J. Basic. Appl. Sci. Res., 2(2)1362-1367, 2012
© 2012, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

A Modified PSO Using Great Deluge Algorithm for Optimization

Sajjad Ghatei[1], Rahim Pasha Khajei[2], Mahdi Samimi Maman[3] and Mohammad Reza Meybodi[4]

[1] Department of Computer Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran
[2] Department of Computer Engineering, Osku Branch, Islamic Azad University, Osku, Iran
[3]Department of Arts, Computing, Engineering and Sciences, Sheffield Hallam University, Sheffield, United Kingdom
[4]Department of Computer Engineering, Amirkabir University of Tehran, Tehran, Iran

ABSTRACT

A modified Particle Swarm Optimization (PSO) using Great Deluge Algorithm (GDA) called MPSO for optimization is presented in this paper. In the proposed algorithm, the range for achieved answers is defined that is the same parameter used in the GDA called "water level". Amount of this range reduces or increases regarding to algorithm's property being used in terms of minimum or maximum during the time. Difference of the proposed algorithm with previous PSOs is that, particles are given a second chance using GDA. So if a particle is trapped in the local optimum may get rid of it. New algorithm has been tested on some standard functions and its performance has been compared with standard PSO. Test results indicate that the proposed method significantly improves the ability of PSO of escaping from the local optimal raise and increases the accuracy and the convergence rate.

KEY WORDS: Modified PSO (Modified Particle Swarm Optimization), Particle Swarm Optimization, Great Deluge Algorithm, Optimization, WL (*Water Level*)

INTRODUCTION

Particle swarm optimization is a stochastic population based optimization algorithm, firstly introduced by Kennedy and Eberhart in 1995 [(Kennedy, J. and R.C., Eberhart, 1995-1), (Kennedy, J. and R.C., Eberhart, 1995-2)]. In PSO algorithm, each member of the population is called a ''particle'', and each particle flies around in the multidimensional search space with a velocity, which is constantly updated by the particle's own experience and the experience of particle's neighbors or the experience of the whole swarm. It has already been applied in many areas, such as function optimization, artificial neural network training, pattern classification and fuzzy system control. The advantages of PSO are that PSO is rapidly converging towards an optimum, simple to compute, easy to implement and free from the complex computation in genetic algorithm (e.g., coding/decoding, crossover and mutation). However, PSO does exhibits some disadvantages: it sometimes is easy to be trapped in local optima, and the convergence rate decreased considerably in the later period of evolution; when reaching a near optimal solution, the algorithm stops optimizing, and thus the accuracy the algorithm can achieve is limited [Yang, X., J., Yuan and H., Mao, 2007].

Various attempts have been made to overcome the problem. Among them, many approaches and strategies are proposed to enhance the performance of PSO via adjusting inertia weight, Such as, Fuzzy adaptive particle swarm optimization [Shi, Y.H. and R.C., Eberhart, 2001], Linearly Decreasing Weight (LDW) [Shi, Y.H. and R.C., Eberhart, 1998], increasing inertia weight [Zhang, Y.L., L.H., Ma, L.Y., Zhang and J.X., Qian, 2003], and randomized inertia weight [(Eberhart, R.C. and Y.H., Shi, 2001), (Zhang, L.P., H.J., Yu, D.Z., Chen and S.X., Hu, 2004)], etc. The strategy of linearly decreasing weight (LDW) is most commonly used, and it can improve the performance of PSO to some extent, but it may be trapped in local optima and fail to attain high search accuracy.

Another method of revelation, which is very similar to the method of simulation annealing [Petrovic, S., Y., Yang and M., Dror, 2007] is great deluge algorithm that was introduced in 1993 by Dueck [Dueck, G., 1993]. In general, this algorithm with an estimated value starts its work and this is considered as a primary answer of the question. In next steps, generating more answers in the range of the primary answer, the algorithm continues its work and results each time are compared with amount of a predetermined parameter that is called "*water level*" (WL). If the produced answer is greater than WL, it is accepted and is rejected otherwise. Amount of the WL increases each time with a specified amount of UP [(Dueck, G., 1993), (Burke, E., Y., Bykov, J., Newall and S., Petrovic, 2003)]. In [Dueck, G., 1993] this algorithm is applied for solving 442 and 532 cities Traveling Salesman Problem. Burke in [(Burke, E., Y., Bykov, J., Newall and S., Petrovic, 2003), (Burke, E., Y., Bykov, J., Newall and S., Petrovic, 2004)] used this algorithm for solving examination time tabling.

***Corresponding Author:** Sajjad Ghatei, Department of Computer Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran. Email: s.ghatei@iau-ahar.ac.ir

In [Nahas, N., D., Ait-Kadi and M., Nourelfath, 2006], Nahas et al. exploit this method as a new local search approach for solving the buffer allocation problem in unreliable production lines. In [Nahas, N., M., Nourelfath and D., Ait-Kadi, 2007-1], authors solve the redundancy allocation problem using a hybrid method of the ant colony and extended great deluge metaheuristics. In [Nahas, N., M., Nourelfath and D., Ait-Kadi, 2007-2], authors have developed a two-phase extended great deluge method to solve efficiently dynamic layout problem. In [Khatab, A., Ait-Kadi, D., Artiba, A., 2008], Khatab et al. exploit this algorithm to solve selective maintenance optimization problem for series– parallel systems.

A modified PSO with great deluge algorithm called MPSO is proposed in this paper. The purpose of this model is to reach to a balance in the search by combining the accuracy of great deluge algorithm with the optimization speed of particle swarm optimization.

This article is segmented as follows: The second part briefly introduces the standard of PSO. In the third part convergence method of the primary PSO and another kind of PSO convergence is introduced. The fourth part includes the proposed model. Test results are shown in the fifth section and the final part focuses on conclusion.

STANDARD PSO

In the original PSO, which is proposed by Kennedy and Eberhart, the velocity and position updating rule is given by Eq. (1) to (2),

$$v_{id}^{t+1} = v_{id}^{t} + c_1 r_1 (pbest_{id}^{t} - x_{id}^{t}) + c_2 r_2 (gbest_{d}^{t} - x_{id}^{t}), \qquad 1$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1}, i = 1,2,\ldots,n, \qquad 2$$

where c_1 and c_2 are constants named acceleration coefficients. r_1 and r_2 are two independent random numbers uniformly distributed in the range of [0, 1]. $v_i \in [-v_{\max}, v_{\max}]$, where $v_{\max}$ is a problem-dependent constant defined in order to clamp the excessive roaming of particles. $pbest_{id}^{t}$ is the best previous position along the d^{th} dimension of particle i in iteration t (memorized by every particle); $gbest_{d}^{t}$ is the best previous position among all the particles along the d^{th} dimension in iteration t (memorized in a common repository).

The original PSO is improved in [Shi, Y.H. and R.C., Eberhart, 1999] by modifying Eq. (3),

$$v_{id}^{t+1} = v_{id}^{t} + c_1 r_1 (pbest_{id}^{t} - x_{id}^{t}) + c_2 r_2 (gbest_{d}^{t} - x_{id}^{t}), \qquad 3$$

where $w>=0$ is defined as inertia weight factor. Empirical studies of PSO with inertia weight have shown that a relatively large w have more global search ability while a relatively small w results in a faster convergence.

THE CONVERGENCE CRITERION OF PSO

Generally, the definition of convergence is that a system or process reaches a stable state. For the population-based optimization algorithm, the convergence of algorithm can be defined in terms of either individual or the whole swarm. For instants, there are two convergence definitions for Genetic Algorithm. Van Den Bergh gave the convergence definition of PSO in [Van den Bergh, F., 2002], stated as follows.

Definition 1. Given a particle position *x(t)* and an arbitrary position *p* in search space, the convergence is defined as Eq. (4),

$$\lim_{t \to \infty} x(t) = p \qquad 4$$

This definition implies that the convergence of particles is that the particle ultimately stops at a certain position p in search space. By analyzing the trajectories of particles, Van Den Bergh concludes that all the particles are convergent to the positions of the global best solutions. This conclusion is of significance, since it reveals an important feature of PSO, i.e., *gbest* is the attractor of the whole swarm. Of course, *gbest* itself changes the algorithm runs.

If all the particles achieve the convergence, the swarm does not change any more, which is the stable state. Thus, it can be stated that the PSO algorithm has achieved convergence. Accordingly, *gbest* will not change. So, another convergence definition of PSO can be given.

Definition 2. Given that the best position of PSO in time *t* or in t^{th} generation is *gbest*(*t*), *gbest** is a fixed position in search space, the convergence definition is written as Eq. (5),

$$\lim_{t \to x} gbest(t) = gbest^{*} \qquad 5$$

Definition 2 implies that, if *gbest* generated by PSO does not change any more, then convergence is achieved. If the *gbest* is the global best solution, then the algorithm attains the global best convergence. Otherwise, the algorithm is stuck in local optima.

MODIFIED PSO WITH GREAT DELUGE ALGORITHM

There is presented a new hybrid model of the particle swarm optimization algorithm and great deluge algorithm called MPSO, in this paper. In the normal PSO, after acquiring a new answer, the achieved answer is compared with the best solution found so far and in case of being better, will be accepted. Whereas in the new algorithm, the achieved solution is compared with both, the best found solution so far as well as with another parameter called "*Water Level*" or *WL*. If it is better than the both, it is accepted as new solution.

In fact, there is a level of acceptance inside the PSO for new solutions and this procedure results in a second chance to particles in case of being trapped in the local optimum to be able of getting rid of there. Considering the nature of the problem in the sense of being minimum or maximum, amount of this level of acceptance decreases or increases over time. This algorithm is basically different with the normal PSO so that it tries to exploit the fundamental method of great deluge local search in the particle swarm optimization algorithm.

The foundation of this method is the particle swarm optimization algorithm and the following some changes have been made. The *WL* parameter is used as an acceptance level and *UP* parameter of the great deluge algorithm is used to determine the admissible range of the answers. *UP* parameter is used in increasing or decreasing the *WL*. New algorithm has been tested on some standard functions and performance of the algorithm compared with PSO standard. Test results indicate that the proposed method significantly raises the ability of PSO to escape from the local optimum and the accuracy and the convergence rate. Pseudo-code of MPSO algorithm is shown in Fig. 1.

```
For each particle i
    Randomly initialize v_i , x_i = p_i
    Evaluate f(p_i)
    P_g = arg max{f(p_i)}
End for
Choose WL and Up
Repeat
    each particle i
     Update particle position x_i According to equation below
        v_i = X[v_i+c_1e_1.(p_g-x_i)+c_2e_2.(p_i-x_i)],
        x_i = x_i+v_i
    evaluate f(x_i)
    if (f(x_i) > f(p_g))
        p_i = x_i
    End if
    if ((f(x_i) > f(p_i)) && (f(x_i) > WL))
        Pg = arg max{f(p_i)}
    End if
    WL = WL + Up
Until termination criterion reached
```

FIGURE 1 Pseudo-code of MPSO

EXPERIMENT AND EVALUATION

Performance of the proposed PSO model is tested on a number of analytical benchmark functions which have been extensively used to compare PSO-type and non-PSO-type meta-heuristic algorithms. This paper utilizes the benchmark function set, shown in Table 1 Reaching a more accurate comparison, experiments are in 10, 20, 30 dimensional spaces.

Table 1. Ackley, Rosenbrock, Sphere and Step benchmark functions

Name	Domain	Function
Ackley	±32	$20+e-20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}}-e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}$
Rosenbrock	±2.04	$\sum_{i=1}^{n-1}(100(x_{i+1}-x_i^2)^2+(x_i-1)^2)$
Sphere	±5.12	$\sum_{i=1}^{n}x_i^2$
Step	±5.12	$\sum_{i=1}^{n}\lfloor x_i \rfloor$

CHARACTERISTICS OF THE PROPOSED PSO

In this method, the coefficient c_1 and c_2 are equal to 2.8 and 1.3 respectively and the maximum dimensions are considered 30. The maximum numbers of iterations for the Ackley function are equal to 15000, 25000 and 40000 for dimensions 10, 20 and 30 respectively. For the Rosenbrock and Sphere functions these values are 5000, 10000 and 15000 respectively and for the Step function these values in all steps is considered 10000. The Inersi weight ω is reached according to the Eq. (6). Number of the particles in both of the algorithms as well as all of the functions is considered 50. The amount of increasing *Water Level* (*WL*) is different in the various functions; but average is assumed 0.0002. It should be noted that the experiments were run 30 times and averaged, best and worst results are in the Tables 2 to 5.

$$w = \frac{2}{|2-(c_1+c_2)-\sqrt{(c_1+c_2)^2-4*(c_1+c_2)}\,|} \qquad 6$$

Table 2. compared result's MPSO and PSO on Ackley function

Algorithm	Dimension	Iteration	Best	Average	Worst
PSO	10	15000	2.6e-15	2.6e-15	2.6e-15
PSO	20	25000	2.6e-15	5.5e-15	2.2e-13
PSO	30	40000	2.07e-5	2.58e-3	2.0946
MPSO	10	15000	2.6e-15	2.6e-15	2.6e-15
MPSO	20	25000	2.6e-15	3.5e-15	2.1e-14
MPSO	30	40000	1.39e-5	2.36e-3	2.1286

Table 3. compared result's MPSO and PSO on Rosenbrock function

Algorithm	Dimension	Iteration	Best	Average	Worst
PSO	10	5000	2.35e-8	2.82e-5	2.22e-3
PSO	20	10000	2.72e-5	2.65e-1	2.5434
PSO	30	15000	5.7350	7.5111	8.1460
MPSO	10	5000	4.9e-10	7.79e-7	3.92e-5
MPSO	20	10000	4.85e-7	7.75e-4	3.54e-1
MPSO	30	15000	5.4336	6.4270	8.6717

Table 4. compared result's MPSO and PSO on Sphere function

Algorithm	Dimension	Iteration	Best	Average	Worst
PSO	10	5000	1.9e-106	1.81e-74	2.64e-68
PSO	20	10000	1.97e-41	2.31e-36	2.84e-33
PSO	30	15000	1.12e-35	8.18e-30	1.19e-25
MPSO	10	5000	2.5e-113	1.08e-86	8.70e-73
MPSO	20	10000	2.33e-43	1.23e-38	3.40e-35
MPSO	30	15000	1.98e-41	4.91e-33	6.40e-30

Table 5. compared result's MPSO and PSO on Step function

Algorithm	Dimension	Iteration	Best	Average	Worst
PSO	10	10000	-60	-60	-60
PSO	20	10000	-118	-111	-108
PSO	30	10000	-178	-169	-167
MPSO	10	10000	-60	-60	-60
MPSO	20	10000	-120	-120	-120
MPSO	30	10000	-180	-176	-172

According to the test results and considering Tables 2 to 5, for Ackley and Sphere functions with the dimension 10, the proposed algorithm has the same results with the PSO standard. But in other cases, the proposed method has been superior, and there is a tangible superiority in the higher dimensions. Figures 2 to 3 shows the fitness function of the average of 30 runs of the proposed method with PSO standard on four standard functions Ackley, Rosenbrock, Sphere and Step is in the dimensions 10, 20 and 30. Experimental results indicate superiority of MPSO over the standard PSO. Obtaining final solution by great deluge algorithm with small steps in each stage leads to a higher accuracy in this algorithm in comparison with other algorithms. The proposed algorithm on the other hand, is strongly capable of escaping from local optimum and can improve the development process owing to exploiting of PSO global search. In fact, the parameter *WL* used in this algorithm has had significant influence in the performance, so that it causes balance between local and global searches.

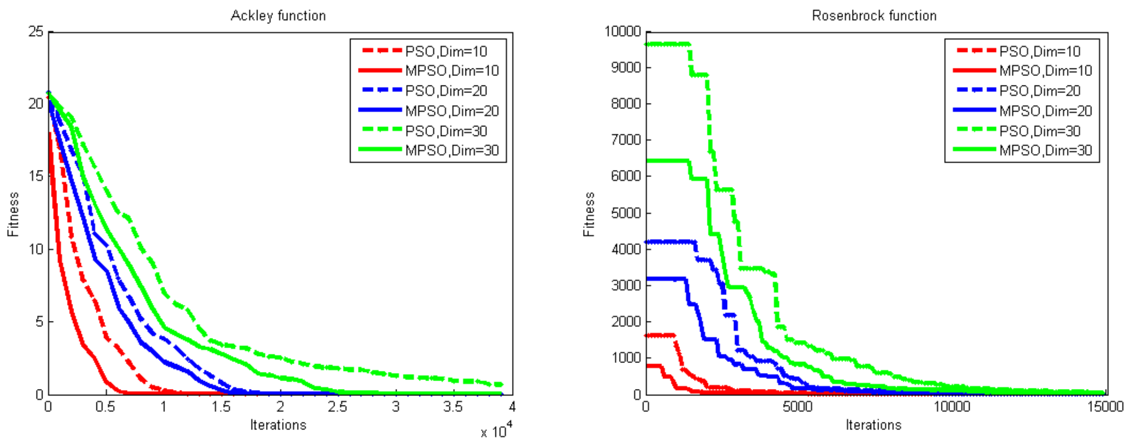

FIGURE 2 Performance MPSO and PSO on Ackley and Rosenbrock functions

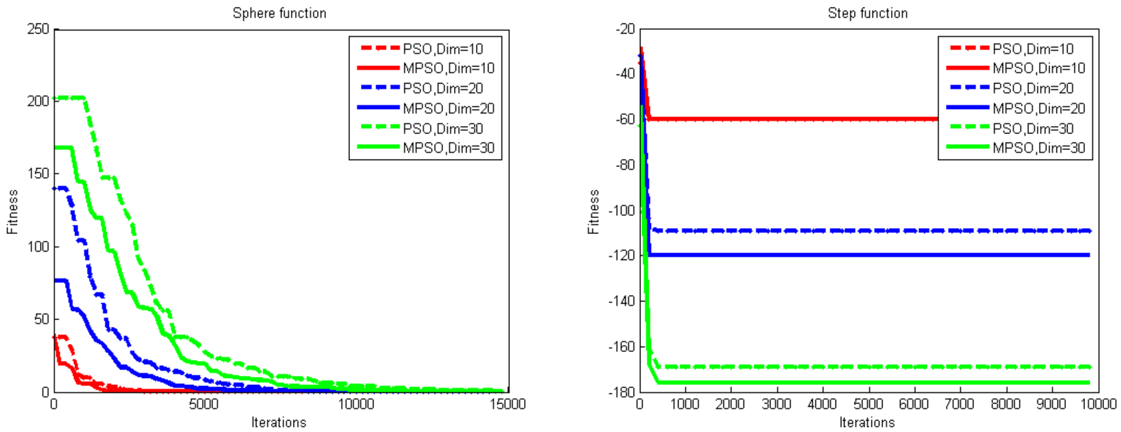

FIGURE 3 Performance MPSO and PSO on Sphere and Step functions

CONCLUSION

There is presented a hybrid model of PSO with great deluge algorithm for optimization in this paper. The purpose of this model is to reach to balance in the search, by combining local search property of the great deluge algorithm and global search property of the particle swarm optimization algorithm. Performed experiments show promising results. Improving the ability of jumping from local optimization, accuracy of convergence and speed, are some improved cases. According to the comparison of the proposed algorithm with other PSO family, it is observed that this algorithm has significant improvement on them.

REFERENCES

1. Kennedy, J. and R.C., Eberhart, 1995. Particle Swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, pp: 1942–1948.
2. Kennedy, J. and R.C., Eberhart, 1995. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS'95), pp: 39–43.
3. Yang, X., J., Yuan and H., Mao, 2007. A modified particle swarm optimizer with dynamic adaptation. Applied Mathematics and Computation, 189: 1205–1213.
4. Shi, Y.H. and R.C., Eberhart, 2001. Fuzzy Adaptive particle swarm optimization. In Proceedings of the IEEE Congress on Evolutionary Computation, Vol. 1, pp: 101–106.
5. Shi, Y.H. and R.C., Eberhart, 1998. A modified particle swarm optimizer. In Proceedings of the IEEE Congress on Evolutionary Computation, pp: 69–73.
6. Zhang, Y.L., L.H., Ma, L.Y., Zhang and J.X., Qian, 2003. On the Convergence Analysis and Parameter Selection in Panicle Swarm Optimization. In Proceedings of the International Conference on Machine Learning and Cybernetics, pp: 1802–1807.
7. Eberhart, R.C. and Y.H., Shi, 2001. Tracking and optimizing dynamic systems with particle swarms, In Proceedings of the IEEE Congress on Evolutionary Computation, pp: 94–100.

8. Zhang, L.P., H.J., Yu, D.Z., Chen and S.X., Hu, 2004. Analysis and improvement of particle swarm optimization algorithm. Information Control, 33: 513–517.
9. Petrovic, S., Y., Yang and M., Dror, 2007. Case-based selection of initialization heuristics for meta-heuristic examination timetabling. Expert Systems with Applications, 33 (3): 772-785.
10. Dueck, G., 1993. New optimization heuristics: The great deluge algorithm and the record-to-record travel. Journal of Computational Physics, 104: 86–92.
11. Burke, E., Y., Bykov, J., Newall and S., Petrovic, 2003. A time predefined approach to course timetabling. Yugoslav Journal of Operations Research, 13 (2): 139–151.
12. Burke, E., Y., Bykov, J., Newall and S., Petrovic, 2004. A time-predefined local search approach to exam timetabling problems. IIE Transaction, 36 (6): 509–528.
13. Nahas, N., D., Ait-Kadi and M., Nourelfath, 2006. A new approach for buffer allocation in unreliable production lines. International Journal of Productions and Economics, 103 (2): 73–81.
14. Nahas, N., M., Nourelfath and D., Ait-Kadi, 2007. Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series–parallel systems. Journal of Reliability Engineering and Systems Safety, 92 (2): 11–22.
15. Nahas, N., M., Nourelfath and D., Ait-Kadi, 2007. A two-phase extended great deluge algorithm for the dynamic layout problem. International conference on industrial engineering and systems management, pp: 567-573.
16. Khatab, A., Ait-Kadi, D., Artiba, A., 2008. Selective maintenance optimization for multi-mission series-parallel systems. European Journal of Operational Research.
17. Shi, Y.H. and R.C., Eberhart, 1999. A new optimizer using particle swarm theory. In Proceedings of the International Conference on Evolutionary Computation, pp: 1945–1950.
18. Van den Bergh, F., 2002. An Analysis of Particle Swarm Optimizers. Ph.D. thesis, University of Pretoria.