J. Basic. Appl. Sci. Res., **2(3)2336-2341, 2012**
© 2012, TextRoad Publication
ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

A New Hybrid Algorithm for Optimization Using PSO and GDA

Sajjad Ghatei[1], Farzad Tigh Panahi[2], Mojtaba Hosseinzadeh[3], Mohammad Rouhi[4], Iman Rezazadeh[5], Ahmad Naebi[6], Zahra Ghatei[7] and Rahim Pasha Khajei[8]

[1,3,4] Department of Computer Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran
[2] Department of Information Technology, National Iranian Gas Company, Iran
[5] Department of Computer Engineering, Meshkinshahr Branch, Islamic Azad University, Meshkinshahr, Iran
[6] Department of Computer Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[7] Department of Computer Engineering, Shabestar Branch, Islamic Azad University, Shabestar, Iran
[8] Department of Computer Engineering, Osku Branch, Islamic Azad University, Osku, Iran

ABSTRACT

In this paper a new combined approach is presented known as PSO-Great Deluge; the main idea of this approach is to combine particle swarm optimization (PSO) with great deluge algorithm. In this approach, global search character of PSO and local search factor of great deluge algorithm are used based on series. At the first step, PSO algorithm is used to search around environment and its results are given to great deluge algorithm to search about taken results accurately. This approach is tested and its efficiency is compared with methods like Genetic Algorithm and standard PSO results. In this paper is shown that this approach has considerable results and usually has good stability.
KEYWORDS: Particle Swarm Optimization, Great Deluge Algorithm, Genetic Algorithm.

1. INTRODUCTION

The aim of optimization is to find amounts of collection of parameters that maximizes or minimizes mean function to certain limits. Whole appropriate amounts are possible solutions and the best amount of these amounts is optimized solution. Optimization is used in machinery, technology, engineering, and computer science [1]. Finding solution for NP problems is very difficult, approaches like heuristic algorithms is a way of solving this problem. Using these approaches, some solutions can be found that are close to results. Some common approaches are: genetic algorithm, particle swarm optimization and simulate annealing [1]. Major use of these approaches is solving optimization problems.

Another heuristic approach so much similar to simulate annealing algorithm is great deluge algorithm introduced by Dueck in 1993 [2]. Defining one parameter as *Water Level* in this algorithm, we determine limitation over taken results and we compare results with parameters every time. In maximization problems, value of the parameter increases and in minimization problems, the value decreases, at each step.

In [2] this algorithm is used to solve TSP problem by 442 and 532 cities. Burke in [3, 4] used this algorithm to solve examination time tabling problem. In [5], Nahas and his colleagues used this algorithm to solve buffer allocation problem in unreliable production lines. In [6], great deluge and ant colony is combined for the discrete facility layout problem. Improving great deluge algorithm, its performance is assessed on dynamic layering problem, in [7]. Khetab and other writers in [8] used this algorithm to optimize the efficiency of Parallel and series system. In [9], kind of refined nonlinear great deluge algorithm is used to solve examination time tabling. The particle swarm optimization algorithm is an optimized approache inspired from social behaviors in nature. Simple protection and scalability in size and good practical efficiency are characters of this algorithm.

In this paper a new model known as PSO-Great Deluge is proposed. In proposed model, global and local characters of the algorithms are used in an efficient way. First in this algorithm, PSO initiate a search process by long steps and then by finding some results, the new results as entranced given to great deluge algorithm, and in this step, this algorithm searched space with small steps.

This paper includes 6 sections: in part 2, PSO algorithm is defined. In part 3, great deluge algorithm is defined, in part 4 proposed algorithm is defined. Part 5 includes tests results and the final part is conclusion.

2. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization [9-13] optimizes an objective function by undertaking a population-based search. The population consists of potential solutions, named particles, which are metaphor of birds in flocks. These particles are randomly initialized and freely fly across the multi dimensional search space. During flight, each particle updates its own velocity and position based on the best experience of its own and the entire population. The various steps involved in particle swarm optimization Algorithm [14] are as follows:

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

Step 1: The velocity and position of all particles are randomly set to within pre-defined ranges.
Step 2: Velocity updating– At each iteration, the velocities of all particles are updated according to,

$$v_i = v_i + c_1 r_1 (p_{i,best} - p_i) + c_2 r_2 (g_{i,best} - p_i) \quad (1)$$

where p_i and v_i are the position and velocity of particle i, respectively; $p_{i,best}$ and $g_{i,best}$ is the position with the 'best' objective value found so far by particle i and the entire population respectively; w is a parameter controlling the dynamics of flying; r_1 and r_2 are random variables in the range [0,1]; c_1 and c_2 are factors controlling the related weighting of corresponding terms. The random variables help the PSO with the ability of stochastic searching.
Step 3: Position updating– The positions of all particles are updated according to,

$$p_i = p_i + v_i \quad (2)$$

After updating, p_i should be checked and limited to the allowed range.
Step 4: Memory updating– Update $p_{i,best}$ and $g_{i,best}$ when condition is met,

$$\begin{aligned} p_{i,best} &= p_i \qquad if\ f(p_i) > f(p_{i,best}) \\ g_{i,best} &= g_i \qquad if\ f(g_i) > f(g_{i,best}) \end{aligned} \quad (3)$$

Where $f(x)$ is the objective function to be optimized.
Step 5: Stopping condition– The algorithm repeats steps 2 to 4 until certain stopping conditions are met, such as a pre-defined number of iterations. Once stopped, the algorithm reports the values of g_{best} and $f(g_{best})$ as its solution.

PSO [14] utilizes several searching points and the searching points gradually get close to the global optimal point using its p_{best} and g_{best}. Initial positions of p_{best} and g_{best} are different. However, using thee different direction of p_{best} and g_{best}, all agents gradually get close to the global optimum. Pseudo code of algorithm PSO shown in Fig.1.

```
For each particle
   initialize particle
End For
Do
     For each particle
         Calculate fitness value of the particle f(p)
           /*updating particle's best fitness value so far*/
            If f(p) is better than p_best
            set current value as the new p_best
      End For
   /*updating population's best fitness value so far)*/
 Set g_best to the best fitness value of all particles
   For each particle
         Calculate particle velocity according equation (1)
         Update particle position according equation (2)
    End For
While maximum iterations OR minimum error criteria is not attained
```

FIGURE 1. Pseudo code of PSO algorithm

3. GREAT DELUGE ALGORITHM

Great deluge algorithm is a comprehensive approach for solving optimization problems which first Dueck suggested in 1993. Like other local search approaches, this approach also replaces common solution (*New_Config*) with best results (*Best_Config*) that have been found by then. This action continues until stop conditions is provided. In this algorithm, new solutions are selected from neighbors. Selection strategy is different from other approaches.

In great deluge algorithm these results are acceptable which their values are equal or better than (for optimization problems) the value of *Water Level* (*WL*). Value of *WL* also rises at a steady pace in every step. Increase of *WL* continues until value of *WL* equals with the best result achieved ever. In this step, the algorithm is repeated several times and if better result is not obtained, it comes to the end. The primary amount of *WL* is equal with the primary results (*f(s)*). Pseudo code of this algorithm is shown in Fig.2.

```
Choose an initial configuration as Old_Config and Best_Config
Choose WL and Up
For n=0 to # of iterations
 Generate a small stochastic perturbation New_Config of the solution
   If Fitness (New_Config) > WL
      If Fitness (New_Config) > (Best_Config)
        Old_Config := New_Config
      End If
   End If
       WL = WL + Up
End For
```

FIGURE 2. Pseudo code of Great Deluge algorithm

4. PROPOSED MODEL PSO-GREAT DELUGE ALGORITHM

In this paper a new combination model of great deluge algorithm and particle swarm optimization algorithm are used. The aim of this model is to reach to balance in search by combining the local search factor of great deluge algorithm and global search factor of PSO algorithm.

First, PSO algorithm with Long steps searches and then by obtaining some results, new results are delivered as input to great deluge algorithm and in this step this algorithm with short steps, searches the problem space. Pseudo code of PSO-Great Deluge algorithm shown in Fig.3.

```
For each particle i
    Randomly initialize v_i, x_i=p_i
    Evaluate f(p_i) and  P_g=arg max{f(p_i)}
End for
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))
        P_g=arg max{f(p_i)}
    End if
Until termination criterion reached Choose the best individual as Old_Config
Choose WL and Up
For n=0 to # of iterations
    Generate a small stochastic perturbation New_Config of the solution
    if Fitness (New_Config) > WL
        Old_Config := New_Config
    End if
WL = WL + Up
End For
```

FIGURE 3. Pseudo code of PSO-Great Deluge algorithm

5. RESULT OF SIMULATION

Tests are taken on 4 standard functions commonly used as patterns of measurement of optimized algorithms in combined and stable space, functions are: Ackley, Rosenbrock, Sphere and Step. A function is checked on then includes optimization of some n-variable function.

Due to an exact comparison of the proposed approach with existing methods, tests are conducted in 10, 20, 30 dimensional spaces. List of these functions and their specifications are in Tab.1.

Tab.1 Standard functions

Function name	Variant confirms	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	$\sum_{i=1}^{n}\lfloor x_i \rfloor$

To assess proposed combined approach, it is compared to other approaches. In this method C_1 and C_2 factors are equal with 2.8, 1.3 and inertia weight w can be found by equation (4). Number of particle swarm and amount of Water Level increase (*UP*) are different in different functions. But their mean 30 and 0.0002 are considered. It should be noticed that tests repeat 30 times and average results, the best and the worst are in table's form 2 to 5.

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

Tab.2 Execution results for Ackley function

Dimension	Algorithm	Best	Average	Worst
10	Genetic	0.4598	0.7866	0.9453
10	PSO	2.66e-15	2.66e-15	2.66e-15
10	PSO-Great Deluge	2.66e-15	2.66e-15	2.66e-15
20	Genetic	1.9800	3.3453	3.8403
20	PSO	2.66e-15	5.54e-15	2.32e-13
20	PSO-Great Deluge	2.66e-15	3.52e-15	2.18e-14
30	Genetic	3.6595	5.7598	6.8743
30	PSO	2.07e-5	2.18e-10	2.0946
30	PSO-Great Deluge	1.39e-5	2.36e-13	2.1286

Tab.3 Execution results for Sphere function

Dimension	Algorithm	Best	Average	Worst
10	Genetic	1.33e-5	0.0409	0.0598
10	PSO	1.97e-106	-74e1.81	-68e2.64
10	PSO-Great Deluge	2.52e-113	1.08e-86	8.7e-73
20	Genetic	1.54e-3	00.3336	0.5133
20	PSO	1.97e-41	2.31e-36	2.84e-33
20	PSO-Great Deluge	2.33e-43	1.23e-38	3.4e-35
30	Genetic	119.1778	143.6159	159.2387
30	PSO	1.12e-35	8.18e-30	1.19e-25
30	PSO-Great Deluge	1.98e-41	4.91e-33	6.40e-30

Tab.4 Execution results for Step function

Dimension	Algorithm	Best	Average	Worst
10	Genetic	-60	-58.4000	-55
10	PSO	60-	-60.0000	-60
10	PSO-Great Deluge	-60	-60.0000	-60
20	Genetic	-120	116.3200	-112
20	PSO	-120	119.5000	-118
20	PSO-Great Deluge	-120	-120	-120
30	Genetic	-172	-168	-165
30	PSO	-178	-170	-167
30	PSO-Great Deluge	-180	-175	-172

Tab.5 Execution results for Rosenbrock function

Dimension	Algorithm	Best	Average	Worst
10	Genetic	1.6973	4.7711	5.3005
10	PSO	2.35e-08	2.82e-05	-03e2.22
10	PSO-Great Deluge	4.97e-10	7.79e-07	-05e3.92
20	Genetic	5.7812	7.1109	8.2866
20	PSO	2.72e-05	2.65e-01	2.5434
20	PSO-Great Deluge	4.85e-07	7.75e-04	3.54e-01
30	Genetic	19.0065	21.2343	23.3072
30	PSO	5.7350	7.5111	8.1460
30	PSO-Great Deluge	5.4336	6.4270	8.6717

As you can see in table's form 2 to 5, in many cases, practice of PSO-Great Deluge algorithm is better than others. In PSO-Great Deluge, last solution comes from great deluge algorithm with short steps so, the accuracy of this algorithm compared with other algorithms increases and in general has better efficiency than other two approaches. In figures 4 to 5, the chart of fitness function in the period of 100000 times repeat from genetic algorithm operation, PSO standard and PSO-Great Deluge are shown for standard function, Ackley, Rosenbrock, Sphere and Step.

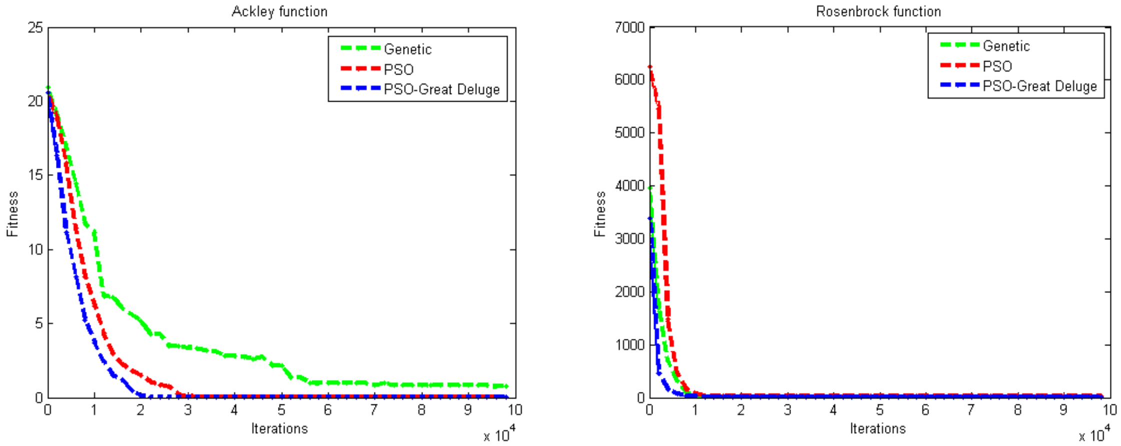

FIGURE 4. Execution result for Ackley and Rosenbrock functions (30 dimensions)

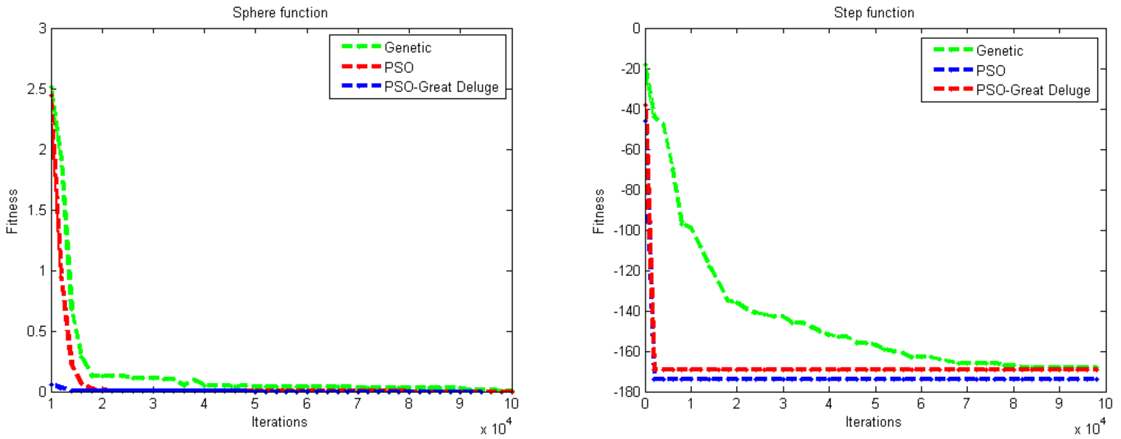

FIGURE 5. Execution result for Sphere and Step functions (30 dimensions)

As we can see in cases proposed algorithm in the contrast to other algorithms are convergent and has the best result compare to other approaches and this conclusion usually has considerable stable. Likewise in some cases other approaches operate well, but in others not.

6. CONCLUSION

In this paper a new hybrid optimization is presented. The aim of this model is to combine local character of the great deluge algorithm and global character of the particle swarm optimization algorithm in order to reach a balance in search.

Both local and global character are used well , first PSO algorithm with long steps starts to search and then by obtaining some results, new results are delivered as entrance to great deluge algorithm, in this step this algorithm searches space with small steps , the results of simulation shows that presented model gives better results compared to genetic and standard PSO.

REFERENCES

1. Dueck, G., 1993. New optimization heuristics: The great deluge algorithm and the record-to-record travel. Journal of Computational Physics, 104: 86–92.

2. 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.

3. 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.

4. 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.

5. 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.

6. 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.
7. Khatab, A., D., Ait-Kadi and A., Artiba, 2008. Selective maintenance optimization for multi-mission series-parallel systems. European Journal of Operation Researches.
8. Shi, Y. and R.C., Eberhart, 2001. Particle Swarm Optimization: developments, applications and resources. Proceedings Congress on Evolutionary Computation, 1: PP. 81-86.
9. Shi, Y. and R.C., Eberhart, 1998. A modified particle swarm optimizer. Proceedings IEEE International Conference on Evolutionary Computation, PP. 69-73.
10. Angeline, P.J., 1998. Using selection to improve particle swarm optimisation. Proceedings IEEE International Conference on Evolutionary Computation, PP. 84-89.
11. Xiao-Feng, X., Z., Wen-Jun and Y., Zhi-Lian, 2002. A dissipative particle swarm optimisation. Proceedings Congress on Evolutionary Computation, 2: 1456-1461.
12. Kennedy, J., 2000. Stereotyping: Improving particle swarm performance with cluster analysis. Proceedings IEEE International Conference on Evolutionary Computation, 2: 1507-1512.
13. Abdelbar, A.M. and S., Abdelshahid, 2003. Swarm optimization with instinct driven particles. Proceedings IEEE Congress on Evolutionary Computation, 2: 777-782.
14. Baumgartner, U., Ch., Magele and W., Renhart, 2004. Pareto optimality and particle swarm optimisation. IEEE Transaction Magnetics, 40: 1172-1175.