

A Modified Particle Swarm Optimization Algorithm for Engineering Optimizations

Shafi Ullah Khan¹

*College of Electrical Engineering, Zhejiang University
Hangzhou, China 310027
E-mail: eesyayang@zju.edu.cn*

Lei Liu, Luyu Wang, Shiyong Yang

*College of Electrical Engineering, Zhejiang University
Hangzhou, China 310027*

Particle Swarm Optimization (PSO) is a population based optimal method and very simple in both theory and numerical implementation. Nowadays, PSO has been recognized as a paradigm for numerical optimizations; however, PSO is easily trapped into a local optimum when solving multidimensional and complex problems. In order to overcome this difficulty, this paper presents a modified PSO with a dynamic inertia weight and an adaptive mutation operator. To verify the proposed PSO, we test it numerically on a set of well known bench mark functions as well as on an engineering problem, as to which it has shown better performance and efficiency while compared to the basic PSO and Beta PSO.

*CENet2015
12-13 September 2015
Shanghai, China*

¹Speaker

1. Introduction

Recently, various improvements have been made in the way of modern optimization. Some promising techniques include, to name but a few, evolutionary programming, artificial immune system, ant colony optimization and artificial bee colony. Such techniques are often referred to as heuristic methods on account of the randomization presented in the initial phase. Using randomized parameters facilitates an algorithm capable of handling linear and nonlinear situations. It can be seen in the literature that such techniques are applicable in many fields. One of these methods is the particle swarm optimization [1]. PSO is much more popular because it is computationally very simple and highly applicable in robotics [2], electrical power systems [3] and other engineering fields [4]; however, there is not any universal algorithm which will find global optimal solutions of all problems. For example, PSO sometimes fails to find the global optimum of higher dimensional or complex problems; consequently, PSO has been groomed to anticipate premature convergence and different modifications are proposed.

One direction in modifying PSO is to vary the parameters; moreover, the inertia weight (w) is an important one. Shi and Eberhart were the first researchers to introduce the inertia weight to balance the local and global searches [5]. A proper adjustment of the inertia weight will increase the convergence speed and also diversity of the particles.

In order to improve the performance, another variation of PSO using a linearly decreased inertia weight was proposed [6]. Hui et. al proposed a new inertia weight mechanism. The inertia weight should be controlled by the Euclidean distance of the particle in question and the best particle of the swarm [7]. In addition, many researchers use different mutation operators. The adaptive mutation operator was introduced by Tang and Zhao [8]. Actually, they added a uniform random number to the global best particle. A power distribution was used [9]. The beta distribution was introduced by Pant et al [10]. They proposed two variations of PSO: AmPSO1 and AmPSO2. In AmPSO1, the global best is mutated, while in AmpSO2 the local best particle is mutated. It was proposed to couple the opposition based PSO with Cauchy mutation [11]. They accelerate the convergence by using opposition based learning to each particle. These mutation particles are used to increase the possibility of escaping from local optimal points. A variation of PSO using a student T mutation was introduced [12]. The global best particle is mutated and compared to the original g_{best} particle. If the mutated particle is the better, it will then replace the original g_{best} particle. Hui Wang proposed that one mutation operator is not enough, so he applied mutation operators such as Gaussian, Cauchy and levy, and dynamically selected a specific mutation [13].

In this paper, we modified the conventional PSO by introducing a variable inertia weight and combining different mutation operators in the same algorithm.

2. A Modified Particle Swarm Optimization

From the literature, it is clear that particles will have zero velocity in some cases. To address this problem, we modified the PSO by assigning random values of the inertia weight and also by introducing a new mutation strategy. To facilitate the description, Fig. 1 gives the flow chart of the modified PSO.

To compromise the balance between the exploration and the exploitation searches, the mutation operator could be conducted on p_{best} and g_{best} . If the fitness value of p_{best} and g_{best} does not improve for some generations (n) (n is a constant), which means that the p_{best} and g_{best} are trapped to local optima. To overcome this difficulty, we mutate the p_{best} and g_{best} . Our new strategy is different from the previous work because it checks the changes of p_{best} and g_{best} instead of the fitness values of each particle.

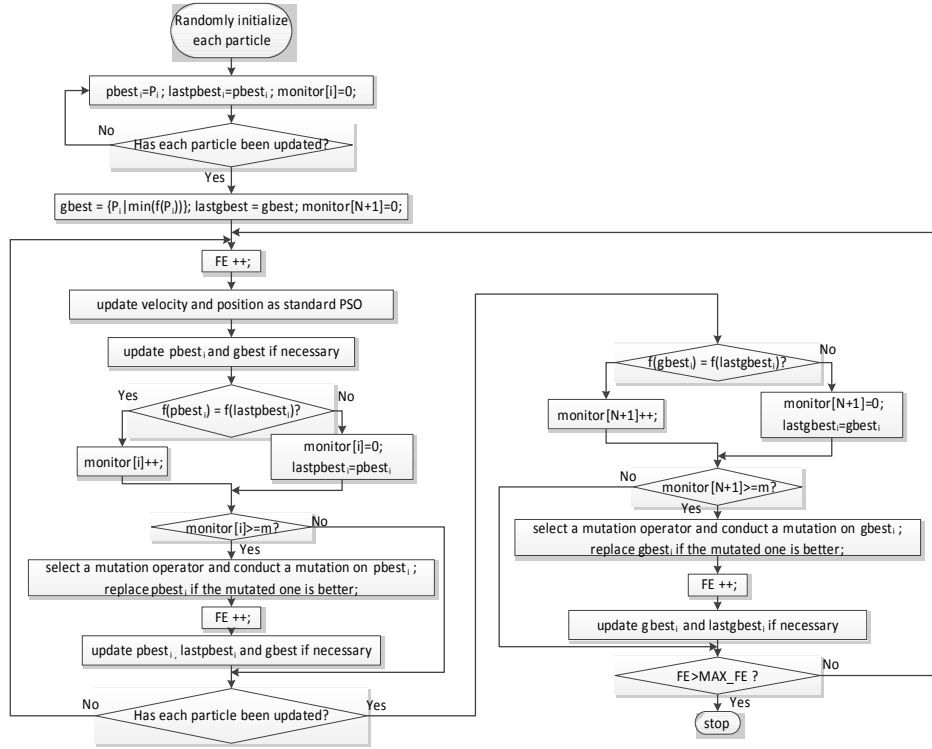


Figure1: Convergence Comparison of Different Algorithms

It should be noted that different mutations are used to control the possibility of premature convergence; however, all of these mutations are problem-dependent. For example, the student T mutation is used for local searches and it may fail when the current search position is very far away from the optimal position. To improve the performance of PSO, a selection ratio is introduced and used to design a special methodology to choose different mutation operators; moreover, the mutation operators used are defined as:

Function mutation

$$Pbest_{ij}^1 = pbest_{ij} + function_j() \quad (2.1)$$

$$Gbest_j^1 = gbest_j + function_j() \quad (2.2)$$

Beta mutation

$$Pbest_{ij}^1 = pbest_{ij} + beta_j() \quad (2.3)$$

$$Pbest_{ij}^1 = gbest_{ij} + beta_j() \quad (2.4)$$

Student T mutation

$$Pbest_{ij}^1 = pbest_{ij} + student_j() \quad (2.5)$$

$$Gbest_j^1 = gbest_j + student_j() \quad (2.6)$$

where $function_j()$, $beta_j()$ and $student_j()$ produce random numbers by using function, Beta and student T distributions, respectively.

In our methodology, we give each mutation operator an independent selection ratio. The mutation ratios for function, beta and student T mutations are, respectively, A , B , and C ($A+B+C=1$).

At the start stage, A , B and C are set to $1/3$ to ensure that each mutation has an equal chance of being selected. In the optimizations, the mutation ratio is updated according to the success rate of the previous mutations of the operator; moreover, the updating formulate for the selection ratios are proposed as:

$$A = q + (1 - 3q)suc_f / suc_m \quad (2.7)$$

$$B = q + (1 - 3q)suc_beta / suc_m \quad (2.8)$$

$$C = q + (1 - 3q) \text{suc_student} / \text{suc_m} \quad (2.9)$$

A , B , and C are then normalized using:

$$D = D / (A + B + C) \quad (2.10)$$

where $D = A, B, C$; suc_c is the number of successful mutations of a mutation operator in its previous mutation operations. The minimum ratio for each mutation operator is predefined by q and its value is 0.05. At the end of each generation, A , B and C are updated. To select the best mutation, we apply the roulette wheel selection method according to the selection ratio of a mutation operator. Obviously, a mutation operator having a longer stay will be chosen of high probability; moreover, to maintain the diversity of the particles, the inertia weight is dynamically varied by using:

$$w = 0.02 + \text{rand}() \quad (2.11)$$

3. Numerical Examples

3.1 Function Test

Firstly, some well-known mathematical test functions are selected to verify the performance of the proposed PSO [14, 15]. The definitions of the functions are given in **Table 1**. All mathematical functions are minimization problems, and the function dimensions are 10 and 30 respectively.

Functions	Decision Scope	Optimum Fitness Value
$f_1 = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$x_i \in [-600, 600]$	0.00
$f_2 = x^2 + 10^6 \cdot \sum_{i=1}^n x_i$	$x_i \in [-10, 10]$	0.00
$f_3 = (10^6)^{i-1} / (n-1) \times x_i^2$	$x_i \in [-100, 100]$	0
$f_4 = \sum_{i=1}^n x_i ^{i+1}$	$x_i \in [-10, 10]$	0.00
$f_5 = \sum_{i=1}^n x_i \sin(x_i) + 0.1 x_i $	$x_i \in [-10, 10]$	0.00

Table 1: Mathematical Test Functions

In order to perform a fair comparison, we take the same number of the swarm size of 20 and 2000 iterations; besides, we take the cognitive coefficient (c_1) and social constants (c_2) to be 2 for all algorithms.

We make comparison among the conventional PSO, the adaptive PSO using beta mutation (Beta PSO) and our proposed PSO [1,10]. We run every algorithm 30 times and compare the find mean performance for these runs. **Table 2** tabulates the average performances of different algorithms for solving the 10 dimensional problems, while **Table 3** represents the corresponding results for the 30 dimensional benchmark problems. View the experimental results as we categorized the algorithms into three classes; (1) the first class-the algorithm shows better performance as compared to others; (2) the second class-the algorithm shows a significant improvement; (3) the third class-the algorithm presents poor performance. From **Table 2** and **Table 3**, we conclude that the proposed PSO is a first class algorithm for the test functions f_2, f_3 , and f_4 . Similarly, the proposed PSO is considered as a second class algorithm for the test functions f_1 and f_5 while the traditional PSO shows poor performance on test functions f_1, f_3, f_4 and f_5 .

Functions	f_1	f_2	f_3	f_4	f_5
Basic PSO	9.240	4.148×10^{-12}	10.6338	8.1324	3.9310
Beta PSO	0.0156	7.489×10^{-30}	$1.2985 \times 10^{+4}$	5.3461	2.1174
Modified PSO	0.0082	1.0972×10^{-143}	1.6332×10^{-36}	4.59×10^{-80}	1.364×10^{-15}

Table 2: Average Performances of Different Algorithms on the 10 Dimensional Test Functions

Functions	f_1	f_2	f_3	f_4	f_5
Basic PSO	0.0240	0.1640	1.2328×10^6	10.3703	3.0467×10^{10}
Beta PSO	0.0176	0.0123	1.2985×10^5	9.2380	$3.114 \times 10^{+2}$
Modified PSO	7.658	1.5374×10^{-60}	9.0139×10^{-12}	3.6913×10^{-66}	7.182×10^{-10}

Table 3: Average Performances of Different Algorithms on the 30 Dimensional Test Functions

Clearly, from the above discussions it is obvious that the proposed approach is more suitable for unimodal as well as multimodal or complex most test functions; moreover, to give an intuitive impression about the convergence of every algorithm, we plot the convergence graph for a typical run in Fig.2 for f_2 due to the space limitations. From the graph we observe that after a small number of generations, the proposed PSO has found the global minimum of the test function, demonstrating that the convergent speed of the proposed modified PSO is very fast. Therefore, the modified PSO is a more robust and efficient global optimizer.

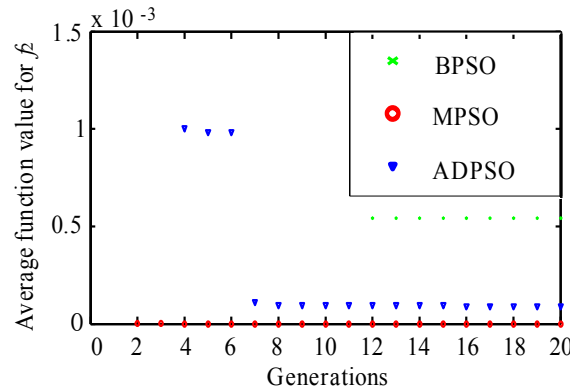


Figure 2: Convergence Comparison of Different Algorithms

3.2 Application

To verify our proposed PSO for engineering problem, we then use it to solve a practical design problem the TEAM workshop problem 22 of a three-parameter optimization problem concerning the configuration of a super-conducting magnetic energy storage (SMES) [16]. The configuration conforms to three objectives;

- 1) the stored energy in the device should be 180 MJ ;
- 2) the magnetic field must not violate the quench condition;
- 3) the stray field should be eliminated as much as possible .

$$\begin{cases} \min f = B_{stray}^2 / B_{norm}^2 + |Energy - E_{ref}| / E_{ref} \\ \text{s.t. } J_i \leq (-6.4 |(B_{max})_i| + 54) (A/mm^2) (i=1,2) \end{cases} \quad (3.1)$$

where $Energy$ is the magnetic energy stored in the SEMC device; $E_{ref} = 180$ MJ and $B_{norm} = 3 \mu T$, J_i and $(B_{max})_i$ is the current density and the maximum magnetic flux intensity in the i^{th} coil respectively, B_{stray}^2 is evaluated along 22 equidistant points along line A and line B , as shown in Fig. 2, and defined as

$$B_{stray}^2 = \sum_{j=1}^{22} |B_{stray, j}|^2 / 22 \quad (3.2)$$

For a fair comparison, we set equal parameters to the maximum generation of 50, swarm size of 15, and the dimension of 3, and c_1 and c_2 of 2. We run each algorithm five times and the average objective function values are recorded in Table 3. The results validate that the performance of the proposed PSO is much better when compared with the other two PSO's.

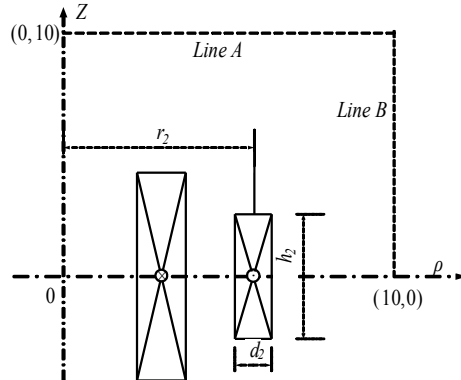


Figure 3: Schematic Diagram of SMES

Problem	Basic PSO	Adaptive PSO(Beta PSO)	MPSO
Objective function	0.0996	0.01163	0.0918

Table 4: Comparison of Different Algorithms on Solving TEAM Workshop Problem 22

4. Conclusion

This paper presents a new methodology for mutations. The mutation operator is conducted on p_{best} and g_{best} by using a roulette wheel selection method according to a specially designed selection ratio. From the experimental results we've observed that the proposed modified PSO is the most suitable one for finding higher quality solutions when compared to the basic PSO and ADPSO (Beta distribution). To sum up, the proposed modified PSO gives the best performance on all of the test functions and case studies.

References

- [1] J. Kennedy, R. Eberhart (1995) *Particles swarm optimization*. Proceeding of IEEE International Conference on Neural Networks, IEEE Press, Piscataway, NJ, 1942-1948
- [2] W. Zhang, Y. Liu (2005) *Adaptive particle swarm optimization for reactive power and voltage control in power systems*. Lecture Notes in Computer Science, LNCS, Springer, Berlin, 3612: 449-452
- [3] J. C. Wen, E. Bompard (2005) *A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimization*. Mathematics and Computers in Simulation, 68(1): 57-65
- [4] V. Vassiliadis and G.Dounias (2009) *Nature-inspired intelligence a review of selected method sand applications*. International Journal on Artificial Intelligence Tools, volume.18(4): 487-516, doi: 10.1142/S021821300900024
- [5] R. C. Eberhart, Y. Shi (2001) *Particle swarm optimization: developments, application and resources*. IEEE Int. Conf. Evolutionary Computation, IEEE Press, Piscataway, NJ, 1:84-88
- [6] X. Liu, Q. Wang, H. Liu, L. Li (2009) *Particle swarm optimization with dynamic inertia weight and mutation*. Third international Conference on Genetic and Evolutionary Computing, IEEE Press, Piscataway, NJ, 620-623

- [7] H. Lu, X. Chen (2011) *A new particle swarm optimization with dynamic inertia weight for solving constrained optimization problems*. Information Technology Journal, 10: 1536-1544, doi: 10.3923/itj.2011.1536.1544
- [8] J. Tang, X. Zhao (2009) *Particle swarm optimization with adaptive mutation*. WASE International Conference on Information on Engineering, IEEE Press, Piscataway, 234-237
- [9] X. Wu, M. Zhong (2009) *Particle swarm optimization based on power mutation*. ISECS International Colloquium on Computing Communication, Control, and Management, IEEE Press, Piscataway, NJ, 464-467
- [10] M. Pant, R Thangarija, A. Abraham (2007) *Particle swarm optimization using adaptive mutation*. 19th International Conference on Database and Expert Systems Applications, IEEE Press, Piscataway, NJ, 97-102
- [11] H. Wang, Y. Liu, S.Y. Zeng, H. Li, C. H. Li (2007) *Opposition based particle swarm algorithm with Cauchy mutation*. IEEE Congress on Evolutionary Computation, IEEE Press, Piscataway, NJ, 4750-4756
- [12] M Imran, T. Hussein, B. Pahat, R Hashim, N. E. A Khalid (2012) *Opposition based particle swarm optimization with Student T mutation (OSTPSO)*. 4th Conference on Data Mining and Optimization, Langkawi, Malaysia, IEEE Press, Piscataway, NJ, 80-85
- [13] H. Wang, W. Wang, Z. Wu (2013) *Particle swarm optimization with adaptive mutation for multimodal optimization*. Journal of Applied Mathematics and Computation volume 221, 296-305
- [14] N. Singh, S. Singh, S. B. Singh, S. Arora (2012) *Half mean particle swarm optimization algorithm*. International Journal of Scientific & Engineering Research, 3(8):1-9
- [15] J. J. Jamian, M. N. Abdullah, H. Mokhlis, M.W. Mustafa, A. H. A. Bakr (2014) *Global particle swarm optimization for high dimensional numerical function analysis*. Journal of Applied Mathematics, Article ID329193,1-14(2013) [http:// www.comumag.co.uk/team.html](http://www.comumag.co.uk/team.html)
- [16] G. Schoenwetter, C. Magele, K. Preis, C. Paul, W. Renhart, K.R. Richer (1995) *Optimization of SMES solenoids with regards to their stray fields*. IEEE Trans. Magn. 31(3): 1940-1943