Journal of Electrical and Electronic Engineering
2017; 5(2): 68-73
http://www.sciencepublishinggroup.com/j/jeee
doi: 10.11648/j.jeee.20170502.17
ISSN: 2329-1613 (Print); ISSN: 2329-1605 (Online)

Improved Particle Swarm Optimization with Controllable Velocity-Updating Mode

Jiao Weidong[1, *], Huang Zhijing[1], Yan Gongbiao[2]

[1]School of Engineering, Zhejiang Normal University, Jinhua, China

[2]Department of Mechanical Engineering, Zhejiang University, Hangzhou, China

Email address:

jiaowd1970@zjnu.cn (Jiao Weidong), 1484660232@qq.com (Huang Zhijing), ygb_zju@163.com (Yan Gongbiao)

[*]Corresponding author

To cite this article:

Jiao Weidong, Huang Zhijing, Yan Gongbiao. Improved Particle Swarm Optimization with Controllable Velocity-Updating Mode. *Journal of Electrical and Electronic Engineering.* Vol. 5, No. 2, 2017, pp. 68-73. doi: 10.11648/j.jeee.20170502.17

Received: March 16, 2017; **Accepted**: April 7, 2017; **Published**: April 12, 2017

Abstract: At the late evolution stage of the basic particle swarm optimization (BPSO), convergence process starts to slow down and the best fitness particle fluctuates around the globally-optimal solution, which may give rise to decrease on convergence precision of the BPSO. Therefore, an improved algorithm for particle swarm optimization was proposed. The modified version of PSO uses a controllable velocity-updating mode to control velocity of evolved particles, which is expected to be useful for tuning the search for the globally-optimal solution. Optimization examples showed that the improved PSO is superior to the BPSO, on not only convergence precision but also computation expense.

Keywords: Particle Swarm Optimization (PSO), Controllable Velocity-Updating Mode, Velocity-Changing Track

1. Introduction

Particle swarm optimization (PSO), an analogy to the choreography of hunting flight of a flock of birds, is a kind of global optimization algorithm [1]. Empirically, a PSO optimizer can rapidly converge at the early evolution stage. But at the late evolution stage, its convergence slows down and the best-fitness particle may fluctuate around the globally-optimal solution, which can lead to decrease of convergence precision. To solve these problems, Shi and Eberhart proposed using a linearly decreased inertia weight w during velocity-updating [2]. In literature [3], a constriction factor was imposed on the velocity term in the position-updating model of the BPSO. Also, fuzzy rule was presented to modify the weight w dynamically, thus global parameters of the swarm optimizer were changed adaptively to find an appropriate balance between search efficiency and optimization precision [4]. In recent years, PSO has been applied to successfully solve optimization tasks in many fields [5-7]. Some improved PSO algorithms have also been developed. Ireneusz proposed a new approach for particle swarm optimization algorithm to self-control on the optimization process, by means of co-evolution or co-operation in natural environments [8]. Cheung *et al*. used supervisors to guide the particles to execute search tasks with different update rules for their position and velocity [9]. Tang and Fang made further improvement on the human-brain simulated particle swarm optimization. They proposed extended memory and new velocity choosing and updating strategies to give the moving direction to each particle more intelligently and help them avoid trapping into local optimum [10].

In this paper, several modes for velocity-updating were defined firstly. Some velocity-changing tracks for swarm evolution were then designed. An improved particle swarm optimizer was proposed. Finally, an optimization example was given to verify the proposed optimization algorithm.

2. The BPSO Algorithm and Some Modified Versions

In the BPSO, a population of particles is firstly initialized with random position and velocity. Then the globally-optimal solution of an given function is searched iteratively. During

each iteration, position and velocity of every particle are updated by means of tracking two extremum. One is individual extremum p_{best}, i.e. the best solution seized by individual particle, and the other is global extremum g_{best}, i.e. the best solution by the swarm. After capturing these two extremum, position and velocity of every particle are updated. Considering the ith evolution, the full model of the BPSO can be written as [1]

$$\begin{cases} v_i = v_{i-1} + c_1 r_1 \left(p_{\text{best},i-1} - p_{\text{present},i-1} \right) + c_2 r_2 \left(g_{\text{best},i-1} - p_{\text{present},i-1} \right), \\ p_{\text{present},i} = p_{\text{present},i-1} + v_i. \quad i = 1, 2, \cdots, N. \end{cases} \tag{1}$$

where v_i and $p_{\text{present}, i}$ are search velocity and present position of particles in the ith evolution respectively. The variable v_i is limited to the range $\pm v_{\max}$. Both r_1 and r_2 are random positive numbers, drawn from a uniform distribution between 0 and 1. c_1 and c_2 are both steering coefficients, and usually $c_1 = c_2 = 2$. N is total amount of evolution.

By continuous studying, evolved particles arrive at location of the globally-optimal solution eventually, and the search process ends. Theoretically, the last output of g_{best} is exactly the globally-optimal solution. However, the evolution process of the BPSO may stagnate and be trapped into a locally-optimal solution, if the globally-optimal solution is not captured during all particles fly to the solution limit p^*. As a result, relatively slow convergence and fluctuation of the best-fitness particle around the globally-optimal solution will be observed in the BPSO.

Some techniques had been applied to solve these problems. In literature [2], Shi and Eberhart proposed using a linearly decreased (LD) inertia weight w_i on the velocity-updating model as

$$v_i = w_i v_{i-1} + c_1 r_1 \left(p_{\text{best},i-1} - p_{\text{present},i-1} \right) + c_2 r_2 \left(g_{\text{best},i-1} - p_{\text{present},i-1} \right). \tag{2}$$

where w_i is limited between the upper limit $w_{\max}$ and the lower limit $w_{\min}$, and

$$w_i = w_{\max} - i \frac{w_{\max} - w_{\min}}{N}. \tag{3}$$

usually $w_i \in [0.1, 0.9]$. Two other weight factors on the individual variation ($p_{\text{best},i\text{-}1}$ - $p_{\text{present},i\text{-}1}$) and the global variation ($g_{\text{best},i\text{-}1}$ - $p_{\text{present},i\text{-}1}$) still change randomly, just as that in the BPSO.

In literature [11], a constriction factor α was imposed on the velocity term in the position-updating model, i.e.

$$p_{\text{present},i} = p_{\text{present},i-1} + \alpha v_i. \tag{4}$$

As a result, the search velocity v_i is constricted to a new bound $\pm \alpha v_{max}$ by the factor α. But, like literature [2], no adequate attention is paid to either the individual variation or the global variation during velocity-updating, which may be unadvisable to improve a swarm optimizer.

3. Constructing a Controllable Particle Swarm Optimizer

3.1. Controllable Velocity-Updating Modes and Velocity-Changing Tracks

Based on the above analysis, we proposed an extended velocity-updating model as

$$v_i = w_i v_{i-1} + c_1 \alpha_i \left(p_{\text{best},i-1} - p_{\text{present},i-1} \right) + c_2 \beta_i \left(g_{\text{best},i-1} - p_{\text{present},i-1} \right). \tag{5}$$

where w_i, α_i and β_i are weight factors on search velocity v_i, on the individual variation ($p_{\text{best,i-1}}$ - $p_{\text{present},i\text{-}1}$) and on the global variation ($g_{\text{best},i\text{-}1}$ - $p_{\text{present},i\text{-}1}$) respectively.

Definition 1. *If the following formula (6) is satisfied, the velocity-updating model (5) is defined as the completely random mode (CRM).*

$$w_i = 1, \quad \alpha_i = \text{rand}, \quad \beta_i = \text{rand}. \tag{6}$$

Definition 2. If the following formula (7) is satisfied, the velocity-updating model (5) is defined as the partial random mode (PRM).

$$w_i = f(i), \quad \alpha_i = \text{rand}, \quad \beta_i = \text{rand}. \tag{7}$$

where $f(i)$ is velocity-changing track satisfying $f(i) \in [w_{\min}, w_{\max}]$. As a matter of fact, the PRM is the same as the one proposed by Shi and Eberhart [2], when the LD track (LDT) in formula (2) is used.

Definition 3. If the following formula (8) is satisfied, the velocity-updating model (5) is defined as the completely controllable mode (CCM).

$$w_i = \alpha_i = \beta_i = f(i), \quad f(i) \in \{w_{\min}, w_{\max}\}. \tag{8}$$

Under the CCM, the velocity-updating process in total evolution is controllable.

Several velocity-changing tracks were designed to optimize the search process as follows, where $\Delta w = w_{\max} - w_{\min}$. They are completely random track (CRT), limited random track (LRT), gaussian radial basis track (GRBT), power exponent track (PET) and square power exponent track (SPET), respectively.

$$\text{CRT:} \quad f(i) = \text{rand}|_i, \tag{9}$$

$$\text{LRT:} \quad f(i) = \Delta w \cdot \text{rand}|_i + w_{\min}, \tag{10}$$

$$\text{GRBT:} \quad f(i) = \Delta w \cdot \text{e}^{\left[-(i-1)^2 / 2\sigma^2 \right]} + w_{\min}, \tag{11}$$

$$\text{PET:} \quad f(i) = \Delta w / \xi^{i-1} + w_{\min}, \tag{12}$$

$$\text{SPET:} \quad f(i) = \frac{\Delta w \cdot \sqrt{\left[N^{2\eta} - (i-1)^{2\eta} \right]}}{N^{\eta}} + w_{\min}. \tag{13}$$

where σ^2, ξ and η are all spread parameters of the response

$f(i)$ to the i, and $\xi > 1$. Like the preceding LDT, three later tracks are strictly monotonically decreasing as the i increasing.

Theoretically, the bound $f(i) \in [w_{\min}, w_{\max}]$ can be easily satisfied by adjusting the spread parameter σ^2, ξ or η in the $f(i)$, when the N is known. In figure 1, typical illustrations of these proposed tracks were depicted. Here x-coordinate is the number of evolution i, and y-coordinate the velocity weight w_i in the ith evolution which was limited between $w_{\min}$ =0.2 and $w_{\max}$ =0.8. The total amount of evolution was selected as N =500.

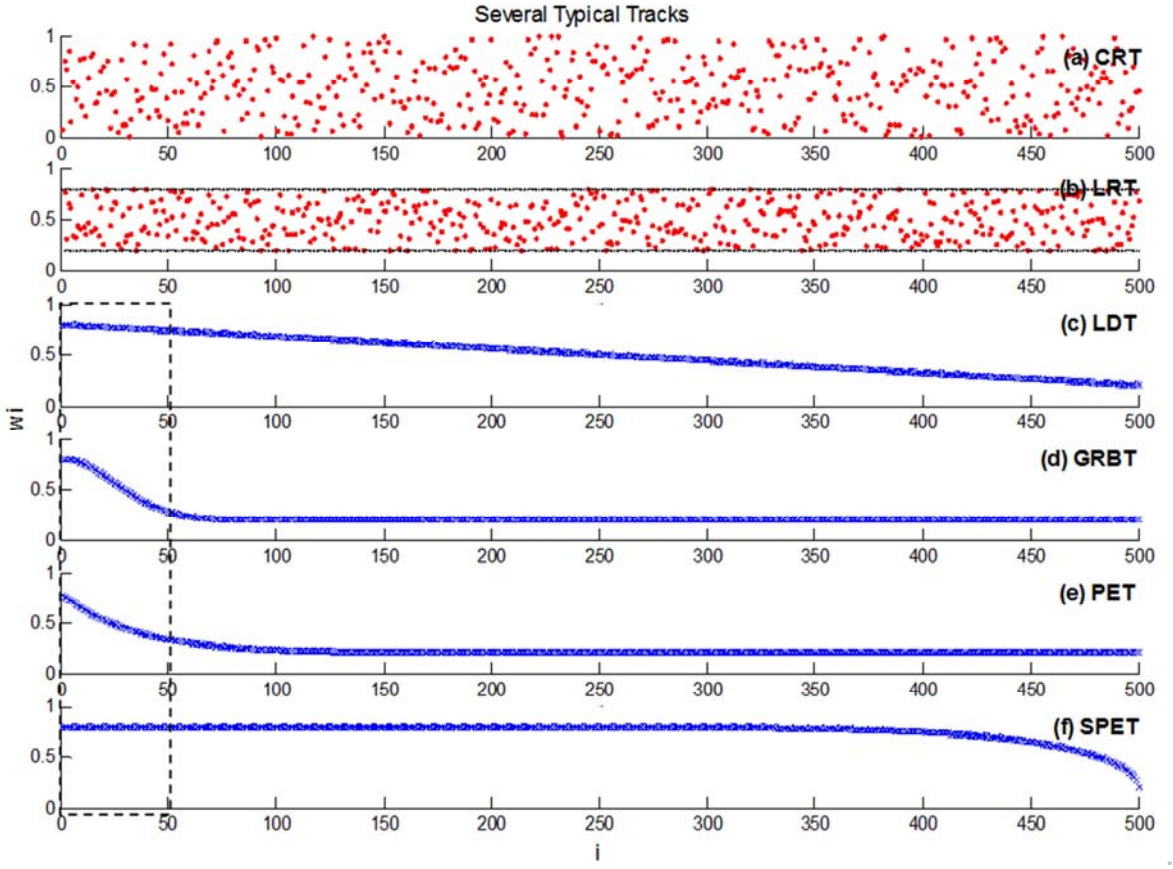

Figure 1. *Illustrations of several typical tracks for velocity-changing.*

The sub-figure 1 (a) was two-dimensional distribution of the CRT, in which the weight w_i randomly changed within [0 1]. The LRT was shown in the sub-figure 1 (b) where the w_i also randomly changed, but bounded by [0.2 0.8]. The LDT was described in the sub-figure 1 (c). Also the GRBT, the PET and the SPET were given in the sub-figure from 1 (d) to 1 (f) respectively, using such parameters setting as σ^2 =600, ξ =1.03 and n =4. Accordingly, their outputs are [0.8 0.2+Δ_2], [0.8 0.2+Δ_2] and [0.8 0.2504] respectively, where both Δ_1 and Δ_2 are very small positive number. In figure 1, every deterministic track has different slope, especially in the first fifty evolutions. Moreover, steep gradient feature can be seen from both the GRBT and the PET (see the framed part by dashed line), below we will see the slope coefficient of a track is very important in a controllable optimizer.

3.2. Performance Evaluation on a Swarm Optimizer

The functions P_i and Q_i are used to evaluate on-line and off-line performances of a swarm optimizer, where F_i is fitness of the ith particle at the jth evolution, and i = 1, 2, …, N, j = 1, 2, …, M. M is size of a population, N is total amount of evolution.

$$\begin{cases} P_i = \log_{10}\left(\dfrac{1}{M}\sum_{j=1}^{M} F_j\right)_i, \\ Q_i = \log_{10}\left[\text{Best}\{F_i, F_2, \cdots, F_M\}\right]. \end{cases} \tag{14}$$

It goes without saying that whether or not a particle has the best-fitness depends on target of an optimization problem, i.e. minimizing or maximizing a given function.

4. An Optimization Example

The optimization problem is formulated as

$$\min_{(x,\,y)} 100\left(x^2 - y\right)^2 \left(2 - x - y^2\right)^2, \quad -2 \le x, y \le 2. \tag{15}$$

Intuitively, its globally-optimal solution is $x = y = 1$, and its extremum is zero.

Firstly, the PSO system was configured as follows. The total amount of evolution N is 500, and the $w_{\min}$ and $w_{\max}$ are

0.2 and 0.8 respectively. Size of population is designedly selected as 30 for testing convergence performance of an optimizer under small-population condition. The best history solution and the best-fitness are both initialized with infinite, and the globally best solution g_{best} with [0 0]. Individual and global steering coefficients are both selected as 2. The maximum velocity is 0.3.

Table 1. *Performance of several typical swarm optimizers on the illustrated example.*

Optimizers	Test No.										Convergence Precision	Computation Expense
	1	2	3	4	5	6	7	8	9	10		
CRM (BPSO)	F	F	F	F	F	F	F	F	F	F	0/10	274
PCM-LRT	S	F	F	S	S	S	F	F	S	S	6/10	365
CCM-LRT	S	S	F	S	F	S	S	S	F	S	7/10	132
PCM-LDT	S	S	S	F	F	S	S	S	F	F	6/10	336
CCM-LDT	S	S	S	F	S	F	S	F	S	S	7/10	235
PCM-GRBT	S	S	S	S	F	F	F	S	S	S	7/10	124
CCM-GRBT	S	F	S	S	S	S	S	F	S	S	8/10	86
PCM-PET	S	S	S	S	S	S	S	F	F	S	8/10	127
CCM-PET	S	S	F	S	S	F	S	S	S	S	8/10	82
PCM-SPET	S	S	S	F	S	S	S	F	S	S	8/10	>500
CCM-SPET	S	S	S	F	S	S	F	S	F	S	7/10	376

Ten tests were randomly made upon several combinations of velocity-updating modes and velocity-changing tracks. Convergence precision and computation expense of these optimizers were listed in table 1. The letter 'S' stands for successful convergence, and the letter 'F' for failed one. Especially, the symbol 'S' implies one almost successful convergence. That is, it will be successful if the amount of evolution N is enough big. Furthermore, on-line and off-line performances of several typical swarm optimizers were given in figure 2. In figure 2, two x-coordinates are both the studying number i. And y-coordinates are the on-line performance *P-online* and the off-line performance *Q-offline* respectively.

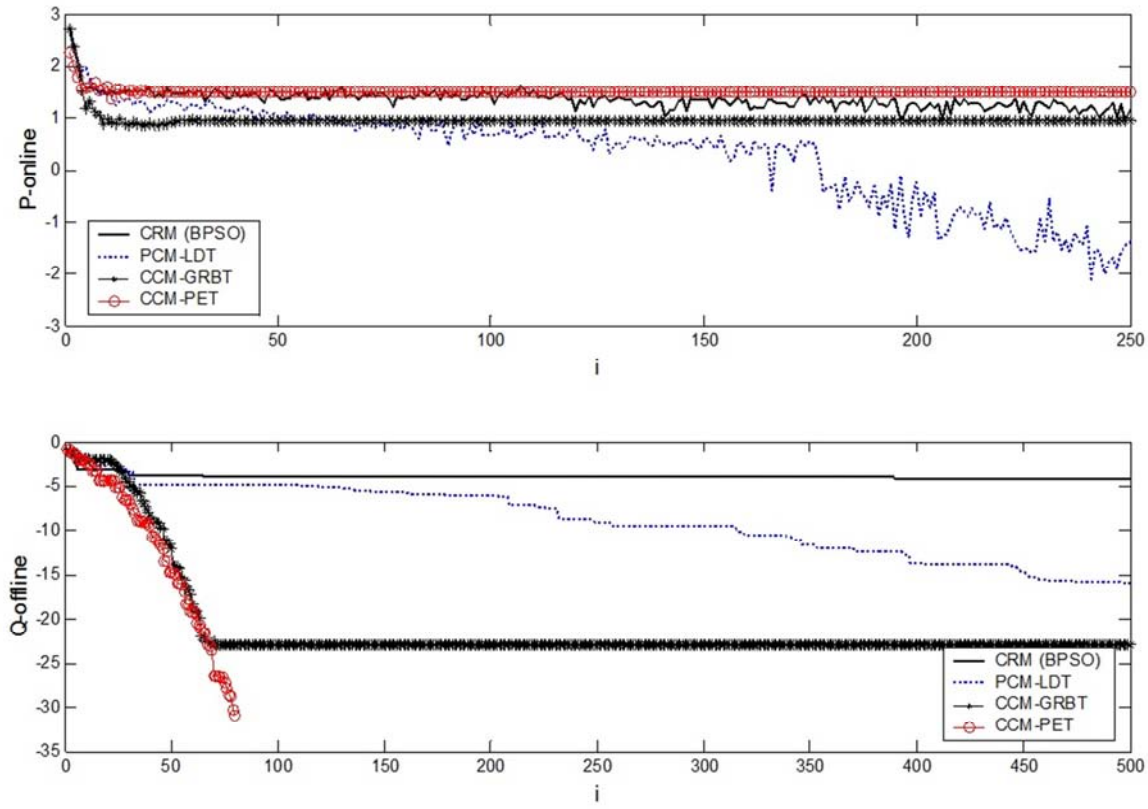

Figure 2. *On-Line and off-line performances of several typical optimizers.*

5. Results and Discussion

Under small-population condition (only thirty individuals), the globally-optimal solution was never captured by the BPSO (CRM) in ten tests, and its averaged computation expense is 274 iterations, seen from table 1. As expected, almost all CCM based optimizers except the CCM-SPET were superior to the BPSO, on both convergence precision

and computation efficiency. It was also shown that the modified PSO by Shi and Eberhart [2] (i.e. the PCM-LDT combination) consumed more computation than the BPSO (from 274 to 336). After using the CCM for velocity-updating, computation cost was considerably reduced from 336 to 235 iterations, and convergence precision was also improved slightly (from 6/10 to 7/10). Although only ten tests are very inadequate for quantitatively evaluating these optimizers, the results can verify their effectiveness and better overall performance qualitatively. The deleterious effects of randomness in the BPSO were indeed controlled to some extent by introducing controllable velocity-updating modes.

It could be also seen from figure 2 that, under keeping good optimization precision, the CCM based swarm optimizers, for example CCM-GRBT (see the black asterisk-solid line) and CCM-PET (see the red circle-solid line), converged faster than the BPSO (see the blue solid line) and PCM-LDT (see the blue dashed line), and had higher computation efficiency.

An interesting phenomenon is worth noticing in experiment that particles always converge to two typical classes after a successful evolution by the CCM based optimizer, i.e. the winners and the losers, see figure 3. In figure 3, all the winning particles were unseen because they has converged to the global optimun (see the red point), but the losers are abandoned and gathered into certain region (see the enclosed part by dashed line). Below, we will explain this phenomenon in detail.

We have proved that the weight factor w, the search velocity v and the particle position y, in the CCM based optimizer, change synchronously and proportionally. Before an evolution starts, every individual in a population is initialized with random position within the solution space, either far from or near to the global optimum. During the evolution, every particle will continuously slow down because certain monotonically decreasing (sometimes steeply gradient, for example the GRB or the PE) track for velocity-updating is used. If a particle is too far from the global optimum initially, it will lose in a competitive evolution eventually. Contrarily, it will win. As a result, particles will present itself a two-clustering distribution after the total evolution ends. Such space diversity phenomenon usually arises in the early evolution stage, under the combinational effect of the CCM and some velocity-changing tracks with monotonic decrease (sometimes steep gradient, for example the GRBT and the PET) feature. Consequently, an interesting but possibly important question can be put forward here. That is, how to deal with those particles doomed to fail in a competitive evolution? Some possible solutions could be suggested as follows: (a) Paying no attention to them. Such solution has no any help to further improving the CCM based optimizer; (b) Stopping updating them timely during the period of evolution, thus saving computation cost. But what time (when) is appropriate to do so? An evaluation criterion for it is indispensable; (c) Using them as a secondary detector to search possibly better solution(s) in unexplored regions of the solution space, by adaptively changing their search velocity. Obviously, a lot of tasks must be done for improvement of the PSO algorithm.

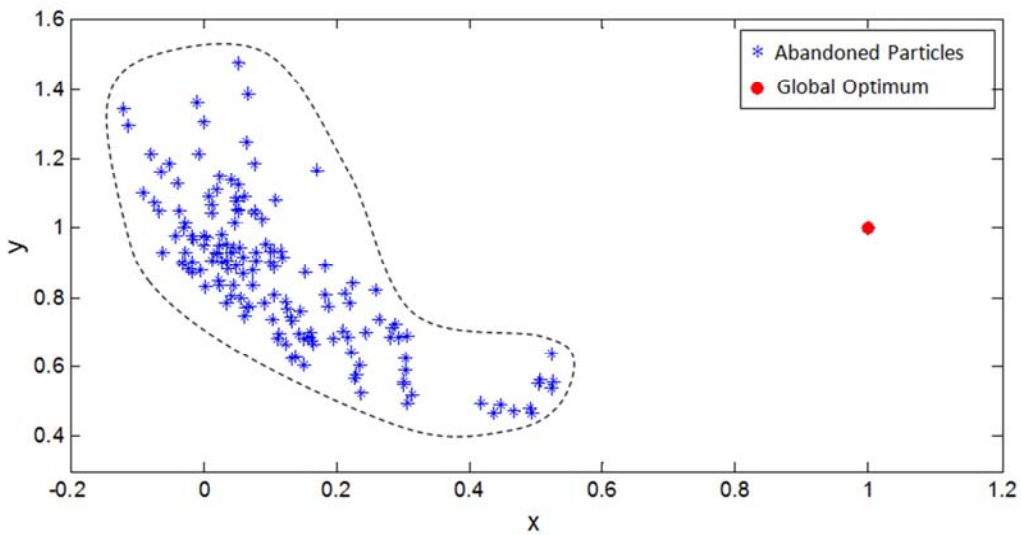

Figure 3. *Two-Clustering distribution in the CCM-PET optimizer with 400 individuals.*

6. Conclusion

In the BPSO, the weight factors upon the individual variation and the global variation change randomly during total evolution. Such deleterious randomness may lead to slow convergence at the late evolution stage and particles fluctuation around the globally-best solution. It was found that performance of the swarm optimizer could be improved remarkably, by using some controllable velocity-updating modes, especially the CCM for velocity-updating. The optimization result indicated that after using the combinations of the CCM and some velocity-changing tracks (for example GRBT and PET), the modified optimizer features relatively high precision and faster convergence. The

reason may be that the CCM enables synchronous and proportional control imposed on the search velocity, the individual variation and the global variation in the velocity-updating model, which is beneficial to location-probing of the globally-optimal solution in total solution space.

Steep gradient features embedded in the forepart of the GRBT and the PET may also contribute to faster convergence at the early evolution stage. Just as that pointed by Marini and Walczak, a swarm of particles may flow through the parameter space defining trajectories which are driven by their own and neighbors' best performances [12]. Intrinsic functional mechanisms of velocity-updating mode on convergence performance of searching particle are thus to be studied in depth. Undoubtedly, further research on these problems will be made in future.

Acknowledgements

This research was supported by 'National Natural Science Foundation of China (Grand #: 51575497 and 51405449)' and 'The Public Welfare Technology Project of Zhejiang Province (Grand #: 2016C31067)'. The authors would also like to express the sincere appreciation to the reviewers for their helpful comments in improving the presentation and quality of the paper.

References

[1] Kennedy J, Eberhart R C. Particle swarm optimization. In *Proc. of the IEEE Conf. on Neural Networks IV*, Perth, IEEE Press, 1995, pp. 1942-1948.

[2] Shi Y, Eberhart R C. A Modified Swarm Optimizer. *IEEE International Conference of Evolutionary computation*, Anchorage, Alaska, 1998, pp. 69-73.

[3] Clerc M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In *Proc. of the ICEC*. Washington, 1999, pp. 1951-1957.

[4] Shi Y, Eberhart R C. Fuzzy adaptive particle swarm optimization. In *Proc. of the Congress on Evolutionary Computation*, Piscataway, IEEE, 2001, pp. 101-106.

[5] Huang Y, Liu Y F, Peng Z M, et al. Research on particle swarm optimization algorithm with characteristic of quantum parallel and its application in parameter estimation for fractional-order chaotic systems. *Acta Phys. Sin.*, 2015, 64 (3): 1-8.

[6] Guo W H, Wang T S. Pre-Impact Configuration Optimization for a Space Robot Capturing Target Satellite. *Journal of Astronautics*, 2015, 36 (4): 390-396.

[7] Zhai T T, Zhu J Q. New Method for First-Order Structure Design of Continuous Zoom Lens System. *Acta Opt. Sin.*, 2015, 35 (7): 1-9.

[8] Ireneusz G. A new approach to particle swarm optimization algorithm. *Expert Systems with Applications*, 2015, 42: 844-854.

[9] Cheung N J, Ding X M, Shen H B. A supervised particle swarm algorithm for real-parameter optimization. *Application Intelligence*, 2015, 43: 825-839.

[10] Tang R L, Fang Y J. Modification of particle swarm optimization with human simulated property. *Neurocomputing*, 2015, 153: 319-331.

[11] Clerc M, Kennedy J. The particle swarm: Explosion stability and convergence in a multi-dimensional complex space. *IEEE Trans. on Evolution Computer*, 2002, 6 (1): 58-73.

[12] Marini F, Walczak B. Particle swarm optimization (PSO): A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 2015, 149: 153-165.