Fast Convergence in Function Optimization Using Modified Velocity Updating in PSO Algorithm

Nanda Dulal Jana[1], Tapas Si[2], and Jaya Sil[3]

[1] Dept. of Information Technology
National Institute of Technology, Durgapur, West Bengal, India
[2] Dept. of Computer Science & Engineering
BankuraUnnayani institute of Engineering, Bankura, West Bengal, India
[3] Dept. of Computer Science & Technology
Bengal Engineering & Science University, Howrah, West Bengal, India
{nanda.jana,c2.tapas}@gmail.com, js@cs.becs.ac.in

Abstract. In this paper, a new version of Particle Swarm Optimization (PSO) Algorithm has been proposed where the velocity update equation of PSO has been modified. A new term is added withthe original velocity update equation by calculating difference between the global best of swarm and local best of particles. The proposed method is applied on eight well known benchmark problems and experimental results are compared with the standard PSO (SPSO). From the experimental results, it has been observed that the newly proposed PSO algorithm outperforms the SPSO in terms of convergence, speed and quality.

1 Introduction

The Particle Swarm Optimization (PSO) is a population based global optimization technique, inspired by the social behavior of bird flocking and fish schooling [1][2]. The PSO algorithm maintains a swarm of particles, called individuals where each particle (individuals) represents a candidate solution. Particles follow a very simple behavior: emulate the success of neighboring particles and own success achieved. The position of particle is therefore influenced by the best particle in a neighborhood, as well as the best solution found by the particle. Initially PSO was designed for continuous optimization problem but shown its capability of handling non-differentiable, discontinuous and multimodal objective functions and has gained increasing popularity in recent years due to its ability to efficiently and effectively tackle several real-world applications [3][4].

Several variations [5][6][7][8][9][10][11] has been proposed to improve the performance and the convergence behavior of PSO algorithms. One class of variations include the modification of velocity update equation in PSO. In T. Ziyu and Z. Dingxue [5] was proposed a new version of PSO without the velocity of the previous iteration and a novel selection of acceleration coefficients was introduced in the algorithm. In [6], cognitive component and social component was replaced by two terms of the linear combination of global best of swarm and personal best of particle.

S.C. Satapathy et al. (Eds.): Proc. of Int. Conf. on Front. of Intell. Comput., AISC 199, pp. 515–521.
DOI: 10.1007/978-3-642-35314-7_58

In this paper, we have proposed a new PSO algorithm, in which velocity update equation has been modified by adding the difference between the global best of swarm and the personal best of particle. Initial experimental results in a benchmark set consisting of eight difficult high dimensional benchmark functions demonstrate that this is a promising approach.

The rest of the paper is organized as follows: Section 2 describes the basic operations of the standard PSO. In section 3, we propose the new PSO algorithm, while in section 4, experimental analysis and results are presented. The paper concludes with a short discussion and some pointers for future work.

2 The Standard Particle Swarm Optimization (SPSO)

The beauty of Particle Swarm Optimization lies in its simplicity and ease of applicability. It is a kind of algorithm to search for the best solution by simulating the movement offlocking birds. It uses a swarm of individual called particles. Each particle has its own position and velocity to move around the search space. The coordinates of each particle represent a possible solution associated with two vectors-the position vector and the velocity vector. Particles have memory and each particle keep track of previous best position and corresponding fitness. The previous best value is called as *pbest*. It also has another value called *gbest*, which is the best value of all the particles *pbest* in the swarm.

Consider a D-dimensional function $f(x)$, want to be optimized

$$Minimize\ f(x),\ where\ f: R^D \rightarrow R$$

The position vector and velocity vector of the *ith* particle is represented by $X_i = (x_{i1}, x_{i2},\ldots\ldots,x_{iD})$ and $V_i = (v_{i1}, v_{i2},\ldots., v_{iD})$, where i denote the swarm size. A swarm consists of a number of particles that proceed (fly) through the search space towards the optimal solution. Each particle update its position based on its own best exploration, overall best swarm exploration and its previous velocity vector according to the following equations:

$$V_i(t+1) = V_i(t) + c_1r_1(pbest_i(t) - X_i(t)) + c_2r_2(gbest_i(t) - X_i(t)) \tag{1}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{2}$$

Where $c_1$and $c_2$are two positive acceleration coefficients, r_1 and r_2 are uniformly distributed random numbers in [0, 1],$X_i = (x_{i1}, x_{i2},\ldots\ldots,x_{iD})$ is the current position of the *ith* particle.$pbest_i= (x_{i1}^{pbest}, x_{i2}^{pbest},\ldots\ldots,x_{iD}^{pbest})$is the best position of the *ith* particle achieved based on its own experience. $gbest_i = (x_1^{gbest}, x_2^{gbest},\ldots\ldots,x_D^{gbest})$is the position of the best particle based on the overall swarms experience. T is the iteration counter.A constant, maximum velocity (V_{max}) is used to limit the velocities of the particles and improve the resolution of the search space. Shi and Eberhart [12][13] proposed to use an "inertia weight" factor ω, in order to overcome the premature convergence of PSO. The resulting velocity update equation (1) becomes:

$$V_i(t+1) = \omega V_i(t) + c_1r_1(pbest_i(t) - X_i(t)) + c_2r_2(gbest_i(t) - X_i(t)) \tag{3}$$

In this paper, the inertia weight version of PSO is regarded as the SPSO.

3 The Proposed Algorithm

The motivation behind designing the new PSO algorithm is to accelerate its convergence rate, success rate and exploration capability in finding global optimum solution. In this approach, the difference between *gbest* and *pbest*is scaled while multiplyingby a constant c_3 and a random number r_3. This term is added with the velocity update equation of SPSO, expressed by (3). The new velocity update equation is defined by equation (4).

$$V_i(t+1)=\omega V_i(t) + c_1 r_1(pbest_i(t) - X_i(t))+ c_2 r_2(gbest_i(t)- X_i(t)) + c_3 r_3(gbest_i(t) -pbest_i(t)) \quad (4)$$

In equation (4), the first term represents current velocity of the particle and can be thought of as momentum term. The second term is responsible for attraction of particles at the current position towards the positive direction of its own best position (*pbest*) i.e. cognitive component. The third term is responsible for the attraction of particles at current position towards the positive direction of the global best position (*gbest*) i.e. social component. The fourth term is responsible for the distance between *gbest* and *pbest*which implies how far-away the *pbest* position is from the *gbest* position. The values of ($gbest_i$- $pbest_i$) are decreasing over the increasing number of iterations (t). Finally,$(gbest_i – pbest_i) \cong 0$ because the *pbest* closely reached to the *gbest* of the swarm.

Moreover, the velocity of the SPSO may be a small value, if both ($pbest$ - X_i) and ($gbest$ - X_i) are small enough. In such situations, exploration capability of SPSO has been lost at some generation. At the early stage of evolution process, ($gbest_i$) - $pbest_i$) preventssuch situation to arrive(exploration capability) in the swarm. But, the loss of diversity for ($gbest_i$ - $pbest_i$) is typically occurred in the latter stage of evolution process because the value of ($gbest_i$ - $pbest_i$)reaches close to zero.

The proposed algorithm is presented below.

Input: Randomly initialized position and velocity of the particles: $X_i(0)$ and $V_i(0)$.
Output: Position of the approximate global optima X^*

while maximum number of iteration (t) or minimum error criterion is not attained

fori= 1 to N (Swarm size)
for j= 1 to D (Dimension of the problem)
Velocity update equation
$V_i(t+1) = \omega V_i(t) + c_1 r_1(pbest_i(t) - X_i(t)) + c_2 r_2(gbest_i(t) - X_i(t)) + c_3 r_3(gbest_i(t) - pbest_i(t))$

Position update equation
$X_i(t+1) = X_i(t) + V_i(t+1)$
end for j
Evaluate fitness of the update position of the particles
Update previous pbest and gbest information, if needed
end for i
end while

4 Results and Discussions

This section compares the performance of the proposed PSO algorithm with the SPSO algorithm. To verify the effectiveness of the proposed approach we have used eight widely known high dimensional benchmark functions with different characteristics [14].The four functions (f_1 – f_4) are high dimensional and scalable benchmark functions. The remaining four functions (f_5 – f_8) are high dimensional and multimodal functions, where the number of local minima increases exponentially with their dimensionality. A brief description of the functions is provided in Table 1. More specifically, D denotes the dimension of the problem, search space is the optimization range box and objective function value is the global minimized value.

The SPSO and the proposed PSO algorithm are coded in MatLab2010b and developed on AMD FX-8150 Eight-Core machine with 4GB RAM under Windows 7 platform. Fifty independent runs with different seed for the generation of random are taken. However, the same seed is used for generating the initial swarm for SPSO and proposed PSO algorithm. We consider*best- run-error* values for function optimization, which is calculatedusing following inequality:

$$|f^*(x) - f(x)| < 0.001 \quad (5)$$

Where $f^*(x)$ is the best known objective function and $f(x)$ is the best objective function found in the corresponding run. Right side of the equation (5)denotes the threshold value representing precision in the objective function. Themaximum number of function evaluations are fixed,say 1, 00,000. The swarm size and dimension are fixed to 40 and 30, respectively. The inertia weight ω is 0.72984 and the acceleration coefficients for SPSO and proposed PSO are set to $c_1 = c_2 = 1.49445$ experimentally.

The quality of solution obtained is measured by the best, mean and standard deviation of the objective function values out of fifty runs. The performance of solution is measured by the mean and standard deviation of the total number of function evaluations out of fifty runs. In Table 2 and Table 3, best-run-error values and number of function evaluations of the SPSO and the proposed PSO are tabulated, considering c_3=c_1 in the proposed algorithm. For c_3= 1, the best-run-error values and number of function evaluations of the SPSO and the proposed PSO are tabulated in Table 4 and Table 5, respectively. The convergence graph for function f8 is given in Fig. 1.

From Table 2, it has been observed that the proposed PSO gives better result than SPSO for the functions f1, f5, f7, f8. Also in Table 3, the proposed PSO shown better performance compared with SPSO. In observing Table 4 and Table 5, it has been concluded that the quality of solution and performance obtained by our proposed PSO is better or equal to the SPSO. Analysis of the results obtained by the proposed PSO algorithm outperforms on multimodal functions than unimodal functions using SPSO.

Table 1. A brief description of the benchmark function set

Test Function	S	fmin
$f_1(x) = \sum_{i=1}^{D} x_i^2$	[-100,100]	0
$f_2(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	[-100,100]	0
$f_3(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{n-1}} x_i^2$	[-100,100]	0
$f_4(x) = [100(x_{i+1} - x_i{}^2)^2 - (1 - x_i\ )^2]$	[-100,100]	0
$f_5(x) = -x_i * \sin(\sqrt{\lvert x_i \rvert})$	[-500,500]	-12569.50
$f_6(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	0
$f_7(x) = -20 * \exp\left(-0.2 * \sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi x_i)\right) + 20 + e$	[-32,32]	0
$f_8(x) = \sum_{i=1}^{D} [x_i - 10\cos(2\pi x_i) + 10]$	[-5.12,5.12]	0

Table 2. Best-run-error values achieved by SPSO and Proposed PSO (for c_3=c_1)

Test#	SPSO		Proposed PSO	
	Mean	Std. Dev.	Mean	Std. Dev.
f1	9.36e-04	**0.0014**	**0.0014**	**0.0014**
f2	**2.84e-04**	**2.64e-04**	3.03e-04	2.77e-04
f3	**9.21e-04**	**7.26e-05**	0.2796	0.3461
f4	**35.5684**	**45.0933**	354.9286	470.18
f5	7.09e+03	597.031	**4.23e+03**	**544.55**
f6	**0.0212**	**0.0223**	0.0232	0.0271
f7	1.5119	0.9891	**0.0244**	**0.0698**
f8	33.76	8.9137	**23.3976**	**8.7277**

Table 3. Number of function evaluations for SPSO and Proposed PSO (for c_3=c_1)

Test#	SPSO		Proposed PSO	
	Mean	Std. Dev.	Mean	Std. Dev.
f1	**13256**	**1.21E+003**	9.38E+004	6.34E+003
f2	**1.88E+003**	**1.01E+003**	3.62E+003	4.35E+003
f3	2.24E+004	2.32E+003	**100000**	**0**
f4	**100000**	**0.00E+000**	**100000**	**0**
f5	**100000**	**0.00E+000**	**100000**	**0**
f6	**7.76E+004**	**3.83E+004**	9.96E+004	1.30E+003
f7	8.54E+004	3.16E+004	**100000**	**0**
f8	100000	0.00E+000	**100000**	**0**

Table 4. Best-run-error values achieved by SPSO and Proposed PSO (for c_3=1)

Test#	SPSO		Proposed PSO	
	Mean	Std. Dev.	Mean	Std. Dev.
f1	9.36e-004	0.0014	**9.44e-04**	**6.11e-05**
f2	**2.84e-04**	**2.64e-04**	3.71e-04	3.28e-04
f3	**9.21e-04**	**7.26e-05**	9.22e-04	**7.26e-05**
f4	**35.5684**	45.0933	50.0599	**40.155**
f5	**7.09e+03**	**597.031**	5.47e+03	636.716
f6	0.0212	0.0223	**0.0157**	**0.0177**
f7	1.5119	0.9891	**0.0241**	**0.1632**
f8	**33.7688**	**8.9137**	37.4701	13.0016

Table 5. Number of function evaluations for SPSO and Proposed PSO (for c_3= 1)

Test#	SPSO		Proposed PSO	
	Mean	Std. Dev.	Mean	Std. Dev.
f1	**13256**	**1.21E+003**	3.67E+04	2.42E+03
f2	1.88E+003	**1.01E+003**	**2656**	2.00E+03
f3	**2.24E+004**	**2.32E+003**	5.51E+04	3.92E+03
f4	**100000**	**0.00E+000**	**100000**	**0.00E+000**
f5	**100000**	**0.00E+000**	**100000**	**0.00E+000**
f6	**7.76E+004**	3.83E+004	8.18E+04	**2.81E+04**
f7	8.54E+004	**3.16E+004**	**5.05E+04**	8.09E+03
f8	**100000**	**0.00E+000**	**100000**	**0.00E+000**

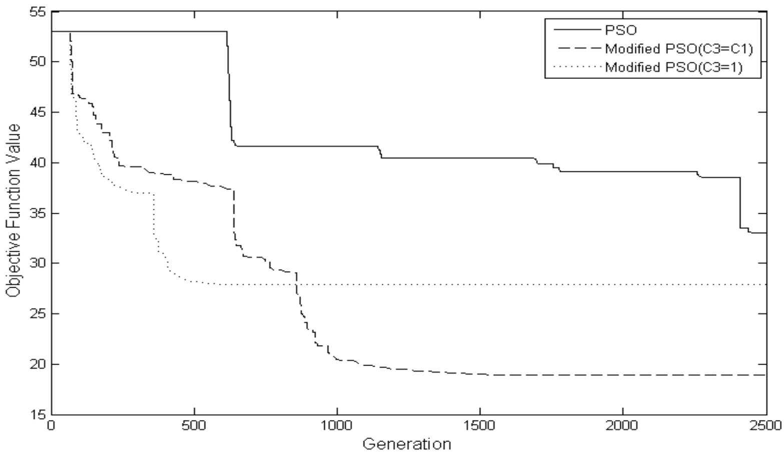

Fig. 1. Convergence graph for function f8

5 Conclusions

In the present study, a new PSO algorithm has been proposed for function optimization. It is based on the basic change in the velocity updating equation. One more term, $(gbest_i - pbest_i)$ is added to the velocity update equation of SPSO. It is tested on eight high dimensional benchmark functions. It is shown that the proposed PSO algorithm outperforms SPSO in terms of efficiency, accuracy and effectiveness. Particularly for multimodal functions, proposed PSO algorithm superior than SPSO. In this paper, we are using only two values ($c_3 = c_1$ and $c_3 = 1$) of the parameter c_3. Therefore, the effective changes of this parameter are notexplored. In a future study parameters fine tuning may be carried out for better performance. Also the application of proposed PSO to the real world problems would be interesting as a future research.

References

1. Eberhart, R.C., Kennedy, J.: A New Optimizer Using Particle Swarm Theory. In: International Symposium on Micromachine and Human Science, pp. 39–43 (1995)
2. Kennedy, J., Eberhart, R.C.: Particle Swarm optimization. In: IEEE International Joint Conference on Neural Networks, pp. 1942–1948. IEEE Press (1995)
3. Clerc, M.: Particle Swarm Optimization. ISTE Publishing Company (2006)
4. Englelbrecht, A.: Computational Intelligence: An Introduction. Halsted Press (2002)
5. Ziyu, T., Dingxue, Z.: A Modified particle Swarm Optimization with an Adaptive acceleration coefficients. In: Asia-Paciffic Conference on Information Processing (2009)
6. Deep, K., Bansal, J.C.: Mean Particle Swarm Optimization for function optimization. International Journal of Computational Intelligence Studies 1(1), 72–92 (2009)
7. Zhan, Z.-H., Zhang, J., Li, Y., Chung, H.S.-H.: Adaptive particle swarm optimization. IEEE Transactions on Systems, Man, and Cybernetics, 1362–1381 (2009)
8. Xinchao, Z.: A perturbed particle swarm algorithm for numerical optimization. Applied Soft Computing, 119–124 (2010)
9. Chen, M.-R., Li, X., Zhang, X., Lu, Y.-Z.: A novel particle swarm optimizer hybridized with external optimization. Applied Soft Computing, 367–373 (2010)
10. Pedersen, M.E.H.: Tuning & Simplifying Heuristically Optimization, Ph.D. thesis, school of Engineering Science, University of Southampton, England (2010)
11. Singh, N., Singh, S.B.: One Half Global Best Position Particle Swarm Optimization Algorithm. International Journal of Scientific & Engineering Research 2(8), 1–10 (2012)
12. Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: Proceedings of the IEEE international Conference on Evolutionary Computation, pp. 69–73 (1998)
13. Shi, Y., Eberhart, R.C.: Parameter Selection in particle swarm Optimization. In: 7th Annual Conference on Evolutionary Programming, San Diego, USA (1998)
14. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation 3, 82–102 (1999)