IEICE TRANS. INF. & SYST., VOL.E94–D, NO.4 APRIL 2011

PAPER

AMT-PSO: An Adaptive Magnification Transformation Based Particle Swarm Optimizer

Junqi ZHANG[†a)], **Lina NI**[††,†], **Chen XIE**[†], **Ying TAN**[†††], *Nonmembers*, *and* **Zheng TANG**[†,††††*b)], *Member*

SUMMARY This paper presents an adaptive magnification transformation based particle swarm optimizer (AMT-PSO) that provides an adaptive search strategy for each particle along the search process. Magnification transformation is a simple but very powerful mechanism, which is inspired by using a convex lens to see things much clearer. The essence of this transformation is to set a magnifier around an area we are interested in, so that we could inspect the area of interest more carefully and precisely. An evolutionary factor, which utilizes the information of population distribution in particle swarm, is used as an index to adaptively tune the magnification scale factor for each particle in each dimension. Furthermore, a perturbation-based elitist learning strategy is utilized to help the swarm's best particle to escape the local optimum and explore the potential better space. The AMT-PSO is evaluated on 15 unimodal and multimodal benchmark functions. The effects of the adaptive magnification transformation mechanism and the elitist learning strategy in AMT-PSO are studied. Results show that the adaptive magnification transformation mechanism provides the main contribution to the proposed AMT-PSO in terms of convergence speed and solution accuracy on four categories of benchmark test functions.

key words: *particle swarm optimizer, magnification transformation, exploitation, exploration, search strategy, adaptive*

1. Introduction

Metaheuristics have been proven to be highly useful for approximately solving hard optimization problems in practice [1]–[4]. The class of the metaheuristics includes simulated annealing, tabu search, evolutionary algorithms like genetic algorithms and evolution strategies, ant colony optimization, particle swarm optimization, estimation of distribution algorithms, scatter search, the greedy randomized adaptive search, multi-start and iterated local search, guided local search, etc. Over the last years, a large number of algorithms combined with various algorithms ideas, such as selection, crossover, mutation, local search, reset, reinitialization, etc., sometimes also from outside of the traditional metaheuristics field, were reported as the hybrid metaheuristics [4].

Particle swarm optimizer (PSO) is a stochastic global optimization technique based on a social interaction metaphor [5], [6], and has been demonstrated to perform well in many practical engineering fields such as function optimization, artificial neural network training, fuzzy system control, blind source separation as well as machine learning [7]. Furthermore, the PSO has also been found to be robust and fast in solving nonlinear, non-differentiable and multimodal problems [8]. The progress of PSO research and the recent achievements for applications to large-scale optimization problems are reviewed in [9].

A suitable adaptive balance between exploitation and exploration searches is the key to the success of the PSO [10]. Exploration is the ability to test various regions in the problem space in order to locate a good optimum, hopefully the global one. Exploitation is the ability to concentrate the search around a promising candidate solution in order to locate the optimum precisely [11]. Generally, the "exploitation search" and "exploration search" mean the local search and global search, respectively. However, because of the complexity, dynamics and randomness involved in the particle swarm optimizer, it was not trivial to adaptively balance exploitation and exploration along the search process directly by parameter selection. The empirical and theoretical analyses of the dynamics of particle swarms have provided some insights into PSO over the last decade [12]–[23]. A linearly decreasing inertia weight was applied over the course of the search by Shi and Eberhart in [24] to enlarge the global search. A constriction factor [16], [25] was introduced to control the dynamic characteristics of the particle swarm, including its exploration versus exploitation propensities.

In [26], we have proposed a novel idea of using a magnifier to reinforce the exploitation search of particles. In this paper, we extend this idea with the definition of the magnification transformation area, adaptive magnification scale factor, extensive analysis of the adaptive magnification transformation effects and substantial experimental results compared with other improved PSOs on fifteen benchmark test functions, resulting in an adaptive magnification transformation mechanism based particle swarm optimizer (AMT-PSO).

To avoid possible local optima in the convergence state,

Manuscript received June 14, 2010.
Manuscript revised October 25, 2010.

[†]The authors are with the Department of Computer Science and Technology, the Key Laboratory of Embedded System and Service Computing, Ministry of Education; Tongji University, Shanghai, 200092, China.

[††]The author is with the College of Info Sci & Engi, Shandong University of Science & Technology, Qingdao, China.

[†††]The author is with the Key Laboratory of Machine Perception, Ministry of Education; Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, China.

[††††]The author is with the Department of Intellectual Information Systems Engineering, University of Toyama, Toyama-shi, 930–8555 Japan.

a) E-mail: zhangjunqi@tongji.edu.cn
b) E-mail: ztang@eng.u-toyama.ac.jp

DOI: 10.1587/transinf.E94.D.786

combinations with auxiliary techniques have been developed elsewhere by introducing operators such as selection, crossover, mutation, local search, reset, reinitialization, etc., into PSO [27]. These hybrid operations are usually implemented in every generation or at a prefixed interval or are controlled by adaptive strategies as a trigger. While these methods have brought improvements in PSO, the performance may be further enhanced if a finer division of the swarm labor is obtained. Magnification transformation is a simple but very powerful mechanism, which is inspired by the use of a convex lens to see things much clearer. The ranges around the best positions among the particle's neighbors are defined as the area of interest to be magnified for much more careful and precise inspection. An evolutionary factor proposed in [27] which utilizes the information of the population distribution in particle swarm, is combined as an index to adaptively tune the magnification scale factor so that the magnification transformation degree for each particle could be dynamically determined along the search process. The particles far away from the swarm's center keep looking for new potential region, while the particles around the swarm center focus on the refinement of the current best solution. Furthermore, a perturbation-based elitist learning strategy, as a commonly used mechanism to avoid prematurity convergence in PSO, is utilized to help the swarm's best particle to escape the local optimum and explore the potential better space. If the swarm's best particle finds a better solution, all other particles would follow the new best position to jump out and converge to the new region. In this way, a finer division of the swarm labor is obtained.

Moreover, tests are carried out to verify the effectiveness of the AMT-PSO, and to compare with other improved PSO algorithms on 15 unimodal and multimodal benchmark functions comprehensively. The effects of the adaptive magnification transformation and the elitist learning strategy in AMT-PSO are also studied. Results show that the proposed AMT-PSO substantially enhances the performance of the PSO paradigm in terms of convergence speed and solution accuracy on four categories of benchmark functions mainly due to the power of the adaptive magnification transformation mechanism.

The remainder of this paper is organized as follows. Section 2 reviews the PSO paradigm briefly. Section 3 elaborates the proposed AMT-PSO. Section 4 gives extensive experimental results to illustrate the effectiveness and efficiency of the proposed method. Finally, concluding remarks are made in Sect. 5.

2. Particle Swarm Optimizer

PSO uses a particle swarm to search for the optimal solution and defines each particle as a potential solution to a problem in N-dimensional space with a memory of its previous best position and the best position among its neighbors, in addition to a velocity component. The canonical PSO formula with inertia weight is controlled by the Eqs. (1) and (2):

$$\begin{aligned} V_i^d(t+1) &= wV_i^d(t) + c_1 r_1 (P_{iB}^d(t) - X_i^d(t)) \\ &\quad + c_2 r_2 (P_{nB}^d(t) - X_i^d(t)) \qquad (1) \\ X_i^d(t+1) &= X_i^d(t) + V_i^d(t+1) \qquad (2) \end{aligned}$$

where V is the velocity and X is the position of particle i. t is the iteration number. $i = 1, 2, \cdots, N$, N is the number of particles in the swarm, $d = 1, 2, \cdots, D$, and D is the dimension of the solution space. w is the inertia weight [24]. The nonnegative acceleration constants c_1 and c_2 reflect the weighting of stochastic acceleration terms, r_1 and r_2 are random numbers uniformly drawn from the interval $[0, 1]$, which are all scalar quantities for each particle i in each dimension d. P_{iB}^d is the position with the best fitness found so far for the ith particle in the dth dimension. P_{nB}^d is the best position in the neighborhood in the dth dimension. In some literature, instead of using P_{nB}^d, $gBest^d$ may be used in the global-version PSO whereas $lBest^d$ may be used in the local-version PSO.

In the constricted version suggested by Clerc and Kennedy [16], Eqs. (1) and (2) are modified to

$$\begin{aligned} V_i^d(t+1) &= \chi(V_i^d(t) + c_1 r_1 (P_{iB}^d(t) - X_i^d(t)) \\ &\quad + c_2 r_2 (P_{nB}^d(t) - X_i^d(t))) \qquad (3) \\ X_i^d(t+1) &= X_i^d(t) + V_i^d(t+1) \qquad (4) \end{aligned}$$

where the constriction coefficient χ is defined in Eq. (5) as follows:

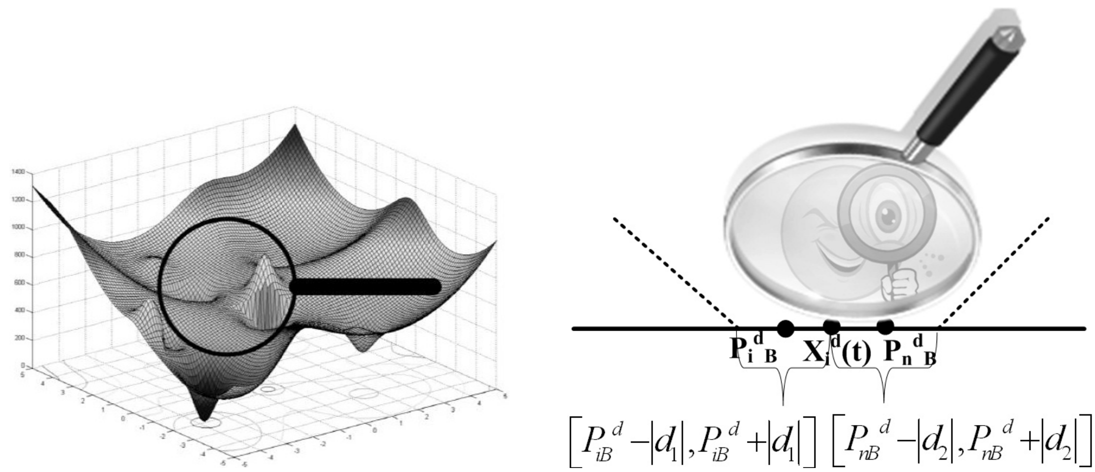
$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \varphi = c_1 + c_2. \qquad (5)$$

Clerc and Kennedy suggested that the values of the constants should be set as: $c_1 = c_2 = 2.05$, $\varphi = 4.1$ and $\chi \approx 0.72984$, so that the convergence of the model can be ensured. The position of every particle in each dimension is independently updated. The only link between the dimensions in the problem space is introduced by the objective function, via P_{iB}^d and P_{nB}^d [11].

3. AMT-PSO

Magnification transformation is a simple but very useful mechanism, which is inspired by using a convex lens to see things much clearer. The essence of this transformation is to set a magnifier around a point we are interested in, so that we could inspect the range around the point more carefully and precisely. There are many widely used transformation strategies, such as Linear Transformations, Non-Linear Transformations which include Fisheye Zoom, Hyperbolic, 3D Pliable Surfaces [28]. For example, the magnification transformation is well used in building screen magnifiers to enlarge the information presented on a visual display in a computer system [29]. However, this magnification transformation technique remains vacant in PSO.

We introduce two factors that are magnification transformation area and magnification scale factor into magnification transformation mechanism to improve the performance of PSO. The magnification transformation area denotes the range to be magnified and is defined as the area just



(a) Sketch map of our AMT-PSO on a composition test function. (b) Magnification transformation area.

Fig. 1 Magnification transformation mechanism and magnification transformation area.

around the best positions among its neighbors in the following. The magnification scale factor adaptively determines the degree of magnification transformation. Figure 1 (a) illustrates the sketch map of the magnification transformation mechanism.

3.1 Magnification Transformation Area

The magnification transformation area ($MTA_i^d(t)$) in PSO is defined in Definition 1, which considers both the personal position P_{iB}^d and the swarm's best position P_{nB}^d in each dimension d in the *t-th* iteration. Figure 1 (b) illustrates the $MTA_i^d(t)$ defined in Definition 1 when the particle position $X_i^d(t)$ is in the middle of the P_{iB}^d and P_{nB}^d.

Definition 1 (Magnification Transformation Area ($MTA_i^d(t)$)): The $MTA_i^d(t)$ for particle i in each dimension d and at the *t-th* iteration is defined as

$$MTA_i^d(t) = [(P_{iB}^d(t) - |d_1|), (P_{iB}^d(t) + |d_1|)] \bigcup [(P_{nB}^d(t) - |d_2|), (P_{nB}^d(t) + |d_2|)] \tag{6}$$

where $d_1 = P_{iB}^d(t) - X_i^d(t)$, $d_2 = P_{nB}^d(t) - X_i^d(t)$.

Let A_L and A_R denote the left and right boundaries of $MTA_i^d(t)$. A_L and A_R can be derived according to Eq. (6) in the following six situations shown in Fig. 2. For example, when $X_i^d(t) = 1$, $P_{nB}^d(t) = 2$ and $P_{iB}^d(t) = 3$ in Situation 1, $A_L = X_i^d(t) = 1$, $A_R = P_{iB}^d(t) + |d_1| = 3 + |d_1| = 3 + (P_{iB}^d(t) - X_i^d(t)) = 3 + (3 - 1) = 5$. The magnification transformation area ($MTA_i^d(t)$) is therefore $[1, 5]$.

3.1.1 Situation 1: $X_i^d(t) \le P_{nB}^d(t) \le P_{iB}^d(t)$

$A_L = X_i^d(t)$, $A_R = P_{iB}^d(t) + |d_1|$.

3.1.2 Situation 2: $X_i^d(t) \le P_{iB}^d(t) \le P_{nB}^d(t)$

$A_L = X_i^d(t)$, $A_R = P_{nB}^d(t) + |d_2|$.

3.1.3 Situation 3: $P_{nB}^d(t) \le P_{iB}^d(t) \le X_i^d(t)$

$A_L = P_{nB}^d(t) - |d_2|$, $A_R = X_i^d(t)$.

3.1.4 Situation 4: $P_{iB}^d(t) \le P_{nB}^d(t) \le X_i^d(t)$.

$A_L = P_{iB}^d(t) - |d_1|$, $A_R = X_i^d(t)$.

3.1.5 Situation 5: $P_{nB}^d(t) \le X_i^d(t) \le P_{iB}^d(t)$

$A_L = P_{nB}^d(t) - |d_2|$, $A_R = P_{iB}^d(t) + |d_1|$.

3.1.6 Situation 6: $P_{iB}^d(t) \le X_i^d(t) \le P_{nB}^d(t)$

$A_L = P_{iB}^d(t) - |d_1|$, $A_R = P_{nB}^d(t) + |d_2|$.

3.2 Magnification Transformation Conditions

The conditions of the magnification transformation on particles depend on the status of the particle positions in the next generation. Whenever particles are going to pass through the magnification transformation area in the next generation, the magnification transformation mechanism would magnify the $MTA_i^d(t)$. In this way, the particles would have a better chance of landing in $MTA_i^d(t)$ and search it in detail. At the same time, the velocities of particles remain constant so that the particles preserve the capability to fly out and continue the exploration search.

Since the particles which will not cross the $MTA_i^d(t)$ in the next generation are busy doing their exploration search, the magnification transformation mechanism will not apply to them.

In this way, the exploration capability of the swarm is maintained while the efficiency of the exploitation search is promoted, which profits from the explicit labor division between the exploration and exploitation searches in AMT-PSO.

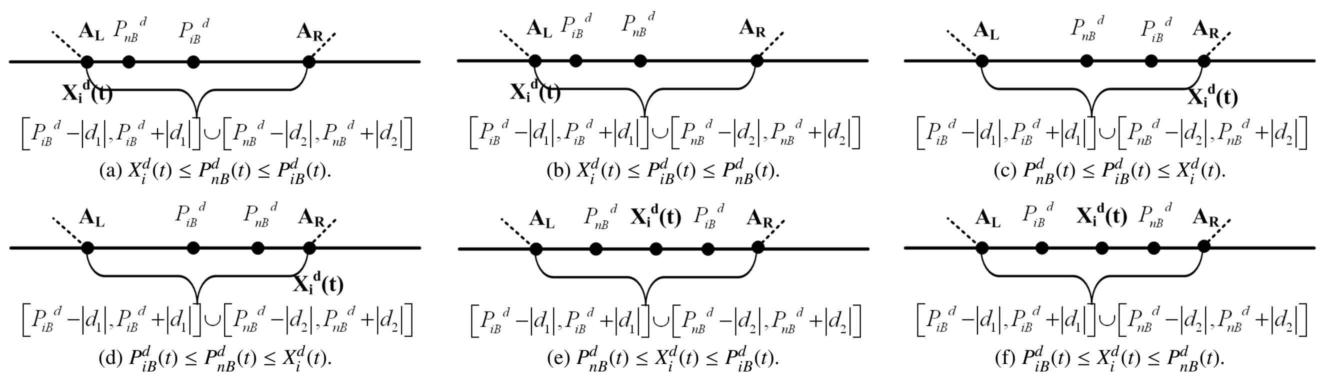


Fig. 2 Situation 1 - 6: The $MTA_i^d(t)$ of particle i in dimension d.

3.3 Adaptive Magnification Scale Factor

The magnification scale factor s is the unique parameter to dynamically tune the search strategy along the search process in AMT-PSO. Every particle i has its own magnification scale factor s_i defined as follows according to all the particles' distribution in the swarm.

Definition 2 (Adaptive Magnification Scale Factor): The adaptive magnification scale factor s_i of particle i is defined according to the particle's distribution in the swarm as

$$s_i = s_{max} - f_i \cdot (s_{max} - s_{min}) \tag{7}$$

where s_{max} and s_{min} are respectively the upper and lower bounds of the magnification scale factor, which will be determined in the experiment part. f_i is the "evolutionary factor" introduced in [27] and computed by

$$f_i = \frac{d_i - d_{min}}{d_{max} - d_{min}} \in [0, 1] \tag{8}$$

where d_i is the mean distance from particle i to other particles, which is given by

$$d_i = \frac{1}{N} \sum_{j=1, j\neq i}^{N} \sqrt{\sum_{k=1}^{D} (x_i^k - x_j^k)^2} \tag{9}$$

where N and D are the population size and the number of dimensions, respectively. d_{max} and d_{min} are the maximum and minimum distances of the mean distances of particles.

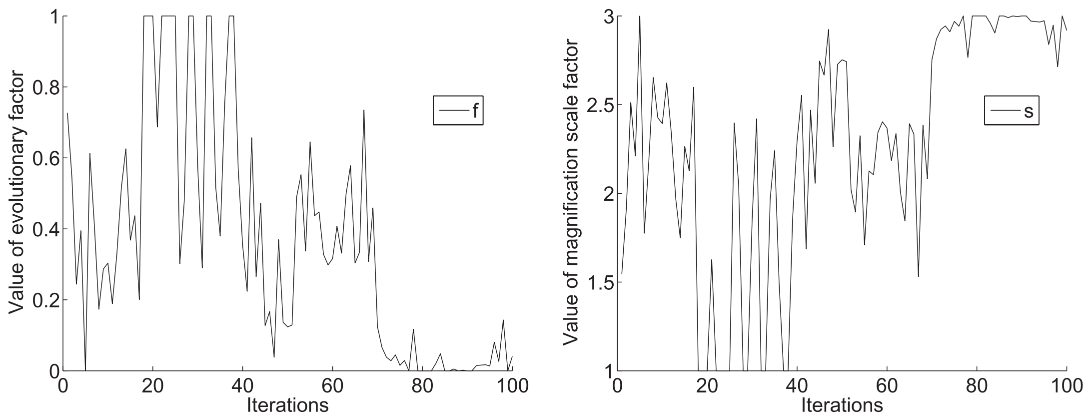
Equation (9) computes the mean distance from each particle to all the other particles. It is reasonable to expect that the mean distance from the globally best particle to other particles would be minimal in the convergence state since the global best tends to be surrounded by the swarm [27]. When a new better optimal region is found, new leaders quickly emerge somewhat far away from the current clustering swarm. This leads the swarm to jump out of the previous optima region to the new region, forming a second convergence. In this way, the search process of the swarm is composed of multiple convergences. The transformation between two successive convergences results from the exploration ability of the swarm and leads the swarm to jump from the local optimum. During each convergence, the swarm exploits the history information and refines the current optimal region for a more accurate solution. Equation (8) computes the distribution information of each particle in the swarm. Bigger f_i means the particle i is far away from the swarm center and explores the space out of the current optimal region. Smaller f_i means the particle i surrounds the swarm center and refines the current optimal region for a more accurate solution. Equation (7) computes the adaptive magnification scale factor s_i of particle i. Small f_i leads to large s_i. Large s_i enhances the exploitation ability of the swarm. In this way, a finer division of the swarm labor is obtained. The particles surrounding the swarm center can pay more attention to refining the current optimal region, while the particles far away from the swarm center can jump out and enhance their abilities to explore the unknown space.

As shown in Fig. 3, according to Eqs. (7) - (9), the particles that converge around the swarm center have bigger magnification scale factor and therefore perform finer search around the current known optimal region due to their smaller evolutionary factor. On the contrary, the particles that are far away from the swarm center have smaller magnification scale factor and therefore jump out to explore the unknown space due to their bigger evolutionary factor.

3.4 Adaptive Magnification Transformation Mechanism

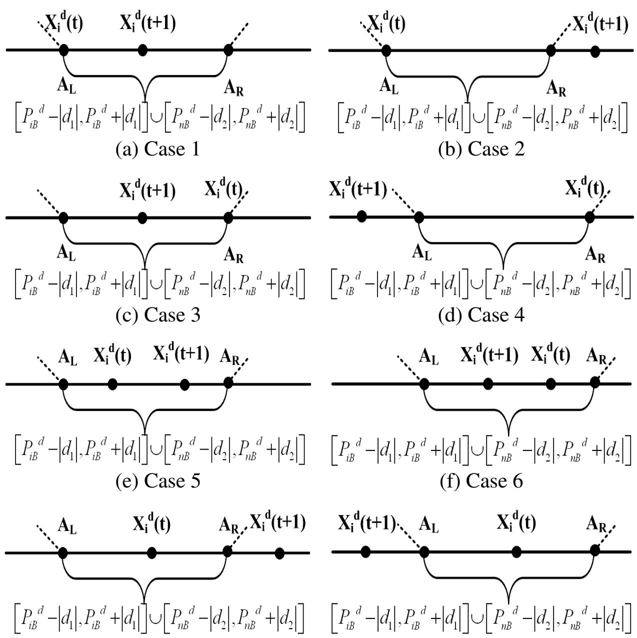
The magnification transformation is elaborated as follows. Let $X_i^d(t)$ be the position of particle i in the current *t-th* generation, $X_i^d(t+1)$ be the position of particle i supposed to be in the next generation without the magnification transformation, which can be fixed by the generation of the two random numbers r_1 and r_2. The magnification transformation of particle i's position X_i^d from the *t-th* to the $(t+1)$*-th* iteration in each dimension can be calculated by Eqs. (10) - (17) in eight cases and schematically illustrated in Figs. 4 (a)- 4 (h), respectively.

The magnification transformation takes effect only in the $MTA_i^d(t)$ and directly magnifies the exploitation by pulling the particles back when the particle flies in the course of the $MTA_i^d(t)$. For example, when $X_i^d(t) = A_L = 1$, $A_R - A_L = 4$, $X_i^d(t+1) = 7$ and $s_i = 2$ in Case 1,



(a) Value of evolutionary factor. (b) Value of adaptive magnification scale factor.

Fig. 3 Evolutionary state information revealed by f and s at run time on multimodal function F_{10} listed in Table 1.



(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4 (e) Case 5 (f) Case 6 (g) Case 7 (h) Case 8

Fig. 4 Case 1 - 8: The position transition of particle i passed through the $MTA_i^d(t)$ along left and right directions.

$((X_i^d(t+1) - X_i^d(t))/s_i) = 3 \le (A_R - A_L) = 4$, $X_i^d(t+1)$ is pulled back into the the $MTA_i^d(t)$ at $X_i^d(t+1) = A_L + (X_i^d(t+1) - X_i^d(t))/s_i) = 4$. When $X_i^d(t+1) = 11$ as in Case 2, $X_i^d(t+1)$ is pulled back but out of the $MTA_i^d(t)$ at $X_i^d(t+1) = X_i^d(t+1) - (A_R - A_L) * (s_i - 1) = 7$. Before the magnification, $X_i^d(t+1) = 11$. After the magnification, the magnification area $[A_R - A_L]$ is magnified $s_i = 2$ times so that the particle has to fly one more magnification area $(A_R - A_L) * (s_i - 1) = (A_R - A_L) * (2 - 1)$. When $s_i = 3$, the particle has to fly two more magnification area $(A_R - A_L) * (s_i - 1) = (A_R - A_L) * (3 - 1)$. Therefore, $X_i^d(t+1) = X_i^d(t+1) - (A_R - A_L) * (s_i - 1)$ in Case 2. In Cases 7 and 8, $(s_i - 1)$ is also used in the same way.

3.4.1 Case 1: $X_i^d(t) = A_L$ *and* $((X_i^d(t+1) - X_i^d(t))/s_i) \le (A_R - A_L)$

$$X_i^d(t+1) = A_L + (X_i^d(t+1) - X_i^d(t))/s_i). \tag{10}$$

3.4.2 Case 2: $X_i^d(t) = A_L$ *and* $((X_i^d(t+1) - X_i^d(t))/s_i) > (A_R - A_L)$

$$X_i^d(t+1) = X_i^d(t+1) - (A_R - A_L) * (s_i - 1). \tag{11}$$

3.4.3 Case 3: $X_i^d(t) = A_R$ *and* $((X_i^d(t) - X_i^d(t+1))/s_i) \le (A_R - A_L)$

$$X_i^d(t+1) = A_L + (X_i^d(t) - X_i^d(t+1))/s_i). \tag{12}$$

3.4.4 Case 4: $X_i^d(t) = A_R$ *and* $((X_i^d(t) - X_i^d(t+1))/s_i) > (A_R - A_L)$

$$X_i^d(t+1) = A_L + (X_i^d(t+1) - X_i^d(t))/s_i). \tag{13}$$

3.4.5 Case 5: $A_L < X_i^d(t) < A_R$ *and* $(X_i^d(t) + (X_i^d(t+1) - X_i^d(t))/s_i \le A_R$

$$X_i^d(t+1) = X_i^d(t) + (X_i^d(t+1) - X_i^d(t))/s_i. \tag{14}$$

3.4.6 Case 6: $A_L < X_i^d(t) < A_R$ *and* $(X_i^d(t) + (X_i^d(t) - X_i^d(t+1))/s_i \ge A_L$

$$X_i^d(t+1) = X_i^d(t) + (X_i^d(t+1) - X_i^d(t))/s_i. \tag{15}$$

3.4.7 Case 7: $A_L < X_i^d(t) < A_R$ *and*

$$(X_i^d(t) + (X_i^d(t+1) - X_i^d(t))/s_i > A_R$$

$$X_i^d(t+1) = X_i^d(t+1) - (A_R - X_i^d(t)) * (s_i - 1). \quad (16)$$

3.4.8 Case 8: $A_L < X_i^d(t) < A_R$ *and*

$$(X_i^d(t) + (X_i^d(t) - X_i^d(t+1))/s_i < A_L$$

$$X_i^d(t+1) = X_i^d(t+1) + (X_i^d(t) - A_L) * (s_i - 1). \quad (17)$$

3.5 Elitist Learning Strategy

PSO emulates the swarm behavior of insects, animals herding, birds flocking, and fish schooling where these swarms search for food in a collaborative manner. Each member in the swarm adapts its search patterns by learning from its own experience and other members' experience [30]. When the swarm can not find the better positions, the swarm's best position may lead the swarm to a local optimum. The swarm's best particle, unlike other particles, has no exemplar to follow and only search around the swarm's best position, which would speed up the premature of the swarm. Therefore, we employ a perturbation-based Elitist Learning Strategy (ELS), as a commonly used mechanism to avoid the prematurity in PSO [27], to help the swarm best particle escape the local optimum and explore the potential better space. If the swarm finds a better solution, all other particles would jump out to follow the best position and converge to the new region.

The ELS randomly chooses one dimension of the swarm's best position, which is denoted as gB^d for the dth dimension. gB^d is perturbed through a Gaussian perturbation

$$gB^d = gB^d + (X_{max}^d - X_{min}^d) \cdot Gaussian(\mu, \sigma^2) \quad (18)$$

where X_{max}^d and X_{min}^d are respectively the upper and lower bounds of the search range. $Gaussian(\mu, \sigma^2)$ is a random number from a Gaussian distribution with a zero mean μ and a standard deviation σ^2.

σ is linearly decreased with the iteration number, which is given by

$$\sigma = \sigma_{max} - (\sigma_{max} - \sigma_{min})\frac{t}{T} \quad (19)$$

where σ_{max} and σ_{min} are respectively the upper and lower bounds of σ, which are suggested in [27] that $\sigma_{max} = 1$ and $\sigma_{min} = 0.1$. t is the current iteration number, T is the maximum iteration number.

3.6 AMT-PSO Algorithm

Figure 5 shows the flow diagram of the proposed AMT-PSO. In the initialization, the initial positions and velocities of particles in the swarm are initialized. Let $\chi \approx 0.72984$ and $c_1 = c_2 = 2.05$. The particles fly in the search space according to the Eqs. (3) and (4). The evolutionary states of particles are computed as indexes to adaptively tune the magnification scale factors. At the same time, the elitist learning strategy is also used to help the best particle escape from the local optima. Whenever particles pass through the $MTA_i^d(t)$, the magnification transformation mechanism will be applied to them. In the next step, the personal best positions of particles, the neighbors best positions, the velocities and positions of particles are all updated. The algorithm can be terminated by a given maximum number G of fitness evaluations (FEs) or a preset solution accuracy. In our experiments, we adopt the former stop criterion.

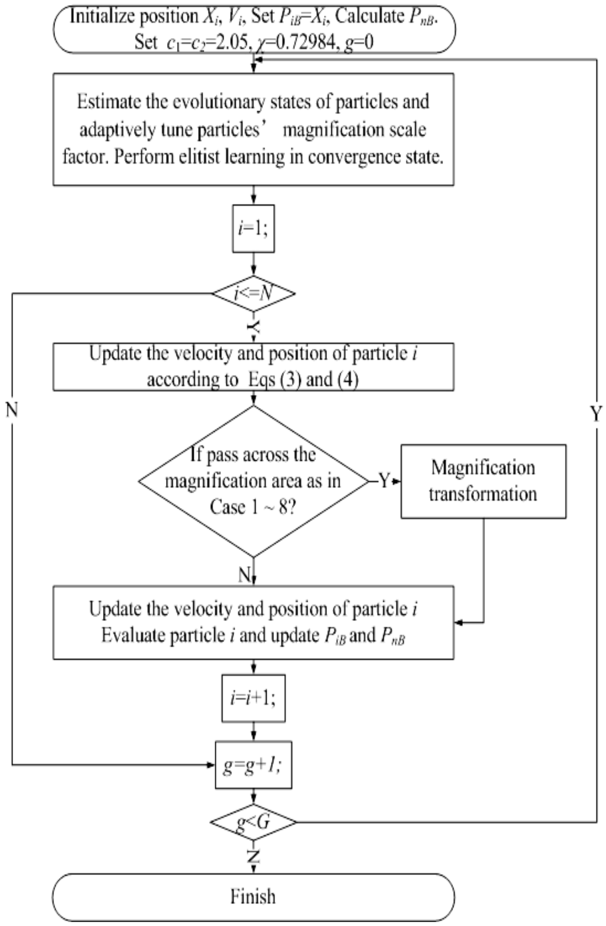


Fig. 5 The flow diagram of the AMT-PSO Algorithm.

An important feature to classify the metaheuristics is the use they make of the search history, that is, whether they use memory or not and how much they use memory [1]. The use of memory is nowadays recognized as one of the fundamental elements of a powerful metaheuristic. The magnification transformation is a metaheuristic that belongs to the memory usage vs. memory-less methods because the usage of the memory, i.e., the known best regions of the swarm, is exploited more finely in the magnification transformation mechanism.

The $MTA_i^d(t)$ contains the history information, that is,

Table 1 List of the benchmark functions and their parameters from CEC'05.

Function Name	Equation	Search Space	f_{bias}	Type
Shifted Sphere	$F_1 = \sum_{i=1}^{D} z_i^2 + f_{bias}, z = x - o$	$[-100, 100]^D$	−450	*UniM*
Shifted Schwefel's Problem 1.2	$F_2 = \sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2 + f_{bias}, z = x - o$	$[-100, 100]^D$	−450	*UniM*
Shifted Rotated High Conditioned Elliptic	$F_3 = \sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}} z_i^2 + f_{bias}, z = (x - o) * M$, M: orthogonal matrix	$[-100, 100]^D$	−450	*UniM*
Shifted Schwefel's Problem 1.2 with Noise in Fitness	$F_4 = \sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2 * (1 + 0.4\lvert N(0,1)\rvert) + f_{bias}, z = x - o$	$[-100, 100]^D$	−450	*UniM*
Schwefel's Problem 2.6 with Global Optimum on Bounds	$F_5 = max\lvert A_i x - B_i\rvert + f_{bias}$, A is a $D * D$ matrix, $B_i = A_i * o$, o is a $D * 1$ vector	$[-100, 100]^D$	−310	*UniM*
Shifted Rosenbrock	$F_6 = \sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f_{bias}, z = x - o + 1$	$[-100, 100]^D$	390	*BMultiM*
Shifted Rotated Griewanks Function without Bounds	$F_7 = \sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} cos(\frac{z_i}{\sqrt{i}}) + 1 + f_{bias}, z = (x - o) * M$, $M = M'(1 + 0.3\lvert N(0,1)\rvert)$, M': linear transformation matrix, condition number=3	$[0, 600]^D$	−180	*BMultiM*
Shifted Rotated Ackleys Function with Global Optimum on Bounds	$F_8 = -20exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}) - exp(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi z_i)) + 20 + e + f_{bias}$, M: linear transformation matrix, $z = (x - o) * M$, condition number=100	$[-32, 32]^D$	−140	*BMultiM*
Shifted Rastrigin's Function	$F_9 = \sum_{i=1}^{D}(z_i^2 - 10cos(2\pi z_i) + 10) + f_{bias}, z = x - o$	$[-5, 5]^D$	−330	*BMultiM*
Shifted Rotated Rastrigin	$F_{10} = \sum_{i=1}^{D}(z_i^2 - 10cos(2\pi z_i) + 10) + f_{bias}$, M: linear transformation matrix, $z = (x - o) * M$, condition number=2	$[-5, 5]^D$	−330	*BMultiM*
Shifted Rotated Weierstrass	$F_{11} = \sum_{i=1}^{D}(\Sigma_{k=0}^{kmax}[\alpha^k cos(2\pi b^k(z_i + 0.5))]) - D\Sigma_{k=0}^{kmax}[\alpha^k cos(2\pi b^k \cdot 0.5)] + f_{bias}$, $\alpha = 0.5, b = 3, k_{max} = 20, z = (x - o) * M$, M: linear transformation matrix, condition number=5	$[-0.5, 0.5]^D$	90	*BMultiM*
Schwefel's Problem 2.13	$F_{12} = \Sigma_{i=1}^{D}(A_i - B_i(x))^2 + f_{bias}, A_i = \Sigma_{i=1}^{D}(\alpha_{ij} sin\alpha_j + b_{ij} cos\alpha_j)$, $B_i(x) = \Sigma_{i=1}^{D}(\alpha_{ij} sinx_j + b_{ij} cosx_j)$	$[-\pi, \pi]^D$	−460	*BMultiM*
Expanded Extended Griewanks plus Rosenbrocks Function (F8F2)	$F_{13} = F8(F2(z_1, z_2)) + F8(F2(z_2, z_3)) + \ldots + F8(F2(z_{D-1}, z_D) + F8(F2(z_{D1}, z_1))$ $+f_{bias}, z = x - o + 1, F8(x) = \Sigma_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} cos(\frac{x_i}{\sqrt{i}}) + 1$, $F2(x) = \sum_{i=1}^{D-1}(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	$[-3, 1]^D$	−130	*EMultiM*
Shifted Rotated Expanded Scaffers F6	$F_{14} = F(z_1, z_2) + F(z_2, z_3) + \ldots + F(z_{D-1}, z_D) + F(z_D, z_1) + f_{bias}$, $z = (x - o) * M, F(x, y) = 0.5 + \frac{(sin^2(\sqrt{x^2+y^2}) - 0.5)}{(1+0.001(x^2+y^2))^2}$, M: linear transformation matrix, condition number=3	$[-100, 100]^D$	−300	*EMultiM*
Hybrid Composition Function	$F_{15} = \Sigma_{i=1}^{n} w_i * f_i'((x - o_i)/\lambda_i * M_i) + bias_i + f_{bias}$, f_i' is composed by five functions, $w_i = exp(-\frac{\Sigma_{k=1}^{D}(x_k - o_{ik})^2)}{2D\sigma_i^2}$, $\sigma_i = 1$ for $i = 1, 2, \ldots, D$, M_i are all identity matrices	$[-5, 5]^D$	120	*HC*

the personal best position and the neighbors' best position. The magnification scale factor (s_i) decides the probability that particles search the $MTA_i^d(t)$. By applying the magnification transformation mechanism, the degree of the history information exploitation increases with the growth of the magnification scale factor. On the contrary, the exploration of the unknown space increases with the decrease of the magnification scale factor (s_i). Therefore, the magnification scale factor (s_i) can be directly used to adaptively balance between the exploitation and the exploration searches along the search process. In this way, the proposed AMT-PSO provides an easy way to balance exploitation and exploration for PSO due to the adaptive and clearer labor division between exploiting the history information and exploring the unknown space.

4. Experimental Results

4.1 Benchmark Functions and Algorithm Configuration

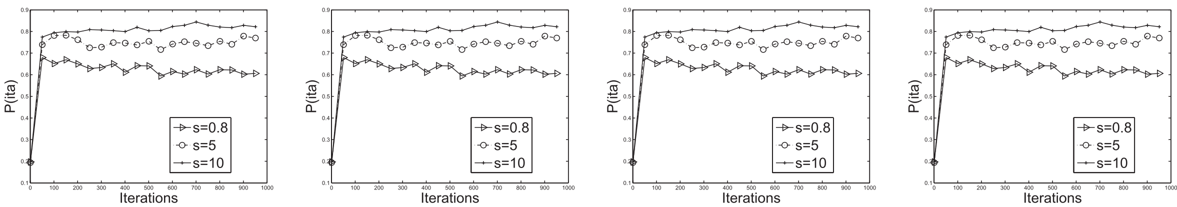
Fifteen benchmark functions from the $CEC'05$ test functions [31] listed in Table 1 are used as the objective functions. o denotes the shifted global optimum. These benchmark functions are unimodal, basic multimodal, extended multimodal and hybrid composition, respectively. In Table 1, *UniM*, *BMultiM*, *EMultiM* and *HC* denote the unimodal, basic multimodal, expanded multimodal and hybrid composition, respectively.

Three improved PSO algorithms have been compared with the AMT-PSO listed in Table 2. The first is GPSO [24], which uses the linear decreasing weight from 0.9 to 0.4, global star topology and $c_1 = c_2 = 2.0$. GPSO is a classical PSO proposed by Yuhui Shi and Eberhart to be compared. The second is LPSO [32], which uses the constricted factors $c_1 = c_2 = 2.05$, $\varphi = 4.1$ and $\chi \approx 0.72984$, and the local ring topology. LPSO is regarded as the standard PSO and has been widely used in PSO applications. The third is MPSO [26], which is a simple idea of using a magnifier to reinforce the exploitation search of particles. MPSO is our first proposed PSO without the definitions of the magnification transformation area and the adaptive magnification scale factor. The magnification transformation area r and the magnification scale factor s are set as $r = 0.3$ and $s = 0.5$ as the same in [26].

For a fair comparison among all the algorithms, these contenders are tested using the standard features of PSO introduced by Daniel Bratton and James Kennedy [33] in the experiments, which use thirty dimensions, non-uniform swarm initialization, and boundary conditions wherein a particle will not be evaluated when it exits the feasible search space. Furthermore, the same population size of 20 that is commonly adopted in PSO [7] is used for comparisons, and all the contenders use the same maximum number of fitness evaluations (FEs) 1.0×10^5 for each test func-

Table 2 PSO algorithms used in the comparison.

Algorithm	Year	Topology	Parameter Setting	Reference
GPSO	1998	Global Star	w : 0.9 − 0.4, $c_1 = c_2 = 2.0$	[24]
LPSO	2002	Local Ring	$w \approx 0.72984$ and $c_1 = c_2 \approx 1.496172$	[16]
MPSO	2008	Global Ring	$w \approx 0.72984$ and $c_1 = c_2 \approx 1.496172$, $r = 0.3$ and $s = 0.5$	[26]
AMT-PSO	-	Local Ring	$w \approx 0.72984$ and $c_1 = c_2 \approx 1.496172$, $s_{min} = 1$ and $s_{max} = 3$	-



(a) $P(ita)$ sequence on F$_3$. (b) $P(ita)$ sequence on F$_5$. (c) $P(ita)$ sequence on F$_9$. (d) $P(ita)$ sequence on F$_{12}$.

Fig. 6 The averaged $P(ita)$ sequences of the AMT-PSO with five particles in five dimensions when optimizing four benchmark functions over 30 independent runs. The swarm evolves for 1000 iterations in each run.

Table 3 Effects of the magnification scale factor on global search quality.

Value of s	F_2	F_4	F_6	F_8	F_{10}	F_{12}	F_{14}
Fixed at 0.8	6.32×10^4	9.00×10^4	7.28×10^9	−118.69	-8.83×10^1	2.00×10^6	−286.01
Fixed at 1	2.57×10^4	3.27×10^4	4.58×10^9	−118.90	-1.93×10^2	1.12×10^6	−286.32
Fixed at 3	-2.26×10^2	$\mathbf{8.21}\times10^3$	4.93×10^2	**−119.02**	-1.91×10^2	6.64×10^5	−286.83
$s_{min} = 0.8$, $s_{max} = 1.0$	3.68×10^4	5.62×10^4	1.27×10^9	−118.86	$\mathbf{-2.18}\times10^2$	1.52×10^6	−286.31
$s_{min} = 1.0$, $s_{max} = 3.0$	$\mathbf{-3.12}\times10^2$	1.30×10^4	$\mathbf{4.61}\times10^{29}$	−119.01	-2.10×10^2	$\mathbf{4.86}\times10^5$	**−286.96**
$s_{min} = 0.8$, $s_{max} = 3.0$	-4.6×10^1	9.49×10^3	2.32×10^8	−118.95	-2.11×10^2	5.76×10^6	−286.85

tion [31]. All the experiments are conducted on the same machine with a Intel (R) Pentium (R) Dual 1.80-GHz CPU, 3-GB memory, and Windows XP2 operating system. For the purpose of statistical errors reduction, each benchmark test function is independently simulated 30 times, and their mean results are used for the comparisons.

4.2 Comparisons on the Solution Accuracy

The performances on the solution accuracy of GPSO, LSPO and MPSO are compared with AMT-PSO. The results are shown in Table 4 in terms of the mean and SD (standard deviation) of the solutions obtained in the 30 independent runs by each algorithm. Boldface in the table indicates the best result among these four contenders.

Table 4 reveals that, when solving unimodal problems, the AMT-PSO achieves the best performance on all the test functions. When solving the basic multimodal problems, the AMT-PSO achieves the highest accuracy on most functions F_8, F_9, F_{10}, F_{11} and F_{12}, and ranks third on F_6 and F_7. The AMT-PSO also achieves the highest accuracy on extended multimodal and hybid composition functions F_{14} and F_{15}, and ranks second on the extended multimodal function F_{13}, which means that the AMT-PSO can successfully jump out of the local optima and surpass all the other algorithms on the complex multimodal problems. The abilities of achieving the highest accurate solutions and avoiding being trapped into the local optima suggest that the AMT-PSO can indeed benefit from the adaptive magnification transformation and the ELS.

4.3 Comparisons on the Convergence Speed

The other salient yardstick for measuring the algorithm performance is the convergence speed in the global optimum achievement. Figure 7 graphically illustrates the comparisons in terms of convergence characteristics of the evolutionary process in solving the 15 different problems. It can be shown that the AMT-PSO generally offers a much higher speed, which suggests the power of the magnification transformation mechanism in refining the solutions.

It can be concluded from Fig. 7 and Table 4 that the AMT-PSO outperforms other contenders both in 12 out of 15 test functions on the solution accuracy, and in 9 out of 15 test functions on the convergence speed. The CPU time is important to measure the computational load, as many existing PSO variants have added extra operations that cost computational time. Because the topology of global ring only computes the global best particle of the swarm while the topology of local ring needs to compute the best particle of the neighbors for every particle, the GPSO and the MPSO cost less CPU time than the LPSO and the AMT-PSO. Although the AMT-PSO needs to calculate the mean distance between every pair of particles in the swarm, the calculation costs negligible CPU time (0.015) compared with the averaged CPU time of the LPSO (0.0145).

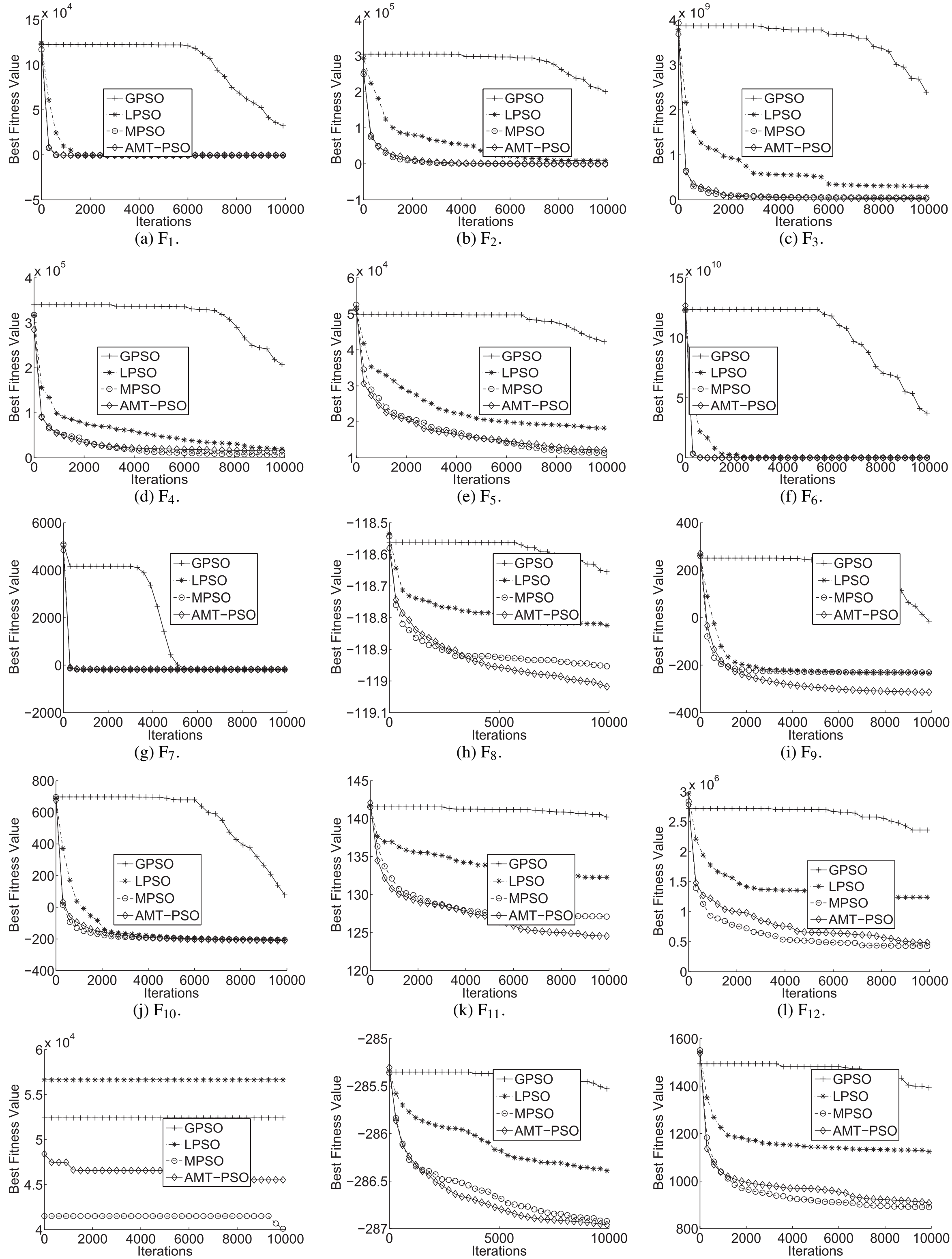


Fig. 7 Convergence curves of the four different PSOs on the benchmark test functions F_1 - F_{15}.

In solving the real-world problems, the fitness evaluation time overwhelms the algorithm's overhead due to the uncertainty of the complexity of them. The mean number of FEs needed to reach acceptable accuracy would be much more interesting than the CPU time. The mean FEs are therefore explicitly presented and compared in Table 4. The

Table 4 Search results of comparisons among four PSOs on the test functions.

Func.	Statistic	GPSO	LPSO	MPSO	AMT-PSO
F_1	Mean	3.15×10^{4}	$\mathbf{-4.50}\times10^{2}$	$\mathbf{-4.50}\times10^{2}$	$\mathbf{-4.50}\times10^{2}$
	Std.Dev	5.45×10^{4}	$\mathbf{5.38}\times10^{-14}$	9.47×10^{-12}	4.44×10^{-10}
F_2	Mean	1.99×10^{5}	8.62×10^{3}	1.31×10^{4}	$\mathbf{-3.12}\times10^{2}$
	Std.Dev	1.83×10^{5}	3.01×10^{4}	2.56×10^{4}	$\mathbf{7.55}\times10^{1}$
F_3	Mean	2.38×10^{9}	2.92×10^{8}	1.93×10^{8}	$\mathbf{2.76}\times10^{7}$
	Std.Dev	1.50×10^{9}	4.85×10^{8}	1.84×10^{8}	$\mathbf{3.87}\times10^{7}$
F_4	Mean	2.68×10^{5}	1.91×10^{4}	3.24×10^{4}	$\mathbf{1.30}\times10^{4}$
	Std.Dev	1.83×10^{5}	1.23×10^{4}	$\mathbf{1.05}\times10^{4}$	1.82×10^{4}
F_5	Mean	4.19×10^{4}	1.82×10^{4}	1.78×10^{4}	$\mathbf{1.21}\times10^{4}$
	Std.Dev	1.20×10^{4}	9.22×10^{3}	6.07×10^{3}	$\mathbf{4.08}\times10^{3}$
F_6	Mean	3.31×10^{10}	$\mathbf{4.57}\times10^{2}$	8.08×10^{8}	4.61×10^{2}
	Std.Dev	5.16×10^{10}	$\mathbf{1.36}\times10^{2}$	1.40×10^{9}	2.59×10^{2}
F_7	Mean	**−179.98**	−179.97	−179.97	−179.88
	Std.Dev	1.69×10^{-2}	$\mathbf{1.48}\times10^{-2}$	1.61×10^{-2}	5.90×10^{-2}
F_8	Mean	-1.18×10^{2}	-1.18×10^{2}	-1.18×10^{2}	$\mathbf{-1.19}\times10^{2}$
	Std.Dev	**0.09**	0.15	0.14	0.27
F_9	Mean	-3.4×10^{1}	-2.34×10^{2}	-1.69×10^{2}	$\mathbf{-3.14}\times10^{2}$
	Std.Dev	2.13×10^{2}	$\mathbf{2.73}\times10^{1}$	5.31×10^{1}	4.19×10^{1}
F_{10}	Mean	6.7×10^{1}	-2.05×10^{2}	-1.44×10^{2}	$\mathbf{-2.10}\times10^{2}$
	Std.Dev	3.47×10^{2}	5.31×10^{1}	5.70×10^{1}	$\mathbf{3.36}\times10^{1}$
F_{11}	Mean	1.40×10^{2}	1.32×10^{2}	1.28×10^{2}	$\mathbf{1.24}\times10^{2}$
	Std.Dev	**1.86**	5.95	3.82	6.50
F_{12}	Mean	2.35×10^{6}	1.23×10^{6}	8.05×10^{5}	$\mathbf{4.86}\times10^{5}$
	Std.Dev	$\mathbf{4.19}\times10^{5}$	7.96×10^{5}	6.01×10^{5}	6.22×10^{5}
F_{13}	Mean	5.24×10^{4}	5.49×10^{4}	$\mathbf{4.29}\times10^{4}$	4.55×10^{4}
	Std.Dev	3.20×10^{4}	3.04×10^{4}	3.08×10^{4}	$\mathbf{2.72}\times10^{4}$
F_{14}	Mean	-28.55×10^{1}	-28.63×10^{1}	-28.66×10^{1}	$\mathbf{-28.69}\times10^{1}$
	Std.Dev	$\mathbf{4.02}\times10^{-1}$	6.16×10^{-1}	4.06×10^{-1}	5.23×10^{-1}
F_{15}	Mean	1.39×10^{3}	1.12×10^{3}	1.01×10^{3}	$\mathbf{9.07}\times10^{2}$
	Std.Dev	$\mathbf{1.10}\times10^{2}$	1.58×10^{2}	2.26×10^{2}	3.22×10^{2}
Accuracy	Total	1	2	2	**12**
Speed	Total	0	0	6	**9**
Time (sec)	Mean	**0.0045**	0.0145	0.0054	0.015
FEs	Mean	8940	5201	6506	**3562**

"acceptance" values are set as $\{-400, -1000, -3\times10^8, 3\times10^4, 3\times10^4, 800, -179.8, -118, -170, -150, 130, 8\times10^5, 4\times10^4, -286, 1000\}$ for the fifteen functions, respectively, to gauge whether a solution found by the non deterministic PSO would be acceptable or not. Noticeably, the AMT-PSO uses the least FEs to reach the acceptable solutions.

4.4 Merits of the Magnification Transformation

Here, the AMT-PSO is tested again to see how it can directly speed up the exploitation of particles. The search behavior of the AMT-PSO is investigated on F_3, F_5, F_9 and F_{12} listed in Table 1. To compare the exploitation capabilities of different magnification scale factors, an "exploitation probability $P(ita)$" proposed in [34], [35] is used here as a yard stick and is denoted by $P(ita)$ as:

$$P(ita) = \int_{A(ita)} f_X(x)dx \tag{20}$$

where $A(ita)$ is defined as the exploitation area and is the same as $MTA_i^d(t)$ in definition 1. The exploitation area is the union of two areas, which are centered around $P_{iB}^d(t)$ and $P_{nB}^d(t)$. The complementary set is defined as the exploration area. $f(x)$ is the density of the probability distribution of the particle's position X in the next iteration. $f(x)$ can be integrated according to the derived density function $f_X(x)$ of the hybrid uniform distribution of the PSO formula. The concrete computational formula of $f(x)$ can be found in [34]. $P(ita)$ represents the exploitation probability of each particle in each dimension based on the accurate theoretical analysis on the sampling distribution in PSO [35].

As can be seen from Fig. 6, the $P(ita)$ of the swarm approximately increases with the growth of the magnification scale factor s. This indicates that the magnification transformation mechanism could directly change the exploitation probability in the swarm. At the same time, due to the Elitist Learning Strategy, the swarm possesses the capability of jumping out the local optima and then converging to another region.

4.5 Sensitivity of the Adaptive Magnification Scale Factor Boundary

The boundaries of magnification scale factor s_{min} and s_{max} may influence the performance of AMT-PSO. To assess the sensitivity of the boundaries of magnification scale factor s_i, six strategies for setting the value of s are tested here using three fixed values (0.8, 1.0, and 3.0) and three adaptive boundaries ($\{s_{min} = 0.8,\ s_{max} = 1.0\}$, $\{s_{min} = 1.0,\ s_{max} = 3.0\}$, $\{s_{min} = 0.8,\ s_{max} = 3.0\}$). The mean results of 30 independent trials are presented in Table 3.

Results show that if s is less than 1.0 (0.8 here), the learning rate is not big enough to help the swarm to refine its solutions, but help the the swarm move away from its known optimal regions so as to jump out of the local optima, which is evident in the performance on all the test functions. However, all other settings, which permits a larger s (e.g. $s = 3$), have delivered the excellent performance, particularly the strategy with a distance-varying s increasing from 1 to 3. This outstanding performance confirms the intuition in AMT-PSO that the outside particles should explore the unknown space and the inner particles should concentrate on refining the current optimal region.

4.6 Merits of the Magnification Transformation Adaptation and Elitist Learning

To quantify the significance of these two mechanisms, the performance of AMT-PSO without adaptation (s=3) or elitist learning is tested under the same running conditions. Results of the mean value and standard deviance on 30 independent trials are presented in Table 5.

It is clear from the results that with the adaptive magnification transformation alone, the AMT-PSO performs best on the unimodal functions. Unfortunately, the AMT-PSO suffers from being trapped into the local optima at the meanwhile. The benchmark functions have various sizes of the search domain, a large number of local optima, global optimum far away from any of the local optima or surrounded by a large local optima, slow or sharp slope, shifted global

Table 5 Merits of magnification transformation adaptation and elitist learning on search quality.

Func.	Statistic	AMT-PSO With ELS & Adaptation	AMT-PSO With only Adaptation	AMT-PSO With ELS	LPSO Without ELS & Adaptation
F_2	Mean	-3.12×10^2	$\mathbf{-3.34}\times10^2$	2.57×10^4	8.62×10^3
	Std.Dev	7.55×10^2	$\mathbf{4.64}\times10^2$	6.38×10^4	3.01×10^4
F_4	Mean	1.30×10^4	$\mathbf{1.02}\times10^4$	3.27×10^4	1.91×10^4
	Std.Dev	1.82×10^4	$\mathbf{5.95}\times10^3$	5.93×10^4	1.23×10^4
F_6	Mean	4.61×10^2	$\mathbf{4.23}\times10^2$	4.58×10^9	4.57×10^2
	Std.Dev	2.59×10^2	5.71×10^2	1.45×10^9	$\mathbf{1.36}\times10^2$
F_8	Mean	$\mathbf{-11.90}\times10^1$	-11.88×10^1	-11.89×10^1	-11.88×10^1
	Std.Dev	27.04×10^{-2}	$\mathbf{15.75}\times10^{-2}$	29.40×10^{-2}	15.78×10^{-2}
F_{10}	Mean	$\mathbf{-2.10}\times10^2$	-2.02×10^2	-1.93×10^2	-2.05×10^2
	Std.Dev	3.36×10^1	$\mathbf{2.98}\times10^1$	9.94×10^1	5.31×10^1
F_{12}	Mean	4.86×10^5	$\mathbf{4.54}\times10^5$	1.12×10^6	1.23×10^6
	Std.Dev	6.22×10^5	$\mathbf{5.45}\times10^5$	8.22×10^5	7.96×10^5
F_{14}	Mean	**-289.96**	-289.85	-289.32	-289.39
	Std.Dev	5.23×10^{-1}	$\mathbf{5.03}\times10^{-1}$	5.67×10^{-1}	6.16×10^{-1}

optimum, etc. The statistical results indicate that the Elitist Learning Strategy improves the ability of PSOs to avoid being trapped into the local optima. However, the PSOs without adaptive magnification perform the worst on all the test functions, which indicates that the adaptive magnification transformation is the main reason for promoting the performance of AMT-PSO.

5. Conclusion

In this paper, PSO has been extended to AMT-PSO. This process in AMT-PSO has been made possible by the adaptive magnification transformation mechanism, which utilizes the information of population distribution in particle swarm. First, the magnification transformation area has been defined as the ranges around the best positions among the particle's neighbors. Then, an evolutionary factor has been used as an index to adaptively tune the magnification scale factor for each particle in each dimension. In this way, the particles far away from the swarm center keep looking for new potential region, while the particles around the swarm center focus on refining current best solution. Furthermore, an Elitist Learning Strategy enables the swarm to jump out of any possible local optima. Thus, the success of AMT-PSO highly profits from the more effective labor division between the exploration and exploitation searches in the swarm. As shown in the benchmark tests, the adaptive control of the magnification transformation mechanism makes the AMT-PSO algorithm much more efficient, offering a substantially improved convergence speed and solution accuracy compared with other improved algorithms on fifteen benchmark test functions.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (NSFC), under Grants No. 60803065, 60804042, 60903194, 60803032 and 60875080. This program is also supported by the NHTRDP of China (863 Program) under Grants No. 2009AA01Z141, the National Basic Research Program of China(973 Program) under Grant No. 2010CB328101, the Program for Changjiang Scholars and Innovative Research Team in University, Program for Young Excellent Talents in Tongji University under Grant No. 2009KJ032.

References

[1] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," ACM Comput. Surv., vol.35, no.3, pp.268–308, 2003.

[2] F. Glover and G.A. Kochenberger, Handbook of metaheuristics, Kluwer, Boston, 2003.

[3] H.H. Hoos and T. Stutzle, Stochastic local search, Morgan Kaufmann, Elsevier, 2005.

[4] G. Raidl, "A unified view on hybrid metaheuristics," Lect. Notes Comput. Sci., vol.4030, pp.1–12, Springer Berlin/Heidelberg, 2006.

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proc. IEEE International Conference on Neural Networks, pp.1942–1948, Perth, Australia, 1995.

[6] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," Proc. 6th International Symposium on Micro Machine and Human Science, pp.39–43, Nagoya, Japan, 1995.

[7] R.C. Eberhart and Y.H. Shi, "Particle swarm optimization: Developments, applications and resources," Proc. IEEE Congress on Evolutionary Computation, vol.1, pp. 81–86, Seoul, South Korea, 2001.

[8] H.W. Ge, Y.C. Liang, Y. Zhou, and X.C. Guo, "A particle swarm optimization-based algorithm for job-shop scheduling problem," Int. J. Comput. Meth., vol.2, no.3, pp.419–430, 2005.

[9] K. Keisuke, "Particle swarm optimization — A survey," IEICE Trans. Inf. & Syst., vol.E92-D, no.7, pp.1354–1361, July 2009.

[10] Holland, Adaption in natural and artificial systems, University of Michigan, 1975.

[11] I.C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," Inf. Process. Lett., vol.85, no.6, pp.317–325, 2003.

[12] J. Kennedy, "The behavior of particle," Proc. 7th Annu. Conf. Evol. Program., pp.581–589, San Diego, CA, March 1998.

[13] E. Ozcan and C.K. Mohan, "Analysis of a simple particle swarm optimization system," Proc. IEEE International Conference on Intelligent Engineering Systems Through Artificial Neural Networks, pp.253–258, 1998.

[14] E. Ozcan, Using genetic algorithms, PhD Dissertation, Department of EECS, Syracuse University, Syracuse (NY), USA, 1998.

[15] E. Ozcan and C.K. Mohan, "Particle swarm optimization: Surfing the waves," Proc. Congress on Evolution Computation, pp.1939–1944, 1999.

[16] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," IEEE Trans. Evol. Comput., vol.6, no.1, pp.58–73, 2002.

[17] F. van den Bergh, An analysis of particle swarm optimizers, PhD thesis, Department of Computer Science, University of Pretoria, South Africa, 2002.

[18] F. van den Bergh and A.P. Engelbrecht, "A study of particle swarm optimization particle trajectories," Inf. Sci., pp.937–971, 2006.

[19] H.M. Emara and H.A.A. Fattah, "Continuous swarm optimization technique with stability analysis," Proc. 2004 American Control Conference, pp.2811–2817, Boston, Massachusetts, 2004.

[20] E.F. Campana, G. Fasano, and A. Pinto, "Dynamic system analysis and initial particles position in particle swarm optimization," IEEE Swarm Intelligencce Symposium, Indianapolis, 2006.

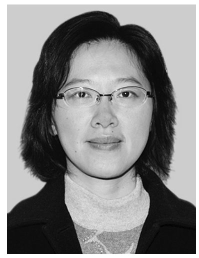
[21] E.F. Campana, G. Fasano, and A. Pinto, "Particle swarm optimization: Efficient globally convergent modifications," Proc. Third European Conference on Computational Mechanics, Solids, Structures and Coupled Problems in Engineering, Lisbon, Portugal, 2006.

[22] M. Clerc, "Stagnation analysis in particle swarm optimization or what happens when nothing happens," Technical Report CSM-460, Department of Computer Science, University of Essex, 2006.

[23] V. Kadirkamanathan, K. Selvarajah, and P.J. Fleming, “Stability analysis of the particle dynamics in particle swarm optimizer,” IEEE Trans. Evol. Comput., vol.10, no.3, pp. 245–255, 2006.
[24] Y.H. Shi and R. Eberhart, “A modified particle swarm optimizer,” IEEE World Congress on Computational Intelligence, pp.69–73, Alaska, ALTEC, 1998.
[25] M. Clerc, “The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization,” Proc. International Center for Electronic Commerce(ICEC), pp.1951–1957, Washington D.C., 1999.
[26] J.Q. Zhang, K. Liu, Y. Tan, and X.G. He, “Magnifier particle swarm optimization for numerical optimization,” Proc. Genetic and Evolutionary Computation Conference, pp.167–168, Atlanta, GA, 2008.
[27] Z.-H. Zhan, J. Zhang, Y. Li, and H.S.-H. Chung, “Adaptive particle swarm optimization,” IEEE Trans. Syst. Man Cybern. B, Cybern., vol.39, no.6, pp.1362–1381, 2009.
[28] T.A. Keahey and E.L. Robertson, “Techniques for non-linear magnification transformations,” Proc. IEEE Symposium on Information Visualization, pp.38–45, 1996.
[29] P. Blenkhorn and D.G. Evans, “A screen magnifier using high level implementation techniques,” IEEE Trans. Neural Systems and Rehabilitation Engineering, vol.14, no.4, pp.501–504, 2006.
[30] J.J. Liang, A.K. Qin, P.N. Suganthan, and S. Baskar, “Comprehensive learning particle swarm optimizer for global optimization of multimodal functions,” IEEE Trans. Evol. Comput., vol.10, no.3, pp.281–296, 2006.
[31] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.P. Chen, A. Auger, and S. Tiwari, “Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization,” http://www.ntu.edu.sg/home/EPNSugan, 2005.
[32] J. Kennedy and R. Mendes, “Population structure and particle swarm performance,” Proc. IEEE Congress on Evolutionary Computation, pp.1671–1676, 2002.
[33] D. Bratton and J. Kennedy, “Defining a standard for particle swarm optimization,” Proc. IEEE Swarm Intelligence Symposium (SIS), pp.120–127, Honolulu, HI, 2007.
[34] J. Zhang, K. Liu, Y. Tan, and X.G. He, “Allocation of local and global search capabilities of particle in canonical pso,” Proc. Genetic and Evolutionary Computation Conference, pp.165–166, Atlanta, GA, 2008.
[35] J. Zhang, Y. Tan, L. Ni, C. Xie, and Z. Tang, “Hybrid uniform distribution of particle swarm optimizer,” IEICE Trans. Fundamentals, vol.E93-A, no.10, pp.1782–1791, Oct. 2010.

Junqi Zhang received his Ph.D. in Computing Science from Fudan University, China, in 2007. He became a postdoctoral research fellow and a lecturer of the Key Laboratory of Machine Perception(Peking University), Ministry of Education, and with Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Beijing, China in 2007. He has received the outstanding post-doctoral award in Peking university, the Financial Support from China Postdoctor Science foundation and the Spacial Grade of the Financial Support from China Postdoctor Science foundation. He is currently a research fellow and lecturer with Department of Computer Science and Technology, Tongji University, Shanghai, China. His research interests include computational intelligence, particle swarm optimization, machine learning, statistical learning, high-dimensional index and multimedia data management. His research is supported by the National Natural Science Foundation of China (NSFC).

Lina Ni is a Ph.D. candidate in National High Performance Center Tongji Branch at Tongji University. Her research interests include optimization, grid computing, service-oriented computing and the theory and application of Petri nets.



Chen Xie is a Ph.D. candidate in Tongji University. Her research interests include intelligent system, pattern recognition and cognitive neuroscience.

Ying Tan is a full professor, advisor of Ph.D. candidates of the Key Laboratory of Machine Perception (Peking University), Ministry of Education, and with Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Beijing, China, and he is the director of Computational Intelligence Laboratory (CIL) of Peking University. He has authored or coauthored more than 150 academic papers in refereed journals and conference proceedings and several books and book chapters. His current research interests include computational intelligence, neural networks, AIS, pattern recognition, machine learning theory and algorithms, swarm intelligence, intelligent information processing as well as their applications in computer security.

Zheng Tang received the B.S. degree from Zhejiang University, Zhejiang, China in 1982 and an M.S. degree and a D.E. degree from Tshinghua University, Beijing, China in 1984 and 1988, respectively. From 1988 to 1989, he was an instructor in the institute of microelectronics at Tshinghua University. From 1990 to 1999, he was an associate professor in the department of electrical and electronic engineering, Miyazaki University, Miyazaki, Japan. In 2000, he joined University of Toyama, Toyama, Japan as a professor of the department of intellectual information systems. Now he is also a distinguished course professor of the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji Uiniversity, Shanghai, China. His current research interests include intellectual information technology, neural networks, and optimizations.