A Novel PSO Model Based on Simulating Human Social Communication Behavior

Yanmin Liu\textsuperscript{1,2} and Ben Niu\textsuperscript{3,4,5}

\textsuperscript{1} School of Economics and Management, Tongji University, Shanghai 200092, China
\textsuperscript{2} School of Mathematics and Computer Science, Zunyi Normal College, Zunyi 563002, China
\textsuperscript{3} College of Management, Shenzhen University, Shenzhen 518060, China
\textsuperscript{4} Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China
\textsuperscript{5} E-Business Technology Institute, The University of Hong Kong, Hong Kong

Correspondence should be addressed to Ben Niu, drniuben@gmail.com

Received 11 May 2012; Revised 22 June 2012; Accepted 25 June 2012

In order to solve the complicated multimodal problems, this paper presents a variant of particle swarm optimizer (PSO) based on the simulation of the human social communication behavior (HSCPSO). In HSCPSO, each particle initially joins a default number of social circles (SC) that consist of some particles, and its learning exemplars include three parts, namely, its own best experience, the experience of the best performing particle in all SCs, and the experiences of the particles of all SCs it is a member of. The learning strategy takes full advantage of the excellent information of each particle to improve the diversity of the swarm to discourage premature convergence. To weight the effects of the particles on the SCs, the worst performing particles will join more SCs to learn from other particles and the best performing particles will leave SCs to reduce their strong influence on other members. Additionally, to insure the effectiveness of solving multimodal problems, the novel parallel hybrid mutation is proposed to improve the particle’s ability to escape from the local optima. Experiments were conducted on a set of classical benchmark functions, and the results demonstrate the good performance of HSCPSO in escaping from the local optima and solving the complex multimodal problems compared with the other PSO variants.

1. Introduction

Particle swarm optimization (PSO), originally introduced by Kennedy and Eberhart [1], has proven to be a powerful competitor to other evolutionary algorithms (e.g., genetic algorithms) [2]. In PSO, these individuals, instead of being manipulated by the evolution operator such as crossover and mutation, are ”evolved” by the cooperation and competition
among the individuals through generations. Each individual in the swarm is called a particle (a point) with a velocity that is dynamically adjusted in the search process according to its own flying experience and the best experience of the swarm.

When solving the unconstrained optimization problem, PSO has empirically turned out to perform well on many optimization problems. However, when it comes to solving complex multimodal problems, PSO may easily get trapped in a local optimum. In order to overcome this defect and improve PSO performance, some researchers proposed several methods [3–20]. In this paper, we present an improved PSO based on human social communication. This strategy ensures the swarm’s diversity against the premature convergence, especially when solving the complex multimodal problems.

This paper is organized as follows. Section 2 presents an overview of the original PSO, as well as a discussion of the previous attempts to improve the PSO performance. Section 3 proposed an improved PSO based on simulation of human communication. Section 4 gives the test functions, the experimental setting, and results. Finally, some conclusions and the future works are discussed in Section 5.

2. Particle Swarm Optimization

2.1. The Original PSO (OPSO)

The original PSO algorithm (OPSO) was inspired by the search behavior of the biological organisms, where each particle moves through the search space by a combination of the best position found so far by itself and its neighbors. In the PSO domain, generally there are two main neighborhood topologies, namely, the global and the local neighborhood that are shown in Figures 1(a) and 1(b), respectively.

The two neighborhood topologies derive two classical PSO variants, namely, the global PSO (GPSO) and the local PSO (LPSO) in which the behaviors of the particles are described by (2.1) and (2.2), respectively. In HSCPSO, the neighborhood topology is somewhat similar to four clusters [4], but not the same as one, as can be seen in Figure 1(c). Here, the black dots represent the particles in each social circle (SC), the circles represent the social circle, and the arrows represent the relationship between SCs. Note that in HSCPSO each particle’s neighborhood is the set of the particles of all SCs that it is a member of (see Section 3.1).
Consider the following:

\[
\begin{align*}
\vec{v}_i(t) &= w \cdot \vec{v}_i(t-1) + \varphi_1 \cdot r_1 (\vec{p}_i - \vec{x}_i(t-1)) + \varphi_2 \cdot r_2 (\vec{p}_g - \vec{x}_i(t-1)), \\
\vec{x}_i(t) &= \vec{x}_i(t-1) + \vec{v}_i(t),
\end{align*}
\]  
(2.1)

\[
\begin{align*}
\vec{v}_i(t) &= w \cdot \vec{v}_i(t-1) + \varphi_1 \cdot r_1 (\vec{p}_i - \vec{x}_i(t-1)) + \varphi_2 \cdot r_2 (\vec{p}_{\text{neighbor}} - \vec{x}_i(t-1)), \\
\vec{x}_i(t) &= \vec{x}_i(t-1) + \vec{v}_i(t),
\end{align*}
\]  
(2.2)

where \( i = 1, \ldots, ps \), \( ps \) is the population size; \( t \) is the iteration counter; \( n \) is the dimension of the search space; \( \varphi_1 \) and \( \varphi_2 \) denote the acceleration coefficients; \( r_1 \) and \( r_2 \) are random vectors with the components uniformly distributed in the range of \([0,1]\); \( \vec{x}_i(t) = (v_{i1}, v_{i2}, \ldots, v_{in}) \) represents the position of the \( i \)th particle at iteration \( t \); \( \vec{v}_i(t) = (v_{i1}, v_{i2}, \ldots, v_{in}) \) represents the velocity of the \( i \)th particle; \( \vec{p}_i \) is the best position yielding the best fitness value for the \( i \)th particle; \( \vec{p}_g \) is the best position discovered by the whole population; \( \vec{p}_{\text{neighbor}} \) is the best position achieved within the neighborhood of current particle \( i \). Note, in this paper the original PSO represent two PSOs, one is PSO with inertia weight and ring topology, and the other is PSO with inertia weight and global topology.

### 2.2. Related Works

Since the introduction of PSO, PSO has attracted a great deal of attention. Many researchers have worked on improving its performance in various ways and derived many interesting variants. In [3], Clerc and Kennedy indicated that a constriction factor \( \chi \) may help to the convergence of PSO. The velocity and position of the \( i \)th particle are updated as follows:

\[
\begin{align*}
\vec{v}_i(t) &= \chi \cdot \{ \vec{v}_i(t-1) + \varphi_1 \cdot r_1 (\vec{p}_i - \vec{x}_i(t-1)) + \varphi_2 \cdot r_2 (\vec{p}_g - \vec{x}_i(t-1)) \}, \\
\vec{x}_i(t) &= \vec{x}_i(t-1) + \vec{v}_i(t),
\end{align*}
\]  
(2.3)

where \( \chi = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \) and \( \phi = \phi_1 + \phi_2, \phi > 4 \).

Kennedy and Mendes [4] claimed that PSO with a small neighborhood might perform better on the complex problems, while the one with a large neighborhood would perform better on the simple problems. In [5], the author proposed a quantum-behaved particle swarm optimization (QPSO) to improve PSO performance. Some researchers [6–9] also combined some techniques to improve PSO performance (e.g., evolutionary operators). However, these improvements are based on a static neighborhood network, which greatly decreases the swarm diversity due to that the particle only learns from the fixed neighborhood. Hence, Suganthan [10] and Hu and Eberhart [11] proposed a dynamic neighborhood topology which transforms gradually from acting like the local neighborhood in the early stage of the search to behaving more like the global neighborhood in the late stage of the search. Liang et al. [12] proposed an improved PSO called CLPSO, which used a novel learning strategy where all particles’ historical best information is used to update a particle’s velocity. In [13], the author proposed the fitness-distance-ratio-based PSO (FDR-PSO) where FDR-PSO selects another particle \( \vec{p}_i \) which is supposed to be a higher fitness value and the nearest to the particle being updated. Mohais et al. [14] proposed a randomly generated
directed graph to define neighborhood which is generated by using two methods, namely, the random edge migration method and the neighborhood restructuring method. Janson and Middendorf [15] arranged the particles in a hierarchy structure, where the best performing particles ascend the tree to influence more particles, replacing relatively worse performing particles which descend the tree. In [16], clubs-based particle swarm optimization (C-PSO) algorithm was proposed, whose membership is dynamically changed to avoid premature convergence and improve performance. In [17], Mendes and Kennedy introduced a fully informed PSO where instead of using the pbest (pbest: personal best position of a given particle, so far) and gbest (gbest: position of the best particle of the entire swarm) positions in the standard algorithm, all the neighbors of the particle are used to update the velocity. The influence of each particle on its neighbors is weighted based on its fitness value and the neighborhood size. In [18], the authors proposed a cooperative particle swarm optimizer (CPSO-H). Although CPSO-H uses one-dimensional (1D) swarms to search each dimension separately, the results of these searches are integrated by a global swarm to significantly improve the performance of the original PSO on multimodal problems. In recent works [19, 20], the authors proposed the dynamic neighborhood topology where the whole population is divided into a number of sub-swarms. These sub-swarms are regrouped frequently by using various regrouping schedules and information exchange among all particles in the whole swarm.

The main differences of our approach with respect to the other proposals existing in the above literatures are as follows.

(i) In HSCPSO, we create the social circle (SC) for the particles analogous to our communities where people can communicate and study to widen the knowledge to each other.

(ii) The learning exemplars of each particle include three parts, namely, its own best experience, the experience of the best performing particle in all SCs, and the experiences of the particles of all SCs it is a member of. This strategy greatly ensures the swarm diversity against the premature convergence.

(iii) The parallel hybrid mutation is used to improve the particle’s ability to escape from the local optima.

3. PSO Based on Simulation of Human Communication Behavior in the Society

3.1. Updating Strategy of Particle Velocity

Based on the simulation of the human social communication behavior, each particle can join more than one SC, and each SC can accommodate any number of particles. Vacant SCs also are allowed. Firstly, each particle randomly joins a predefined number of SCs, which is called the default social circle level (DSC). At each run, the worst performing particles are more socialized through joining more SCs to improve their knowledge, while the best performing particles are less socialized through leaving an SC to reduce their strong influences on other members, which leads to the fact that DSC is dynamically adjusted in terms of the particle’s performance. During this cycle of leaving and joining SCs, the particles that no longer show the extreme performance in its neighborhood will gradually return to DSC. The speed of regaining DSC will decide the algorithm performance, so the check is made every $n$ iteration (here, $n$ is called the gap iteration for adjusting DSC) to find the particles that have SC level
The best performing particle in SC
The worst performing particle in SC
Particle
SC: social circle \((i = 1, 2, \ldots, 6)\)

Figure 2: The snapshot of the SCs at iteration \(t\).
and Rastrigin functions, respectively; the iteration of each run is set as 1000. The mean values of the results are plotted in Figure 3(a).

As all test functions are the minimization problems, the smaller the mean value is, the better the performance works. From Figure 3, we can observe that the learning strategy of CPSO suffers the premature convergence, while the strategy of HSCPSO not only ensures the diversity of the swarm, but also improves the swarm ability to escape from the local optima.

In the real world, everyone has the ability to make himself be one member in any SC to widen the visual field. In a similar way, we hypothesize that each particle in the swarm has the same ability as human in the society. Based on our previous work \cite{20}, the following updating equation of the velocity and position is employed in HSCPSO:

\begin{align}
\overline{v}_i(t) &= w \cdot \overline{v}_i(t-1) + \varphi_1 \cdot r_1(\overline{p}_i - \overline{x}_i(t-1)) + \varphi_2 \cdot r_2(\overline{p}_{bin(i)} - \overline{x}_i(t-1)) \\
&\quad + \varphi_3 \cdot r_3(\overline{p}_g - \overline{x}_i(t-1)), \\
\overline{x}_i(t) &= \overline{x}_i(t-1) + \overline{v}_i(t),
\end{align}

where \(\overline{p}_i = (p_{i1}, p_{i2}, \ldots, p_{in})\) is the best position for the \(i\)th particle; \(\overline{p}_g\) denotes the experience of the best performing particle in all SCs; \(\overline{p}_{bin(i)}\) is called the comprehensive strategy in which the particles’ historical best information of the SCs that particle \(i\) is a member of is used to update a particle’s velocity. Here, \(\varphi_1=0.972\), \(\varphi_2=0.972\), \(\varphi_3=0.972\), and \(w = 0.729\). In the comprehensive strategy, the pseudocode of the learning exemplar choice is shown in Algorithm 1.

### 3.2. Parallel Hybrid Mutation

In \cite{4}, the author has concluded that PSO converges rapidly in the first search process and then gradually slows down or stagnates. The phenomenon is caused by the loss of diversity in the swarm. In order to conquer the default, some researchers \cite{7–9} applied the evolutionary
%SC_i denotes the set of the particles of all SCs that particle i is a member of
%fit(p) represents the corresponding fitness value of an array p

(1) Function comprehensive_learning()
(2) For each particle (i = 1 : ps)
(3) For each dimension (k = 1 : n) % n = particle dimension
(4) flag(i, k) = 0
(5) For each particle (j = 1 : ps) % ps = swarm size
(6) If (j is the member in SC_i) && (|fit(p_i) - fit(p_j)| / |p_{ik} - x_{ik}| > flag(i, k))
(7) p(i, k) = |fit(p_i) - fit(p_j)| / |p_{ik} - x_{ik}| % pi denotes any particle’s pbest
(8) x_{index} = j % the particle ID
(9) End If
(10) Next j
(11) p_{pbest}(i, k) = x_{index} % the k dimension exemplar of the particle i
(12) Next k
(13) Output
(14) Next i
(15) End function

Algorithm 1

$\exp(5(i - 1)/ps - 1) - 1 \right\} \over \exp(5) - 1}$. \hspace{1cm} (3.2)

Figure 4(b) shows an example of mc assigned for 40 particles. Each particle has a mutation ability ranging from 0.05 to 0.5.

(ii) Choose the mode of the mutation, as follows in Algorithm 2.

Algorithm 2
Here, $p_u$ is called the mutation factor that denotes the ratio of the uniform distribution mutation and correspondingly, $1 - p_u$ is the ratio of the gaussian distribution mutation. The Gaussian($\sigma$) returns a random number drawn from a Gaussian distribution with a standard deviation $\sigma$. ceil($p$) rounds the elements of $p$ to the nearest integers greater than or equal to $p$. Here, three main mutation factors (Linear, Exponential, and Sigmoid defined in (3.3), (3.4) and (3.5)) are adopted, and their shapes with maximum generation 2000 are shown in Figure 4(a):

\[
p_u(t) = 1 - \frac{t}{\text{gen}},
\]

\[
p_u(t) = 1 - \left( \exp\left( \frac{t \log 2}{\text{gen}} \right) - 1 \right),
\]

\[
f(r, t) = \frac{1}{1 + \exp(-rt)},
\]

\[
p_u(t) = 1 - f\left(t - \frac{\text{gen}}{2}, r\right),
\]

where $t$ denotes the current generation; gen is the maximum generation.

In order to test which mutation factor is appropriate to our algorithm, the following experiment is conducted. On Sphere, Rosenbrock, Griewanks, Ackley, Rastrigin, noncont, and Rastrigin, HSCPSO with the different mutation factor is run 30 times on each function, and the iteration is set as 2000. The experimental results are standardized to the same scale with (3.7). The results are presented in Table 1, where $p_u$ represents the type of mutation factor; $X_{\text{end}}$ is the standardized value after 2000 iterations; Linear represents the linear mutation factor; Exponent denotes the exponential mutation factor; Sigmoid is Sigmoid mutation factor; No denotes no mutation factor. We can observe that HSCPSO with linear mutation factor achieves the best result, thus the linear mutation factor is adopted in HSCPSO.
Table 1: Standardized value of different mutation factors.

<table>
<thead>
<tr>
<th>Linear</th>
<th>$X_{end}$</th>
<th>$p_u$</th>
<th>Exponent</th>
<th>$X_{end}$</th>
<th>$p_u$</th>
<th>$X_{end}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.7e-06</td>
<td>Sigm0.1</td>
<td>3.8e-05</td>
<td>Sigm0.001</td>
<td>5.9e-04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.9e-04</td>
<td>Sigm0.01</td>
<td>7.2e-04</td>
<td>No</td>
<td>6.8e-03</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.3. **HSCPSO’s Parameter Settings and Analysis of the Swarm’s Diversity**

In this section, we will discuss four parameters, namely, default social circle level (DSC), the gap iteration ($n$) for adjusting DSC, and minimum and maximum of allowed SC number.

3.3.1. **Default Social Circle Level (DSC)**

In order to explore the different DSC effects on HSCPSO, five ten-dimensional test functions are used (Sphere, Rosenbrock, Ackley, Griewanks, and Rastrigin functions). When dealing with experimental data, it is impossible to combine the raw results from the PSOs with the different DSC for the different functions, as they are scaled differently. Thus, Mendes’s method [17] is used to deal with the raw results which are standardized to the same scale as follows:

$$X = \frac{x_{ij} - \mu_{ij}}{\sigma_{ij}},$$

where $x_{ij}$, $\mu_{ij}$, and $\sigma_{ij}$ denote the trial results, mean, and standard deviation of the $i$th test function in the $j$th algorithm, respectively. Note that the parameter $j$ represents the algorithms with the different DSC.

By trial and error, we found that the different DSC yielded the different results as can be seen in Figure 5(a). As all test functions are the minimization problems, the smaller
the standardizing value is, the better the performance works (i.e., the larger the DSC is, the better the performance goes). However, given a second thought, we found the phenomenon is not logical because in the real-life people have no enough energy to take part in all social activities. Therefore, another experiment (the experiment setting and the method of the data processing are the same as one in Figure 5) is conducted to test the impact of DSC. Figure 5(b) gives the simulation results, and we found that the smaller the DSC is, the slower the convergence speed is, and vice versa. The slower convergence rate, rather than the faster convergence speed, is obviously beneficial to the ability to escape from a local optimum [4]. Based on the above analysis, the parameter DSC is set as 15 in our algorithm. Furthermore, in order to ensure the effectiveness of the DSC choice statistically, we adopted the nonparametric Wilcoxon rank sum tests to determine whether the difference is significant or not. The criterion is the following:

\[
\text{result} = \begin{cases} 
1 & \text{if } p < \alpha, \\
0 & \text{if } p \geq \alpha.
\end{cases}
\]  

(3.8)

If \( p \) is less than \( \alpha \), two group numbers are statistically different. If \( p \) is equal or greater than \( \alpha \), it means that two group numbers are not statistically different. From Table 2, we can observe that the performance of DSC-15 is statistically different from that of the other DSC values except for DSC-10.

### 3.3.3. Minimum and Maximum of Allowed SC Number

At each run, the number of the SCs that a particle is a member of is dynamically changed according to its own performance, which will influence the HSCPSO performance. Additionally, in PSO, as the first search process for the global best position requires the exploration of the possible solutions, the LPSO algorithm is chosen in this stage. While the
later search process requires the exploitation of the best found possible solutions by the early search process, the GPSO algorithm is chosen to meet this requirement. Thus, given the above characteristics, in our algorithm, we made the minimum and maximum of allowed SC number dynamically change with the iteration elapsed and empirically proposed (3.9) to set them, respectively:

\[
\begin{align*}
    f_{\min} &= 4 + \text{floor}(\frac{t}{100}) \quad t \in [\text{Maxgen}], \\
    f_{\max} &= 30 - \text{floor}(\frac{t}{100}) \quad t \in [\text{Maxgen}],
\end{align*}
\]

where the floor operation is round towards minus infinity; \( t \) is the iteration counter; \( |\text{Maxgen}| \) is the total number of the iteration. Figure 7(a) represents the minimum and maximum of allowed SC number with the iteration elapsed, respectively, and we can observe that the dynamic minimum and maximum of allowed SC number (dynamic max-min) improves the swarm diversity compared with the fixed max-min. Note that the measure method of the diversity is presented in [20].

3.3.4. Search Bounds Limitation

There are the bounds on the variables’ ranges in many practical problems, and all test functions used in this paper have the bounds. Thus, in order to make the particles fly within the search range, some researchers use the equation \( x_i(t) = \min[x_{\max}, \max(x_{\min}, x_i)] \) to restrict a particle on the border. Here, a different method, but similar to the above-mentioned one, is presented; that is, only when a particle is within the search range, we calculate its fitness.
value and update its $\overline{p_{\text{bin}(i)}}$, $\overline{p_g}$, and $\overline{p}$. As all learning exemplars are within the search range, a particle will eventually return to the search range.

### 3.3.5. Analysis of Swarm’s Diversity

In order to compare the particle diversity of OPSO with that of HSCPSO, we omit the previous velocity $w \cdot \overline{v_i}(t-1)$ and make $\varphi_1$ and $\varphi_2$ equal to one. Thus, the following velocity updating equation is employed in the experiment:

\[ \overline{v_i}(t) \rightarrow r_1(\overline{p_i} - \overline{x_i}(t-1)) + r_2(\overline{p_g} - \overline{x_i}(t-1)), \]
\[ \overline{v_i}(t) \rightarrow r_1(\overline{p_i} - \overline{x_i}(t-1)) + r_2(\overline{p_{\text{bin}(i)}} - \overline{x_i}(t-1)). \]

(3.10)

In terms of (3.10), we run HSCPSO and OPSO on Rosenbrock and Rastrigin functions to analyze the number of the best particle (NBP) at each iteration. Figure 8 gives the scatter plot about the index of the best performing particles associated with the numbers of the iteration. For example, a dot (20, 4000) represents that the index 20’s particle has the best global fitness in the iteration number 4000. The velocity updating strategy of HSCPSO has more NBP than OPSO, which shows that the strategy of HSCPSO increases the swarm’s diversity compared with OPSO [16]. The pseudocode of the HSCPSO is shown in Algorithm 3.

### 4. Experiment Settings and Results

#### 4.1. Test Functions and Parameter Settings of PSO Variants

To compare with the performance of all algorithms, Sphere, Rosenbrock, Acley, Griewanks, Rastrigin, and Rastrigin_noncont functions are selected to test the convergence characteristics,
and Ackley, Rastrigin, Rastrigin-noncont, and Rosenbrock functions are rotated with Salomon’s algorithm [21] to increase the optimization difficulty and test the algorithm robustness (here, the predefined threshold is 0.05, 50, 2, and 3 rep.). Table 3 represents the properties of these functions. Note that in Table 3, the values listed in the search space column are used to specify the range of the initial random particles’ position; the \( x^* \) denotes the global optimum, and the \( f(x^*) \) is the corresponding fitness value.

In order to make these different algorithms comparable, the population size is set at 30 for all PSOs, and each test function is run 30 times. At each run, the maximum fitness evaluation (FEs) is set at \( 3 \times 10^4 \) for the unrotated test functions and \( 6 \times 10^4 \) for the rotated. The comparative algorithms and their parameters settings are listed as below.

(i) Local version PSO with constriction factor (CF-LPSO) [3],

(ii) Global version PSO with constriction factor (CF-GPSO) [3],

(iii) Fitness-distance-ratio-based PSO (FDR-PSO) [13],

(iv) Fully informed particle swarm optimization (FIPS) [17],

---

**Algorithm 3**

(1) Begin
(2) Initialize particles’ position, velocity, \( pbest_i, gbest \) and the related parameters
(3) Initialize default social circle of which the particle \( i \) is a member
(4) \( V_{\text{max}} = 0.25 (X_{\text{max}} - X_{\text{min}}) \)
(5) Initialize \( p_u \)
(6) while (fitcount < Max_FES) \&\& (k < iteration)
(7)    For each particle
(8)        Locate the best particle position of the SCs that particle \( i \) is a member of
(9)        Computer \( P_{\text{ps}} \) in terms of (3.1).
(10)       For each dimension
(11)          Updating the particles’ velocity and position in terms of (3.1)
(12)          \textbf{If} fitness \( (x_i) < \text{fitness}(pbest_i) \)
(13)          \hspace{1cm} \texttt{pbest}_i = x_i, \texttt{pbestval}_i = \text{fitness}(x_i)
(14)          \textbf{End If}
(15)          \textbf{If} ceil(mC + \text{rand} – 1) == 1
(16)          \hspace{1cm} \text{If} \text{rand} \leq p_u
(17)          \hspace{1cm} \hspace{1cm} x(i,d) = (1 + \text{rand}) \times \text{pos}(i,d) \hspace{1cm} \textbf{Else} \hspace{1cm} x(i,d) = \text{Gaussian}() \times \text{pos}(i,d)
(18)          \hspace{1cm} \textbf{End If}
(19)          \textbf{End If}
(20)        Next dimension
(21)    Next particle
(22)    Update the member of SC in terms of Figure 2(b)
(23)    \textbf{If} \text{(mod}(k,n)) \&\& \text{[SC]} \leftrightarrow \text{DSC}
(24)    Update particle’s DSC level.
(25)    \textbf{End If}
(26)    Update each particle neighbor.
(27)    Index = find \( \text{pbestval}_i \) //Find a function which finds a particle’s position
(28)    \textbf{If} \text{fitness}(gbest) > \text{min} \( \text{pbestval}_i \)
(29)    \hspace{1cm} gbestval = \text{pbestval}_i, \text{gbest} = x_{\text{index}}
(30)    \textbf{End If}
(31)    End While
(32) End
Table 3: Dimension, search space, and global optimum of the test functions.

<table>
<thead>
<tr>
<th>Test functions and formulas</th>
<th>n</th>
<th>Global optimum ($x^*$)</th>
<th>$f(x^*)$</th>
<th>search Space</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sphere ($f_1$)</strong></td>
<td>30</td>
<td>[0, 0, ..., 0]</td>
<td>0</td>
<td>[-100, 100]$^30$</td>
</tr>
<tr>
<td><strong>Rosenbrock ($f_2$)</strong></td>
<td>30</td>
<td>[1.1, 1.1, ..., 1]</td>
<td>0</td>
<td>[-2.048, 2.048]$^30$</td>
</tr>
<tr>
<td><strong>Ackley ($f_3$)</strong></td>
<td>30</td>
<td>[0, 0, ..., 0]</td>
<td>0</td>
<td>[-32.768, 32.768]$^30$</td>
</tr>
<tr>
<td><strong>Griewanks ($f_4$)</strong></td>
<td>30</td>
<td>[0, 0, ..., 0]</td>
<td>0</td>
<td>[-600, 600]$^30$</td>
</tr>
<tr>
<td><strong>Rastrigin ($f_5$)</strong></td>
<td>30</td>
<td>[0, 0, ..., 0]</td>
<td>0</td>
<td>[-5.12, 5.12]$^30$</td>
</tr>
<tr>
<td><strong>Rastrigin-noncont ($f_6$)</strong></td>
<td>30</td>
<td>[0, 0, ..., 0]</td>
<td>0</td>
<td>[-5.12, 5.12]$^30$</td>
</tr>
<tr>
<td><strong>Ackley ($f_7$)</strong></td>
<td>30</td>
<td>[0, 0, ..., 0]</td>
<td>0</td>
<td>[-32.768, 32.768]$^30$</td>
</tr>
<tr>
<td><strong>Rastrigin ($f_8$)</strong></td>
<td>30</td>
<td>[0, 0, ..., 0]</td>
<td>0</td>
<td>[-5.12, 5.12]$^30$</td>
</tr>
<tr>
<td><strong>Rastrigin-noncont ($f_9$)</strong></td>
<td>30</td>
<td>[0, 0, ..., 0]</td>
<td>0</td>
<td>[-5.12, 5.12]$^30$</td>
</tr>
<tr>
<td><strong>Rosenbrock ($f_{10}$)</strong></td>
<td>30</td>
<td>[0, 0, ..., 0]</td>
<td>0</td>
<td>[-2.048, 2.048]$^30$</td>
</tr>
</tbody>
</table>
Figure 8: The best particle in the swarm with iteration elapsed.

(v) Cooperative particle swarm optimization (CPSO-H_k) [18],

(vi) Comprehensive learning particle swarm optimizer (CLPSO) [12].

The fully informed PSO (FIPS) with a U-ring topology that achieved the highest success rate was used. CPSO-H_k is a cooperative PSO model combined with the standard PSO, in which k = 6 is used in this paper. \( \psi_1 = 1.492, \psi_2 = 1.492, \) and \( w = 0.729 \) are used in all PSOs except HSCPSO.

4.2. Fixed Iteration Results and Analysis

Tables 4 and 5 present the means and 95% confidence interval after \( 3 \times 10^4 \) and \( 6 \times 10^4 \) function evaluations. The best results among seven PSO algorithms are presented in bold. In addition, to determine whether the result obtained by HSCPSO is statistically different from the results of the other six PSO variants, the Wilcoxon rank sum tests are conducted between the HSCPSO result and the best result achieved by the other five PSO variants on each test problem, and the test results are presented in the last row of Tables 4 and 5. Note that an \( h \) value of one implies that the performance of the two algorithms is statistically different with 95% certainty, whereas \( h \) value of zero indicates that the performance is not statistically different. Figures 9 and 10 present the convergence characteristics in terms of the best fitness value of the median run of each algorithm for each test function.
early on, whereas CLPSO and HSCPSO belong to the performance leaders. As a result of the five PSO variants. In the rotated case, the FDR-PSO algorithm is trapped in local optima the unrotated case, HSCPSO takes the lead, and the FDR-PSO performs better than the other convergence. Note that in the rotated case, there is little e

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
<th>$f_5$</th>
<th>$f_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CF-LPSO</td>
<td>4.1167e - 045</td>
<td>2.5761e + 001</td>
<td>3.5339e - 004</td>
<td>1.2321e - 002</td>
<td>4.2783e + 001</td>
<td>7.0021e + 000</td>
</tr>
<tr>
<td>± 2.340e - 045</td>
<td>± 3.1156e + 001</td>
<td>± 6.1254e - 004</td>
<td>± 6.2135e - 002</td>
<td>± 3.5631e + 001</td>
<td>± 1.2031e + 000</td>
<td></td>
</tr>
<tr>
<td>CF-GPSO</td>
<td>7.1619e - 072</td>
<td>2.2596e + 001</td>
<td>1.3404e + 000</td>
<td>7.7928e - 001</td>
<td>4.5768e + 001</td>
<td>5.0004e + 000</td>
</tr>
<tr>
<td>± 1.342e - 072</td>
<td>± 4.7652e + 001</td>
<td>± 7.8543e + 000</td>
<td>± 3.7452e - 001</td>
<td>± 6.3256e + 001</td>
<td>± 2.0128e + 000</td>
<td></td>
</tr>
<tr>
<td>FDR-PSO</td>
<td>4.2434e - 021</td>
<td>2.5387e + 001</td>
<td>3.6685e - 008</td>
<td>5.1549e + 002</td>
<td>4.7758e + 001</td>
<td>3.0011e + 000</td>
</tr>
<tr>
<td>± 1.096e - 021</td>
<td>± 2.0341e + 001</td>
<td>± 5.3412e - 008</td>
<td>± 7.1321e - 002</td>
<td>± 3.6134e + 001</td>
<td>± 3.2863e + 000</td>
<td></td>
</tr>
<tr>
<td>HSCPSO</td>
<td>2.6302e - 013</td>
<td>2.9948e + 001</td>
<td>1.8719e + 000</td>
<td>1.0885e + 000</td>
<td>1.4509e + 002</td>
<td>6.9801e + 000</td>
</tr>
<tr>
<td>± 2.431e - 013</td>
<td>± 2.0987e + 001</td>
<td>± 6.9432e + 000</td>
<td>± 4.8312e + 000</td>
<td>± 2.6753e + 000</td>
<td>± 2.1109e + 000</td>
<td></td>
</tr>
<tr>
<td>CPSO</td>
<td>3.6684e - 006</td>
<td>6.3901e + 000</td>
<td>3.0047e - 002</td>
<td>3.3829e + 001</td>
<td>3.0242e + 000</td>
<td></td>
</tr>
<tr>
<td>± 3.1233e - 006</td>
<td>± 1.0245e + 000</td>
<td>± 2.8965e - 002</td>
<td>± 3.8901e - 002</td>
<td>± 2.7953e + 001</td>
<td>± 1.9023e + 000</td>
<td></td>
</tr>
<tr>
<td>CLPSO</td>
<td>8.2953e - 013</td>
<td>7.2600e + 001</td>
<td>1.5160e + 000</td>
<td>1.1316e + 002</td>
<td>1.2702e + 001</td>
<td>8.1642e - 003</td>
</tr>
<tr>
<td>± 2.6712e - 013</td>
<td>± 2.1456e + 001</td>
<td>± 1.1463e + 000</td>
<td>± 1.0932e - 002</td>
<td>± 1.6954e + 001</td>
<td>± 3.9206e - 003</td>
<td></td>
</tr>
<tr>
<td>HSCPSO</td>
<td>2.3335e - 012</td>
<td>2.4682e - 001</td>
<td>1.3323e - 014</td>
<td>7.3275e - 015</td>
<td>7.2479e - 003</td>
<td>1.3488e - 005</td>
</tr>
<tr>
<td>± 3.9123e - 012</td>
<td>± 1.6324e - 001</td>
<td>± 7.9854e - 014</td>
<td>± 3.0451e - 015</td>
<td>± 3.4521e - 003</td>
<td>± 1.0028e - 005</td>
<td></td>
</tr>
</tbody>
</table>

| Result  | 0 | 1 | 1 | 1 | 1 |

From the above experimental results, we can observe that Sphere function is easily optimized by CF-GPSO and CF-LPSO, while the other five algorithms show the slower convergence speed. Rosenbrock function has a narrow valley from the perceived local optima to the global optimum. In the unrotated case, HSCPSO may avoid the premature convergence. Note that in the rotated case, there is little effect on all algorithms.

There are many local minima positioned in a regular grid on the multimodal Ackley. In the unrotated case, HSCPSO takes the lead, and the FDR-PSO performs better than the other five PSO variants. In the rotated case, the FDR-PSO algorithm is trapped in local optima early on, whereas CLPSO and HSCPSO belong to the performance leaders. As a result of the
Figure 9: Convergence characteristics of the unrotated test functions. (a) Sphere; (b) Rosenbrock; (c) Ackley; (d) Griekwanks; (e) Rastrigin; (f) Rastrigin-noncont.
comprehensive learning strategy in CLPSO, it can discourage premature convergence. At the same time, this learning strategy of HSCPSO can also ensure the diversity of the swarm to be preserved to discourage the premature convergence.

Rastrigin function exhibits a pattern similar to that observed with Ackley function. In the unrotated case, HSCPSO performs very well, but its performance rapidly deteriorates when the search space is rotated. In the rotated case, CLPSO takes the lead. Note that CLPSO is still able to improve the performance in the rotated case.

On the unrotated Rastrigin-noncont function, HSCPSO presents an excellent performance compared with other algorithms, and after $2.6 \times 10^4$ function evaluation CLPSO has the strongest ability to escape from local optima. In the unrotated case, all algorithms quickly get trapped in a local minimum besides CLPSO and HSCPSO. These two algorithms can avoid the premature convergence and escape from local minima.

Altogether, according to Wilcoxon rank sum tests, we can observe that the HSCPSO algorithm can perform better than the other six algorithms on functions $f_2$, $f_3$, $f_4$, $f_5$, $f_6$, $f_9$, and $f_{10}$. Although HSCPSO has no best performance in $f_7$ and $f_8$, it shows almost the same convergence character as one of CLPSO.

### 4.3. Robustness Analysis

Tables 6 and 7 show the results of the robustness analysis. Here, the “robustness” is used to test the stability of the algorithm optimization ability in different conditions (the rotated and
unrotated test functions) by a certain criterion. In HSCPSO, the criterion is that the algorithm succeeded in the ability of reaching a specified threshold. A robust algorithm is the one that manages to reach the threshold consistently whether in the rotated or the unrotated. The “Success rate” column lists the rate of algorithm reaching threshold in 60 times run. The “FES” column means the number of the function evaluations when reaching the threshold, and only the dates of the successful runs are used to compute “Success rate.”

As can be seen in Tables 6 and 7, none of all PSOs had any difficulty in reaching the threshold on the Rosenbrock function during 30 runs in any case. CF-GPSO has some difficulties in the unrotated and the rotated Ackley function, but CF-LPSO reaches the threshold on the unrotated Ackley. FDR-PSO and CPSO failed on the rotated Ackley function, but in the unrotated case, they consistently reached the threshold. The FIPS completely failed in both the unrotated and the rotated cases. It is interesting to note that CLPSO reached the threshold during all the runs on the rotated Ackley function. However, in the unrotated case, it did not get a perfect success rate. Only HSCPSO consistently reached the threshold in both the unrotated and rotated cases.
The Rastrigin-noncont function is hard to be solved for the majority algorithms, as can be seen in Tables 6 and 7. CLPSO and HSCPSO consistently reached the threshold in the unrotated case, while in the rotated case, only HSCPSO could achieve a perfect success rate.

On Rastrigin function, HSCPSO and CLPSO consistently reached the threshold on both the unrotated and the rotated cases. CPSO and FDR-PSO reached the threshold in the unrotated case, but they failed in the rotated case. CF-LPSO, FIPS, and CF-GPSO algorithms had some difficulties in both the unrotated and the rotated cases.

Altogether, on Rosenbrock, Ackley, Rastrigin, and Rastrigin-noncont functions, HSCPSO shows the stability of the optimization ability in the different conditions. CLPSO, FDR, CPSO, and FIPS consistently reached the threshold on most of test functions, and they were slightly less robust. CF-LPSO and CF-GPSO seemed to be unreliable on the multimodal benchmark functions.

5. Conclusions and Future Works

This paper proposed an improved PSO based on the simulation of the human social behavior (HSCPSO for short) where each particle can adjust its learning strategy in terms of the current condition self-adaptively. From the convergence character and the robustness analysis, we can conclude that HSCPSO significantly improves the ability to solve the complicated multimodal problems compared with the other PSO versions. In addition, from Wilcoxon rank sum tests, these results achieved by HSCPSO are statistically different from the second best result. Although the HSCPSO algorithm outperformed the other PSO variants on most of the test functions evaluated in this paper, it can be regarded as an effective improvement in the PSO domain.

In the future, we will focus on (i) constructing the model for solving the relevant parameters of PSO, (ii) testing the proposed algorithm effectiveness with more multimodal test problems and several composition functions that are more difficult to be optimized, (iii) applying the proposed algorithm to some practices to verify its effectiveness, and (iv) in the future this proposed algorithm will be tested on CEC 2005 problems.

Acknowledgments

This work is supported by The National Natural Science Foundation of China (Grants nos. 71001072, 71271140, 71210107016), China Postdoctoral Science Foundation (Grant nos. 20100480705, 2012T50584), the Science and Technology Fund of Guizhou Province ([2012] 234000), Zunyi Normal College Research Funded Project (2012 BSJJ19), Shanghai Postdoctoral Fund Project (12R2141600), Science and Technology Project of Shenzhen (Grant no. JC201005280492A), and the Natural Science Foundation of Guangdong Province (Grant no. 9451806001002294).

References


Submit your manuscripts at http://www.hindawi.com