Information Sciences 293 (2015) 370–382

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems

Yuhua Li [a,b], Zhi-Hui Zhan [b,c,*], Shujin Lin [d], Jun Zhang [c], Xiaonan Luo [a,b]

[a] National Engineering Research Center of Digital Life, Guangzhou 510006, China
[b] School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China
[c] Department of Computer Science, Sun Yat-sen University, Guangzhou 510006, China
[d] School of Communication and Design, Sun Yat-sen University, Guangzhou 510006, China

ARTICLE INFO

Article history:
Received 21 February 2014
Received in revised form 26 August 2014
Accepted 7 September 2014
Available online 28 September 2014

Keywords:
Particle swarm optimization (PSO)
Competition
Cooperation
Information sharing
Global optimization problems

ABSTRACT

This paper proposes an information sharing mechanism (ISM) to improve the performance of particle swarm optimization (PSO). The ISM allows each particle to share its best search information, so that all the other particles can take advantage of the shared information by communicating with it. In this way, the particles could enhance the mutual interaction with the others sufficiently and heighten their search ability greatly by using the search information of the whole swarm. Also, a competitive and cooperative (CC) operator is designed for a particle to utilize the shared information in a proper and efficient way. As the ISM share the search information among all the particles, it is an appropriate way to mix up information of the whole swarm for a better exploration of the landscape. Therefore, the competitive and cooperative PSO with ISM (CCPSO-ISM) is capable to prevent the premature convergence when solving global optimization problems. The satisfactory performance of CCPSO-ISM is evaluated by comparing it with other variants of PSOs on a set of 16 global optimization functions. Moreover, the effectiveness and efficiency of CCPSO-ISM is validated under different test environments such as biased initialization, coordinate rotated and high dimensionality.

1. Introduction

Inspired by the swarm behaviors of birds flocking and fish schooling, the particle swarm optimization (PSO) was first introduced by Kennedy and Eberhart in 1995 [16]. A particle in PSO uses the information of its historical best position and its neighborhood's best position to adjust its flying velocity to search for the global optimum in the solution space. However, the algorithm is not very efficient when solving complex problems because it is easy to be trapped into local optima [21,28].

The easiness of getting trapped into local optima is caused by that PSO does not sufficiently utilize its population's search information to guide the search direction. Therefore PSO has difficulty in solving complex problems [28]. The original PSO is

* Corresponding author. School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China.
E-mail address: zhanzhh@mail.sysu.edu.cn (Z.-H. Zhan).

http://dx.doi.org/10.1016/j.ins.2014.09.030
0020-0255/

a global version PSO (GPSO) where all the particles are attracted by the same globally best particle and the swarm has tendency to fast converge to the current globally best point. However, as GPSO only uses the search information of the globally best particle to guide the search direction, it may be premature convergence due to the lack of diversity [22]. Therefore, GPSO is not very efficient when solving complex multimodal functions because particles cannot efficiently utilize search information of the whole swarm to find out the global optimum.

How to cope with the "attraction" phenomenon of the globally best particle in PSO and make the particles have access to more information is a critical issue in improving the performance of PSO. Kennedy and Mendes [17] introduced a local version PSO (LPSO) to handle this drawback. Particle in LPSO is influenced by its historically best position and the local neighborhood's best position. As different particles have different neighborhoods, more local best particles are used to guide the search direction. Therefore, the information used in LPSO is richer than that in GPSO. LPSO is less prone to be trapped in local optima, but usually converges more slowly [17]. However, LPSO is still not very efficient in solving complex multimodal functions because each particle still uses the information only from two exemplars, the personal best position and its neighborhood's best position. For particles have access to more search information to guide the search direction, a "fully informed particle swarm" (FIPS) is proposed in [24]. In FIPS, all the particles in the neighborhood make contributions to guide the search direction. FIPS hence uses more information from the neighbors and leads to good performance.

The performance of GPSO, LPSO and FIPS with different degrees of information sharing has indicated that the more information is efficiently utilized to guide the flying, the better performance the PSO algorithm will have. Therefore, an information sharing mechanism (ISM) is proposed in this paper to let the particles share their best search information with all the other particles, and a competitive and cooperative (CC) operator is designed to use the shared information properly and efficiently. Thus, a competitive and cooperative PSO with ISM (CCPSO-ISM) is developed to enhance the PSO performance.

The ISM is inspired by the "blackboard" idea [13]. In the ISM, a "blackboard" is used as information pool where each particle can post information, or read information. In every iteration, the particles post their historically best information to the blackboard. Any particle can access and utilize the search information provided by other particles. This way, the degree of information share is much higher than GPSO, LPSO, or FIPS. The blackboard idea is similar to the "archive" idea which has been widely used in multiobjective optimization approaches [19,43,40] to store the found nondominated solutions, and is also similar to the harmony memory strategy in harmony search algorithm [12,35]. The archive strategy and harmony memory strategy have been proven to bring better performance to optimization approaches. Therefore we can expect good performance of the blackboard strategy because it makes the search information sufficiently shared. More efficiently, the additional memory required by the blackboard idea used in this paper is almost negligible because the historically best information of each particle is stored by the particle itself. The blackboard mechanism makes all the search information shared, therefore is helpful to mix up information of the whole swarm for a better exploration of the landscape. In order to use the shared information in a proper and efficient way to improve the PSO's performance, the CC operator is designed, which is loosely inspired by the corresponding competition and cooperation behaviors in human society [6]. CCPSO-ISM is shown to have good performance by testing on global optimization problems, especially on complex multimodal functions. Moreover, it is also promising on functions with biased initialization ranges, coordinate rotation and high dimensionality.

The reminder of this paper is organized as follows. In the next section, we give a brief review of traditional PSO together with its recent developments and the previous work related to the information sharing. In Section 3, the algorithm named CCPSO-ISM is developed based on the ISM and the CC operator. Section 4 presents experimental results, comparisons and discussions. Section 5 makes further investigation on the performance of CCPSO-ISM under different environments, followed by conclusions and future work in Section 6.

2. PSO and its developments

2.1. Framework of PSO algorithm

PSO uses a swarm of particles to represent the potential solutions of the optimization problem and lets the particles fly in the search space to search for the global optimum. Assume that the particles search in a *D*-dimension hyperspace, a particle *i* has a position vector $X_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$ which represents the current solution and a velocity vector $V_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]$ which is used to adjust the position. Moreover, each particle has a memory of a vector called the personal historically best $pBest_i$ to store the best position that the particle has found so far. The best $pBest_i$ in the particle *i*'s neighborhood is regarded as $nBest_i$ (for convenience, *gBest* is used in GPSO and $lBest_i$ is used in LPSO). The velocity and position of each particle *i* are first initialized randomly and will be updated by the influences of its own $pBest_i$ and the corresponding $nBest_i$ as

$$v_{id} = \omega v_{id} + c_1 r_{1d}(pBest_{id} - x_{id}) + c_2 r_{2d}(nBest_{id} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

In Eq. (1), the velocity is updated. The ω is the inertia weight introduced by Shi and Eberhart [33] in order to balance the abilities of global search and local search; c_1 and c_2 are the acceleration coefficients which indicate the influence of the particle's historically best position and its neighborhood's best position, respectively [8]; r_{1d} and r_{2d} are two randomly generated number in range [0, 1] with the uniform distribution, where the subscript d means that the random values for different dimensions are generated independently. Moreover, a position variant $V_{max,d}$ is used to clamp the maximal absolute value of updated velocity $|v_{id}|$.

In Eq. (2), the new position of the particle is obtained. Here, the new position may be out of the search region sometimes. If the new position is out of the search range, it will not be evaluated. This way, the particles which are out of range will have worse fitness and can be drawn back into the search region eventually.

After updating the velocity and position, the particles within the search range are evaluated. The $pBest_i$ of each particle is replaced by its current search position if and only if the fitness of current position is better than $pBest_i$. The $nBest_i$ is also replaced if a position which is better than the current $nBest_i$ has been found in the neighborhood of particle i.

Evolutionary process goes on and the updated velocity and position will be obtained again by the new $pBest_i$ and new $nBest_i$. The evolutionary process is iterative and will end when the stop conditions are met.

2.2. Developments of PSO

As the PSO has become more attractive and has been utilized in lots of real world applications [22,38,25,5,23,18,20,32], many researches have been working to improve the algorithm performance and various PSOs have been proposed.

Some researches focused on the parameter studies such as the inertia weight [33,34,41] and the acceleration coefficients [41,29]. Also, some researchers concentrated on combining PSO with other evolutionary operators and techniques to improve PSO's performance. Since Angeline proposed to apply selection operator into PSO [2], the combination of PSO with other optimization algorithms has become very attractive [7,1]. Inspired by biological mechanisms, niche technology [4] and speciation technology [26] are introduced into PSO to prevent particles to be too close to each other so that PSO can locate as many optimal solutions as possible. On the other hand, different topology structures, such as the star, ring, pyramid, and von Neumann structures [17] have been studied to improve the algorithms performance. PSO variants enhanced by orthogonal learning strategy [42], neighborhood search [36], centripetal strategy [3], multi-layer search strategy [37], self-adaptive strategy [9], and intermediate disturbance strategy [11] have attracted great attentions in recent years in PSO community.

2.3. Related work on information sharing

How to utilizing existing information to better adjust particles' velocities has become a promising and significant research topic in PSO. Besides the information sharing methods in GPSO, LPSO and FIPS, some other information sharing strategies are reviewed by Engelbrecht in [10]. Stereotyping method [14], fitness-distance-ratio method [27], barebones method [15], and general information sharing method [30] are summarized and their advantages and disadvantages are discussed. In the stereotyping PSO, the information comes from the cluster group [14], but the clustering adds computational complexity in the PSO and the best number of clusters is always problem-dependent [10]. In FIPS, the particle is fully informed by all the particles in the neighborhood [24], but the FIPS has the disadvantage that it may cancel the influence of each particle by the summing up of multiple influences [10]. In [27], the information of the fitness-distance-ratio PSO (FDR-PSO) provides a particle with maximal FDR to influence its flying velocity. However, the calculation of the additional particle on each dimension for each particle is time-consuming. In the information sharing strategy used in [30], the individuals with fitness which is better than the average fitness are collected in the best-performing list (BPL) and the individuals are attracted towards the nearest member in the BPL. However, the approach is developed for general swarms not specifically for PSO [10]. The comprehensive learning PSO (CLPSO) proposed in [21] can enhance the population diversity by letting each dimension learn from different particles. The CLPSO performs well on multimodal functions, but is less promising in unimodal functions.

3. CCPSO-ISM

3.1. ISM

The ISM idea is simple and easy to be implemented. Similar to [13], the ISM is implemented by using a sharing device, called "blackboard", where an individual particle posts information and reads information. To keep the algorithm simple, the blackboard has limited capacity. That is, the older, worse information will be overwritten by the newer, better one. In each iteration, all the particles post their current personal historically best information $pBest_i$ to the blackboard. Note that the blackboard keeps only the newest best information of each particle, and hence the information post in last iteration is overwritten by the information in this iteration. This way, the particles in the swarm can communicate with each other by posting and reading the shared information. Also, the particles can use this shared information for better search when necessary. The following subsection will design the CC operator to properly and efficiently use the shared information to enhance the search ability.

3.2. CC operator

The CC operator is based on the ISM that collects the sharing information. In order to simplify the implementation, a vector named *ccBest* (competition and cooperation best) is introduced into each particle. This vector is used to store the shared information the particle gets from the "blackboard". In each iteration, when updating the velocity, the particle i is influenced by its corresponding $ccBest_i$ as shown in Eq. (3) below instead of $pBest_i$ and $nBest_i$ used in Eq. (1). In Eq. (3), ω is the inertia weight as in Eq. (1), the c is the acceleration coefficient set as 2.0 and the r_d is a random value in the range [0,1].

$$v_{id} = \omega v_{id} + cr_d(ccBest_{id} - x_{id}) \quad (3)$$

The construction of $ccBest$ is described as follows. The vector $ccBest_i$ is initialized as $pBest_i$ at the beginning and is used to guide the flying velocity as in Eq. (3). However, when the particle i has been trapped for a specified number of iterations G, this means the current guidance information in $ccBest_i$ is no longer effective to lead the particle to a better search region. Thus the trapped particle will call for cooperation from other particles. But who will be the correct cooperators and how to manipulate the cooperative operation become important issues. Inspired by the phenomenon in the human society that people always have different cooperators in different aspects, we allow the particle to have different cooperators in different dimensions. In details, for each dimension d of the trapped particle i, the particle will communicate with all the particles through the "blackboard" and randomly select K particles (possibly including itself) to compete. The winner with the best $pBest$ fitness (particle k, for example) will become the cooperator.

The competition pressure can be controlled by K, which can also be regarded as the neighborhood size of particle i. In this paper, a novel strategy that uses a dynamic neighborhood size can be designed in order to balance the exploration and exploitation abilities. The K is obtained as

$$K = \left\lceil \frac{t}{T} \times SIZE \right\rceil \quad (4)$$

where t, T and $SIZE$ are the current iteration, maximal iteration and population size respectively. Eq. (4) shows that K increases from 1 to the whole swarm. Even though there may be other strategies to control the K value or just set it as constant, we think that a linearly increase K value is somehow consistent with the claim in [17] that smaller neighborhood is better for exploration in the early phase whilst larger one is propitious to convergence in the late phase.

After the determination of the cooperator, CCPSO-ISM uses a cooperation probability P to control the cooperation behavior, and only when a random value generated uniformly in range [0,1] is lower than P, the cooperation behavior takes place successfully. Once the cooperation behavior happens, the value of $ccBest_{id}$ will be replaced by $pBest_{kd}$ of the corresponding cooperator. Otherwise, $ccBest_{id}$ will be replaced by $pBest_{id}$ of the particle itself.

As $ccBest$ is the result of competition and cooperation among the whole swarm, it represents for the search information not only the particle itself, the neighborhoods' best particle, the globally best particle, but also any other particles. Hence, $ccBest$ on the one hand keeps the concept of original PSO, and on the other hand can enhance diversity to weaken the "attraction" phenomenon of the globally best particle because all the information of the swarm is used. It should be noted that the personal historically best $pBest_i$ is still stored by each particle because this information is to be posted to the "blackboard" in each iteration. Additionally, the globally best position which is important to indicate the global convergence will not be lost during the evolutionary process.

3.3. Parameter investigations of CCPSO-ISM

CCPSO-ISM adds two new parameters into original PSO in its CC operator. The two new parameters are the stagnated iterations G and the cooperation probability P. We will investigate these two parameters through experiments based on Sphere, Rosenbrock, Schewefel, Rastrigin, Ackley and Griewank functions. All these functions are given in Table 1, where the search range, acceptable error value, and global optimum are also presented. We run 30 independent trials on each function when investigate different parameter configurations. The population size is 20 and the maximal iteration is 5000. The mean results are recorded for comparison.

Firstly, the parameter G is investigated. The value of G may affect the performance of CCPSO-ISM. A smaller value for G may destroy the normal stagnation phenomenon of the algorithm whilst a larger value of G may result in the waste of computation on the local optima. In order to obtain an insightful view of how the stagnated iteration G affects the performance of CCPSO-ISM, this paper tests different G from 0 to 10. As we have not investigated the value of parameter P, we adopt 0.05 for P. The mean fitness values of 30 runs when CCPSO-ISM using different G are plotted in Fig. 1(a). It should be noted that the global optimum of the Schewefel function is -12596.5, and therefore the mean fitness values of this function are often smaller than $-10{,}000$ even with different G, while all the other functions are with 0 as the global optimum. Therefore the vertical axis of the figure is broken into two segments to make the plot possible. The figure shows that it is better to use a smaller G for Sphere function which is a simple unimodal function mainly owing to that it is better to change the guidance as soon as possible to enhance the hill-climbing. However, such a property is not evident for the difficult unimodal function (Rosenbrock function) and the multimodal functions. The plots in the figure indicate that a value of 5 for G is better when all the six functions are considered. Therefore, we recommend setting $G = 5$ to balance the performance of CCPSO-ISM on both unimodal and multimodal functions.

Secondly, we investigate the parameter P. P is used to control the cooperation behavior. That is, it is the probability that impacts whether the particle learns from cooperator's historically best. By considering the impact of P, we can imagine that parameter P has something common with parameter G. A smaller G may affect the results somehow like a larger P because

Table 1
The 16 test functions for comparison, the first 4 functions are unimodal functions, and followed by 6 multimodal functions with many local optima, the last 6 are multimodal functions with a few local optima, more details of these functions referred to [39].

Test functions	D	Search range	f_{min}	Accept	Name
$f_1(x) = \sum_{i=1}^{D} x_i^2$	30	$[-100, 100]^D$	0	0.01	Sphere
$f_2(x) = \sum_{i=1}^{D-1}\left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	30	$[-10, 10]^D$	0	100	Rosenbrock
$f_3(x) = \sum_{i=1}^{D}(\lfloor x_i + 0.5 \rfloor)^2$	30	$[-100,100]^D$	0	0	Step
$f_4(x) = \sum_{i=1}^{D} ix^4 + random[0,1)$	30	$[-1.28,1.28]^D$	0	0.01	Quadric noise
$f_5(x) = \sum_{i=1}^{D} -x_i \sin\left(\sqrt{x_i}\right)$	30	$[-500, 500]^D$	−12596.5	−10,000	Schwefel
$f_6(x) = \sum_{i=1}^{D}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	30	$[-5.12, 5.12]^D$	0	50	Rastrigrin
$f_7(x) = -20\exp\left(-0.2\sqrt{1/D\sum_{i=1}^{D} x_i^2}\right) - \exp\left(1/D\sum_{i=1}^{D}\cos 2\pi x_i\right) + 20 + e$	30	$[-32, 32]^D$	0	0.01	Ackley
$f_8(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^D$	0	0.01	Griewank
$f_9(x) = \frac{\pi}{D}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$	30	$[-50, 50]^D$	0	0.01	Generalized penalized 1
$f_{10}(x) = \frac{1}{10}\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2[1 + \sin^2(2\pi x_D)]\right\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	30	$[-50, 50]^D$	0	0.01	Generalized penalized 2
$f_{11}(x) = 0.5 + \frac{\left(\sin\sqrt{x_1^2 + x_2^2}\right)^2 - 0.5}{\left(1.0 + 0.001\left(x_1^2 + x_2^2\right)\right)^2}$	2	$[-100, 100]^D$	0	0	Schaffers f6
$f_{12}(x) = \left[\frac{1}{500} + \sum_{j=1}^{25}\left(j + \sum_{i=1}^{2}(x_i - a_{ij})^6\right)^{-1}\right]^{-1}$	2	$[-65.536, 65.536]^D$	0.988004	0.988004	Shekels foxholes
$f_{13}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1\left(b_i{}^2 + b_i x_2\right)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	$[-5, 5]^D$	0.0003075	0.0005	Kowalik
$f_{14}(x) = -\sum_{i=1}^{5}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]^D$	−10.1532	−10	Shekels family 1
$f_{15}(x) = -\sum_{i=1}^{7}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]^D$	−10.4029	−10	Shekels family 2
$f_{16}(x) = -\sum_{i=1}^{10}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]^D$	−10.5364	−10	Shekels family 3

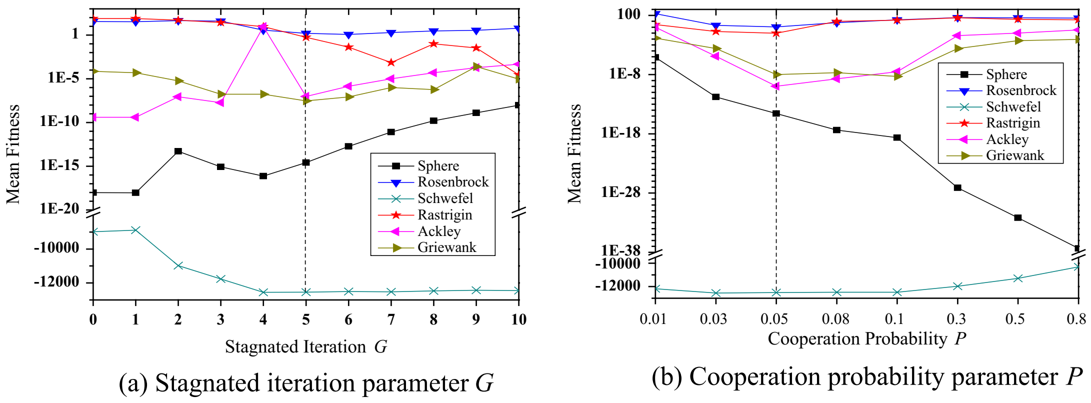

Fig. 1. Parameter investigation results in CCPSO-ISM.

they both make the cooperation behavior among the swarm more frequent and easier. The results plotted in Fig. 1(b) support this hypothesis in a certain sense. For example, while optimizing the Sphere function, a smaller G or a larger P can enhance search performance. Fig. 1(b) shows that for a general configuration, neither too small P nor too large P is better, but the value of 0.05 for P is good for most of the test functions. Hence, we will use 0.05 as the cooperation probability in CCPSO-ISM because it can balance the performance of CCPSO-ISM on different kinds of functions.

Table 2
Results comparison of variant PSOs (Boldface means the best value among all the PSOs).

Functions		GPSO	LPSO	FDR-PSO	FIPS	CLPSO	CCPSO-ISM
f_1	Mean	3.03×10^{-52}	1.43×10^{-28}	**7.14×10^{-117}**	**3.97×10^{-30}**	3.37×10^{-19}	6.61×10^{-35}
	Std. dev	9.55×10^{-52}	6.79×10^{-28}	**5.05×10^{-116}**	6.82×10^{-30}	3.80×10^{-19}	6.56×10^{-35}
f_2	Mean	22.28	22.95	1.10	22.52	11.46	0.07
	Std. dev	18.54	14.37	1.72	0.46	11.44	0.19
f_3	Mean	**0**	**0**	**0**	**0**	**0**	**0**
	Std. dev	**0**	**0**	**0**	**0**	**0**	**0**
f_4	Mean	7.14×10^{-3}	1.46×10^{-2}	4.11×10^{-3}	**2.38×10^{-3}**	4.03×10^{-3}	6.71×10^{-3}
	Std. dev	2.50×10^{-3}	4.08×10^{-3}	2.20×10^{-3}	**5.79×10^{-4}**	1.19×10^{-3}	1.71×10^{-3}
f_5	Mean	−9925.01	−9518.86	−9071.14	−10050.4	**−12552.91**	−12538.69
	Std. dev	536.07	432.51	489.17	823.30	**41.54**	62.46
f_6	Mean	27.02	35.72	31.72	28.43	1.23×10^{-11}	**0**
	Std. dev	7.55	8.61	9.62	9.69	2.29×10^{-11}	**0**
f_7	Mean	1.44×10^{-14}	1.91×10^{-14}	1.51×10^{-14}	**1.25×10^{-14}**	1.92×10^{-12}	1.40×10^{-14}
	Std. dev	2.88×10^{-14}	5.96×10^{-15}	8.37×10^{-15}	3.21×10^{-14}	1.17×10^{-12}	**1.65×10^{-15}**
f_8	Mean	2.62×10^{-2}	1.27×10^{-2}	1.35×10^{-2}	1.97×10^{-4}	9.29×10^{-13}	**6.84×10^{-14}**
	Std. dev	2.48×10^{-2}	1.64×10^{-2}	1.62×10^{-2}	1.39×10^{-3}	3.81×10^{-12}	**1.69×10^{-13}**
f_9	Mean	1.66×10^{-2}	1.59×10^{-30}	8.29×10^{-3}	1.07×10^{-31}	1.57×10^{-21}	**1.57×10^{-32}**
	Std. dev	4.37×10^{-2}	4.22×10^{-30}	2.84×10^{-2}	3.98×10^{-32}	2.18×10^{-21}	**1.11×10^{-47}**
f_{10}	Mean	2.2×10^{-3}	5.36×10^{-28}	2.86×10^{-3}	1.35×10^{-30}	1.37×10^{-20}	**1.35×10^{-32}**
	Std. dev	4.44×10^{-3}	1.66×10^{-27}	4.87×10^{-3}	4.55×10^{-31}	1.67×10^{-20}	**1.38×10^{-47}**
f_{11}	Mean	**0**	1.94×10^{-4}	**0**	1.94×10^{-4}	3.12×10^{-9}	**0**
	Std. dev	**0**	1.37×10^{-3}	**0**	1.37×10^{-3}	1.55×10^{-8}	**0**
f_{12}	Mean	**0.998**	**0.998**	**0.998**	1.215	**0.998**	**0.998**
	Std. dev	**1.12×10^{-15}**	**1.12×10^{-15}**	**1.12×10^{-15}**	0.982	**1.12×10^{-15}**	**1.12×10^{-15}**
f_{13}	Mean	1.68×10^{-3}	**3.07×10^{-4}**	1.64×10^{-3}	3.28×10^{-4}	4.48×10^{-4}	4.38×10^{-4}
	Std. dev	4.78×10^{-3}	**1.59×10^{-17}**	4.79×10^{-3}	1.44×10^{-4}	1.14×10^{-4}	1.14×10^{-4}
f_{14}	Mean	−6.7503	−9.8017	−5.5036	−9.2041	**−10.1532**	**−10.1532**
	Std. dev	3.52515	1.43344	3.45745	2.41188	**1.26×10^{-14}**	**1.26×10^{-14}**
f_{15}	Mean	−8.0983	**−10.4029**	−7.2395	−10.2499	**−10.4029**	**−10.4029**
	Std. dev	3.4340	8.87×10^{-15}	3.7696	1.08	**7.18×10^{-15}**	**7.18×10^{-15}**
f_{16}	Mean	−7.7852	**−10.5364**	−7.5083	**−10.5364**	−10.4292	**−10.5364**
	Std. dev	3.7395	**8.97×10^{-15}**	3.7943	1.07×10^{-14}	0.7581	**8.97×10^{-15}**

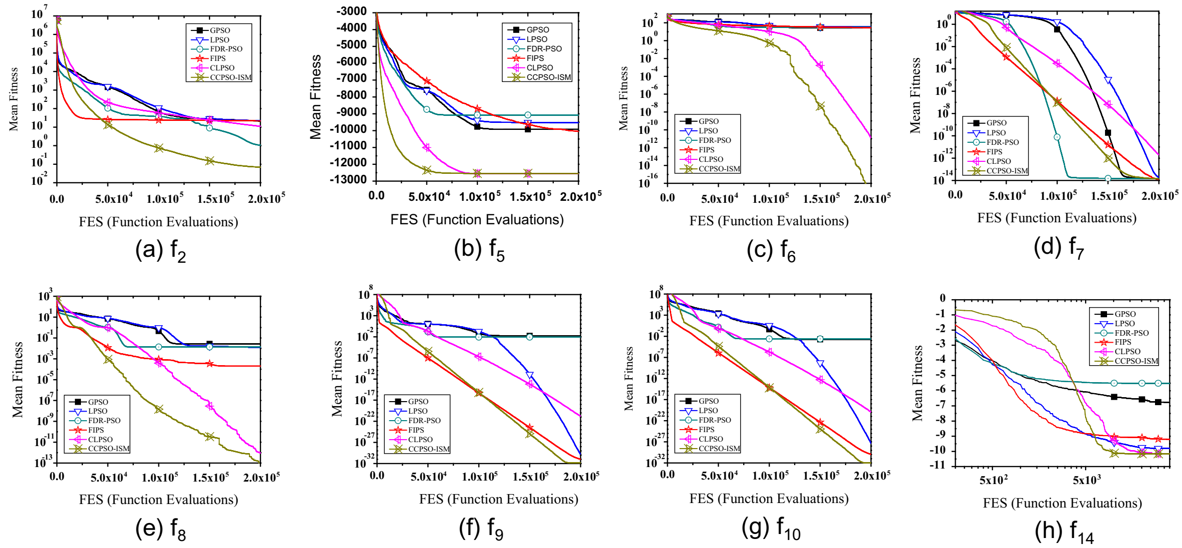

Fig. 2. The mean evolutionary curves of different PSO algorithms on some selected test functions. (a) f_2, (b) f_5, (c) f_6, (d) f_7, (e) f_8, (f) f_9, (g) f_{10}, (h) f_{14}.

4. Experimental studies

4.1. Benchmark functions and PSOs configurations

In order to show the advantages of CCPSO-ISM, we choose 16 benchmark functions [39] to test the performance of CCPSO-ISM and compare the results obtained by CCPSO-ISM and other PSOs. These 16 test functions listed in Table 1 include 4 unimodal functions, 6 complex multimodal functions with many local optima and 6 multimodal functions with only a few local optima. The details of these benchmark functions are given in Table 1 and more information can be referred to [39].

We choose five PSOs that use different methods to utilize the search information to compare with CCPSO-ISM. The five PSO algorithms are GPSO [33], LPSO [17], FDR-PSO [27], FIPS [24] and CLPSO [21], which are paradigms that use the search information in different fashions. In order to make a fair comparison among the algorithms, the parameter configurations of different PSOs are set according to that proposed by their authors respectively. For CCPSO-ISM, the inertia weight $\omega = 0.6$, acceleration coefficient $c = 2.0$, stagnated iteration $G = 5$, and cooperation probability $P = 0.05$. Moreover, all the algorithms use the same population size of 20. All the PSOs use the same maximal number of 2×10^5 function evaluations (FEs) for each test function. For the purpose of reducing the influences of stochastic error, we run 50 independent trials on each test function and record the mean results for comparison.

4.2. Result comparisons on solution accuracy

The experimental results in Table 2 contain the mean fitness and the standard deviation of the 50 trials on each function with different PSO algorithms. The results show that CCPSO-ISM has a stronger ability to reach the global optimum on most of the tested functions. CCPSO-ISM obtains the best solutions on the functions f_2, f_3, f_6, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{14}, f_{15} and f_{16}, 11 out of the 16 functions.

Fig. 2 give the evolutionary curves of each algorithm in the same figure to show the convergence characteristics of different PSOs on some selected functions. The curves can also suggest the convergence speed and solution quality. As the functions are minimization problems, the left the curve is closer to, the faster convergence speed it indicates, and the lower point it has when the algorithm terminates, the better solution it obtains at last. Moreover, we can see from the curves whether the algorithm is trapped into the local optima while the iteration goes on. An algorithm which is easy to be premature or be trapped into local optima will keep a relative aclinic line on the later iterations (function evaluations) whilst an algorithm has ability to jump out of local optima to refine the results can keep a gradient line until it finds the global optimum. The figures show that CCPSO-ISM is not likely to be trapped into local optima but can reach the global optimum gradually during the process, especially on the complex multimodal functions f_5 to f_{10}.

4.3. Result comparisons on convergence speed

The experimental results indicate that CCPSO-ISM can get better solutions than other PSO algorithms, especially on complex multimodal problems. However, it is also interesting to compare the convergence speed of algorithms. The data in

Table 3
Convergence speed comparison among different PSO algorithms (Boldface means the best value).

Functions		GPSO	LPSO	FDR-PSO	FIPS	CLPSO	CCPSO-ISM
f_1	Speed (FEs)	106743	117,730	63,549	32,485	71,297	**35,020**
	Time (s)	0.96	1.12	4.74	0.35	0.66	**0.27**
	Ratio (%)	**100**	**100**	**100**	**100**	**100**	**100**
f_2	Speed (FEs)	94,216	102,557	51,580	**12,922**	76,852	32,276
	Time (s)	0.92	1.05	3.88	**0.15**	0.75	0.27
	Ratio (%)	**100**	**100**	**100**	**100**	**100**	**100**
f_3	Speed (FEs)	93,625	122,506	51,413	**19,562**	39,384	20,229
	Time (s)	1.17	1.6	4.01	0.27	0.51	**0.23**
	Ratio (%)	**100**	**100**	**100**	**100**	**100**	**100**
f_4	Speed (FEs)	166,203	188,299	80,523	**46,967**	100,858	133,985
	Time (s)	1.57	1.93	6.02	**0.52**	0.98	1.12
	Ratio (%)	88	16	98	**100**	**100**	98
f_5	Speed (FEs)	91,011	96,602	51,789	121,938	38,802	**14,350**
	Time (s)	2.32	2.06	6.53	2.22	0.95	**0.21**
	Ratio (%)	52	14	6	64	**100**	**100**
f_6	Speed (FEs)	93,690	99,473	47,758	89,219	54,704	**17,819**
	Time (s)	1.22	1.4	3.76	1.29	0.77	**0.22**
	Ratio (%)	100	96	96	**100**	**100**	**100**
f_7	Speed (FEs)	110,705	125,543	66,489	**38,434**	75,584	48,964
	Time (s)	1.39	1.82	5.18	0.59	0.98	0.56
	Ratio (%)	**100**	**100**	**100**	**100**	**100**	**100**
f_8	Speed (FEs)	111,043	132,489	65,612	45,875	80,784	**40,667**
	Time (s)	1.45	1.84	5.15	0.69	1.08	**0.49**
	Ratio (%)	38	58	54	**100**	**100**	**100**
f_9	Speed (FEs)	98,918	106,550	54,106	**19,689**	58,403	32,032
	Time (s)	2.14	2.6	4.69	**0.47**	1.33	0.66
	Ratio (%)	86	**100**	92	**100**	**100**	**100**
f_{10}	Speed (FEs)	110,014	122,578	61,840	**26,605**	68,030	38,539
	Time (s)	2.57	2.7	5.46	**0.57**	1.58	0.82
	Ratio (%)	80	**100**	74	**100**	**100**	**100**
f_{11}	Speed (FEs)	50,005	58,915	33,250	**13,708**	66,088	42,352
	Time (s)	0.65	0.69	3.2	**0.13**	0.83	0.31
	Ratio (%)	**100**	98	**100**	98	96	**100**
f_{12}	Speed (FEs)	6199	7362	4424	**3729**	6618	6456
	Time (s)	0.17	0.21	0.49	**0.09**	0.19	0.13
	Ratio (%)	**100**	**100**	**100**	94	**100**	**100**
f_{13}	Speed (FEs)	43,726	55,044	25,834	**18,919**	49,775	58,976
	Time (s)	0.7	0.76	2.88	**0.22**	0.67	0.52
	Ratio (%)	76	**100**	82	98	68	68
f_{14}	Speed (FEs)	14,479	14,151	**3562**	7020	12,262	7077
	Time (s)	0.24	0.22	0.39	0.08	0.17	**0.06**
	Ratio (%)	50	94	34	86	**100**	**100**
f_{15}	Speed (FEs)	10,954	9459	3180	**2012**	10,179	7202
	Time (s)	0.18	0.15	0.32	**0.02**	0.14	0.06
	Ratio (%)	68	**100**	58	98	**100**	**100**
f_{16}	Speed (FEs)	8889	7810	3118	**2369**	10,879	7272
	Time (s)	0.15	0.14	0.33	**0.03**	0.16	0.06
	Ratio (%)	64	**100**	60	**100**	98	**100**
Mean reliability		81.375%	86%	78.375%	96.125%	97.625%	97.875%

Table 3 shows the convergence speed of each PSO by giving the FEs and running time to reach the Accept solution (acceptable accuracy for each test function that presented in Table 1). Additionally, successful ratio is given in Table 3. We can observe from Table 3 that CCPSO-ISM reaches the Accept solutions fastest on functions f_1, f_5, f_6 and f_8 when measured on the FEs. When measured on the running time, CCPSO-ISM is the fastest on functions f_1, f_3, f_5, f_6, f_7, f_8 and f_{14} (all the experiments are carried out on the same machine with a Celeron 2.26 GHz CPU, 256 MB memory and the Windows XP2 operating system). Nevertheless, the most attractive advantage of CCPSO-ISM is that it can obtain the *Accept* solutions with much higher successful ratio. It reaches the ratio of 100% on 14 out of the 16 tested functions. The highest mean reliability with 97.875% makes CCPSO-ISM the most reliable algorithm.

4.4. Result discussions

(1) Unimodal functions

While optimizing unimodal functions, for example, from f_1 to f_4, GPSO always does better than LPSO. This is because that GPSO uses the globally best particle to guide all the particles to fast converge convergence on unimodal functions. FDR-PSO adds a nearest-better particle as an exemplar to guide the flying. Moreover, the c_1 and c_2 used in FDR-PSO are both 1.0 instead of 2.0. Both the best "fitness-distance-ratio" particle and the smaller acceleration coefficients may help to refine the solution accuracy. Therefore, FDR-PSO has better performance on unimodal functions. On the contrary, CLPSO is not much promising for solving unimodal functions. The conclusions in [21] and the experimental results in this paper both are in agree with this fact. The result that CCPSO-ISM is not better than GPSO and FDR-PSO for the simple unimodal function may be caused by weakening the influence of the globally best particle. However, CCPSO-ISM can still achieve higher accurate solution than LPSO, FIPS and CLPSO on the Sphere function. The advantages of CCPSO-ISM become more evident when more complex functions are tested. Like the Rosenbrock function (f_2), although it is a unimodal function, it is very difficult to find the global optimum for that the minimal point is on the long narrow valley [21]. We can observe that only CCPSO-ISM can find the globally best optimum for the Rosenbrock function, therefore, CCPSO-ISM outperforms all the other PSOs on this function.

(2) Multimodal functions with many local optima

The capabilities of avoiding local optima and reaching the global optimum on multimodal functions are very important for global optimization algorithms. Traditional PSO has disadvantages in solving multimodal functions for that the algorithm is easy to be trapped by local optima. The results presented in Table 2 show that GPSO and FDR-PSO do much worse than LPSO, FIPS, CLPSO and CCPSO-ISM mainly due to that the later ones use more information from the swarm to maintain the diversity when solving multimodal functions. Especially, CCPSO-ISM is very promising on multimodal functions. Compared with all the other PSOs, CCPSO-ISM can reach the global optimum on all the complex multimodal function from f_5 to f_{10} and outperforms all the other PSOs on functions f_6, f_8, f_9 and f_{10}. More importantly, only CCPSO-ISM can reach the optimum value 0 on the Rastrigrin function (f_6).

Multimodal functions are difficult to solve because the global optimum can be very far from the local optima (like the Schwefel function f_5), or can be surrounded by a considerable amount of local optima (like the Rastrigin function f_6), or the function has linkages between variables (like the Griewank function f_8). Hence, any algorithm that wants to find the global optimum need to have the ability of jumping out of local optima by using all the information of the swarm to maintain the diversity. Traditional PSO is easy to be attracted by the only globally best particle. Therefore it is difficult to jump out of the current globally best region which is more likely to be a local optimum. CCPSO-ISM lets the particle cooperate with any other particle on different dimension. The guidance vector *ccBest* in CCPSO-ISM is the competition and cooperation results of the whole swarm not that of one particle (for example, the globally best particle). Such a mechanism can avoid local optima for that the whole information in the swarm is used. Hence the ISM can mix up information of the whole swarm for a better exploration of the landscape. Also, the figures in Fig. 2 support the claim that CCPSO-ISM has the ability to jump out of the local optima and search for better region on complex multimodal problems. The curves of CCPSO-ISM indicate that the algorithm is able to improve the solutions steadily for a long time on the multimodal functions whilst many other PSOs appear to fall into poor local optima quite early and cannot jump out during the running.

(3) Multimodal functions with only a few local optima

On solving functions f_{11} to f_{16}, we can also observe the advantages of CCPSO-ISM as shown in Table 2. Functions f_{11} to f_{16} are simpler although they are still multimodal functions because the dimension is lower and the local optima are fewer. We can see from the experimental results that most of the algorithms can found the global optimum for these problems. On function f_{11}, all the PSOs can obtain the global optimum except that LPSO, FIPS and CLPSO are trapped. FIPS is also trapped in f_{12} whilst other algorithms are not. It is interesting to observe that different PSOs can get different results for f_{13}. Functions f_{14} to f_{16} are the Shekel's family functions, and CCPSO-ISM performs much better than other algorithms. CCPSO-ISM obtains the global optima on all these three functions, and only CLPSO can have a comparable performance with CCPSO-ISM on these functions. The experimental results show that CCPSO-ISM is a much promising algorithm in solving this kind of problems.

Table 4
Biased initialization ranges for test functions.

Functions	Symmetric initialization	Biased initialization	Global optimization
f_1	$[-100, 100]^D$	$[50, 100]^D$	$[0, 0, \cdots, 0]$
f_2	$[-10, 10]^D$	$[5, 10]^D$	$[1, 1, \cdots, 1]$
f_5	$[-500, 500]^D$	$[-500, -250]^D$	$[420.96, 420.96, \cdots, 420.96]$
f_6	$[-5.12, 5.12]^D$	$[2.56, 5.12]^D$	$[0, 0, \cdots, 0]$
f_8	$[-600, 600]^D$	$[300, 600]^D$	$[0, 0, \cdots, 0]$
f_{12}	$[-65.536, 65.536]^D$	$[32.768, 65.536]^D$	[-32, -32]

Table 5
Results comparison on biased initialized functions (Boldface means the best value).

Functions	PSO		CCPSO-ISM	
	Mean	Std. dev	Mean	Std. dev
f_1	$\mathbf{3.76 \times 10^{-53}}$	$\mathbf{1.12 \times 10^{-52}}$	8.20×10^{-36}	5.80×10^{-35}
f_2	28.801	25.8169	**8.63897**	**8.53605**
f_5	−8843.61	241.272	**− 10371.3**	**59.3881**
f_6	30.8437	8.6715	$\mathbf{2.49 \times 10^{-16}}$	$\mathbf{1.76 \times 10^{-15}}$
f_8	2.35×10^{-2}	2.60×10^{-2}	$\mathbf{6.89 \times 10^{-14}}$	$\mathbf{2.90 \times 10^{-13}}$
f_{12}	**0.998004**	$\mathbf{1.59 \times 10^{-16}}$	**0.998004**	1.7×10^{-16}

Table 6
Results comparison on rotated multimodal functions (Boldface means the best value).

Functions	PSO		CCPSO-ISM	
	Mean	Std. dev	Mean	Std. dev
f_5	−7524.2	664.389	**−8555.20**	**384.30**
f_6	61.6077	16.4053	**43.35**	**13.71**
f_7	1.92294	0.657,078	$\mathbf{3.67 \times 10^{-4}}$	$\mathbf{2.15 \times 10^{-3}}$
f_8	1.39×10^{-2}	1.55×10^{-2}	$\mathbf{6.43 \times 10^{-6}}$	$\mathbf{2.48 \times 10^{-5}}$

5. Further performance tests on CCPSO-ISM

This section further tests CCPSO-ISM under different conditions. The asymmetric initialization ranges for each function which do not contain the globally best point will be used; the rotated multimodal functions are tested; and higher dimensional functions are also tested, for the purpose of testing the performance of CCPSO-ISM under different environments. We use GPSO as the traditional PSO (PSO for short) to compare with CCPSO-ISM here in the following simulations.

5.1. Biased initializations of test functions

We use the schema proposed by Angeline [2], in which the particles are initialized just to a portion of the space that does not contain the global optimal point. Six functions are used for test, including the unimodal function f_1 and f_2, and the multimodal function f_5, f_6, f_8 and f_{12}. The biased initialization ranges are shown in Table 4. Also, the test of each function is run for 50 independent trials and each trail runs for 2×10^5 FEs.

Table 5 gives the results. The results indicate that both PSO and CCPSO-ISM perform well with or without biased initializations. These are in reasonably good agreement with the conclusions of Angeline that PSO are only slightly affected by the initialization schemas [2]. Therefore, CCPSO-ISM can obtain promising solutions even though it is initialized without containing the global optimal region.

5.2. Rotated multimodal test functions

The use of rotated functions to test the performance of algorithms is also emphasized by many researchers [21,31]. Traditional test functions are often criticized for the lack of linkages among variables and they can be optimized by algorithms dimension by dimension. The most popular method to rotate a function is to left multiply an orthogonal matrix. More details of the rotation method can be referred to [21,31].

We rotated the multimodal functions of f_5, f_6, f_7 and f_8 in this way and reevaluated them in both PSO and CCPSO-ISM. Results are obtained by 50 independent trials (2×10^5 FEs for each trial) and are given in Table 6.

Results in Table 6 show that both PSO and CCPSO-ISM are affected by the rotation of test functions. They have difficulty in finding the global optimum of function f_5 and f_6 after rotation. However, CCPSO-ISM can still perform better than PSO on these two rotated functions. Moreover, CCPSO-ISM can still obtain the global optimum on function f_7 and f_8 even though they are rotated, whilst traditional PSO is trapped.

5.3. Different dimensions of test functions

We also test the performance of CCPSO-ISM on functions with different dimensions. As we can expect, when the dimension of the problem increases, it becomes harder and harder to find the global optimum, especially for the multimodal functions whose number of local optima increase exponentially as the dimension increases. CCPSO-ISM has adequate information among the swarm by using the ISM and CC operator. Therefore CCPSO-ISM is expected to reach the global optimum much

Table 7
Results comparison between PSO and CCPSO-ISM on f_1 with different dimensions.

Dimensions		3	5	10	15	20	30	50	100
MAX. FEs		2×10^4	2×10^4	5×10^4	1×10^5	5×10^5	2×10^5	5×10^5	1×10^6
PSO	Mean	6.04×10^{-60}	7.64×10^{-43}	3.29×10^{-55}	2.12×10^{-69}	3.01×10^{-47}	1.95×10^{-52}	5.59×10^{-60}	8.59×10^{-36}
	FEs	4633	7362	21,414	44,921	50,927	106,100	276,023	624,608
CCPSO-ISM	Mean	1.10×10^{-19}	6.23×10^{-12}	1.87×10^{-18}	4.73×10^{-29}	8.93×10^{-23}	6.99×10^{-35}	3.32×10^{-61}	1.71×10^{-71}
	FEs	3984	7112	13,762	19,825	24,880	35,105	53,488	95,090
Speed ratio		1.16	1.04	1.56	2.27	2.05	3.02	5.16	6.57
Solutions ratio		5.47×10^{-41}	1.23×10^{-31}	1.75×10^{-37}	4.49×10^{-41}	3.37×10^{-25}	2.79×10^{-18}	16.8	5.03×10^{35}

Table 8
Results comparison between PSO and CCPSO-ISM on f_6 with different dimensions.

Dimensions		3	5	10	15	20	30	50	100
MAX. FEs		2×10^4	2×10^4	5×10^4	1×10^5	5×10^5	2×10^5	5×10^5	1×10^6
PSO	Mean	0	0.437804	3.50227	7.68766	15.004	30.7044	51.4393	167.592
	FEs	39	39	1546	19,601	34,138	93,825	569,699	–
CCPSO-ISM	Mean	2.09×10^{-13}	3.77×10^{-7}	4.74×10^{-11}	0	9.48×10^{-11}	0	3.98×10^{-2}	53.9886
	FEs	37	72	2172	4835	8096	18,298	89,446	1,772,955
Speed ratio		1.05	0.54	0.71	4.05	4.22	5.13	6.37	–

easier than traditional PSO. Experimental results in Section 4 have shown the advantages of CCPSO-ISM on most of the functions with 30 dimensions. In order to find out the impact of dimension, we set different dimension number on the unimodal function f_1 and the multimodal function f_6 and carry out experiments. The dimension increases from 3 to 100, including 3, 5, 10, 15, 20, 30, 50 and 100. For the complex level of different dimensions, we set different maximal FEs for different dimensions. The results of f_1 and f_6 are given in Tables 7 and 8 respectively. Here, we still use 0.01 as the acceptable solution for f_1 and 50 for f_6. The final solutions and speed (average FEs number to obtain the acceptable solution) are average values of 50 independent trials.

It can be seen from the results that the advantages of CCPSO-ISM become more and more evident as the dimension increases according to the Row "Speed Ratio" and Row "Solution Ratio" in Table 7, and the Row "Speed Ratio" in Table 8. The value of "Solution Ratio" is the quotient of the solution obtained by PSO and the solution obtained by CCPSO-ISM. As the functions are minimization problems, the larger the value is, the better result CCPSO-ISM obtains than PSO. The value of "Speed Ratio" is the quotient of the FEs needed by PSO and the one needed by CCPSO-ISM to obtain an acceptable solution. Therefore, the larger the value is, the faster CCPSO-ISM outperforms PSO. For example, in Table 7 for f_1, when the dimension is 30, solution ratio is 2.79×10^{-18} as the result of $(1.95\times10^{-52})/(6.99\times10^{-35})$ and speed ratio is 3.02 as the result of 106,100/35,105. Due to the existence of solution 0 to f_6, we do not provide the solution ratio in Table 8. The Speed Ratio value of the 100 dimensional f_6 is not given in Table 8 because PSO cannot reach the Accept solution within the maximal FEs. The experimental results show that CCPSO-ISM is a promising algorithm not only on low but also high dimensional problems, not only on unimodal functions but also on multimodal functions. Hence, CCPSO-ISM is a promising algorithm to be used in a wider scope of real-world applications.

6. Conclusion

This paper has designed the ISM for PSO to make all the best information in the swarm accessible by all the particles. This is useful to cope with the lack of information sharing in traditional PSOs. In order to use the shared information more properly and efficiently, the CC operator was designed based on competition and cooperation behaviors in human society. The implementation of CCPSO-ISM was given in details and the advantages of CCPSO-ISM were shown by comparing it with some other PSO algorithms on 16 global numerical benchmark functions. The experimental results firstly show that even though CCPSO-ISM is not as good as GPSO or FDR-PSO in solving simple unimodal functions, it does much better than LPSO, FIPS and CLPSO. Secondly, CCPSO-ISM has strong global search ability by using all the information of the swarm to guide search direction, indicating that the ISM is an appropriate way to mix up information of the whole swarm for a better exploration of the landscape. Lastly, CCPSO-ISM has very satisfactory performance on solving complex problems and outperforms other PSOs on most of the benchmark functions, especially on multimodal functions with many local optima.

We also provided a further test on the proposed algorithm CCPSO-ISM under more difficult environments in order to verify its effectiveness and efficiency. The results show that CCPSO-ISM is slightly affected by the biased initialization ranges of functions and can still have better performance on the rotated problems when compared with traditional PSO. Moreover, the advantages of CCPSO-ISM become more and more evident with the increment of the problems' complexity. In the future work, we will apply CCPSO-ISM to multi-objective optimization problem and test the CCPSO-ISM performance on real-world application problems.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (61232011, 61320106008, 61103162, 61402545), and the National High-Technology Research and Development Program (863 Program) of China (No. 2013AA01A212).

References

[1] P.S. Andrews, An investigation into mutation operators for particle swarm optimization, in: Proccedings of the IEEE Congress on Evolutionary Computation, 2006, pp. 1044–1051.
[2] P.J. Angeline, Using selection to improve particle swarm optimization, in: Proceedings of IEEE International Conference on Evolutionary Computation, 1998, pp. 1945–1950.
[3] Z. Beheshti, S.M. Hj Shamsuddin, CAPSO: centripetal accelerated particle swarm optimization, Inform. Sci. 258 (2014) 54–79.
[4] R. Brits, A.P. Engelbrecht, F. van den Bergh, Locating multiple optima using particle swarm optimization, Appl. Math. Comput. 189 (2007) 1859–1883.
[5] J.C. Chang, A robust adaptive array beamformer using particle swarm optimization for space–time code division multiple access systems, Inform. Sci. 278 (2014) 174–186.
[6] W.R. Charlesworth, Co-operation and competition: contributions to an evolutionary and developmental model, Int. J. Behav. Dev. 19 (1996) 25–38.
[7] M.Y. Chen, A hybrid anfis model for business failure prediction utilizing particle swarm optimization and subtractive clustering, Inform. Sci. 220 (2013) 180–195.
[8] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.
[9] S.M. Elsayed, R.A. Sarker, E. Mezura-Montes, Self-adaptive mix of particle swarm methodologies for constrained optimization, Inform. Sci. 277 (2014) 216–233.
[10] A.P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, John Wiley & Sons, 2006.
[11] H. Gao, S. Kwong, J. Yang, J. Cao, Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation, Inform. Sci. 250 (2013) 82–112.
[12] M. Hadwan, M. Ayob, N.R. Sabar, R. Qu, A harmony search algorithm for nurse rostering problems, Inform. Sci. 233 (2013) 126–140.
[13] T. Hogg, C.P. Williams, Solving the really hard problems with cooperative search, in: Proceedings of the National Conference on Artificial Intelligence, 1993, pp. 231–236.
[14] J. Kennedy, Stereotyping: improving particle swarm performance with cluster analysis, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2000, pp. 1507–1512.
[15] J. Kennedy, Bare bones particle swarms, in: Proceedings of the IEEE Swarm Intelligence Symposium, 2003, pp. 80–87.
[16] J. Kennedy, R. Eberhart, et al., Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.
[17] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2002, pp. 1671–1676.
[18] G. Koulinas, L. Kotsikas, K. Anagnostopoulos, A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem, Inform. Sci. 277 (2014) 680–693.
[19] K. Li, S. Kwong, J. Cao, M. Li, J. Zheng, R. Shen, Achieving balance between proximity and diversity in multi-objective evolutionary algorithm, Inform. Sci. 182 (2012) 220–242.
[20] X.L. Li, X.D. He, A hybrid particle swarm optimization method for structure learning of probabilistic relational models, Inform. Sci. 283 (2014) 258–266.
[21] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (2006) 281–295.
[22] F.Q. Lu, M. Huang, W.K. Ching, T.K. Siu, Credit portfolio management using two-level particle swarm optimization, Inform. Sci. 237 (2013) 162–175.
[23] Q. Lu, Q.L. Han, S. Liu, A finite-time particle swarm optimization algorithm for odor source localization, Inform. Sci. 277 (2014) 111–140.
[24] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, IEEE Trans. Evol. Comput. 8 (2004) 204–210.
[25] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, P.N. Suganthan, A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization, Inform. Sci. 209 (2012) 16–36.
[26] D. Parrott, X. Li, Locating and tracking multiple dynamic optima by a particle swarm model using speciation, IEEE Trans. Evol. Comput. 10 (2006) 440–458.
[27] T. Peram, K. Veeramachaneni, C.K. Mohan, Fitness-distance-ratio based particle swarm optimization, in: Proceedings of the IEEE Swarm Intelligence Symposium, IEEE, 2003, pp. 174–181.
[28] B.Y. Qu, J.J. Liang, P.N. Suganthan, Niching particle swarm optimization with local search for multi-modal optimization, Inform. Sci. 197 (2012) 131–143.
[29] A. Ratnaweera, S. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (2004) 240–255.
[30] T. Ray, K. Liew, A swarm with an effective information sharing mechanism for unconstrained and constrained single objective optimisation problems, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2001, pp. 75–80.
[31] R. Salomon, Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions, BioSystems 39 (1996) 263–278.
[32] M. Shen, Z.H. Zhan, W.N. Chen, Y.J. Gong, J. Zhang, Y. Li, Bi-velocity discrete particle swarm optimization and its application to multicast routing problem in communication networks, IEEE Trans. Ind. Electron. 61 (2014) 7141–7151.
[33] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE International Conference on Evolutionary Computation, 1998, pp. 69–73.
[34] P.K. Tripathi, S. Bandyopadhyay, S.K. Pal, Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients, Inform. Sci. 177 (2007) 5033–5049.
[35] A.M. Turky, S. Abdullah, A multi-population harmony search algorithm with external archive for dynamic optimization problems, Inform. Sci. 272 (2014) 84–95.
[36] H. Wang, H. Sun, C. Li, S. Rahnamayan, J.S. Pan, Diversity enhanced particle swarm optimization with neighborhood search, Inform. Sci. 223 (2013) 119–135.
[37] L. Wang, B. Yang, Y. Chen, Improving particle swarm optimization using multi-layer searching strategy, Inform. Sci. 274 (2014) 70–94.
[38] X.Z. Wang, Y.L. He, L.C. Dong, H.Y. Zhao, Particle swarm optimization for determining fuzzy measures from data, Inform. Sci. 181 (2011) 4230–4252.
[39] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (1999) 82–102.

[40] Z.H. Zhan, J. Li, J. Cao, J. Zhang, H.H. Chung, Y.H. Shi, Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems, IEEE Trans. Cybernet. 43 (2013) 445–463.
[41] Z.H. Zhan, J. Zhang, Y. Li, H. Chung, Adaptive particle swarm optimization, IEEE Trans. Syst. Man Cybernet. Part B: Cybernet. 39 (2009) 1362–1381.
[42] Z.H. Zhan, J. Zhang, Y. Li, Y.H. Shi, Orthogonal learning particle swarm optimization, IEEE Trans. Evol. Comput. 15 (2011) 832–847.
[43] Y. Zhang, D.W. Gong, Z. Ding, A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch, Inform. Sci. 192 (2012) 213–227.