2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

An Adaptive PSO Based on Motivation Mechanism and Acceleration Restraint Operator

Jiangshao Gu and Xuanhua Shi
School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China
{jsgu, xhshi}@hust.edu.cn

***Abstract*—To obtain precise solutions in optimization problems and decrease the risk of being trapped in local optima, researchers have studied on various improved particle swarm optimizations (PSO) and made a series of achievements. However, these methods focus on artificially altering the physical rules of motion, rather than strengthening the individual self-learning and adjustment during the optimization process, which is the original motive of the swarm-based evolutionary algorithms. In this paper, we propose a fresh self-adaptive variant, MMARO-PSO, which employs motivation mechanism to simulate the behavior of intelligent organisms more vividly. We manage to simplify the update formulas and give each term a definite bio-psychic sense. Furthermore, we introduce a vectorized operator to restrain particle's acceleration, instead of the inertia weight parameter in conventional methods. Large number of experiments were conducted and the results illustrate that these innovations make the technique perform more consistently to find a better balance between global exploration and local exploitation, compared with the existing versions, e.g. SPSO, e1-PSO, ARFPSO, and (k,l)PSO.**

Keywords*—acceleration restraint operator; adaptive; motivation mechanism; optimization problems; particle swarm optimization

I. INTRODUCTION

Particle swarm optimization (PSO) introduced by Kennedy and Eberhart in 1995 [6], [11] provides a meta-heuristic search technique to solve binary-parameter and real-parameter optimization problems. It was inspired by the social behavior in natural world, for instance, bird flocking and fish schooling. The abstract particle in this method contains two main properties, position and velocity, to simulate the inhabiting or foraging processes. During the run, the best or near-optimal solutions of a problem will be spontaneously exhumed. On account of the superior rapidity and validity relative to other deterministic or stochastic methods, PSO and its variants have been applied in masses of engineering fields.

Due to the inherent characteristics, PSO is likely to fall into local optima during the diversity-losing procedure, especially in high-dimensional or constrained conditions, which limits its application prospects. In the previous studies, researchers usually concentrated on discovering more high-effective update formulas for time-varying parameters, from where a lot of complicated non-linear equations were developed [1], [8], [13]. In another line of research, local versions of PSO that construct some topological structures to define the neighborhood of a single particle, showed outstanding performance in many scenarios because it slowed down the flow of global-best information [9], [12]. It is worth noting that these modifications focus on artificially altering the physical rules of neighborhood searching, which emblems that we attempt to adjust the law of nature to our wishes, rather than promote the level of individual intelligence. We have reasonable grounding to believe that several finite improvements in this way lead to prominent performance promotion according to Darwin's theory of evolution, thus the related work of in-depth study has a promising future.

The purpose of the paper is to introduce an improved adaptive PSO based on motivation mechanism and acceleration restraint operator (MMARO-PSO) that alters the standard version of PSO in consideration of the reasons above. Firstly, we propose a new model substituting motivation factors which can be tuned referring to previous experience for the coefficients in traditional update formulas. Using heuristic methods to calculate the factors' update volume and the amount of attenuation, a preliminary version called MM-PSO takes shape. Secondly, we explain the reason why inertia weight decreasing method provides better dynamic balance, and then we provide a more physically meaningful technique, a location-based acceleration restraint operator, leading to the final version MMARO-PSO. Compared with existing variants using diverse methods such as SPSO, e1-PSO, ARFPSO, and (k,l)PSO, experimental results demonstrate MMARO-PSO has more consistent and remarkable optimization capacity.

The remainder of the paper is organized as follows: Section II briefly looks back at historical developments and representative variants of PSO. Section III describes the considerations and implementations of algorithms we propose in detail. The experiment setup, parameter selection and comparison with other competitors will be displayed in Section IV. And Section V concluding the contributions of this paper meanwhile outlining the future work follows.

II. BACKGROUND

In this section, we first formulate the standard version of PSO (SPSO) and then introduce some of its variants.

A. Standard Particle Swarm Optimization

PSO is a meta-heuristic swarm-based search algorithm where a group of N particles standing for candidate solutions fly through a D-dimensional search space according to several update formulas in order to find the optimal solution. Every particle has a position given by a vector $\overrightarrow{X_i}(t)=(X_{i,1},X_{i,2},...,X_{i,D})$ and a velocity given by $\overrightarrow{V_i}(t)=(V_{i,1},V_{i,2},...,V_{i,D})$, where i means the i-th particle. According to a position vector we can calculate the value of objective function $f(\overrightarrow{X_i}(t))$, which is also regarded as the fitness value of the particle at current time. The particle has a tendency to keep track of the fittest position it has gone through, usually called personal best ($\overrightarrow{pbest_i}(t)$), and the best solution the swarm have visited so far, usually called global best ($\overrightarrow{gbest(t)}$) (The corresponding term in local version is called local best, $\overrightarrow{lbest(t)}$.), which is formulated in (1).

$$\overrightarrow{gbest}(t)=\{\overrightarrow{pbest_i}(t)\mid \text{maximize} f(\overrightarrow{pbest_i}(t)), i\in[1,N]\} \quad (1)$$

In each time step t, every particle adjusts its position and velocity according to

$$\begin{aligned}V_{ij}(t)=\omega V_{ij}(t-1)&+c_1r_{1j}(t)(pbest_{ij}(t-1)-X_{ij}(t-1))\\&+c_2r_{2j}(t)(gbest_j(t-1)-X_{ij}(t-1))\end{aligned} \quad (2)$$

$$X_{ij}(t)=X_{ij}(t-1)+V_{ij}(t) \quad (3)$$

where $V_{ij}(t)$ stands for i-th particle's velocity in dimension j at time step t, correspondingly $X_{ij}(t)$ stands for i-th particle's position in dimension j at time step t. Parameters c_1 and c_2 are acceleration coefficients to scale the influence of historical memory, which are originally defined as constants. Random real-number parameters $r_{1j}(t)$ and $r_{2j}(t)$ are distributed uniformly in the range [0, 1.0], i.e. $r_{1j}(t),r_{2j}(t)\sim U(0,1)$, in order to introduce stochastic factors. In general, we comprehend the first term as the cognitive part in that it represents the personal experience, while the second term as the social part in that it represents the collective memory. ω is the inertia weight invented by Shi and Eberhart [15] to keep a balance between local and global search characters, usually range [0,1.0], and was expanded to a time-varying parameter in the following researches. It is empirically confirmed that an inertia weight ω declining from 0.9 to 0.4 regularly provides satisfying performance.

In practice, we usually define the domain of the function optimized in case of losing actual meaning, naturally leading to limit the particle's location. For similar consideration, velocity is constrained in each iteration to prevent it from uncontrolled explosion, namely

$$X_{ij}(t)=\begin{cases}X_{ij}^{'}(t), & X_j^{\min}\le X_{ij}^{'}(t)\le X_j^{\max}\\ X_j^{\min}, & X_{ij}^{'}(t)<X_j^{\min}\\ X_j^{\max}, & X_j^{\max}<X_{ij}^{'}(t)\end{cases} \quad (4)$$

$$V_{ij}(t)=\begin{cases}V_{ij}^{'}(t), & V_j^{\min}\le V_{ij}^{'}(t)\le V_j^{\max}\\ V_j^{\min}, & V_{ij}^{'}(t)<V_j^{\min}\\ V_j^{\max}, & V_j^{\max}<V_{ij}^{'}(t)\end{cases} \quad (5)$$

where $X_j^{\min}$, $X_j^{\max}$, $V_j^{\min}$, $V_j^{\max}$ are normally defined as constants at the same order of magnitude empirically. $X_{ij}^{'}(t)$ and $V_{ij}^{'}(t)$ are results calculated with (2) and (3), respectively.

There are two simplified PSO models which remove one of acceleration coefficients created by Kennedy [10]. In the case $c_1\equiv 0$, PSO becomes the social-only model (sPSO). Relatively in the case $c_2\equiv 0$, PSO becomes the cognition-only model (cPSO) shown in

$$V_{ij}(t)=\omega V_{ij}(t-1)+c_2r_{2j}(t)(gbest_j(t-1)-X_{ij}(t-1)) \quad , for\ sPSO \quad (6)$$

$$V_{ij}(t)=\omega V_{ij}(t-1)+c_1r_{1j}(t)(pbest_{ij}(t-1)-X_{ij}(t-1)) \quad , for\ cPSO \quad (7)$$

Although in some specific circumstances the simplified versions outperform SPSO, they can't work consistently in the majority time. They play as choices of behaviors for heterogeneous PSO (HPSO) in recent studies [7], [14].

B. Particle Swarm Optimization Variants

There have been plenty of retrofitting versions proposed in last two decades since PSO algorithm was introduced. Here we choose some representative ones as opponents to evaluate the improvements of our methods.

Chen *et al*. [3] created 2 sets of nonlinearity time-varying formulas putting natural exponential function into use. In many experimental environments the one called e1-PSO performed better than the other one, and it performed competitively compared with a series of similar formulas according to [2]. It is on behalf of a classic idea to lessen inertia weight with the iterative process.

Chen and Liu [4] developed an improved particle swarm optimization based on adaptive rejection factor (ARFPSO), which provides an adaptive control factor computed by average distance and maximal distance amongst swarm to increase the ability to escape from local optima. With the stochastic mechanism, ARFPSO can avoid premature phenomenon validly on account of controlling the distribution of personal best positions $\overrightarrow{pbest_i}(t)$ on the basis of global best position $\overrightarrow{gbest(t)}$. Test results find evidences of its superiority especially on complex multimodal problems.

Zhou [17] invented a novel method called (k,l)PSO to make inertia weight varies from an individual to another. In classic algorithm framework, the inertia weight is a constant for different particles, which does not meet the need of behavior diversity for the particles at different positions. In this method, the best k particles are selected and the global best is generated by roulette method based on their fitness value. The inertia weight of each particle adjusts to the distance between itself and the global best particle. The effectiveness of (k,l)PSO is confirmed by some experiments.

In order to observe the effect of our innovations specifically, we implement the algorithms we propose based on the global version of PSO, and these variants are also formed on the basis of global version. Their different starting points make them suitable as the performance references for MMARO-PSO. Note that the heuristic ideas in our method have good portability and we can guarantee that they will bring significant performance gains in local versions as well.

III. Proposed PSO Variant

Different from the existing variants, there are two obvious features in MMARO-PSO. Firstly, motivation mechanism is introduced to reflect the individual psychological situation on selecting the most beneficial action trend. Additionally, we provide a vectorized operator to give more intuitive and precise control on the rate of particle's travel direction change.

A. Motivation Mechanism

In SPSO, $\overrightarrow{pbest_i}(t)$ and $\overrightarrow{gbest}(t)$ are involved in velocity update formula as optimization guidance, whose influence is decided by acceleration coefficients c_1 and c_2. Obviously this model is not comprehensive adequately to imitate the organisms' behavior, so there is considerable scope for enhancements. As to intelligent lives, more frequent and more remarkable fitness updates will arouse greater enthusiasm to explore the corresponding region in the short-term, and the attraction will subside gradually in the long-term when the area has been fully developed. If the attraction is low enough, individual will even fly away from the area to look for potential better solutions. This mechanism can not only accelerate the partial convergences, but also enhance the diversity of global search. We manage to improve the technique without the cost of complexity.

Enthusiasm losing is analogous to forgetting process wherefore we can consult the researches conducted by Ebbinghaus [5] on forgetting curve to design our model. The following formula can roughly describe it.

$$R(t) = e^{-\frac{t}{S}} \tag{8}$$

where R is memory retention, S is the relative strength of memory, and t is time. Discretize it as

$$R(t) = e^{-\frac{1}{S}} \times e^{-\frac{t-1}{S}} = e^{-\frac{1}{S}} \times R(t-1) = a \times R(t-1) \tag{9}$$

where we can conclude that the memory attenuates proportionally with time. Thereby we propose a new model introducing motivation factors which is updated with the ratio of recent optimization amount to reference quantity, to depict the psychological state of the particles and measure the effect of different heuristic information, given by

$$\begin{aligned}\overrightarrow{V_i}(t) = \delta[k_i(t), \overrightarrow{V_i}(t-1), c_{i,0}(t)\overrightarrow{V_i}(t-1) \\ + c_{i,1}(t) r_{1,j}(t)(\overrightarrow{pBest_i}(t-1) - \overrightarrow{X_i}(t-1)) \\ + c_{i,2}(t) r_{2,j}(t)(\overrightarrow{gBest}(t-1) - \overrightarrow{X_i}(t-1))]\end{aligned} \tag{10}$$

$$c_{i,j}(t) = c_j^{\min} + (c_j^{\max} - c_j^{\min}) \times MF_{i,j}(t),\ j = 0, 1, 2 \tag{11}$$

$$\begin{aligned}MF_{i,j}(t) = \min(1.0,\ att \times MF_{i,j}(t-1) + delta_{i,j}(t)) \\ ,\ j = 0, 1, 2\end{aligned} \tag{12}$$

$$delta_{i,0}(t) = \frac{\max(0,\ f(\overrightarrow{X_i}(t-1)) - f(\overrightarrow{X_i}(t)))}{\max\_update_i(t)} \tag{13}$$

$$delta_{i,1}(t) = \frac{\max(0,\ f(\overrightarrow{pBest_i}(t-1)) - f(\overrightarrow{pBest_i}(t)))}{\sqrt{\max\_update_i(t) \times sum\_update_i(t)}} \tag{14}$$

$$delta_{i,2}(t) = \frac{\max(0,\ f(\overrightarrow{gBest}(t-1)) - f(\overrightarrow{gBest}(t)))}{\sqrt[3]{\max\_update_i(t) \times sum\_update_i(t) \times \max\_sum\_update(t)}} \tag{15}$$

$$\begin{aligned}\max\_update_i(t) = \max\{\max(0,\ f(\overrightarrow{X_i}(\tau-1)) - f(\overrightarrow{X_i}(\tau))) \\ ,\ \tau \in [1, t]\}\end{aligned} \tag{16}$$

$$\begin{aligned}sum\_update_i(t) = \max\{\max(0,\ f(\overrightarrow{X_0}(\tau-1)) - f(\overrightarrow{X_i}(\tau))) \\ ,\ \tau \in [1, t]\}\end{aligned} \tag{17}$$

$$\begin{aligned}\max\_sum\_update(t) = \max\{sum\_update_i(t) \\ ,\ i \in [1, N]\}\end{aligned} \tag{18}$$

where $\delta[k, \vec{V}, \vec{V}']$ is an operator acting on the velocity vector aimed at controlling accelerated degree which will be described in detail in the next part. The inertia weight ω is relaxed whereas new coefficients $c_{i,0}(t)$, $c_{i,1}(t)$, and $c_{i,2}(t)$ determinate the propensity of different behaviors to be adopted, which change linearly with motivation factors $MF_{i,j}(t)$ in the predefined ranges. $MF_{i,j}(t)$ are reference quantities in the range [0, 1.0], which attenuate by the scale factor att and increase by quantities of stimulus $delta_{i,j}(t)$ in each time step. $delta_{i,j}(t)$ are calculated according to the idea to take the ratio of real-time update amount to corresponding historical reference. More specifically, $\max\_update_i(t)$ equals the maximum one-time update quantity of the i-th particle from beginning to time step t. $sum\_update_i(t)$ means the cumulative optimization amount of the i-th particle, and $\max\_sum\_update(t)$ is the global maximum of $sum\_update_i(t)$.

Particle tends to keep current gliding state in a downhill section, which is considered in (13). Continuous updates will increase $MF_{i,0}(t)$ dramatically enhancing the opportunity to slide into the lowest point rapidly, which is usually ignored in the existing studies. Formula (14) and (15) express that the update of $\overrightarrow{pbest_i}(t)$ and $\overrightarrow{gbest}(t)$ will draw particle's attention to exploit the local and global best regions, respectively. Formula (14) seems to be more easy-to-comprehend if the denominator is simplified to $sum\_update_i(t)$. However, we take notice that the proportional definition is likely to neglect the profit of some absolute updates which appear insignificant in comparison with $sum\_update_i(t)$, so that we choose the geometric mean value of it and single update reference $\max\_update_i(t)$. Formula (15) can be understood in the same way noting that the involvement of individual-related terms strengthen the diversity and robustness of the iteration strategy.

We use formulas (10)-(18) to substitute for original update rules, leading to a promoted elementary version called MM-PSO. The purpose is to provide more practice options and meanwhile direct the parameter tuning more specifically.

B. Acceleration Restraint Operator

Previous researches show that an inertia weight ω decreasing with the evolution process gives properer balance during the run than the constant version. It's widely believed that larger ω result in stronger ability of exploration in large range while smaller ω is more suitable for exploitation near the global minimum. Now we provide some suggestive explanations.

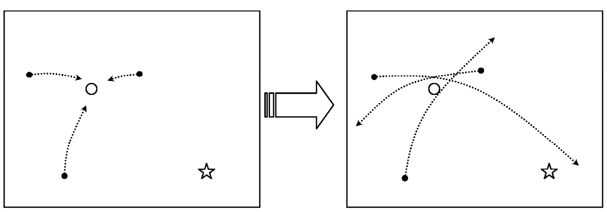


Fig. 1. Evolution process with a larger inertia weight.

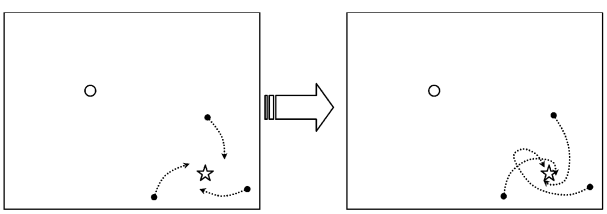


Fig. 2. Evolution process with a smaller inertia weight.

In Fig. 1 and 2, the hollow circles stand for local optima while the global minima are expressed as the stars. The dotted lines show the flight paths of the swarm. As Fig. 1 shows, the velocity isn't changed obviously in each time step with a large ω, so that a particle's flying track is similar to a straight line. Even though the whole swarm gather in the vicinity of local optimum, particles tend to be uniformly distributed in a wide search space after some generations because of the randomness of speed direction. Oppositely in Fig. 2, particle's forward orientation will turning sensitively with the attractiveness of $\overrightarrow{pbest_i}(t)$ or $\overrightarrow{gbest}(t)$, making the swarm converge to global optimum speedily with a greater curvature.

Based on the above we can find that a crucial effect of ω is to control the rate of heading direction change, thereby adjusts the exploration/exploitation balance. However the update regulations have at least two inappropriate points:

- There is no restriction on the norm of acceleration, meaning that the particles can obtain infinite momentum, which does not meet the real dynamical system that particles are in.
- There is no coupling between the residual amount of velocity and the cognition-learning part or the social-learning part by tuning inertia weight ω to control the balance. This weakens the unity of program, and perturbs the probabilistic expectation of velocity even worse.

Based on the discussions above, we propose an acceleration restraint operator given by

$$\delta[k,\vec{V},\vec{V}'] = \vec{V} + \min(1.0, k\times\frac{|\vec{V}|}{|\vec{V}'-\vec{V}|})(\vec{V}'-\vec{V}) \tag{19}$$

$$k_i(t) = k^{\min} + (k^{\max}-k^{\min})\times\frac{|\overrightarrow{X_i}(t)-\overrightarrow{centre}(t)|}{\max D} \tag{20}$$

$$\overrightarrow{centre}(t) = \frac{\sum_{i=1}^{N}\overrightarrow{pBest_i}(t)}{N} \tag{21}$$

$$\max D = |\overrightarrow{X^{\max}} - \overrightarrow{X^{\min}}| \tag{22}$$

where $k_i(t)$ is the acceleration restraint factor in (10), which changes linearly with the distance between single particle and the mean-optimal position $\overrightarrow{centre}(t)$, in current time step. $\overrightarrow{X^{\max}} = (X_1^{\max}, X_2^{\max}, ..., X_D^{\max})$ and $\overrightarrow{X^{\min}} = (X_1^{\min}, X_2^{\min}, ..., X_D^{\min})$ specify the domain of feasible solutions, and $\max D$ pre-calculated gives the maximum distance amongst particles. The intuitive sense of these formulas is to limit the change in velocity within $k_i(t)$ times of $\vec{V}$, to control the curvature adaptively at different stages of the search.

Rather than the time-varying strategy, we adopt the location-based strategy referring to quantum-behaved particle swarm optimization (QPSO) [16], another PSO's variant. In QPSO, researchers adjust a particle's wave function according to the distance between it and the average centre of $\overrightarrow{pbest_i}(t)$, which leading to waiting effect which enhances the cooperation between particles. In our framework of classical physics, acceleration restraint factor $k_i(t)$ changes linearly with distance similarly defined. The waiting effect can be shown in

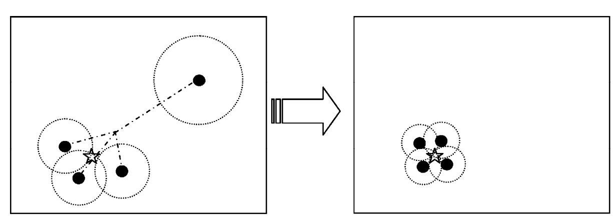


Fig. 3. Illustration of waiting effect in MMARO-PSO.

where the dotted circles indicate the particles' search ranges roughly. It is easy to find that as a result of the withdrawn particles, the centre is dragged away from the population around the optimum, making those particles search in a larger range, which seems that they are waiting for the distant brothers. With the acceleration restraint operator, we propose the final version of our algorithm which is shown in TABLE I.

TABLE I
PSEUDOCODE FOR THE ALGORITHM PROPOSED

MMARO-PSO Algorithm
1. For each particle (i=1, 2, . . ., N):
1.1. Initialize position $\overrightarrow{X_i}(0)$ and velocity $\overrightarrow{V_i}(0)$
1.2. Evaluate fitness value $f(\overrightarrow{X_i}(0))$
1.3. Set $\overrightarrow{pbest_i}(0)=\overrightarrow{X_i}(0)$ and update $\overrightarrow{gbest}(0)$ with (1)
1.4. Set $MF_{i,j}(0)=1.0$
2. For each particle (i=1, 2, . . ., N):
2.1. Calculate $\overrightarrow{V_i}(t)$ with motivation factors and acceleration restraint operator using (10)-(11) and (19)-(22)
2.2. Get new $\overrightarrow{X_i}(t)$ and evaluate fitness $f(\overrightarrow{X_i}(t))$
2.3. Update $\overrightarrow{pbest_i}(t)$ and $\overrightarrow{gbest}(t)$ if $f(\overrightarrow{X_i}(t))$ is better than before
2.4. Update $MF_{i,j}(t)$ using (12)-(18)
3. Go to 2 until the iteration limit or tolerance error is met

IV. EXPERIMENTS AND RESULTS

In this section, we first make a brief description of benchmark functions, then introduce the methodology and parameter settings of MMARO-PSO, and finally make an intensive comparison with some other competitors. Each algorithm was executed on functions in 5, 10, 20 dimensions for $D\times100$ iterations and repeated 10 times independently, with fixed population size 30. The remaining parameters of the competitors were set to the recommended values in [3], [4], [17].

The experimental environment was a laptop computer with a 2.4GHz Intel Pentium dual-core processor, 2.0GB of RAM, running Microsoft Windows XP Professional. The simulation programs were developed using Dev-C++ 4.9.9.2.

A. Benchmark Functions

The experiment setup was constructed with six unimodal or multimodal benchmark functions widely adopted to investigate the performance of PSO for years. The functional properties of them can be easily found in relevant literatures so we omit this part. We shifted the search ranges to avoid the optimum positions concentrate in the middle. The detailed expressions are listed as follows.

Ackley(f_1):

$$f(\vec{X})=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D}X_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{D}\cos(2\pi X_i)\right)+20+e$$

Search and Initialization range: [-20, 40]

The global minimum: $\overrightarrow{X^*}=(0, 0, ..., 0)$, $f(\overrightarrow{X^*})=0$

Griewank(f_2): $$f(\vec{X})=\frac{1}{4000}\left(\sqrt{\sum_{i=1}^{D}X_i^2}\right)-\left(\prod_{i=1}^{D}\cos(\frac{X_i}{\sqrt{i}})\right)+1$$

Search and Initialization range: [-600, 400]

The global minimum: $\overrightarrow{X^*}=(0, 0, ..., 0)$, $f(\overrightarrow{X^*})=0$

Rastrigin(f_3): $$f(\vec{X})=\sum_{i=1}^{D}\left(X_i^2-10\cos(2\pi X_i)+10\right)$$

Search and Initialization range: [$-\pi$, $\frac{\pi}{2}$]

The global minimum: $\overrightarrow{X^*}=(0, 0, ..., 0)$, $f(\overrightarrow{X^*})=0$

Rosenbrock(f_4): $$f(\vec{X})=\sum_{i=1}^{D-1}\left(100\left(X_{i+1}-X_i^2\right)^2+(X_i-1)^2\right)$$

Search and Initialization range: [-25, 40]

The global minimum: $\overrightarrow{X^*}=(1, 1, ..., 1)$, $f(\overrightarrow{X^*})=0$

Schwefel(f_5): $$f(\vec{X})=418.9829\text{D}-\sum_{i=1}^{D}X_i\sin\left(\sqrt{|X_i|}\right)$$

Search and Initialization range: [-500, 500]

The global minimum:

$\overrightarrow{X^*}=(420.9687, 420.9687, ..., 420.9687)$, $f(\overrightarrow{X^*})=0$

Sphere(f_6): $$f(\vec{X})=\sum_{i=1}^{D}\left(X_i^2\right)$$

Search and Initialization range: [-200, 150]

The global minimum: $\overrightarrow{X^*}=(0, 0, ..., 0)$, $f(\overrightarrow{X^*})=0$

B. Parameter Tuning

Progressively, we first tuned the parameters of MM-PSO to ensure that motivation factor working in the best state, and second adjusted the acceleration restraint based on the settings in the prior step.

In MM-PSO, the major parameters include $c_j^{\min}$ and $c_j^{\max}$ (j=0,1,2), as well as the attenuation coefficient att. The motivation coefficients are confined to the interval [$c_j^{\min}$, $c_j^{\max}$], while att determines the attraction fading rate which react individual psychological state. $c_{i,1-2}^{\min}$ and $c_{i,1-2}^{\max}$ can refer to the tuning experience of acceleration coefficients in SPSO, generally believed that 2.0 may result in nice performance. We recommend $c_{i,0}^{\min}$ and $c_{i,0}^{\max}$ set around 1.0 so that particles can realize acceleration and deceleration according to the changeable situations. Some typical settings of parameters and the corresponding results are listed in TABLE II.

We can sum that $c_{i,0}^{\min}$=0.8, $c_{i,0}^{\max}$=1.2, $c_{i,1-2}^{\min}$=1.6, $c_{i,1-2}^{\max}$=2.4, and att=0.9 provide satisfactory performance in most testing functions, so we use these settings as default.

TABLE II
BEST PERFORMANCE OF SEVERAL PARAMETER SETTINGS

$c_{i,0}^{\min}$	$c_{i,0}^{\max}$	$c_{i,1-2}^{\min}$	$c_{i,1-2}^{\max}$	*att*	Benchmark Functions (10D)			
					f_2	f_3	f_4	f_5
0.4	0.8	0.8	1.6	0.8	2.78E-01	6.27E+00	2.10E+01	9.58E+02
0.4	0.8	0.8	1.6	0.9	1.18E-01	1.00E+01	1.72E+02	8.32E+02
0.4	0.8	0.8	1.6	1	0.00E+00	6.96E+00	6.20E+00	5.92E+02
0.8	1.2	1.6	2.4	0.8	0.00E+00	1.86E+01	4.31E+00	5.22E+02
0.8	1.2	1.6	2.4	0.9	**0.00E+00**	**0.00E+00**	**6.70E-10**	**7.99E+02**
0.8	1.2	1.6	2.4	1	0.00E+00	0.00E+00	8.89E-07	9.73E+02
1.2	1.6	2.4	3.2	0.8	0.00E+00	1.36E+01	1.15E-03	1.65E+03
1.2	1.6	2.4	3.2	0.9	0.00E+00	0.00E+00	1.55E-03	1.46E+03
1.2	1.6	2.4	3.2	1	0.00E+00	0.00E+00	4.46E-03	1.91E+03

On the other hand, the acceleration restraint factor $k_i(t)$ is an important variable which directly controls the balance of search strategy. $k^{\min}$ should be greater than 0 so that particles always have the ability to change flying direction. $k^{\max}$ reflects the restrictive relation between the convergence rate and search diversity. We enumerate some simulation results in TABLE III from which we believe that $k^{\min}$=0.1 and $k^{\max}$=10.0 may provide preferable performance most often.

TABLE III
BEST PERFORMANCE OF SEVERAL PARAMETER SETTINGS

$k^{\min}$	$k^{\max}$	Benchmark Functions (10D)			
		f_2	f_3	f_4	f_5
0.1	1	6.29E-01	9.00E+00	1.45E+02	5.04E+02
0.1	10	**0.00E+00**	**0.00E+00**	**1.34E-01**	**2.80E+02**
0.1	100	0.00E+00	1.01E+01	1.21E-07	5.80E+02
1	1	1.77E-01	8.66E+00	5.82E+01	7.66E+02
1	10	0.00E+00	2.10E+01	8.33E-05	8.85E+02
1	100	0.00E+00	2.18E+00	2.49E-07	6.58E+02
10	1	0.00E+00	1.69E+01	1.82E-08	5.35E+02
10	10	0.00E+00	1.35E+01	4.38E-06	9.01E+02
10	100	0.00E+00	3.72E+00	4.14E-06	6.04E+02

However, it needs to be emphasized that the parameters we adopted perform well in most instances empirically, whereas the niche-targeting fine tuning should not be neglected. We have found that some calibrations for specific examples enhance the results by several orders of magnitude.

C. Performance Comparison

To compare the capabilities of MM-PSO and MMARO-PSO with other competitors, we ran each program on benchmark functions in different dimensions. The detailed performance indexes are enumerated in Table IV, V, and VI.

We can easily find that MM-PSO and MMARO-PSO didn't significantly improve the running time compared to SPSO in the same scale, which is a huge advantage over ARFPSO and (k,l)PSO. Our algorithms outperformed other variants in most of the time, especially on f_2, f_4, and f_6, where our methods achieved or come extremely close to the optimum. It can also be seen that e1-PSO and (k,l)PSO performed not ideally in the search ranges we designed with an off-center optimal value. Note that on f_5-*5D*, MM-PSO was trapped into local optimization whereas MMARO-PSO avoided the situation, which shows the effect of acceleration restraint operator to enhance the program's stability and robustness. In addition, the consistent results confirm the extensibility for MM-PSO and MMARO-PSO in high-dimensional occasions to some extent.

Fig. 4 and 5 give the convergence plots for the convenience of observing the optimization processes. Note that since the error values reached zero quickly in some instances, a small amount 10^{-15} was added to each item so as to take the logarithms. As can be seen, MM-PSO and MMARO-PSO were not the fastest way to convergence in the early times. However, when other approaches stagnated gradually, the methods we propose not just had the ability to continue to optimize, more surprisingly, they usually reached the exact optimal value expeditiously in several iterations, which is believed due to the motivation mechanism. We can also find that the use of acceleration restraint operator reduces the rate of convergence, which embodies the contradiction of speed and low risk.

In addition, we draw the motivation curves to display the mental process of a single particle in Fig. 6 and 7. In the early stage, motivation factors evolved at a high level to guide the movement of the particles, then reduced gradually making the swarm to stabilize. The rises of curves meant that the particle approached an under-explored space with a potential better optimum position, and induced the particle to exploit the region in turn which met our expectations.

V. CONCLUSION AND FUTURE WORK

This paper describes an improved adaptive PSO based on motivation mechanism and acceleration restraint operator (MMARO-PSO), which introduces motivation factors to simulate psychological behavior of social organisms and constrain the acceleration of particles adaptively using a vectorized operator. The experimental results on several typical benchmark functions show that the proposed algorithms give better performance on altering the exploration/exploitation balance compared with existing variants of PSO.

Future work focuses on modifying the update formulas of motivation factors to perfect the heuristic mechanism, making targeted extensions so as to deal with the constrained and multi-objective conditions, and applying the technique to various practical fields.

REFERENCES

[1] W. Al-Hassan, M. B. Fayek, and S. I. Shaheen, "Psosa: An optimized particle swarm technique for solving the urban planning problem," in *the 2006 International Conference on Computer Engineering and Systems*, pp. 401–405. IEEE, 2007.

[2] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," in *Proc. of Third World Congress on Nature and Biologically Inspired Computing*, 2011, pp. 640–647.

[3] G. Chen, X. Huang, J. Jia, and Z. Min, "Natural exponential inertia weight strategy in particle swarm optimization," in *the Sixth World Congress on Intelligent Control and Automation*, IEEE, 2006, vol. 1, pp. 3672–3675.

[4] M. Chen and Y. M. Liu, "Improved particle swarm optimization based on adaptive rejection factor," in *Journal of Computer Applications*, 2013, 33(8):2269–2272 (in Chinese with English abstract).

[5] H. Ebbinghaus, *Memory: A contribution to experimental psychology*. New York: Dover, 1964 (Original work published 1885; translated 1913).

[6] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.

[7] A. P. Engelbrecht, "Heterogeneous particle swarm optimization," in *Proc. of 7th International Conference on Swarm Intelligence*, 2010, pp. 191–202.

[8] Y. Feng, G. F. Teng, A. X. Wang, and Y. M. Yao, "Chaotic inertia weight in particle swarm optimization," in *Innovative Computing, Information and Control, 2007. ICICIC'07. Second International Conference on*, pp. 475. IEEE, 2008.

[9] C. M. Fernandes, A. C. Rosa, J. L. Laredo, C. Cotta, and J. J. Merelo, "A study on time-varying partially connected topologies for the particle swarm," in *Proc. of the IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 2450–2456.

[10] J. Kennedy, "The particle swarm: social adaptation of knowledge," in *Proc. of the IEEE International Conference on Evolutionary Computation*, 1997, pp. 303–308.

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of the IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[12] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. of the IEEE Congress on Evolutionary Computation (CEC)*, 2002, pp. 1671–1676.

[13] R. F. Malik, T. A. Rahman, S. Z. M. Hashim, and R. Ngah, "New particle swarm optimizer with sigmoid increasing inertia weight," in *International Journal of Computer Science and Security (IJCSS)*, 2007, 1(2):35.

[14] F. V. Nepomuceno and A. P. Engelbrecht, "A self-adaptive heterogeneous PSO for real-parameter optimization," in *Proc. of the IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 361–368.

[15] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. of the IEEE International Conference on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 1998, pp. 69–73.

[16] J. Sun, W. Xu, and B. Feng, "A global search strategy of quantum-behaved particle swarm optimization," in *the IEEE conference on Cybernetics and Intelligent Systems*, 2004, pp. 111–116.

[17] M. Zhou, "Distance-based adaptive k particle swarm optimization," in *Computer Engineering and Applications*, 2011, 47(15):43–45 (in Chinese with English abstract).

TABLE IV
THE PERFORMANCE COMPARISON IN 5 DIMENSIONS

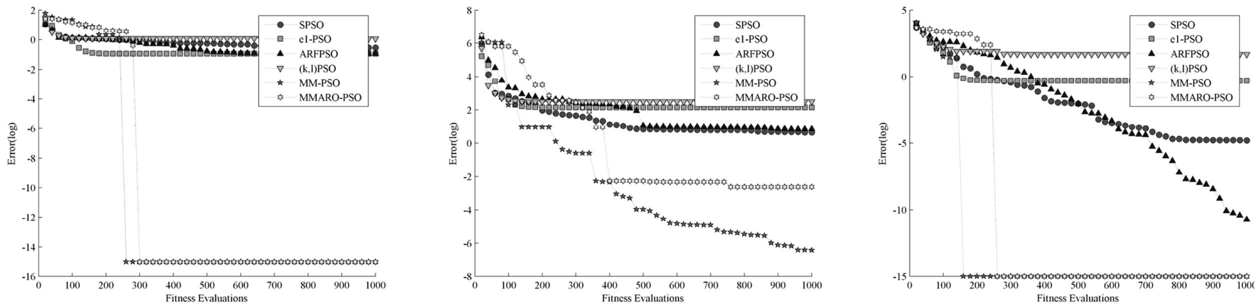
Algorithm	Comparison Index	Benchmark Functions (5D)						Total Time (secs)
		f_1	f_2	f_3	f_4	f_5	f_6	
SPSO	Best	5.96E-05	0.00E+00	9.95E-01	1.30E-04	1.18E+02	0.00E+00	3.265
	Mean	6.00E+00	1.16E-01	2.83E+00	8.81E+02	4.02E+02	1.17E-07	
	Std	9.12E+00	9.26E-02	2.11E+00	1.59E+03	1.78E+02	1.69E-07	
e1-PSO	Best	4.00E-15	3.94E-02	1.99E+00	0.00E+00	6.36E-05	2.28E-68	3.296
	Mean	4.32E+00	1.49E-01	3.88E+00	9.62E+01	4.02E+02	2.05E-62	
	Std	7.83E+00	6.93E-02	2.24E+00	2.00E+02	2.05E+02	6.14E-62	
ARFPSO	Best	4.44E-16	0.00E+00	0.00E+00	9.81E-07	6.36E-05	0.00E+00	7.187
	Mean	9.35E-05	1.47E-01	2.19E+00	5.67E+00	4.04E+02	1.95E-08	
	Std	1.37E-04	1.28E-01	2.17E+00	9.43E+00	1.62E+02	2.14E-08	
(k,l)PSO	Best	4.44E-16	0.00E+00	2.84E-07	3.26E-02	6.36E-05	0.00E+00	10.406
	Mean	3.95E+00	9.07E-02	2.29E+00	6.72E+03	3.09E+02	7.92E-04	
	Std	7.90E+00	1.09E-01	1.61E+00	1.86E+04	1.70E+02	2.33E-03	
MM-PSO	Best	4.44E-16	**0.00E+00**	0.00E+00	**3.64E-10**	2.95E+01	**0.00E+00**	3.656
	Mean	3.96E+00	**0.00E+00**	2.72E+00	**1.26E+02**	4.09E+02	**0.00E+00**	
	Std	7.92E+00	**0.00E+00**	2.99E+00	**3.76E+02**	1.79E+02	**0.00E+00**	
MMARO-PSO	Best	4.44E-16	**0.00E+00**	0.00E+00	**3.76E-06**	2.28E-03	**0.00E+00**	4.218
	Mean	7.15E+00	**0.00E+00**	3.51E+00	**4.91E+02**	3.03E+02	**0.00E+00**	
	Std	8.45E+00	**0.00E+00**	4.10E+00	**1.20E+03**	2.32E+02	**0.00E+00**	

TABLE V
THE PERFORMANCE COMPARISON IN 10 DIMENSIONS

Algorithm	Comparison Index	Benchmark Functions (10D)						Total Time (secs)
		f_1	f_2	f_3	f_4	f_5	f_6	
SPSO	Best	2.00E+01	9.20E-02	9.95E-01	4.94E+00	2.37E+02	4.27E-06	9.921
	Mean	2.00E+01	2.93E-01	1.53E+01	9.12E+02	7.22E+02	1.68E-04	
	Std	1.96E-03	2.14E-01	7.28E+00	1.57E+03	2.72E+02	1.64E-04	
e1-PSO	Best	2.00E+01	9.56E-02	5.97E+00	7.36E+00	7.71E+02	2.01E-02	10
	Mean	2.00E+01	3.98E-01	1.17E+01	1.27E+04	1.05E+03	1.15E+00	
	Std	1.37E-03	3.82E-01	3.63E+00	2.49E+04	2.03E+02	1.83E+00	
ARFPSO	Best	1.21E-06	1.16E-01	2.98E+00	1.33E-01	2.38E+02	1.58E-13	19.765
	Mean	1.26E+01	1.95E-01	1.31E+01	2.89E+01	7.29E+02	2.82E-11	
	Std	9.20E+00	5.55E-02	7.54E+00	7.93E+01	3.02E+02	4.17E-11	
(k,l)PSO	Best	2.00E+01	3.42E-01	8.20E+00	3.81E-02	5.75E+02	0.00E+00	35.046
	Mean	2.00E+01	8.56E-01	1.78E+01	1.68E+03	8.88E+02	8.35E+01	
	Std	3.36E-03	3.96E-01	7.20E+00	1.75E+03	2.88E+02	1.00E+02	
MM-PSO	Best	2.00E+01	**0.00E+00**	0.00E+00	**6.70E-10**	7.99E+02	**0.00E+00**	10.562
	Mean	2.00E+01	**0.00E+00**	2.82E+01	**6.28E+03**	1.34E+03	**0.00E+00**	
	Std	1.19E-02	**0.00E+00**	1.33E+01	**1.87E+04**	3.50E+02	**0.00E+00**	
MMARO-PSO	Best	2.00E+01	**0.00E+00**	0.00E+00	**1.34E-01**	2.80E+02	**0.00E+00**	12.234
	Mean	2.00E+01	**0.00E+00**	2.66E+01	**1.34E+04**	1.14E+03	**0.00E+00**	
	Std	7.55E-04	**0.00E+00**	1.21E+01	**2.46E+04**	3.62E+02	**0.00E+00**	

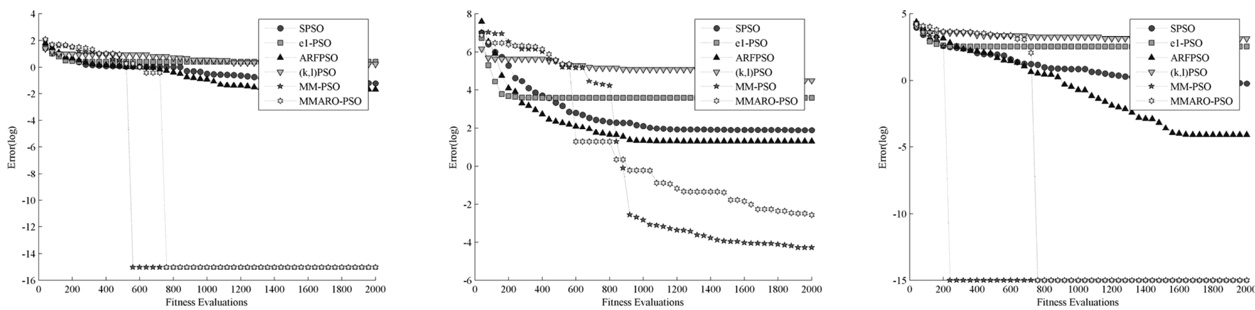
TABLE VI
THE PERFORMANCE COMPARISON IN 20 DIMENSIONS

Algorithm	Comparison Index	Benchmark Functions (20D)						Total Time (secs)
		f_1	f_2	f_3	f_4	f_5	f_6	
SPSO	Best	2.00E+01	1.46E-01	1.10E+01	2.33E+01	1.44E+03	4.05E-01	31.421
	Mean	2.00E+01	4.35E+00	3.72E+01	1.02E+03	2.62E+03	1.35E+00	
	Std	5.53E-08	1.20E+01	1.51E+01	1.58E+03	8.02E+02	9.54E-01	
e1-PSO	Best	2.00E+01	1.23E+00	2.78E+01	2.56E+03	2.26E+03	1.81E+02	31.656
	Mean	2.00E+01	6.54E+00	4.25E+01	5.66E+04	2.75E+03	5.49E+02	
	Std	6.76E-07	1.20E+01	8.50E+00	6.63E+04	4.23E+02	2.37E+02	
ARFPSO	Best	2.00E+01	1.01E-02	1.89E+01	9.09E+00	1.05E+03	1.93E-05	61.671
	Mean	2.00E+01	4.29E-02	3.76E+01	8.49E+02	1.80E+03	5.29E-03	
	Std	2.05E-04	2.81E-02	1.22E+01	1.33E+03	6.36E+02	7.22E-03	
(k,l)PSO	Best	2.00E+01	0.00E+00	1.30E+01	1.81E+03	1.37E+03	4.72E+02	151.312
	Mean	2.00E+01	1.15E+00	5.02E+01	3.44E+04	2.38E+03	3.77E+03	
	Std	3.52E-07	5.51E-01	2.73E+01	3.37E+04	7.12E+02	6.59E+03	
MM-PSO	Best	2.00E+01	**0.00E+00**	7.92E+01	**6.38E-09**	2.95E+03	**0.00E+00**	36.39
	Mean	2.00E+01	**2.02E+01**	1.10E+02	**1.15E+07**	3.50E+03	**8.16E+02**	
	Std	3.15E-06	**2.71E+01**	1.56E+01	**1.76E+07**	3.72E+02	**1.28E+03**	
MMARO-PSO	Best	2.00E+01	**0.00E+00**	5.27E+01	**4.94E-02**	2.67E+03	**0.00E+00**	38.75
	Mean	2.00E+01	**1.21E+01**	9.68E+01	**3.85E+06**	3.56E+03	**4.25E+02**	
	Std	1.21E-04	**1.85E+01**	2.19E+01	**1.15E+07**	7.79E+02	**8.52E+02**	



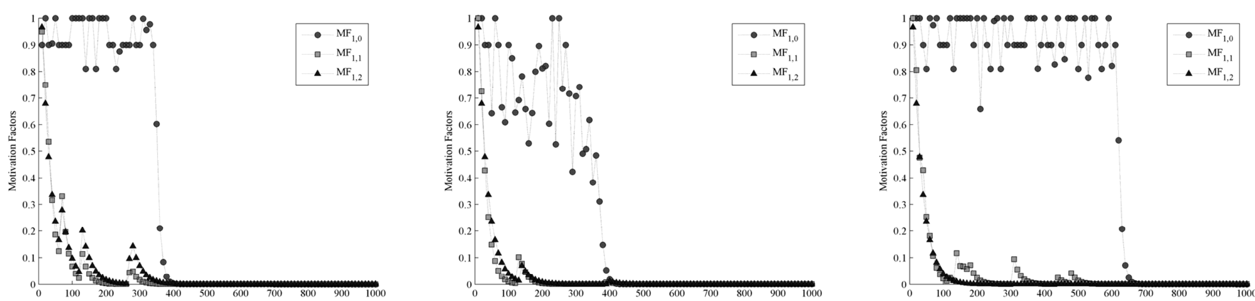
(a) f_2 (b) f_4 (c) f_6

Fig. 4. Convergence plots of all methods for the functions in 10D. Note that in our algorithms, the optimal values sometimes dropped suddenly reaching the exact optimal solutions, which indicates that our heuristic considerations worked.



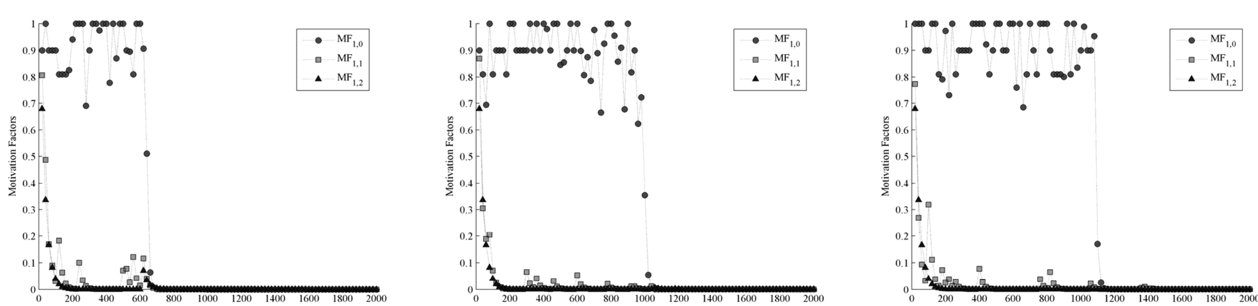
(a) f_2 (b) f_4 (c) f_6

Fig. 5. Convergence plots of all methods for the functions in 20D.



(a) f_2 (b) f_4 (c) f_6

Fig. 6. Motivation curves of MMARO-PSO for the functions in 10D. The undulations symbolize the alternations in the particle's mental status and decision-making perspective in the dynamic procedure.



(a) f_2 (b) f_4 (c) f_6

Fig. 7. Motivation curves of MMARO-PSO for the functions in 20D.