2015 Third International Conference on Artificial Intelligence, Modelling and Simulation

Improved Particle Swarm Optimization Based on Velocity Clamping and Particle Penalization

Musaed Alhussein
College of Computer and Information Sciences
King Saud University
Riyadh, Saudi Arabia
musaed@ksu.edu.sa

Syed Irtaza Haider
College of Computer and Information Sciences
King Saud University
Riyadh, Saudi Arabia
sirtaza@ksu.edu.sa

***Abstract*—The idea of particle swarm optimization falls under the domain of swarm intelligence. Particle swarm optimization technique is widely used for finding the global minima of well-known benchmark functions. The main idea behind this technique is that working in a group improves the performance of a system. A modified particle swarm optimization technique is proposed in this paper and tested on seven standard benchmark functions. The two major modifications are introduced in the standard particle swarm optimization; modify the velocity of a particle such that the particle remains within the confine limits of clamp velocity, and penalize the particle velocity, if the sum of the velocity vector and position vector results in breaching the boundary limits of search space. The results of the modified PSO are compared with the two versions of standard PSO; constant inertial weight with no velocity clamping and linearly decreasing inertial weight with no velocity clamping.**

Keywords*— *particle swarm optimization, swarm intelligence, velocity clamp, inertial weight, minimization problem

I. INTRODUCTION

Particle swarm optimization (PSO) is an optimization technique proposed by Dr. Kennedy and Dr. Eberhart in 1995 [1]. The inspiration behind this optimization technique came from the behavioral and problem solving capabilities of social animals [2]. It is a well-known fact that the efficiency of a group increases by the social sharing of information within the group. Swarm intelligence (SI) was first introduced by Gerardo Beni and Jing Wang, in the context of cellular robotics in 1989. The concept was based on the fact that working in a group can give better results. The idea of swarm intelligence and social interaction was used by Dr. Kennedy and Dr. Eberhart in 1995, and introduces an optimization technique named as particle swarm optimization (PSO). In this technique random agents interact with each other to find a solution of a fitness function. Nowadays, this stochastic approach is the basis for the optimization of continuous non-linear functions.

In 1998, Eberhart and Shi modified the original PSO and introduces a factor called the inertial weight [4]. It determines the contribution of particles velocity in the previous iteration, to its velocity at the current iteration. The study shows that, if the inertial weight is kept large during the initial iterations, the particles extensively explore the search space, i.e. it facilitates the global search. By gradually decreasing the inertial weight to a much smaller value, the system would be more exploitative i.e. it facilitates the local search. There is a possibility that the particles prematurely converges to the local minima, so to solve this problem an adaptive inertial weight strategy is proposed to improve the search capability of the particles. Some of the other inertial weight strategies are natural exponent, chaotic, sigmoid increasing, oscillating and logarithm decreasing inertial weight [5]. In 2000, Eberhart and Shi introduces a new parameter known as constriction factor to optimize the multidimensional functions. They proposed that the performance of the PSO can be significantly improved by utilizing the constriction factor while limiting the velocity [6].

Different variants of PSO have been explored and discussed in the past few years [7-11]. In this paper, following modifications are proposed in order to improve the performance of PSO; a custom inertial weight is implemented that helps in avoiding stagnation and ensures fast convergence, velocity clamping and particle penalization is introduced in order to confine the particles within the search space.

This paper is organized as follows. Section II explains the standard PSO algorithm and its parameters. Section III presents the modified particle swarm optimization technique. The standard benchmark functions are presented in section IV whereas the results and discussion is provided in section V.

II. STANDARD PSO

In PSO, individuals called "particles" changes their "position" with time. Each particle consists of a position and velocity vector. The algorithm searches through a multi-dimensional space and starts with the random initialization of a position and velocity vector of each particle. Consider a j^{th} particle in D-dimensional space, the position vector can be represented as,

$$X_j = (x_{j1}, x_{j2}, x_{j3}, \dots, x_{jD}) \quad (1)$$

Similarly, the velocity vector can be represented as,

$$V_j = (v_{j1}, v_{j2}, v_{j3}, \dots, v_{jD}) \quad (2)$$

The current position vector of each particle is passed to a fitness function that results in a fitness value. If the fitness value is better than the previous best value, update the local best position of a particle. The global best is updated based on the best fitness value found by any of the neighbor. The velocity vector of each particle is updated based on the particle's local best position and the global best and expressed as,

978-1-4673-8675-3/15 $31.00 © 2015 IEEE
DOI 10.1109/AIMS.2015.20

$$v_j^i = \left(w * v_j^{i-1}\right) + c_1 * r_1 * \left(pbest_j^{i-1} - X_j^{i-1}\right) + c_2 * r_2 * \left(gbest_j^{i-1} - X_j^{i-1}\right) \quad (3)$$

In the above equation, w is the inertial weight, v_j^i is the velocity of the jth particle in ith iteration, c_1 and c_2 are the local and global acceleration constants, r_1 and r_2 are uniformly distributed random numbers, $pbest_j^{i-1}$ is the particles local best position and $gbest_j^{i-1}$ is the global best of the swarm. The updated velocity vector is then used to update the position vector of each particle using the following expression,

$$x_j^i = x_j^{i-1} + v_j^i \quad (4)$$

where, x_j^i is the position of jth particle in the ith iteration. The stopping criteria for the fitness function evaluation is the maximum function evaluations and expressed as,

$$FE = 5000 * D \quad (5)$$

where, FE is the maximum function evaluations and D is the dimension.

In standard particle swarm optimization, inertial weight and acceleration constants play an important role in order to observe the convergence. Inertial weight controls the influence of previous velocity on the current velocity of a particle, and serves as a tradeoff between the local and global search abilities of the swarm. In the standard PSO algorithm, the value of the inertial weight was kept constant. Later on, many variants of controlling the inertial weight have been introduced by the researchers to facilitate the convergence. The acceleration constants c_1 and c_2 helps each particle in the swarm to get pulled towards the local or global best position respectively. The values of the acceleration constant must satisfy the condition $c_1+c_2 \leq 4$, otherwise usually PSO doesn't converge and there is a chance of particle stuck in the local minima. In order to maintain the diversity of the population, uniformly distributed random numbers, r_1 and r_2 are generated in the range of [0, 1].

III. Modified PSO

This work proposes an approach that is based on the following modifications; modifying the particle velocity such that the velocity remains within the limits of [V_{max}, V_{min}], and penalizing the particle velocity, if the sum of the velocity vector and position vector results in the new position of particle outside the boundary limits of the search space. These two modifications will make sure that the velocity of the particle is within the confined limits along with the position of the particle within the boundary limits of the search space. The V_{max} and V_{min} parameters are obtained using the following expression,

$$V_{max} = lambda * (Max_{SS} - Min_{SS})$$
$$V_{min} = lambda * (Max_{SS} - Min_{SS}) \quad (6)$$

In the above expressions, Max_{SS} and Min_{SS} are the search space limits and is different for each benchmark function, lambda is a factor that is used to clamp the velocity of a particle. The proposed approach compares the velocity of the ith particle in the Dth dimension with V_{max}, V_{min} and modifies the velocity of the particle based on the following expression,

$$\begin{cases} V_{i,D} > Vmax \;\; then \; V_{i,D} = Vmax \\ V_{i,D} < Vmin \;\; then \; V_{i,D} = Vmin \end{cases} \quad (7)$$

The penalizing algorithm works as follows,

$$V_{i,D} + X_{i,D} > Max_{SS} \;\; or \; V_{i,D} + X_{i,D} < Min_{SS}$$
$$then \; V_{i,D} = 0, X_{i,D} = -X_{i,D} \quad (8)$$

In the above expression, $X_{i,D}$ is the position of the ith particle in Dth dimension. If the resultant vector of sum of the velocity and position vector lies outside the search space, then the velocity of the particle is penalized and set to zero, along with the change in the direction of the position vector. After penalizing the particle velocity, the updated particle position is evaluated using the standard PSO position update expression. In this paper, three algorithms are simulated; Algorithm1-Standard PSO with constant inertial weight and no velocity clamping, Algorithm2-Standard PSO with linearly decreasing inertial weight and no velocity clamping, Algorithm3-Modified PSO with velocity clamping and inertial weight control. Algorithm1 uses the value of inertial weight 0.7, whereas the expression for the inertial weight for Algorithm2 can be expressed as,

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{FE}\right) * i \quad (9)$$

In the above expression, i is the current function evaluation. The value of w_{max} and w_{min} has been optimized to achieve the best results. One of the promising feature of the algorithm3 is to avoid the stagnation and converge very fast to find the minima of any standard problem. In order to do this, the inertial weight was controlled using (10), and the coefficients were optimized such that to obtain the best results. Following expression is used to evaluate the inertial weight for each iteration,

$$w = a * \exp(b * i) + c * \exp(d * i) \quad (10)$$

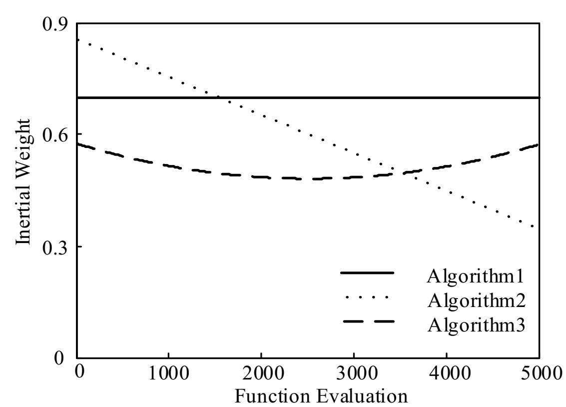

Fig. 1. Inertial Weight Control

FE is the maximum number of function evaluations, which in our case is the termination criteria. The values of the coefficients are mentioned in Table II. Fig. 1 shows the inertial weight control for Algorithm1, Algorithm2 and Algorithm3.

IV. Benchmark Functions

The standard benchmark functions are widely used to test the performance of an algorithm. These benchmark functions may vary from simple unimodal functions to complex multi-modal functions. Table I shows the well-known bench mark functions used to test the performance of general PSO and modified PSO. These functions are generalized for any number of dimensions. This paper presents the behavior of the standard and modified PSO on 30 dimension.

TABLE I. Benchmark Functions

Function	Equation f(x)	Range
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	[-100,100]
Ackley	$f_2(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi x_i)\right) + 20 + \exp(1)$	[-32,32]
Rastrigin	$f_3(x) = 10D + \sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i))$	[-5.12,5.12]
Rosenbrock	$f_4(x) = \sum_{i=1}^{D-1}[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	[-30,30]
Griewank	$f_5(x) = \sum_{i=1}^{D}\frac{x_i^2}{4000} - \prod\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-100,100]
Salomon	$f_6(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i^2}$	[-100,100]
Alpine1	$f_7(x) = \sum_{i=1}^{D}\lvert x_i\sin(x_i) + 0.1x_i\rvert$	[-10.10]

Table II indicates the values of the parameters that are set before the start of the simulation, where FE is the termination criteria.

TABLE II. Simulation Setup

Parameter	Value
Particle Size, PS	30
Dimension, D	30
Function Evaluations, FE	5000*D
$[w_{max}, w_{min}]$	[0.855, 0.345]
$c_1 = c_2$	2
{a,b}	{4.45E-01, -8.18E-06}
{c,d}	{1.30E-01, -8.17E-06}

V. Results

It can be seen from the Table III that the Algorithm1 shows the worst results. It is not recommended to use a constant inertial weight without the velocity clamping, or otherwise the particles will breach the premises of the search space and the algorithm will never be able to converge. When the inertial weight is modified according to (9), there is an improvement in the results as it can be seen in the result section of algorithm2. This method keeps looking for the global minima throughout the simulation time but the convergence is very slow. By introducing the inertial weight control, velocity control and particle penalization, as mentioned in (10), (7) and (8) respectively, sharp convergence can be achieved along with much better results compare to algorithm1 and algorithm2. Algorithm1.1 and Algorithm2.1 are an extension of the Algorithm1 and Algorithm2 respectively, and they utilizes the modification proposed in Section IV. Fig.2 shows the snapshot of all the standard benchmark functions when the above mentioned algorithms are simulated and the dimension is set to 30. It can be seen from the figure that the PSO with constant inertial weight is neither able to converge nor able to find the minima for any standard problem. PSO with linearly decreasing inertial weight keeps looking for the minima throughout the simulation time but the convergence is very slow. The proposed algorithm converges very fast and is able to find the better results compared to the previous techniques.

TABLE III. Summary Of Results

Function		Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 1.1	Algorithm 2.1
Sphere	Mean	1.22E+03	3.77E-30	1.81E-52	1.07E-02	5.48E-34
	STD	3.29E+03	6.78E-30	3.52E-52	2.19E-02	9.60E-34
Ackley	Mean	1.97E+01	2.69E-14	7.99E-15	2.35E-02	1.19E-14
	STD	9.79E-01	8.59E-14	1.58E-30	1.49E-02	3.35E-15
Rastrigin	Mean	1.99E+02	4.45E+01	4.57E+01	5.02E+01	3.33E+01
	STD	1.08E+02	1.04E+01	1.23E+01	1.46E+01	7.76E+00
Rosenbrock	Mean	2.30E+08	9.73E+01	8.01E+01	4.19E+02	8.27E+01
	STD	2.12E+08	9.40E+01	6.06E+01	2.48E+02	7.00E+01
Griewank	Mean	1.30E+02	9.09E-03	7.63E-03	4.95E-02	1.62E-02
	STD	2.23E+02	1.23E-02	8.69E-03	3.88E-02	1.58E-02
Salomon	Mean	1.04E+01	4.60E-01	3.80E-01	1.53E+00	4.60E-01
	STD	7.56E+00	4.90E-02	4.00E-02	2.00E-01	1.20E-01
Alpine1	Mean	3.00E+01	8.78E-14	1.09E-14	2.04E-02	4.22E-14
	STD	1.33E+01	8.59E-14	1.18E-14	3.57E-02	5.05E-14

The integrity of the proposed modification is further verified by using them along with the algorithm1 and algorithm2. The results can be seen in the Table III, under algorithm1.1 and algorithm2.1 respectively. For these algorithms the convergence is very fast but they are not able to compete with the results of algorithm3.

VI. Conclusion

This paper proposes a modification in the standard particle swarm optimization algorithm for the standard benchmark minimization problems. The proposed technique is validated and verified on seven most common minimization benchmark functions. The study comprises of two variants of PSO; with and without velocity clamping, along with three inertial weight variants; constant, linearly decreasing and custom controlled. The proposed algorithm not only modifies the velocity but also penalizes the particle if it breaches the premises of search space and make sure that particle stays within the bounds. It can be concluded from the results that the proposed algorithm results in very fast convergence along with the much better results compare to PSO with constant or linearly decreasing inertial weight.

Acknowledgement

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia for funding this work through the research group project no. RG-1436-016.

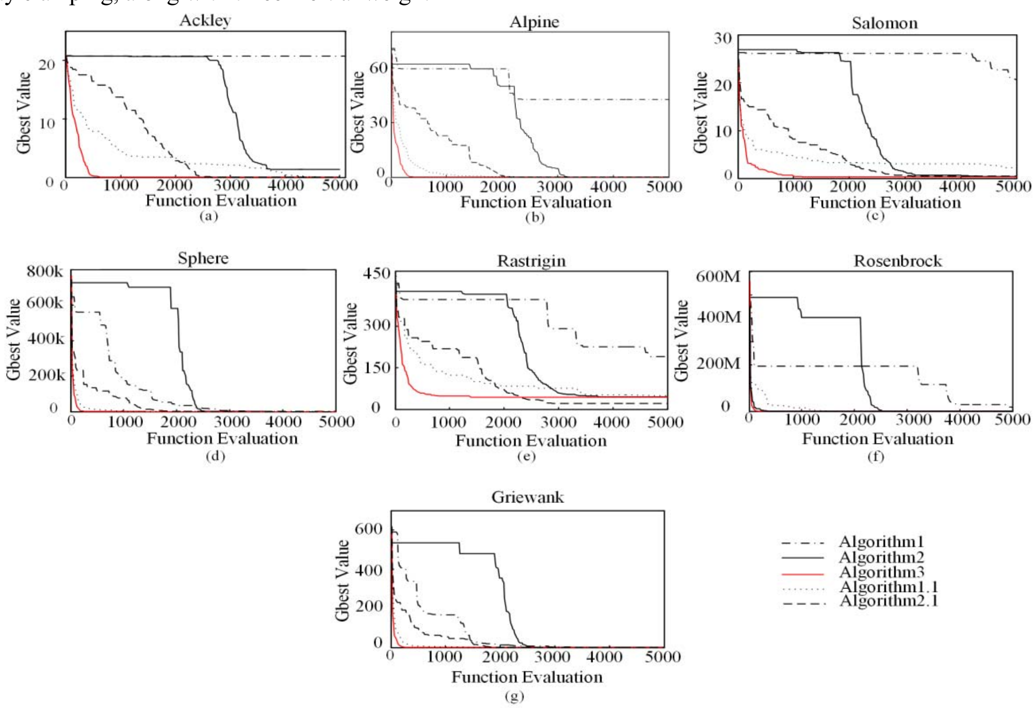

Fig. 2. Benchmark Function Evaluation for D=30, (a) Ackley, (b) Alpine, (c) Salomon, (d) Sphere, (e) Rastrigin, (f) Rosenbrock, (g) Griewank

References

[1] Ebarhart and Kennedy, "Particle Swarm Optimization," 1995.

[2] J. Toner and Y. Tu, “Flocks, Herds, and Schools: A Quantitative Theory of Flocking”, Physical Review E, vol.58, no.4, pp.4828-4858, October. 1998.

[3] Y. Shi and R. Ebarhart, "A modified particle swarm optimizer," 1998.

[4] J. Kennedy and R. C. Eberhart, “Swarm Intelligence”, Morgan Kaufmann Publisher, 2001.

[5] Bansal, J.C.; Singh, P.K.; Saraswat, M.; Verma, A.; Jadon, S.S.; Abraham, A., "Inertia Weight strategies in Particle Swarm Optimization," in Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress, pp.633-640, Oct. 2011.

[6] Eberhart, R. C., and Shi, Y, "Comparing inertia weights and constriction factors in particle swarm optimization," Proc. Proceedings of the IEEE Conference on Evolutionary Computation, San Diego, CA., pp. 84-88 2000.

[7] Muhammad Imran, Rathiah Hashima and Noor Elaiza Abd Khalidb, “An Overview of Particle Swarm Optimization Varients”, Proceedia Engineering, Volume 53, pp.491-496, 2013.

[8] Wei Hong Lim, Nor Ashidi Mat Isa, “Particle swarm optimization with increasing topology connectivity”, in Engineering Applications of Artificial Intelligence, vol. 27, pp. 80-102, Oct 2013.

[9] Mohammad Reza Bonyadi, Zbigniew Michalewicz and Xiaodong Li, “An analysis of the velocity updating rule of the particle swarm optimization algorithm”, in Journal of Heuristics, vol. 20, Issue 4, pp 417-452, Aug 2014.

[10] Simone A. Ludwig, “Repulsive Self-Adaptive Acceleration Particle Swarm Optimization Approach”, in Journal of Artificial Intelligence and Soft Computing Research, vol. 4, Issue 3, pp. 189-204, 2014.

[11] Aderemi Oluyinka Adewumi and Akugbe Martins Arasomwan, “Improved Particle Swarm Optimization based on Greedy and Adaptive Features”, in Swarm Intelligence (SIS), in IEEE Symposium on Swarm Intelligence, pp.1-6, Dec. 2014.