

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

An Improved Particle Swarm Optimization

Wenjuan Zeng, Haibo Gao and Wang Jing

College of Information Science and Engineering, Hunan International Economics University,
Changsha, 410205, China

Abstract: The basic and improved algorithms of PSO are focused on how to search effectively the optimal solution in the solution space by using one of the particle swarm. However, the particles are always chasing the global optimal point and such points are currently found on their way of search, rapidly leading their speed down to zero and hence being restrained in the local minimum. Consequently, there are the convergence or early maturity of particles. The improved PSO is based on the enlightenment of Back-Propagation (BP) neural network while the improvement is similar to the smooth weight through low-pass filter. The test of classical functions show that the PSO provides a promotion in the convergence precision and make better a certain extent in the calculation velocity.

Key words: PSO, smooth weight, improved particle swarm, back-propagation, optimization algorithm

INTRODUCTION

Since the 1990s, computing problems to be solved through behavioral simulation of biological populations has become a new hotspot and has formed a theoretical system as the core of the swarm intelligence and has achieved breakthroughs in a number of practical applications. As a typical implementation model of swarm intelligence, it is widespread concern by the academic community that colony optimization algorithm as simulation process for ant community to collect food (Colomi *et al.*, 1991) (ant colony optimization) and particle swarm optimization algorithm as simulation of flock movement patterns (Arumugam and Rao, 2008) (particle swarm optimization).

Particle Swarm Optimization algorithm (PSO) is proposed in 1995 by the American social psychologist James Kennedy and Russell Eberhart, an electrical engineer. The basic idea is affected and inspired by their early findings on the behavior of bird populations. Biologist Frank Heppner's biological population model was applied.

PSO is similar to genetic algorithms and particle swarm optimization is also based on the individual collaboration and competition to complete the complex search of the optimal solution in the search space and it is an evolutionary computing technology which is based on swarm intelligence methods. However, PSO does not make cross (crossover), variation (mutation) and other operations in the genetic algorithm but the particles search in the solution space to follow the optimal particle,

so it has the advantages of simple and easy to achieve and it is need to be adjusted without too many parameters.

PARTICLE SWARM OPTIMIZATION

Elementary particle swarm optimization mathematical model: PSO basic concepts are derived from the study of artificial life and the flock predatory behavior. Imagine such a scene; a flock of birds search food in a random, only a piece of food in this area, all birds know where the food is but they do not know how far away the current location is from the food. Then the optimal strategy to find food in the simplest and most effective is to search around the area of the recent bird from food.

The PSO algorithm is inspired from biological populations behavior which is used to solve optimization problems. In PSO, each potential solution of the optimization problem can be thought of as a point on the N-dimensional search space, we call particles, all particles have a fitness value which is determined by the objective function. Each particle speed determines the direction and distance where they fly, the particles are then following the current optimum particles to search in the solution space.

In the search space of the N-dimension target, a community is composed of M particles, wherein the *i*th particle is expressed as N-dimensional vector $X_i = (X_{i1}, X_{i2}, \dots, X_{iN})$, $i = 1, 2, \dots, m$, i.e., the *i*-th particle in the location of the N-dimensional space. Its fitness value can be calculated by X_i into an objective function, X_i 's pros or

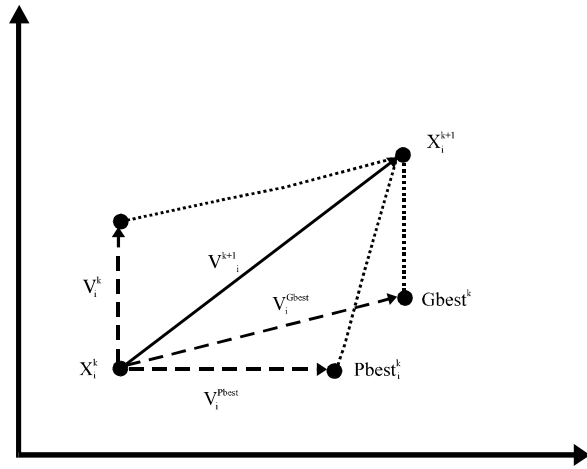


Fig. 1: PSO algorithm schematic diagram

cons is measured based on the size of the fitness value. The flying speed of the i -th particle is N -dimensional vector $V_i = (v_{i1}, v_{i2}, \dots, v_{iN})$, denoted by the algorithm shown in Fig. 1. The particle is operated by the following equations:

$$V_{id}^{k+1} = w \times V_{id}^k + c_1 \times \text{rand}() \times [X_{id}^k - X_{id(\text{pbest})}^k] + c_2 \times \text{rand}() \times [X_{id}^k - X_{id(\text{gbest})}^k] \quad (1)$$

$$X_{id}^{k+1} = X_{id}^k + V_{id}^{k+1} \quad (2)$$

Particle swarm optimization parameters

Particle population size (M): The selection of the particles population size is depending on the specific problem but generally 20-50 is set in the number of particles. In fact, for most of the problems, good results can be achieved enough by 10 particles but for the more difficult issues or specific types of problems, the number of particles can be taken up to 100 or 200.

Length (N) of the particles: This is the number of dimensions of the solution space.

Maximum speed of the particles: The velocity of the particle in space has a maximum speed limit value in each dimension, that is used to clamp so that the speed is controlled in the velocity of the particle range (Xiang *et al.*, 2007) and to determine the search granularity of the problem space.

Acceleration constant c_1 , c_2 : To adjust the direction of the pbest and gbest flight and the maximum step, the particles individual experience and group experience are

decided to influence on the particle trajectory. They reflect the exchange of information between the particle swarm. If, the particles is only group experience, its convergence speed is faster but it is easy to fall into local optimum, particle groups does not share information, a scale for the M group is equivalent to running a single particle, it is difficult to get the most optimal solution, so the general settings. Changing these constants will change the "tension" of the system, a lower c_1 , c_2 values make the particles hovering in the area away from the target and higher c_1 , c_2 value is generating steep movement or over the target area. Shi and Eberhart (1998) suggested that, in order to balance the role of random factors, general settings, most algorithms have adopted this suggestion.

Rand: That is a random number in the range (0,1).

Iteration termination condition: The maximum number of iterations and the calculation accuracy are set in general.

Particle swarm optimization steps:

- Step 1:** Initialize the particle swarm to given the size M of population and to set the N-dimension of the solution space. Each particle's position and its speed are randomly generated
- Step 2:** The fitness value of each particle is calculated by each benchmark functions
- Step 3:** Update individual extreme to evaluate the fitness value of each particle. The i th particle current fitness value is compare with the adaptation value of the particle individual extremum, if the former is superior, update and otherwise remain unchanged
- Step 4:** Update the global extremum, the best is selected from all, as a global minimum
- Step 5:** Update the speed and position by the Eq. 1 and 2 update the speed and position of each particle
- Step 6:** Check whether to meet the terminating conditions, if it satisfies, to exit, otherwise, to go to step 2

PROBLEM OF THE BASIC PSO ALGORITHM

The search range of the particles is limited in particle convergence. To broaden the search, it is necessary to increase the number of particles in the particle swarm or weaken the chase of the particles of the entire particle swarm search to global optimum. The increasing number of particles will result in that computational complexity is increased while there is again a shortcomings of small

algorithm and easy to convergence in the weakening chase of the global optimum particle.

Since the basic PSO algorithm relies on the cooperation and competition between groups, particle itself has no mutation mechanism and thus a single particle is difficult by itself to jump out of local extreme restraint once after there is a local extreme restraint, the help of other particles successful discovery is needed in this time. In fact, PSO algorithm optimization capability is mainly from the interaction and mutual influence between the particles. If you remove the interaction and mutual influence between the particles from the algorithm, the PSO algorithm optimization ability becomes very limited.

Tests indicates out that in the initial stage of the algorithm running, convergence speed is faster and trajectory is sinusoidal swing but after running for some time, the speed began to slow down or even stagnation (Zhou *et al.*, 2013; Yu *et al.*, 2005). When the speed of all the particles is almost 0, at this point, particle swarm lost the ability to further evolution, the algorithm execution can be considered to have been converged. In many cases (such as complex multimodal function optimization), the algorithm does not converge to the global minimum and it may not reach even the local extreme. This phenomenon is known as premature convergence or stagnation. When this phenomenon occurs, the particle swarm shows a high degree aggregation and because of a serious lack of diversity, particle swarm will be impossible to escape the rallying point for a long time or forever. So, a lot of the particle swarm optimization algorithm is focused on improving the diversity of PSO, making particle swarm throughout the iterative process to maintain the ability to further optimize.

IMPROVED PARTICLE SWARM OPTIMIZATION

Improved particle swarm optimization gets inspiration from the Back-Propagation (BP) neural network. BP algorithm is used to deal with multi-level neural network method. It uses the gradient descent method to minimize the error between the actual output results and predicted output ones. But it exists in the vicinity of local optima stagnation and oscillation, thus falling into local optimum, not easy to get global optimal solution through the following methods to improve on this issue. Such improvements as the use of the low-pass filter to smoothing weight (5). Its mathematical expression is as follows:

$$y^t = (1 - \lambda) \times x^t + \lambda \times y^{t-1} \quad (3)$$

where, $\lambda \in (0, 1)$, x is a signal sequence which needs to be smoothed, x^t is the signal value of the time t , y^t is the output of the filter in time t . The higher the λ value is took, the better the smoothing properties.

Therefore, the introduction of a new parameter PSO velocity update equation is as follows:

$$V_{id}^{k+1} = (1 - \lambda) \{ V_{id}^k + c_1 \times \text{rand}() \times [X_{id}^k - X_{id(\text{pbest})}^k] + c_2 \times \text{rand}() \times [X_{id}^k - X_{id(\text{gbest})}^k] \} + \lambda \times V_{id}^{k-1} \quad (4)$$

$$X_{id}^{k+1} = X_{id}^k + V_{id}^{k+1} \quad (5)$$

Compared with the original PSO equation, the improved PSO algorithm increases the coefficient:

$$\lambda \times V_{id}^{k-1}$$

which, λ is introducing items. This improvement is compared with existing methods to improve the speed of update equations and is very different. This is because the original particle velocity update equation is a first-order differential equation while the improved PSO particle update rate equation is for the second-order differential equation.

The main advantage of this improved PSO are as following:

- In this study, an improved PSO still has a strong operational both in the mathematical expression and in the implementation of the program, the absence of the introduction complex operations and data structures, so the operation is relatively easy
- Smoothed particle trajectories, eliminating late oscillation in the iteration which does not converge and stagnate into a local optimum
- The improved PSO can, almost, be applied to all existing improved PSO algorithm with clear particle velocity update equation, such as the inertia weight change type PSO (Shi and Eberhart, 1998), the shrinkage factor change type PSO (Clerc, 1999), hybrid PSO (Angeline, 1998)

EXPERIMENTAL EVALUATION

Test function: In order to verify the performance of the improved PSO algorithm, researchers in the field generally use the following test function and Fig. 2 is the image of these functions:

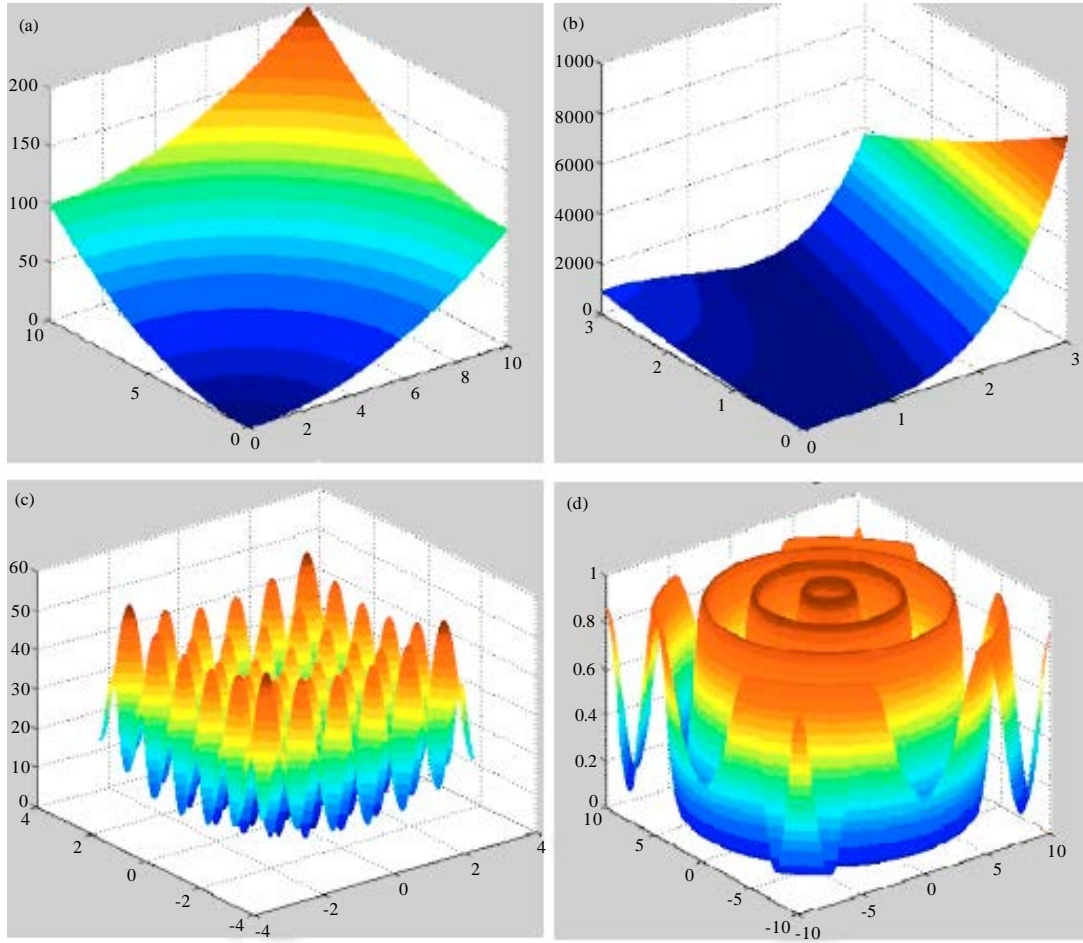


Fig. 2(a-d): Test functions, (a) Sphere function, (b) Rosenbrock function, (c) Rastrigrin function and (d) Schaffer function (Yang, 2006; Tong, 2006)

Sphere function:

$$f_1(x) = \sum_{i=1}^n x_i^2 \quad (6)$$

Global minimum point in $x^* = (0, 0, \dots, 0)$ the global minimum $f(x^*) = 0$.

Rosenbrock function:

$$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (7)$$

Rosenbrock function is a unimodal function, the strong coupling between the variables. Global minimum point in $x^* = (0, 0, \dots, 0)$ the global minimum $f(x^*) = 0$.

Rastrigrin function:

$$f_3(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10] \quad (8)$$

Rastrigrin function is a multimodal function, many sinusoidal protrusions local minima, the variables are independent of each other, global minimum point in $x^* = (0, 0, \dots, 0)$, the global minimum $f(x^*) = 0$.

Schaffer function:

$$f_4(x) = 0.5 + \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} \quad (9)$$

Schaffer function is a two-dimensional function, global minimum point in $x^* = (0, 0, \dots, 0)$, the global minimum $f(x^*) = 0$.

IMPROVED PSO ALGORITHM EVALUATION INDICATORS

In order to characterize the performance of the improved PSO algorithm, we need to introduce evaluation index (Yu *et al.*, 2005; Zhou *et al.*, 2013). This study is introduced that the following four evaluation indicators are; the convergence precision error, algorithm success ratio, convergence speed and iteration convergence curve. Characterization wherein the convergence accuracy error is the deviation between the theoretical optimal value of the objective function and the actual results, the smaller the value, the higher the accuracy of the calculation. The success rate of the PSO algorithm is as follows:

$$\beta = \frac{\text{Successfully_running_time}}{\text{Algorithm_total_running_time}} \quad (10)$$

The successful running time is algorithm time from convergence to a given benchmark global optimum to meeting the end of the algorithm accuracy. The convergence speed γ is characterized by the spent time which algorithm converges to the global optimal value. Iterative convergence curve characterizes the change with the number of iterations and trajectory changes in the value of the objective function. Iterative convergence curve characterizes the change with the number of iterations, and trajectory changes in the value of the objective function. By the iterative convergence curve contrast, clear conclusion can be obtained intuitively.

SIMULATION RESULTS

The calculation results are shown in Fig. 3-6.

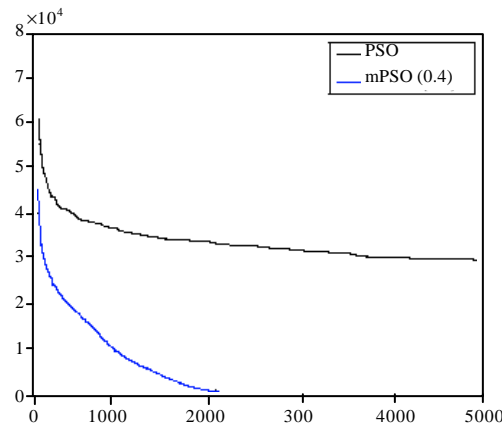


Fig. 3: Sphere function calculation results when $\lambda = 0.4$

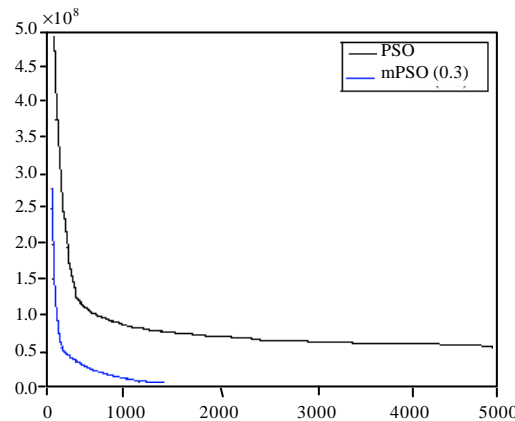


Fig. 4: Rosenbrock function calculation results when $\lambda = 0.3$

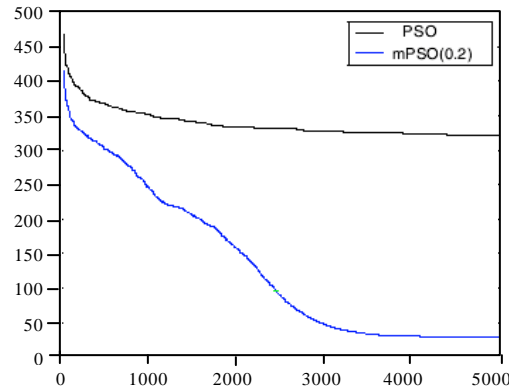


Fig. 5: Rastrigri function calculation results when $\lambda = 0.2$

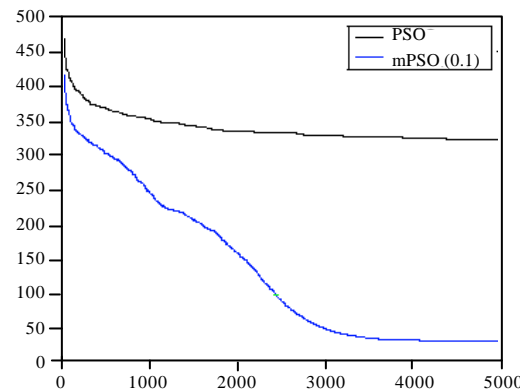


Fig. 6: Schaffer function calculation results when $\lambda = 0.1$

CONCLUSION

Improved PSO, in this study, has reduced the local iteration degree oscillation of the PSO algorithm in an iterative process because it can be trapped into local optima. By comparing the calculation results, the improved PSO algorithm accelerate the convergence rate and improve the calculation speed. As a result of the update to the particle velocity equation, the making speed equation becomes second order difference equations and these are improving the movement of particles in space controllability which greatly eliminates the oscillations in the late iterations. Meanwhile, the experiment shows that when λ changes from 0.2-0.4 in the improved PSO algorithm, improved PSO algorithm converges to the optimum with fewer iterations. The improved PSO of $\lambda = 0$ is evolved in basic particle swarm

algorithm of the weights $\omega = 1$. It is visible that improved PSO algorithm in this study has good versatility.

Since, the PSO for handling large-scale optimization problems has powerful processing capability, therefore, high dimension, nonlinear applications in research and PSO algorithm has been an extremely wide range of applications. Therefore, for the study of issues which is related to the application system (Eberhart and Shi, 2004; Ling *et al.*, 2008), the application of improved PSO algorithm and which is optimized analysis for application systems, it has become a necessary and viable means.

ACKNOWLEDGMENT

The study is funded by National Natural Science Foundation of China (NO. 51304076).

REFERENCES

- Angeline, P.J., 1998. Using selection to improve particle swarm optimization. Proceedings of the International Conference Evolutionary Computation Intelligence, May 4-9, 1998, Anchorage, AK., USA., pp: 84-89.
- Arumugam, M.S. and M.V.C. Rao, 2008. On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems. *Applied Soft Comput.*, 8: 324-336.
- Clerc, M., 1999. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. Proceedings of the Congress on Evolutionary Computation, July 6-9, 1999, Washington, DC., USA., pp: 1951-1957.
- Colomi, A., M. Dorigo and V. Maniezzo, 1991. Distributed optimization by ant colonies. Proceedings of the 1st European Conference on Artificial Life, December 11-13, 1991, Paris, France, pp: 134-142.
- Eberhart, R.C. and Y. Shi, 2004. Guest editorial special issue on particle swarm optimization. *IEEE Trans. Evol. Comput.*, 8: 201-203.
- Ling, S.H., H.H.C. Iu, K.Y. Chan, H.K. Lam, B.C.W. Yeung and F.H. Leung, 2008. Hybrid particle swarm optimization with wavelet mutation and its industrial applications. *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, 38: 743-764.
- Shi, Y. and R. Eberhart, 1998. A modified particle swarm optimizer. Proceedings of the World Congress on Computational Intelligence and IEEE International Conference on Evolutionary Computation, May 4-9, 1998, Anchorage, AK., pp: 69-73.
- Tong, Z., 2006. Particle Swarm Optimization Algorithm and Improved. Nanjing University of Science and Technology, Nanjing, China.
- Xiang, T., J. Wang and X. Liao, 2007. An improved particle swarm optimizer with momentum. Proceedings of the IEEE Congress on Evolutionary Computation, September 25-28, 2007, Singapore, pp: 2241-3345.
- Yang, L., 2006. Particle Swarm Optimization Algorithm with Improved. Harbin Institute of Technology, Harbin, China.
- Yu, H.J., N. Xu, L.P. Zhang and S.X. Hu, 2005. Research on hybrid particle swarm optimization. *Inform. Control*, 34: 500-509.
- Zhou, X.Y., Z.J. Wu, H. Wang, K.S. Li and H.Y. Zhang, 2013. Elite opposition-based particle swarm optimization. *Acta Electron. Sin.*, 41: 1647-1652.