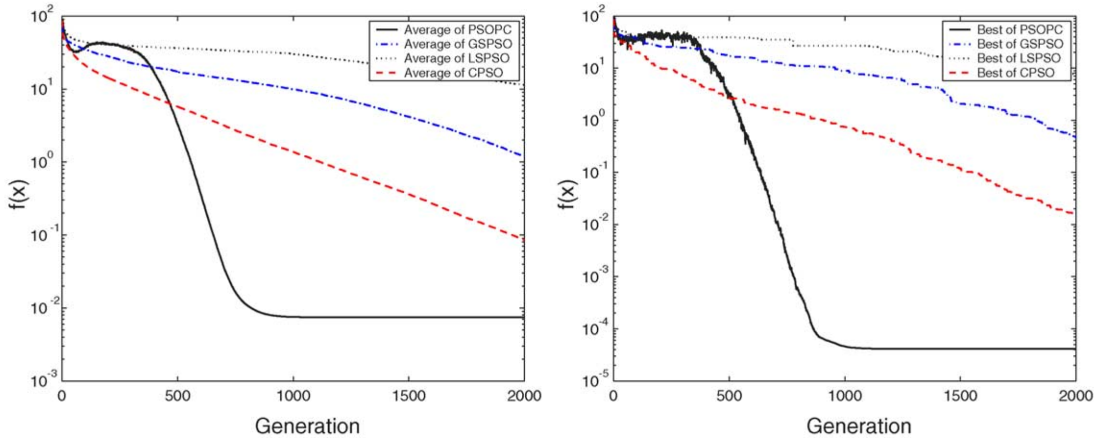

Fig. 5. f_3, Schwefel's Problem 2.21.

The performance of CPSO is better than GSPSO on all the unimodel benchmark functions (functions f_1–f_4). But GSPSO results in good performance on the multimodel benchmark functions (functions f_5–f_{10}). Although it is believed that LSPSO is able to "flow around" local optima (Kennedy and Mendes, 2002), our experimental results have indicated that GSPSO and CPSO exhibit better global convergence performance. The search performance of four algorithms tested here can be ordered as PSOPC $>$ GSPSO $\sim$ CPSO $>$ LSPSO.

Figs. 3–12 show the search progress of the average values and the best solutions found by the four algorithms over 50 runs for functions f_1–f_{10}. From these figures, for most of the benchmark functions, PSOPC quickly found the near optima in the early search process.

For unimodel functions (function f_1–f_4), the convergence rates are more important than the final results of optimization as there are other methods such as gradient-based search methods that are designed specially to optimize unimodal functions (Yao et al., 1999). From Figs. 3–6, it can be seen that PSOPC has a faster convergence rate than other three algorithms.

Functions f_5–f_{10} are multimodal functions that are very difficult to optimize, since the number of local minima increases exponentially as the function dimension increases (Törn and Zilinskas (1989), Schwefel

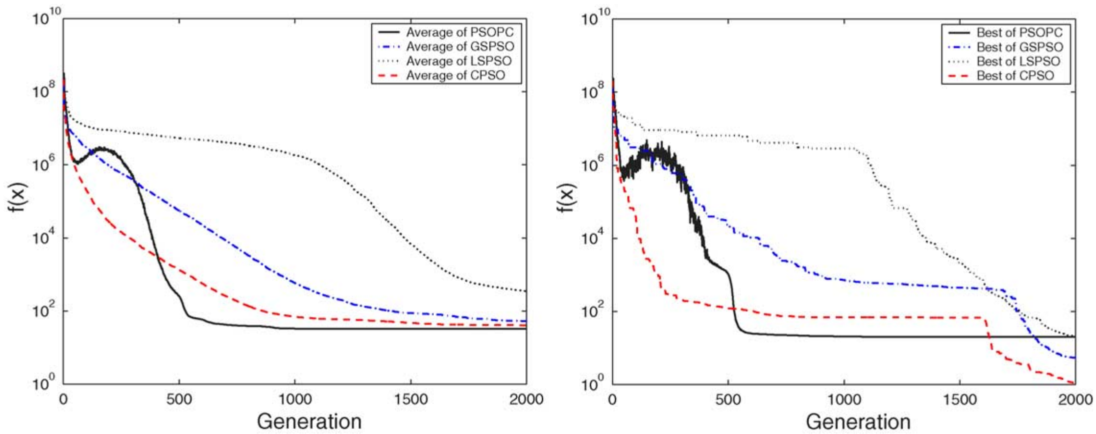

Fig. 6. f_4, Generalized Rosenbrock function.

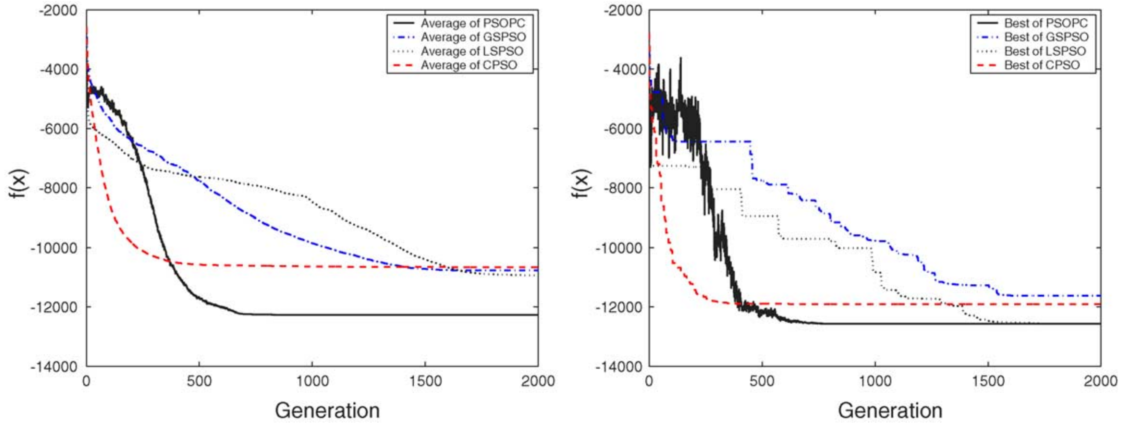

Fig. 7. f_5, Generalized Schwefel's Problem 2.26.

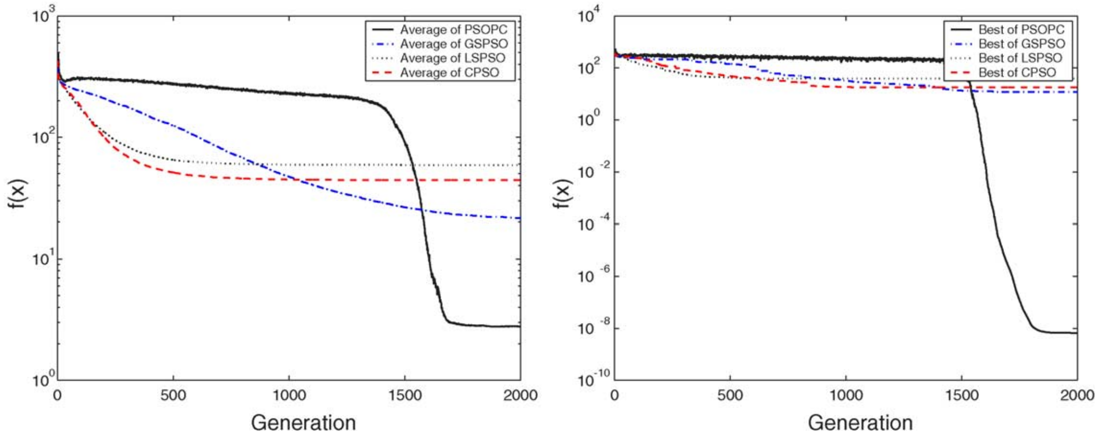

Fig. 8. f_6, Generalized Rastrigin's function.

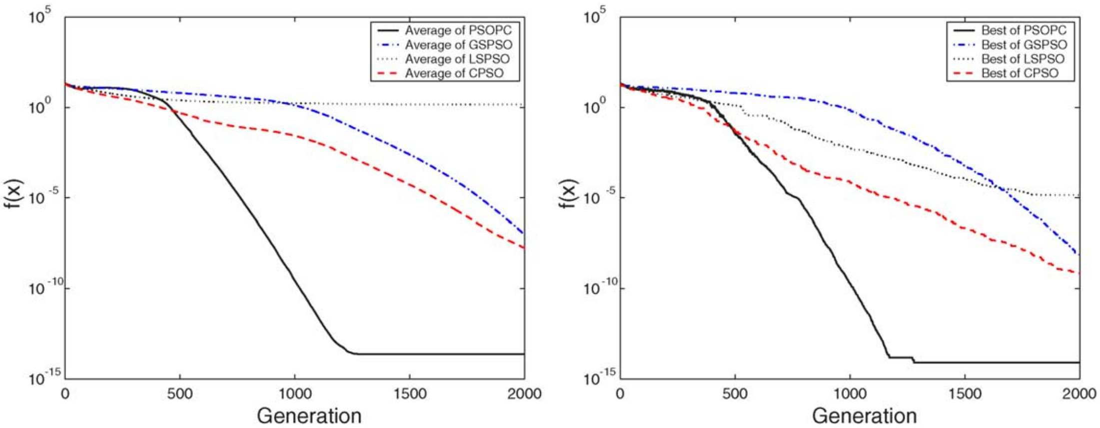

Fig. 9. f_7, Ackley's function.

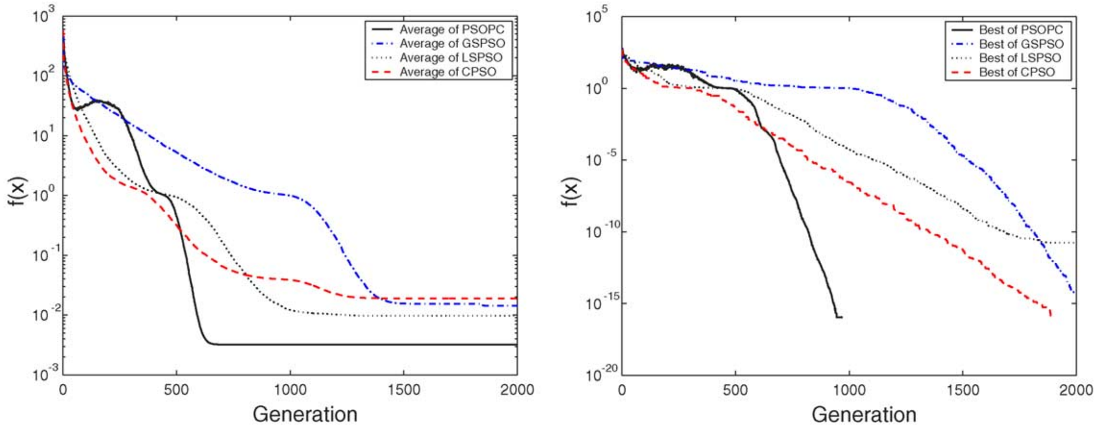

Fig. 10. f_8, Generalized Griewank function.

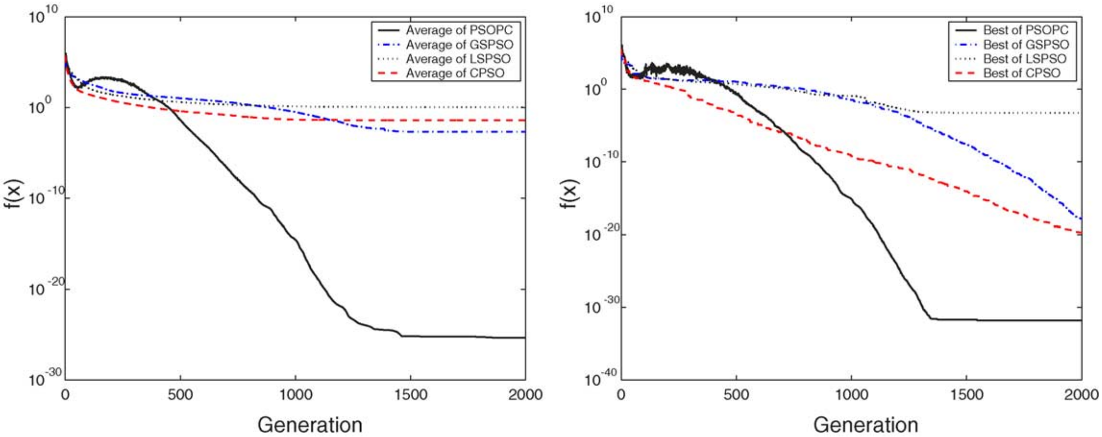

Fig. 11. f_9, Penalized function P8.

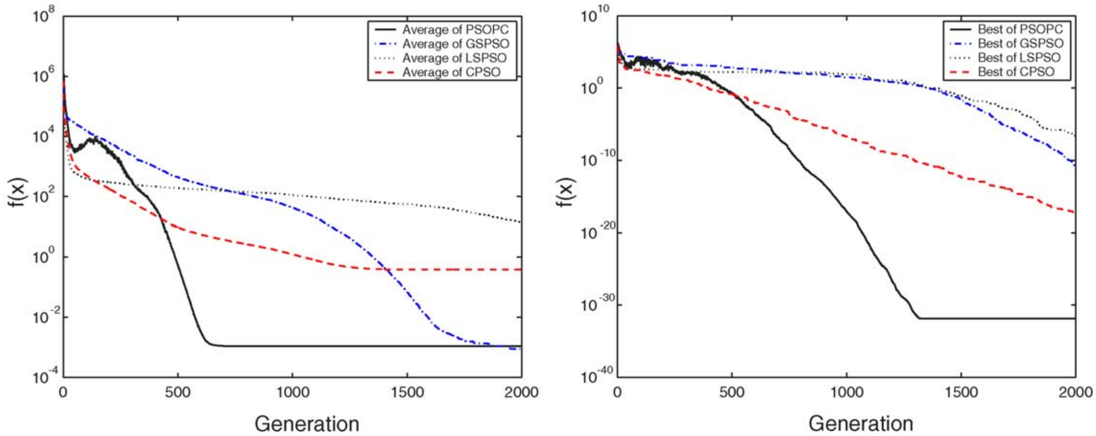

Fig. 12. f_{10}, Penalized function P16.

(1995)). The search process of four algorithms for f_5–f_{10} is shown by Figs. 7–12. According to these figures, for most of the functions (f_5, f_6, f_8, f_9 and f_{10}), PSOPC converges near global minima while the other three algorithms were trapped by poor local minima and then stagnated. The only exception is about function f_7, for which GSPSO and CPSO did not fully converge when the maximum generations was reached.

5. Discussion

Arithmetically, this passive congregation operator can be regarded as a stochastic variable that introduces perturbations to the search process. Fieldsend and Singh (2002) also introduced a stochastic variable into the standard PSO, which is referred to as *turbulence* in their paper. The velocity-updating equation is given by

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) + r_3 \tag{11}$$

where r_3 is a random variable $r_3 \sim U(0, 0.1R)$, and R is the absolute range of the model parameter.

From our experience, a large R will help the swarm escape local minima but may also cause the search process to diverge. A too small R may have no impact on search performance. The value of R is also problem-specific, e.g., a suitable R for some benchmark functions will deteriorate the search performance on other functions. Therefore, finding a proper value of R is necessary for the best solution of an optimization problem.

Compared with the turbulence factor r_3, the passive congregation operator $c_3 r_3 (R_i^k - X_i^k)$ is more adaptive to different optimization problems. For each individual, the turbulence (perturbation) is proportional to the distance between itself and a randomly selected neighborhood rather than an external random number. In the early search process, the distances between individuals are large, therefore, the turbulence is large, which may allow the swarm to avoid converging to a poor local minimum. As the generations increase, the distances between individuals become smaller, therefore, the turbulence becomes smaller, which enables the swarm to refine solutions.

Kennedy and Mendes (2002) investigated population topologies of PSO systematically. In their study, two sociometric variables, the number of neighbors for each node in the population k and the number of neighbors in common C, were varied to generated different topologies. One experiment, called random graphs, is implemented to generate different topologies by randomly initialized different k, C, standard deviation of k (std$k$), and standard deviation of C (std$C$), and then optimized by a method with a cooling mechanism that was inspired by simulated annealing. Since the PSO algorithm used in their work is the CPSO as defined in Eq. (5), the only factor affected by k, C, std$k$, and std$C$ is P_g. Therefore, the essential result of their experiment is most likely finding a proper selection scheme for P_g rather than introducing a new information sharing mechanism into swarms as PSOPC does.

6. Summary

In this paper, a new PSO with passive congregation (PSOPC) has been presented based on the standard PSO. By introducing passive congregation, information can be transferred among individuals that will help individuals to avoid misjudging information and becoming trapped by poor local minima. The only coefficient introduced into the standard PSO is the passive congregation coefficient c_3. A generic value of c_3 was selected by experiments.

A set of 10 benchmark functions has been used to test PSOPC in comparison with GSPSO, LSPSO, and CPSO. Among them, four functions were unimodal and six were multimodal. For most of the multimodal benchmark functions, PSOPC found better results than those generated by the other three standard PSO variants. For most of the unimodal functions, of which the convergence rate is more important than the final results, our PSOPC outperformed the other three algorithms in terms of accuracy and convergence rate. We also applied two-tailed tests to evaluate the statistical significance of differences between PSOPC and the other three algorithms. The results indicated that for 6 out of 10 benchmark functions, PSOPC performed significantly better than all other three standard PSO variants.

Acknowledgements

We wish to dedicate this paper to the memory of Dr. Ray C. Paton, who passed away on July 29, 2004. He is greatly missed by us all. Thanks to three anonymous referees for the comments on an earlier version of this paper and Dr. David Fogel for his comments to improve the quality of this paper.

References

Alexander, R.D., 1974. The evolution of social behaviour. Annu. Rev. Ecol. Syst. 5, 325–383.

Angeline, P., 1998. Evolutionary optimization versus particle swarm optimization: philosophy and performance difference. In: Proceedings of the Evolutionary Programming Conference, San Diago, USA.

Bonabeau, E., Dorigo, M., Theraulaz, G., 1999. Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press.

Chellapilla, K., September 1998. Combining mutation operators in evolutionary programming. IEEE Trans. Evolut. Comput. 2 (3), 91–96.

Clerc, M., Kennedy, J., 2002. The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space. IEEE Trans. Evolut. Comput. 6 (1), 58–73.

Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., Chretien, L., 1991. The dynamics of collective sorting: robot-like ant and ant-like robot. In: Proceedings of the First Conference on Simulation of Adaptive Behavior: from Animals to Animals. pp. 356–365.

Eberhart, R., Shi, Y., 2000. Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation (CEC2000). pp. 84–88.

Eberhart, R.C., Shi, Y., 2001. Particle swarm optimization: developments, applications and resources. In: Proceedings of IEEE International Conference on Evolutionary Computation. pp. 81–86.

Fieldsend, J.E., Singh, S., 2002. A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence. In: Proceedings of UK Workshop on Computational Intelligence (UKCI'02). pp. 37–44.

Fogel, D.B., 1991. System Identification Through Simulated Evolution: A Machine Learning Approach to Modeling. Ginn Press, 160 Gould Street, Needham Heights, MA 01294.

Gordon, D.M., Paul, R.E., Thorpe, K., 1993. What is the function of encounter pattern in ant colonies? Anim. Behav. 45, 1083–1100.

Hamilton, W.D., 1971. Geometry for the selfish herd. J. Theor. Biol. 31, 295–311.

Hilborn, R., 1991. Modelling the stability of fish schools: exchange of individual fish between schools of skipjack tuna (*Katsuwonus pelamis*). Can. J. Fish. Aquat. Sci. 48, 1080–1091.

Kennedy, J., 2000. Stereotyping: improving particle swarm performance with cluster analysis. In: Proceedings of the IEEE International Conference on Evolutionary Computation. pp. 1507–1512.

Kennedy, J., Eberhart, R., 1942–1948. Particle swarm optimization. In: IEEE International Conference on Neural Networks, IEEE Press, vol. 4.

Kennedy, J., Eberhart, R.C., 2001. Swarm Intelligence. Morgan Kaufmann Publishers.

Kennedy, J., Mendes, R., 2002. Population structure and particle swarm performance. In: Proceedings of the 2002 Congress on Evolutionary Computation CEC2002, IEEE Press, pp. 1671–1676.

Løbjerg, M., Rasmussen, T.K., Krink, K., 2001. Hybrid particle swarm optimizer with breeding and subpopulations. In: Proceedings of the Third Genetic and Evolutionary Computation Conference (GECCO-2001), vol. 1. pp. 469–476.

Lumer, E., Faieta, B., 1994. Diversity and adaptation in population of clustering ants. In: Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animals, pp. 499–508.

Magurran, A.E., Higham, A., 1988. Information transfer across fish shoals under predator threat. Ethology 78, 153–158.

Parrish, J.K., Hamner, W.M., 1997. Animal Groups in Three Dimensions. Cambridge University Press, Cambridge, UK.

Pitcher, T.J., Parrish, J.K., 1993. Functions of shoaling behaviour in teleosts. In: Pitcher, T.J. (Ed.), Behaviour of Teleost Fishes. Chapman and Hall, London, pp. 363–439.

Schwefel, H.P., 1995. Evolution and Optimum Seeking. John Wiley and Sons, New York.

Shi, Y., Eberhart, R.C., 1997. A modified particle swarm optimiser. In: Proceedings of the IEEE International Conference on Evolutionary Computation. pp. 303–308.

Shi, Y., Eberhart, R.C., 1998. Parameter selection in particle swarm optimization. In: Evolutionary Programming VII (1998). Lecture Notes in Computer Science, vol. 1447. Springer, pp. 591–600.

Shi, Y., Eberhart, R.C., 2001. Fuzzy adaptive particle swarm optimization. In: Proceedings of the IEEE International Conference on Evolutionary Computation. pp. 101–106.

Törn, A., Zilinskas, A., 1989. Global optimisation. Lecture Notes in Computer Science, vol. 350. Springer-Verlag, Berlin.

Yao, X., Liu, Y., Liu, G., 1999. Evolutionary programming made faster. IEEE Trans. Evolut. Comput. 3 (2), 82–102.