Magnifier Particle Swarm Optimization For Numerical Optimization

Junqi Zhang Kun Liu Ying Tan* Xingui He
Key Laboratory of Machine Perception (Peking University), Ministry of Education, and Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Beijing, China
ytan@pku.edu.cn

ABSTRACT

A novel particle swarm optimization algorithm based on magnification transformation, called magnifier particle swarm optimization (MPSO), is proposed for the first time in this paper. In the MPSO, we enlarge the range around the best individual of each generation like using a magnifier, while the velocity of particles unchanged. In such a way, MPSO achieves much faster convergence performance and better optimization solving capability than the conventional standard particle swarm optimization and latest clonal PSO by a number of simulations. A detailed description and explanation of the MPSO algorithm are given in the paper. Experiments on fourteen benchmark test functions are conducted and shows the inspiring success that the proposed MPSO speeds up the convergence tremendously, while keeping a good search capability of global solution with much more accuracy. Experiments on fourteen benchmark test functions are conducted to demonstrate that the proposed MPSO algorithm is able to speedup the evolution process distinctly and improve the performance of global optimizer greatly.

Categories and Subject Descriptors: I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search

General Terms: Algorithms, Performance.

Keywords: Particle Swarm Optimization(PSO), Swarm Intelligence, Global Numerical Optimization.

1. INTRODUCTION

The particle swarm optimization (PSO) is a stochastic global optimization technique inspired by social behavior of bird flocking or fish schooling [1]. In conventional PSO algorithm, the update formula for each particle's velocity and position in conventional standard PSO is written as

$$V_{id}(t+1) = wV_{id}(t) + c_1 r_1 (P_{iBd}(t) - X_{id}(t)) + c_2 r_2 (P_{gBd}(t) - X_{id}(t)), \quad (1)$$

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

where $i = 1, 2, \cdots, n$, n is the number of particles in the swarm, $d = 1, 2, \cdots, D$, and D is the dimension of solution space. The learning factors c_1 and c_2 are nonnegative constants, r_1 and r_2 are random numbers uniformly distributed in the interval [0,1], $V_{id} \in [-V_{max}, V_{max}]$, where V_{max} is a designated maximum velocity which is a constant preset by users according to the objective optimization function. The parameter $w \in [0, 1]$ is the inertia weight [2]. The position and the velocity of each particle are updated according to its own previous best position ($P_{iBd}(t)$) and the current best position of all particles ($P_{gBd}(t)$) in each iteration. For convenience, we call the PSO expressed in Eqs. (1) and (2) as standard PSO (abbreviated as SPSO) in this paper.

Magnification transformation is a very simple but very useful strategy, 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. For example, this transformation is well used in building screen magnifiers to enlarge the information presented on a visual display in a computer system [3]. In this paper, a simple magnification transformation is introduced into PSO, resulting in a novel magnifier PSO (MPSO, for short). In stretching method, attention was paid to the top part of the fitness function to eliminate undesired local minima by a two-stage transformation. We will focus on the bottom of the fitness landscape since the range around the best individual deserves a better check, and the probability that the actual global best particle lying in that range is probably greater than others in search space. In our algorithm, the original function is not changed, just the range mentioned above is enlarged via magnification transformation, while keeping the velocity of particles unchanged. In this speed, it will speed up the local search while maintain their global search capability.

$$\tilde{x}(t+1) = x(t+1) - (2*r/s - 2*r) \quad if \quad x(t) < L \quad and \quad x(t+1) > R, \quad (3)$$

$$\tilde{x}(t+1) = x(t+1) + (2*r/s - 2*r) \quad if \quad x(t) > R \quad and \quad x(t+1) < L, \quad (4)$$

$$\tilde{x}(t+1) = x(t+1) - [(R - x(t))/s - (R - x(t))] \quad if \quad L < x(t) < R \quad and \quad x(t+1) > R, \quad (5)$$

$$\tilde{x}(t+1) = x(t+1) + [(x(t) - L)/s - (x(t) - L)] \quad if \quad L < x(t) < R \quad and \quad x(t+1) < L. \quad (6)$$

2. MPSO METHOD

In each generation, a range around the best individual is set up in each dimension. If the particles in the swarm would pass through the range in the next generation, we use a magnifier operator to enlarge the range without changing the velocity of particles. Thus the particles would get a better chance to land into the range, which is able to check the area around the current best individual more precisely. For those particles who were already going to land into

*Corresponding author.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

Table 1: Statistical means and standard deviations of the solutions of fourteen benchmark test functions, listed in Table 1, given by the MPSO, CPSO and the SPSO over 50 independent runs.

Functions	FEs	MPSO's M ± S	CPSO's M ± S	SPSO's M ± S
Shaffer f6	26,600	**1.000 ± 0**	0.995993 ± 0.004688	0.992647 ± 0.003122
Sphere	48,480	**0 ± 0**	2080.392090 ± 1376.428640	40378.2 ± 4138.8
Rosenbrock	1,200,000	12.192243 ± 21.495943	**1.012707 ± 1.727705**	45.646805 ± 54.516615
Griewangk	1,200,000	53.230202 ± 15.117090	**7.312949 ± 3.521576**	18.804737 ± 4.953855
Rastrigrin	1,200,000	**0.012906 ± 0.012880**	0.037030 ± 0.027916	0.015851 ± 0.015683
Ellipse	56,320	**0 ± 0**	403974.18 ± 292686.82	2296746.5 ± 498215.4
Cigar	62,080	**0 ± 0**	18274214 ± 8592084.6	93314232 ± 11030717
Tablet	48,720	**0 ± 0**	2586.096191 ± 1943.306358	16214.034180 ± 2760.936218
SumCan	1,200,000	0.000005 ± 0	0.000004 ± 0.000000	**0.0000030 ± 0**
Schwefel	72960	**0 ± 0**	289262.9 ± 286777.1	6190796.5 ± 818203.97
Ackley	1,200,000	0.124778 ± 0.375480	**0 ± 0**	0.999836 ± 4.358183
Griewangk RT	1,200,000	**0.000006 ± 0.000012**	0.025361 ± 0.034384	0.034504 ± 0.02725
Schwefel RT	1,200,000	**1.898376 ± 0.000001**	1.898377 ± 0.000001	1.898395 ± 0.000029
Ackley RT	1,200,000	**0.057799 ± 0.251938**	0.494281 ± 0.733514	0.475809 ± 0.633685

the range, we do not use the magnifier operator to them, because they already shew interests to the range. On the other hand, we keep the velocity of the particles unchanged so that they are able to fly out of the range in a certain generations for maintaining the global search ability in SPSO. In each situation, the position after using the magnifier operator in the MPSO will be calculated by Eqs. (3) - (6), respectively. r is the radius of the interval whose left and right boundary are indicated by L and R. s is the scale which decides the magnification to enlarge the range.

3. EXPERIMENTS

To test and verify the performance of our proposed MPSO, and make a comparison with SPSO and clonal PSO (CPSO), fourteen benchmark functions and the corresponding parameters listed in our lab site $http://www.cil.pku.edu.cn/resources/benchmark\_pso/$. We fixed the number of particles in a swarm to be 40 for convenient comparisons later on. We used an tentative method to determine the set of parameters [4]. The performances of the MPSO on four functions with different scales of the magnifier operator show that s should not be too small, because we must make sure that the particles will not fall into the range too easily and hard to fly out, which lead to a prematurity, and $s = 0.5$ will be a suitable decision. On the other hand, r should reduce along with the growth of generations, because we want the best individual to converge inside the range. So, we should fix an initial value for r, from which r reduces linearly to zero. The iterative equation of r is expressed by Equ. (7).

$$r = r * (1 - k/M), \quad (7)$$

where k is the current iteration number and M is the maximum iteration number we set. The performances of the MPSO with different initial value of r show that the initial value of r should be 0.3.

In order to verify the validation and efficiency of our MPSO, in Table 1, we give the statistical means and standard deviations of our obtained solutions of the fourteen benchmark test functions, by using MPSO, CPSO and SPSO, over 50 independent runs, respectively. Thereinto, FEs denotes the number of fitness value evaluations of swarm. It has been seen from the averaged solutions that our proposed MPSO outperforms CPSO and SPSO dramatically on most of the functions.

4. DISCUSSION

The essence of MPSO is to adjust the particles to search the solution space more pertinently. We increase the probability of particles landing into the range around x_{gB}, maintain the probability of particles flying far from x_{gB}, and decrease the probability of particles wandering around the range containing x_{gB}. In such a way, we give all the particles only two choices, either landing very near x_{gB} to enhance the local search ability or landing far from x_{gB} to keep the global search capability. So, MPSO simply makes the particles search the space more pertinently and efficiently to improve both the convergent speed and global search performance without adding much computational cost.

Acknowledgment

This work is partially supported by the National High Technology Research and Development Program of China (863 Program), with grant No.2007AA01Z453, and partially supported by the National Natural Science Foundation of China under grant No.60673020. This work is also financially supported by the Research Fund for the Doctoral Program of Higher Education (RFDP) in China and and China Postdoctoral Science Foundation funded project grants No.20070420257.

5. REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, IEEE Service Center, Vol. 4. Piscataway, NJ, 1995, pp. 1942-1948.

[2] Y. H. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer", IEEE World Congress on Computational Intelligence, Alaska, ALTEC, vol. 1, 1998, pp. 69-73.

[3] Paul Blenkhorn and David Gareth Evans, "A Screen Magnifier Using 'High Level' Implementation Techniques", IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 14, no. 4, december 2006, pp. 501-504.

[4] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection", Information Processing Letters 85, 2003, pp. 317-325