

Search Performance Improvement for PSO in High Dimensional Space

Toshiharu Hatanaka¹, Takeshi Korenaga¹, Nobuhiko Kondo²
and Katsuji Uosaki³

¹*Department of Information and Physical Sciences, Osaka University*

²*Institute of Intelligent Information and Communications Technology, Konan University*

³*Department of Management and Information Sciences, Fukui University of Technology
Japan*

1. Introduction

Particle swarm optimisation (PSO) was developed by Kennedy and Eberhart in 1995 (Kennedy & Eberhart, 1995) inspired by the collective behaviour of natural birds or fish. PSO is a stochastic optimisation technique that uses a behaviour of population composed by many search points called particle. In spite of easy implementation in computer algorithms, it is well known as a powerful numerical optimizer. In the typical PSO algorithms, a set of particles searches the optimal solution in the problem space efficiently, by sharing the common attractor called global best. There are many modified versions of PSO by improving convergence property to a certain problem. While, a standard PSO is defined by Bratton and Kennedy (Bratton & Kennedy, 2007) to give a real standard for PSO studies. PSO seems as one of the evolutionary computations (ECs), and it has been shown that PSO is comparable to a genetic algorithm (Angeline, 1998). Thus, a lot of studies have demonstrated the effectiveness of PSO family in optimizing various continuous and discrete optimization problems. And a plenty of applications of PSO, such as the neural network training, PID controller tuning, electric system optimisation have been studied and achieved well results (Kennedy, 1997).

However, PSO is often failed in searching the global optimal solution in the case of the objective function has a large number of dimensions. The reason of this phenomenon is not only existence of the local optimal solutions, the velocities of the particles sometimes lapsed into the degeneracy, so that the successive range is restricted in the sub-plane of the whole search hyper-plane. The sub-plane that is defined by finite number of particle velocities is a partial space in the whole search space. The issue of local optima in PSO has been studied and proposed several modifications on the basic particle driven equation (Parsopoulos et al., 2001; Hendtlass, 2005; Liang et al., 2006). There used a kind of adaptation technique or randomized method (e.g. mutation in evolutionary computations) to keep particles velocities or to accelerate them. Although such improvements work well and have ability to avoid fall in the local optima, the problem of early convergence by the degeneracy of some dimensions is still remaining, even if there are no local optima. Hence the PSO algorithm does not always work well for the high-dimensional function.

From this point of view, the purpose of this paper is to improve performance of the PSO algorithm in case of high-dimensional optimization. To avoid such convergence with finite number of particles, we propose a novel PSO model, called as the Rotated Particle Swarm (RPS), where we introduce a coordinate conversion method. The numerical simulation results show the RPS is efficient in optimizing high-dimensional functions.

The remaining parts of this chapter organized as the following. In Section 2, PSO is briefly introduced, and then the early convergence phenomenon is illustrated. The proposed novel PSO model is shown in Section 3. Some results of the numerical studies for the benchmark problems are presented in Section 4, and we mention about some remarks in the last section.

2. Particle Swarm Optimization

Each particle, it is a member of the population, has its own position x and velocity v . A velocity decides a movement direction of a particle. The particles fly around the problem space, searching for the position of optima. Each particle memorizes two positions in order to find a favourite position in the search space. One is its own best position called the personal best and the other is the global best that is the best among all particles, denoted by p and g , respectively. Then, $p^{(i)}$ indicates the best position found by i -th particle from the first time step and $g^{(i)}$ indicates the best position among all p_i in the neighbour particle of i -th particle. Neighbour particle is defined by the topology of particles, which represents the network structure of population. Memories are utilized in adjusting the velocity to find better solutions.

In one of the standard versions of PSO algorithm, the velocity and position are updated at each time step, according to the following two equations,

$$v_d^{(i)} = \chi(v_d^{(i)} + \phi_{1,d}r_{1,d}(p_d^{(i)} - x_d^{(i)}) + \phi_{2,d}r_{2,d}(g_d^{(i)} - x_d^{(i)})) \quad (1)$$

$$x_d^{(i)} = x_d^{(i)} + v_d^{(i)} \quad (2)$$

Here, χ is the constriction coefficient, which prevents explosion of the velocity and balances between exploration and exploitation. The coefficients $r_{1,d}$ and $r_{2,d}$ are random values uniformly distributed over the range $[0, 1]$. These parameters are often set as $\phi_{1,d} = \phi_{2,d} = 2.05$ and $\chi = 0.7298$ (Kennedy & Clerc, 2002). $v_d^{(i)}$ indicates d -th element of velocity vector of i -th particle, and $x_d^{(i)}$ indicates d -th element of position. $p_d^{(i)}$ and $g_d^{(i)}$ represent d -th elements of $p^{(i)}$ and $g^{(i)}$ respectively.

Some theoretical analyses of particle trajectories derived from Eq. (1) and (2) have been performed, where PSO algorithm is simplified (e.g., only one particle, one dimension, no stochastic elements) (Ozcan & Mohan, 1998; Ozcan & Mohan, 1999; Kennedy & Clerc, 2002). In those studies, it is shown that each particle oscillates around the weighted average position of its p and g , and settle down in an equilibrium state where velocities are considerably small until new p or g is found by particle. Note that the particles converge are not always local or global optima. We consider the effect of this kind of convergence property on high-dimensional optimization. Now, we present the experimental results

where a standard PSO algorithm is applied to high-dimension Sphere function. Sphere function is one of the standard test functions without local optima. It is defined as follows,

$$f_1(x) = \sum_{d=1}^D x_d^2 \quad (3)$$

where D represents the number of dimensions.

Population size is set to 20 particles. Star topology, where all particles are interconnected, is used. The average fitness (function value) for 10 dimensions case at each time step over 10 trials is shown in Fig.1.

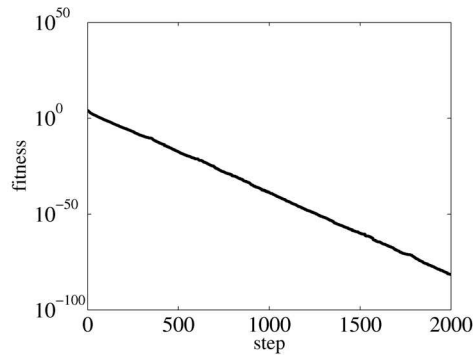


Figure 1. Function value of the global best particle, in case of dimension size $D=10$

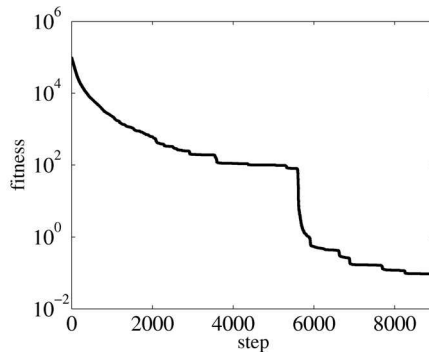


Figure 2. Function value of the global best particle, in case of dimension size $D=100$

Figure 2 shows the average fitness in the same conditions except for the dimension. In this case, a dimension is 100. The function value in 10 dimensions keeps decreasing as the search proceeds, while the improvement of the fitness value in 100 dimensions gets slow. Note that both vertical and horizontal axes are not same order. For higher dimension, the convergence speed is very low, or it does not converge.

We examined the other topologies such as four clusters, pyramid or ring. They have low connectivity to keep diversity in population. Though they yield better results than the Star topology, performance gets worse as the dimension of dimensions increases. When PSO

does not work well in optimizing high-dimensional function like this result, it is frequently observed that diversity of positions and memories of the whole population in some dimensions gets lost without trapping local optima.

It is conjectured that such premature convergence occurs because the velocity updating by Eq. (1) depends only on information of the same dimension. Once the diversity of positions and memories of certain dimension gets low, particles have difficulty in searching in the direction of the dimension. In the next section, we propose the novel velocity model that utilizes information of other dimensions.

3. Rotated Particle Swarm

Now, to consider the conventional way to update the velocity, we rewrite Eq. (1) by vectors and matrixes, such as,

$$v_i = \chi(v_i + \Phi_1(p_i - x_i) + \Phi_2(g_i - x_i)), \tag{4}$$

where

$$\Phi_1 = \text{diag}(\phi_{1,1}r_{1,1}, \phi_{1,2}r_{1,2}, \dots, \phi_{1,D}r_{1,D}) \tag{5}$$

$$\Phi_2 = \text{diag}(\phi_{2,1}r_{2,1}, \phi_{2,2}r_{2,2}, \dots, \phi_{2,D}r_{2,D}). \tag{6}$$

Fig.3 illustrates sample space by $\Phi_1(p^{(i)} - x^{(i)})$, the second term of Eq. (1), in 2 dimensions.

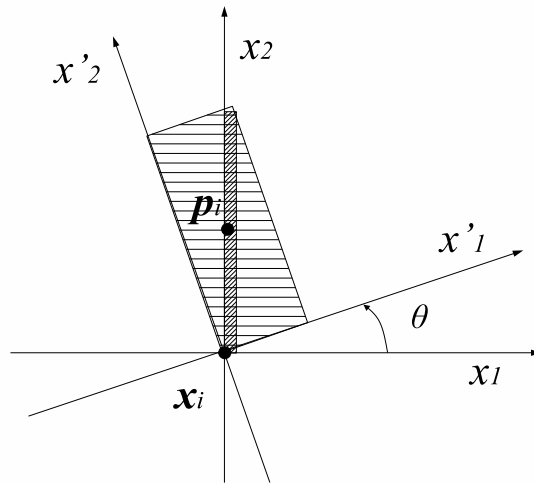


Figure 3. Geometric illustration of sample space by $\Phi_1(p^{(i)} - x^{(i)})$ in 2 dimensions

In $x_1 - x_2$ coordinate system where x_i and p_i are aligned parallel to the axis, the sample space has only one degree of freedom. On the other hand, the sample space is two-

dimensional in the $x'_1 - x'_2$ coordinate system. The same can be said about $\Phi_2(g^{(i)} - x^{(i)})$. The original PSO algorithm was designed by emulating birds seeking food. Birds probably never change the strategy to seek according to whether food exists in the true north or in the northeast. Consequently, particles can search optima even if axes are rotated. We introduce the coordinate conversion to the velocity update.

Before the values are sampled by $\Phi_1(p^{(i)} - x^{(i)})$ and $\Phi_2(g^{(i)} - x^{(i)})$, the coordinate system is rotated. Thus information of other dimensions is employed in calculating each component of velocity. In the proposed method, the velocity update equation Eq. (5) is substituted into

$$v_i = \chi(v_i + A^{-1}\Phi_1A(p_i - x_i) + A^{-1}\Phi_2A(g_i - x_i)) \tag{7}$$

where A is $D \times D$ matrix to rotate the axes. For the remainder of this paper, we refer to the PSO algorithm with this proposed velocity update method as the Rotated Particle Swarm (RPS). The RPS is designed for high-dimensional optimization. Therefore matrix computation is time-consuming if axes are arbitrarily rotated. To avoid it, in this study, certain number of axes are randomly selected without duplication and paired respectively. Only pairs of selected axes are rotated by θ . Most of the elements of A determined by this means are zero. For example, if $D = 5$ and selected pairs of axes are $x_1 - x_2$ and $x_3 - x_5$,

$$A = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 & 0 \\ 0 & 0 & \cos \theta & 0 & -\sin \theta \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \sin \theta & 0 & \cos \theta \end{pmatrix} \tag{8}$$

4. Numerical Simulation

The following well-known test functions, i.e. Sphere function, Quatric function (De Jong F4), Rosenbrock function, Griewank Function, and Rastrigin function) are used to evaluate convergence property of the proposed method.

Sphere function: $f_1(x) = \sum_{d=1}^D x_d^2 \tag{9}$

Quatric function: $f_2(x) = \sum_{d=1}^D dx_d^4 \tag{10}$

Rosenbrock function: $f_3(x) = \sum_{d=1}^{D-1} 100(x_{d+1} - x_d^2)^2 + (x_d - 1)^2 \tag{11}$

Griewank function: $f_4(x) = \frac{1}{4000} \sum_{d=1}^D x_d^2 - \prod_{d=1}^D \cos(\frac{x_d}{\sqrt{d}}) + 1 \tag{12}$

Rastrigin function: $f_5(x) = \sum_{d=1}^D x_d^2 + 10 - 10 \cos(2\pi x_d) \tag{13}$

Sphere function, Quatric function and Rosenbrock function are unimodal. The others are multimodal functions. Rosenbrock function has dependence among variables.

For each function, search range and initialization range were defined as listed in Table.1. Rosenbrock function has its global optimum at $[1, 1]^D$ and the others have at the origin. We use asymmetric initialization method, in which initial population is distributed only in a portion of the search range (Angeline, 1998). Optimal population size is problem-dependent. In this study, population size is set to 20 which is commonly used (Bergh & Engelbrecht, 2001). The canonical PSO and the RPS are tested with Star and Von Neumann topology. Von Neumann topology is grid-structured and has been shown to outperform other topologies in various problems (Kennedy & Mendes, 2002). In the RPS algorithm, the angle of rotation θ is set to $\pi/5$ and number of axes to be rotated is 40% of number of dimensions. The number of dimension D is set to 30, 100, and 400. Each experiment is run 20 times in each condition and the fitness at each time step is averaged.

In the results of Sphere function shown in Fig.3 - 5, in these figures, the bold lines show the convergence properties of the conventional PSO and the thin lines show the convergence properties of the proposed PSO. The solid lines indicate using the star topology and the dash lines indicate using Von-Neumann topology.

We can see a great difference in convergence ability between the RPS and the canonical PSO. Especially in $D=400$ though it becomes difficult for the canonical PSO to keep converging to the optimum, the fitness of RPS keeps decreasing. Similarly, in the case of Quatric, Rosenbrock and Griewank shown in Fig.6-14, for every functions, a convergence speed and final obtained fitness of the RPS get relatively good compared with the canonical PSO as the number of dimension increases.

Function	Search range	Range of the initial population
Sphere	$[-50, 50]^D$	$[25, 40]^D$
Quatric	$[-20, 20]^D$	$[10, 16]^D$
Rosenbrock	$[-100, 100]^D$	$[50, 80]^D$
Griewank	$[-600, 600]^D$	$[300, 500]^D$
Rastrigin	$[-5.12, 5.12]^D$	$[1, 4.5]^D$

Table 1. Search range and initialization for each function

5. Conclusion

The purpose of this study is to improve the early convergence of the particle swarm optimization in high-dimensional function optimization problems by the degeneracy. We have proposed the novel particle driven model, called Rotated Particle Swarm (RPS). It employs a coordinate conversion where information of other dimensions is utilized to keep diversity of each dimension. It is very simple technique and it is able to apply to any modified PSO model. The experimental results have shown that the proposed RPS is more efficient in optimizing high-dimensional functions than a standard PSO. The proposed RPS indicated remarkable improvement in convergence for high-dimensional space, especially in unimodal functions. An appropriate selection of rotated angles and dimensions are the future study, however it is envisioned that the performance of the proposed algorithm has robustness for such parameter settings. To compare the proposed method to the other modifications and to develop more powerful algorithm by combining with local optima technique are now under investigation.

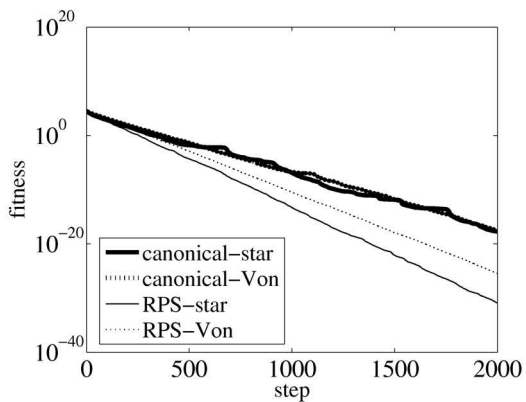


Figure 4. Sphere function ($D=30$)

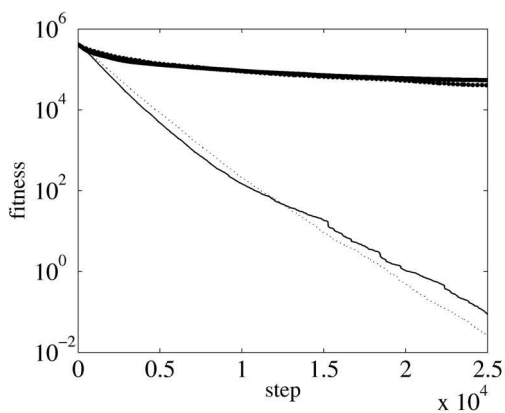


Figure 5. Sphere function ($D=100$)

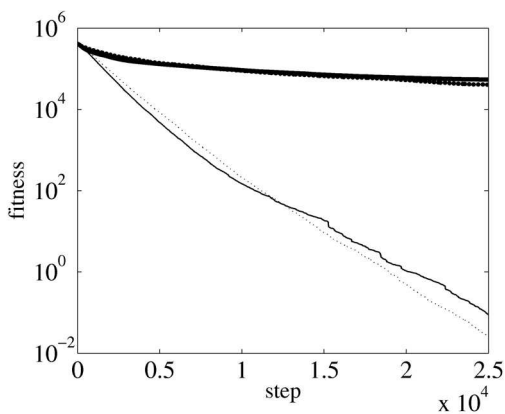
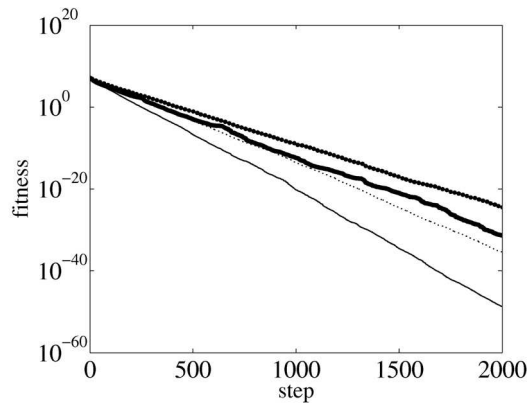
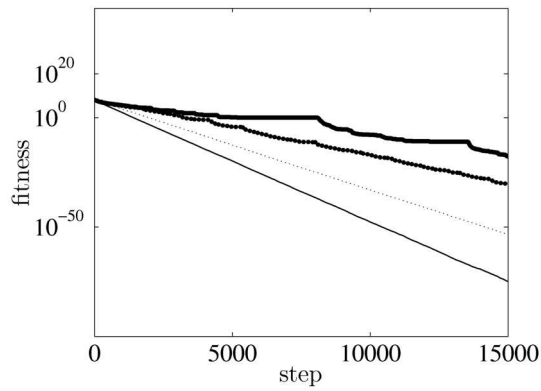
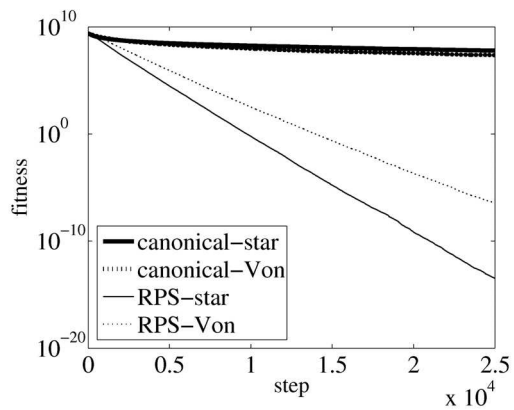


Figure 6. Sphere function ($D=400$)

Figure 7. Quatric function ($D=30$)Figure 8. Quatric function ($D=100$)Figure 9. Quatric function ($D=400$)

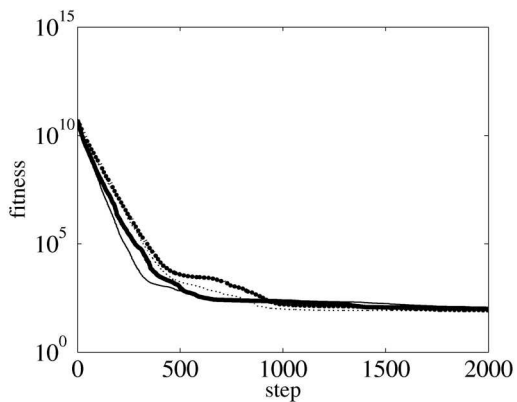


Figure 10. Rosenbrock function ($D=30$)

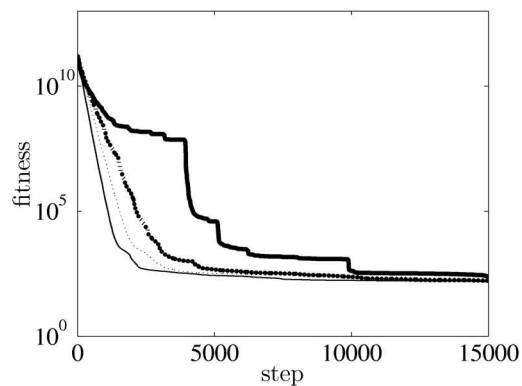


Figure 11. Rosenbrock function ($D=100$)

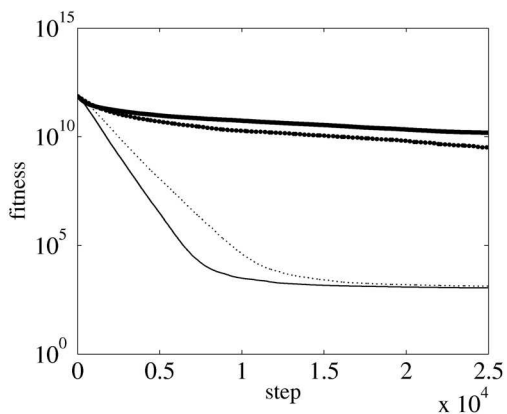
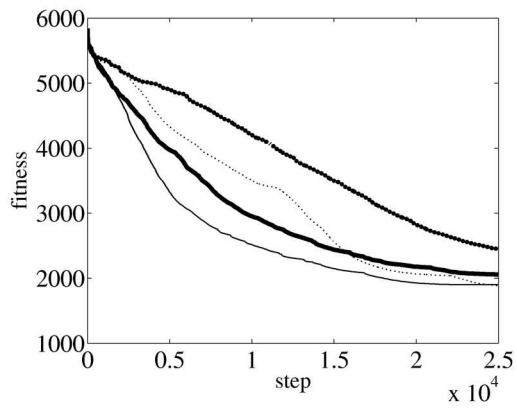
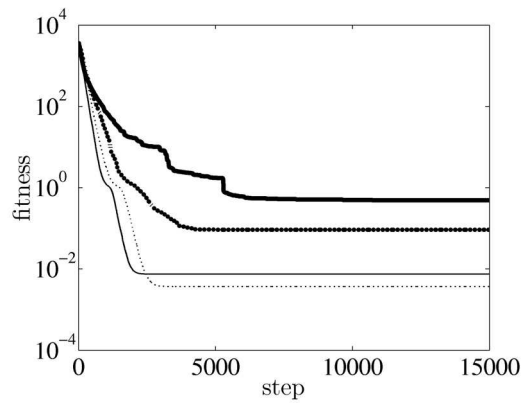
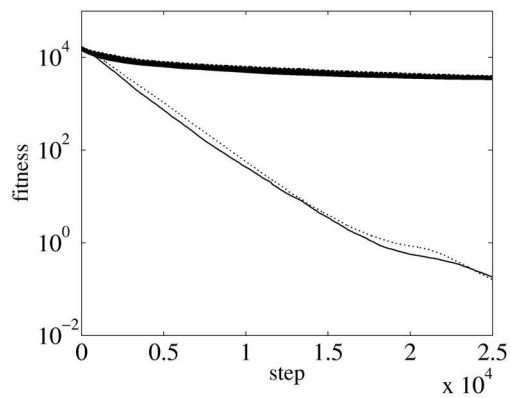
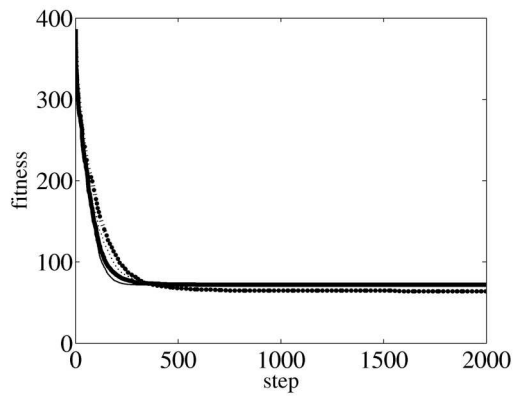
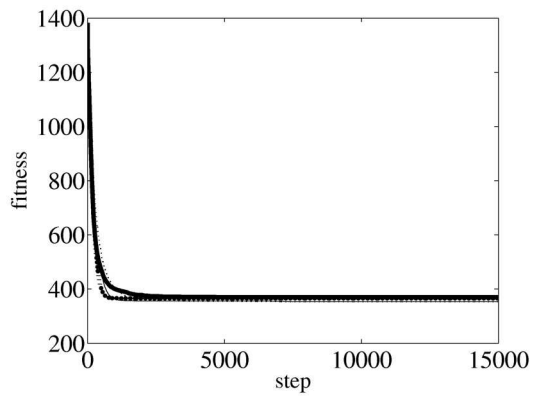
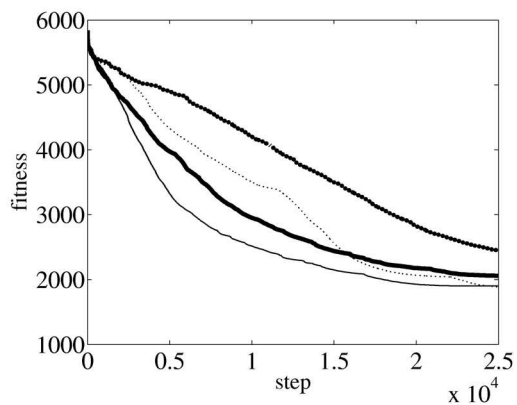


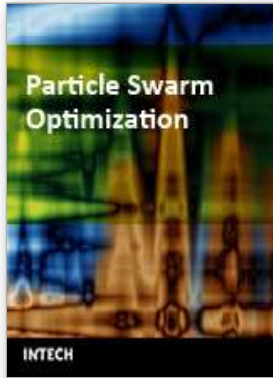
Figure 12. Rosenbrock function ($D=400$)

Figure 13. Griewank function ($D=30$)Figure 14. Griewank function ($D=30$)Figure 15. Griewank function ($D=30$)

Figure 16. Rastrigin function ($D=30$)Figure 17. Rastrigin function ($D=100$)Figure 18. Rastrigin function ($D=400$)

6. References

- Angeline, P.J. (1998). Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences, *Evolutionary Programming VII, in Lecture Notes in Computer Science*, Vol. 1447, pp. 601-610, 3-540-64891-7, Springer, London
- Bratton, D. & Kennedy, J. (2007). Defining a Standard for Particle Swarm Optimization, *Proceedings of Swarm Intelligence Symposium 2007*, pp. 120-127, 1-4244-0708-7, Honolulu, April 2007
- Hendtlass, T. (2005). A particle swarm algorithm for high dimensional, multi-optima problem spaces, *Proceedings of Swarm Intelligence Symposium 2005*. pp. 149-154, Pasadena, June 2005
- Kennedy, J. (1997). The particle swarm: Social Adaptation of Knowledge, *Proceedings of IEEE International Conference on Evolutionary Computation*, pp.303-308, 0-7803-3949-5, Indianapolis, April 1997
- Kennedy, J. & Clerc, M. (2002). The particle swarm: Explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, 58-73, 1089-778X
- Kennedy, J. & Eberhart, R.C. (1995). Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942-1948, 0-7803-2768-3, Perth, November 1995
- Kennedy, J. & Mendes, R. (2002). Population structure and particle swarm performance, *Proceedings of 2002 IEEE Congress on Evolutionary Computation*, Vol. 2, pp.1671-1676, Honolulu, May 2002
- Kennedy, J. & Spears, W.M. (1998). Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator, *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 78--83, 0-7803-4869-9, Anchorage, May 1998
- Liang, J.J., Qin, A.K., Suganthan, P.N. & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE Transactions on Evolutionary Computation*, Vol.10, No.3, 281-295, 1089-778X
- Ozcan, E. & Mohan, C.K. (1998). Analysis of a Simple Particle Swarm Optimization System, *Intelligent Engineering Systems Through Artificial Neural Networks*, Vol. 8, pp. 253-258, 0-7918-0051-2
- Ozcan, E. & Mohan, C.K. (1999). Particle swarm optimization: Surfing the waves, *Proceedings of 1999 IEEE Congress on Evolutionary Computation*, Vol. 3, pp.1939-1944, Washington, July 1999
- Parsopoulos, K., Plagianakos, V. P. , Magoulas, G. D. & Vrahatis, M. N. (2001). Stretching technique for obtaining global minimizers through particle swarm optimization, *Proceedings of the Particle Swarm Optimization Workshop*, pp. 22-29, Indianapolis, 2001
- van den Bergh, F. & A.P. Engelbrecht, A.P. (2001). Effects of Swarm Size on Cooperative Particle Swarm Optimizers, *Proceedings of the Genetic and Evolutionary Computation Conference*, pp.892-899, San Francisco, July 2001



Particle Swarm Optimization

Edited by Aleksandar Lazinica

ISBN 978-953-7619-48-0

Hard cover, 476 pages

Publisher InTech

Published online 01, January, 2009

Published in print edition January, 2009

Particle swarm optimization (PSO) is a population based stochastic optimization technique influenced by the social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. This book represents the contributions of the top researchers in this field and will serve as a valuable tool for professionals in this interdisciplinary field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Toshiharu Hatanaka, Takeshi Korenaga, Nobuhiko Kondo and Katsuji Uosaki (2009). Search Performance Improvement for PSO in High Dimensional Space, Particle Swarm Optimization, Aleksandar Lazinica (Ed.), ISBN: 978-953-7619-48-0, InTech, Available from:
http://www.intechopen.com/books/particle_swarm_optimization/search_performance_improvement_for_pso_in_high_dimensional_space

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.