Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 03, March-2017

Multi Objective Particle Swarm Optimization

Ashvini Kulkarni
Department of Electronics and Telecommunication
International Institute of Information Technology, Pune, India

Dr. Datta Parle
Principal Consultant
Infosys Pvt. Ltd. Pune, India

***Abstract* — several optimization techniques are proposed in artificial intelligence. This paper we contrast performance of Swarm Intelligence based PSO search strategy to optimize the multiple objective functions. Experimental analysis also demonstrated the effect of the inertia weight for multiple objective functions in the algorithm. And optimize time for all particles are detected and calculated by Particle Swarm Optimization.**

Keywords—PSO,Inertia Weight,Multi Object PSO(MOPSO)

I. INTRODUCTION

Swarm intelligence is systematic method of obtaining objective function for the self organize and decentralized particles. Swarm intelligence is focused on the collective behavior of the particles that proceed for the local behavior by mutual communicating in particles and update the position to achieve the global position. Swarm intelligence discipline has many algorithms to meets the objective function [1].

In this paper we present the performance of multi object PSO algorithm with different inertia weight, constriction factor and iteration repercussion to achieve objective function.

The remainder of the paper is organized a follows. Section II describes the basic formulation of an Swarm Intelligence. Section III describes Particle Swarm Intelligence algorithm. In section IV consists of multi objective Particle Swarm Optimization and implementation. Section V consists of analysis of PSO and test result in section VI. Conclusion and future work are discussed in section VII.

II. SWARM INTELLIGENCE

Swarm Intelligence is distributed system of self interactive particles with self organized and de-centralize system.Swarm intelligence which involves incorporating prior information about the particles that are dealing with other particle and evaluation of different particles that update the position of particles. The computational complexity for these algorithms is usually much higher. For swarm representation and updating in given frame we have selected one of the method as Particle Swarm Optimization [2][3][4].As PSO techniques are much more similar to that of the evolutionary computation technique like Genetic Algorithm (GA) for system initialization and population based particle position update. The basic difference is based on Swarm Intelligence and GA is through mutation and crossover. The detailed description for PSO is provided in section III.

III. PARCILE SWARM OPTIMIZATION

PSO is inspired by the flocking and schooling patterns of birds and fish, Particle Swarm Optimization (PSO) was invented by Russell Eberhart and James Kennedy in 1995.The main objective of this algorithm is to solve optimization problems. Over a number of iterations, a group of particles have their position values and adjusted over to the member whose value is closest to the target at any given moment. This is similar to a flock of birds flying over an area where they can smell a hidden source of food. The one who is closest to the food chirps the loudest and the other birds swing around in his direction. If any of the other birds comes closer to the target than the first, it chirps louder and the others veer over toward him. This tightening pattern continues until one of the birds happens upon the food. It's an algorithm that's simple and easy to implement.

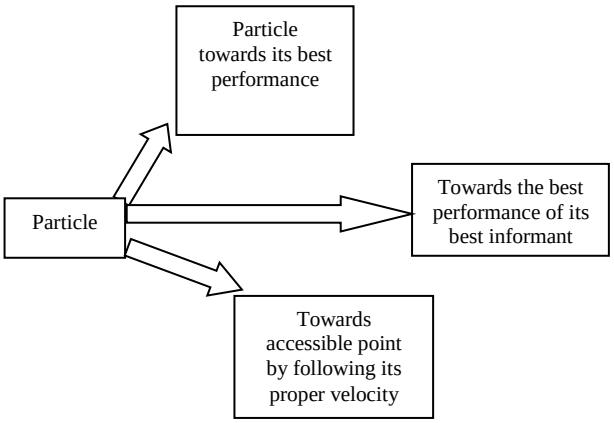

Fig. 1. Fundamenta elements for calculation of the next displacement of a particle

In PSO, nature of the particle movement is based on the three fundamental elements for the calculation of the next displacement of a particle:according to its velocity,towards individual particle best performance and best performance of its best informant.On the basis of three elements movement of the particle is updated.Accordingly three caes are there for the movement as first Particle tend to follow its own way.It has null confidence to receive information from informants and the particles will be satisfied with its more or less for the next displacement of the particle.Second case as particles are conservative and update with its best performance in unceasingly manner. And the third case is when particles do not have any confidence but it moves to its best informants.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 03, March-2017

PSO algorithm estimates the particle position using each 2D coordinate axis in this. In PSO actually is to evaluate the optimization for the target to get the benefit of minimizing efforts and maximize the solution. This algorithm addresses swarm position towards possible particle location, as follows [5]:

$$x_d^{k+1} = x_d^k + v_d^{k+1} \quad (1)$$

$$v_d = c_1 v_d + c_2 (p_d - v_d) + c_3 (g_d - v_d) \quad (2)$$

Where,

v_d = The velocity vector of particle in dimension at time.

x_d = The position vector of particle in dimension at time.

k = Iteration

Above equation is used to give initialize particle position and initialize the velocity for every iteration.
In PSO three fundamental elements for the calculation of next displacement of a particle as according to its velocity, towards particle performance and best performance of its informant so that particles new position is updated by[6]:

$$v^k(t) = v_{old}^k + r_1 + r_2 \quad (3)$$

$$x^k(t) = x^k(t-1) + v^k(t) \quad (4)$$

$$v_{old}^k = \omega v^k(t-1) \quad (5)$$

$$r_1 = c_1 * rand() * \left[P_{best}^k - x^k(t-1)\right] \quad (6)$$

$$r_2 = c_2 * rand() * \left[g_{best}^k - x^k(t-1)\right] \quad (7)$$

Where c_1 and c_2 are parameters that the vector terms, ω serves as intertia weight factor, $rand()$ is uniform distribution random generator in [0,1], P_{best}^k is best particle performance for k^{th} iteration and g_{best}^k is best performance of its best informant.
Every particle converge to the target with inertia weights, the convergence speed in the particle swarm optimization algorithm with the convergence agent is much quicker. In fact, when the proper c_1 and c_2 is decided, the two calculation methods are identical [7]. So, the particle swarm optimization algorithm with convergence agent can be regarded as a special example of the particle swarm optimization algorithm with inertia weights.For the particle update from old position to converge to new position for every iteration till all reaches to the final objective function is given by:

$$v_d^{k+1} = v_d^k + c_1 r_1^k \left(pbest_d^k - x_d^k\right) + c_2 r_2^k \left(gbest_d^k - x_d^k\right) \quad (8)$$

Where,

$pbest_d^k$ = The personal best position of particle in dimension found from initialization through time t

$gbest_d^k$ = The global best position of particle in dimension found from initialization through time t

c_1 and c_2 = Positive acceleration constants which are used to level the contribution of the cognitive and social components respectively.

r_1^k and r_2^k = Random numbers from uniform distribution at time t.

A. *Particles Inertia weight*

An Inertia weight ω is a proportional agent that is related with the speed of last time, and the formula for the change of the speed is the following:

$$v_d^{k+1} = \omega v_d^k + c_1 r_1^k \left(pbest_d^k - x_d^k\right) + c_2 r_2^k \left(gbest_d^k - x_d^k\right) \quad (9)$$

The influence that the last speed has on the current speed can be controlled by inertia weights. The inertia weight ω is the bigger, the PSO's searching ability for the whole bigger and the smaller ω is, the bigger the PSO's searching ability for the partial. Generally, ω is equal to 1, so at the later period of the several generations, there is a lack of the searching ability for the partial. Experimental results show that PSO has the biggest speed of convergence when ω is between 0.8 and 1.2. While experimenting, ω is confined from 0.9 to 0.4 according to the linear decrease, which makes PSO search for the bigger space at the beginning and locate the position quickly where there is the most optimist solution. As ω is decreasing, the speed of the particle will also slow down to search for the delicate partial. The method quickens the speed of the convergence, and the function of the PSO is improved. When the problem that is to be solved is very complex, this method makes PSO's searching ability for the whole at the later period after several generation is not adequate, the most optimist solution cannot be found, so the inertia weights can be used to work out the problem[8].

B. *Convergence of particles*

Convergence of the particles is given by to change positions and the speed. Convergence factor is given by:

$$v_{id=} \chi\{c_1 v_d + c_2 (p_d - v_d) + c_3 (g_d - v_d)\} \quad (10)$$

Where,

$$\chi = 2/\left|2 - \Phi - \sqrt{\Phi^2 - 4\Phi}\right| \quad (11)$$

$$\Phi = c_1 + c_2 \quad (12)$$

C. *Swarm Size*

The size of the swarm is fixed once for all. The more particles, the faster the search will be in terms of the number of iterations. For iteration, the number of evaluations is equal to the number of particles. If particle size is kept small resulting large time to find the objective function.

D. *Initialization of particles*

A search space is defined for the continues problem in the form of $[x_{min}, x_{max}]^D$.Initialization is just a randomly placing particles according to uniform distribution in search space.

E. *Equation of Motion*

The dimension of the search space is *D*. The current position of a particle in this space at the moment *t* is given by a vector *x*(*t*), with *D* components. Its current velocity is *v*(*t*). The best position found up to now by this particle is given by a vector *p*(*t*). Lastly, the best position found by informants of the particle is indicated by a vector *g*(*t*)[9].The *d*th component of one of these vectors is indicated by the index *d* for each dimension *d*. For movement of particles c_1,c_2 plays vital role.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 03, March-2017

The first rule stipulates that $c1$ must have an absolute value less than 1. We pd = xd = gd ,with each increment, velocity is simply multiplied by $c1$. If its absolute value is greater than 1, velocity increases unceasingly and convergence is impossible. Note that, in theory, this coefficient we will assume it to be positive. In practice, this coefficient should be neither too small, which induces a premature convergence, nor too large, which, on the contrary, can slow down convergence excessively.

The second rule states simply that the parameter c_{max} should not be too large, a value of about 1.5 to 1.7 being regarded as effective in the majority of cases. In fact, the recommended values are very close to those deduced later from mathematical analyses showing that for a good convergence the values from $c_1$and c_2 should not be independently selected .The existence of this relation between these two parameters will help us later establish performance maps in only two variables: a parameter ϕ and the size of the swarm.

F. Proximity Distribution

This assigns random variables for displacements obtained while varying independently c_2 and c_3 between 0 and c_{max}. Let us call p the vector whose d^{th} component is:

$$c_1 r_1^k \left(pbest_d^k - x_d^k\right) \quad (13)$$

g the vector whose d^{th} component is:

$$c_2 r_2^k \left(gbest_d^k - x_d^k\right) \quad (14)$$

The distribution of the possible points in the proximities of p and g is uniform.

IV. MULTIOBJECTIVE PSO

Multi objective PSO is a class of PSO in which multiple solutions are evaluated with one or more objective.PSO is population based approach. Algorithm for multi objective PSO:

1) Specify no. of particles, size of the particles, iteration and objective function.
2) Assume the initial velocity of the particle.
3) Each particles move towards new position if local best position of each particle is better than present position.
4) If local best position is better to reach the target then all other particles also follows the same position and global best position is updated.
5) Go to the step 2 and update local and global best position.

V. TEST DESCRIPTION

The test cases are used for evaluation of particles to reach to the objective function. These test cases also describe the number of particles and their relation to reach to the objective function based on the different parameter viz. iteration, inertia weight, multiple objective function and optimum time which affect particle swarm optimization performance.

We first optimize single objective function where all particles are moving towards it. In next part, same numbers of particles are optimized to the multiple objective functions. In further part same objective function is further analyzed for the multiple parameters for the multi-PSO function.

In this analysis we have taken single objective function as:

$$objective\ function = (x-20)^2 + (y-15)^2 \quad (15)$$

and three multiple objective function as:

$$1st\ objective\ function = (x-20)^2 + (y-15)^2 \quad (16)$$

$$2nd\ objective\ function = (x-15)^2 + (y-18)^2 \quad (17)$$

$$3rd\ objective\ function = (x-16)^2 + (y-15)^2 \quad (18)$$

In simple PSO with one objective function is tested for six particles shown in blue color. And for PSO with multi objective function is tested for four particles which are shown in red, green and blue color. All objective function have tested on MATLAB 14a tool.

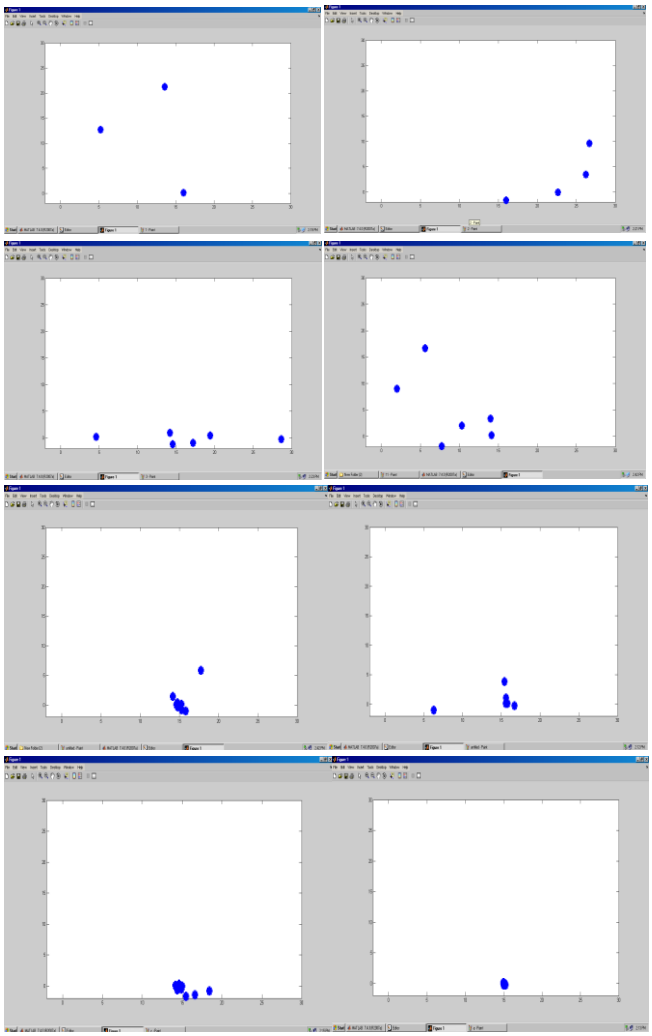

Fig. 1. Trajectory of particles with single objective function

Figure 1 illustrates the single objective function according to the equation number (15).All the particles are converging to the objective function.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 03, March-2017

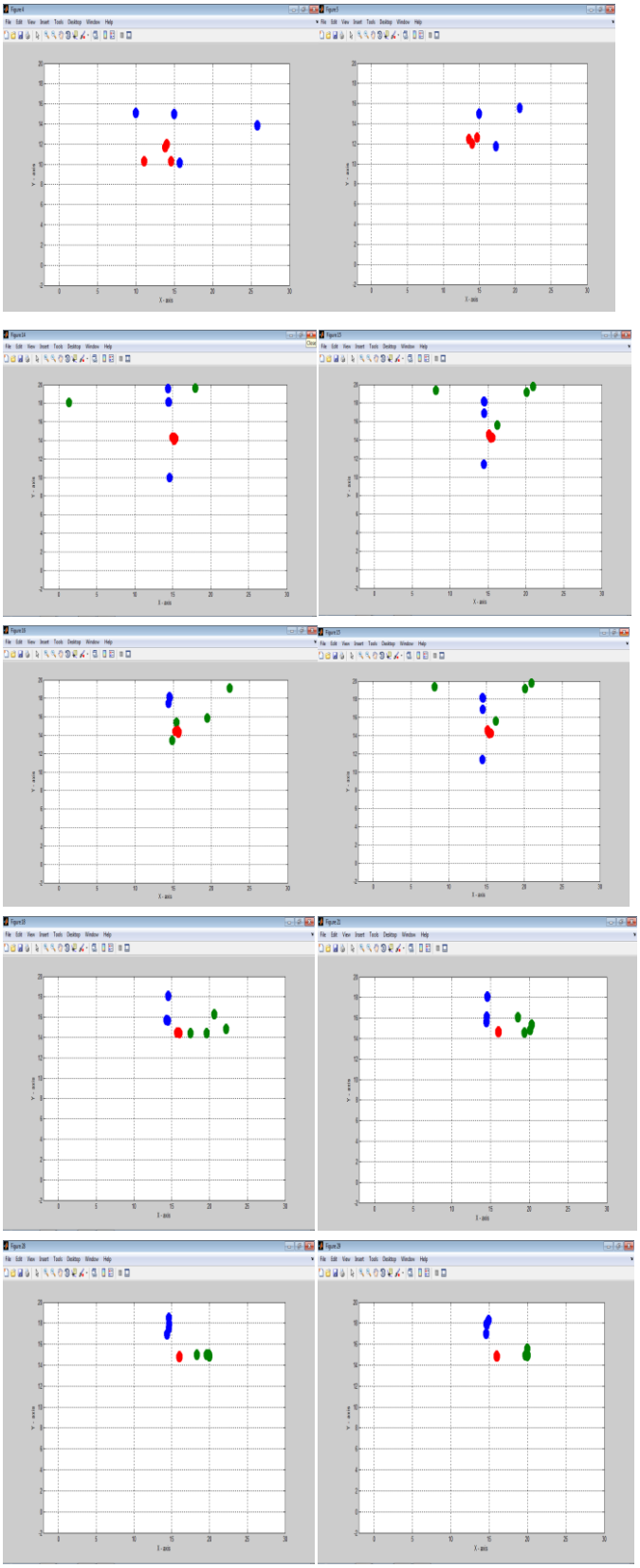

Fig. 2. Trajectory of multiple particles with multiple objrvtive function

Figure 2 illustrates the single objective function according to the euation number (16),(17),(18).PSO is analyzied according to the particle weight,iteration

VI. RESULT

Objective function we have taken as specified in equation (15).This objective function is for the four particles and all particles are moving towards objective function by changing the local updating position and global updating position in every iteration.As a result we have observed that particles are converged according to the objective function.The intertia weight of particles affects the converging time.in single objective function and multiobjective function.For the MOPSO inertia weight is considered between 0.4 to 0.8 for all objective function and analyzed with increasing iteration as a result when inertia weight is increase,optimized time for particles to reach to objective function is gradually increased. MOPSO is not much affected by particle weight at the same time MOPSO's optimize time may get change if number of particles are increased as shown in table II.If number of iterations are increased in most cases converging time may get decreased because of tendency of particles to share their position information.

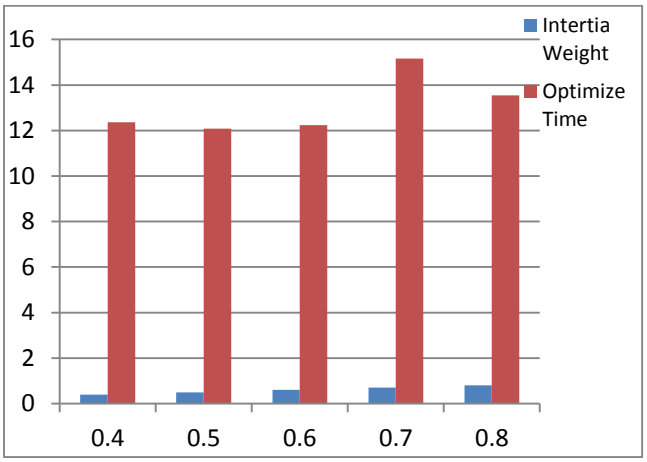

Fig. 3. Plot of Particle Converging to the objective function with different iteration and its optimize time with 0.4 to 0.8 inertia of particles.

Figure 3 shows the performance comparison of particles converging to the objective function where all particles are having inertia weight of from 0.4 to 0.8.This figure shows the inertia weight of particles on x-axis and time on y-axis for 50 iterations.

TABLE I. PERFORMANCE OF SINGLE OBJECTIVE PSO AND MULTIOBJECTIVE PSO

Objective function	No. of particles	Elapsed Time
Single	*9*	*13.28*
Multiple	9	*12.46*

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 03, March-2017

TABLE II. PERFORMANCE OF MPSO

MPSO				
No. of particles	Iteration	Optimize Time for objective functions		
		Intertia Weight 0.8	Intertia Weight 0.7	Intertia Weight 0.6
4	50	13.54	15.16	12.23
	100	25.26	24.25	24.89
	150	39.59	36.21	36.79
	200	51.14	48.6	48.79

VII. CONCLUSION

In this paper, we have described and evaluated the use of multi objective PSO (MOPSO) approach to improve converging performance of PSO algorithm. We have shown that the single objective function is converging and multi objective function converging. In the trajectory of particles weight affects the optimized time. We have also shown that different particle weight effect with time due to which particles are moderately giving time for same number of iterations. Hence we conclude that from the evaluation the PSO is very fast for converging to the objective function.PSO is not much affected by particle weight at the same time PSO's optimize time may get change if number of particles are increased as shown in table No.II. If number of iterations are increased in most cases converging time may get decreased because of tendency of particles to share their position information. Furthermore this framework
naturally extended to path tracking PSO.

VIII. REFERENCES

[1] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference Neural Networks, 1995, pp. 1942–1948

[2] K.E. Parsopoulos, M.N. Vrahatis, Particle swarm optimization method in multiobjective problems, in: Proceedings of the 2002 ACM

[3] S. Mostaghim, J. Teich, Strategies for Finding Good Local Guides in Multi-objective Particle Swarm Optimization (SIS'03), IEEE

[4] Engelbrecht, A. P.,Fundamentals of Computational Swarm Intelligence, John Wiley & Sons, 2005.

[5] Coello, C. A. C., G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," IEEE Trans. Evolutionary Computat., Vol. 8, 256–279, 2004.

[6] N. Srinivas and K. Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," Evolutionary Computation, 1994, 2, pp. 221-248.

[7] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," IEEE Transactions on Evolutionary Computation, 1999, 3, pp. 257-271.

[8] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," Evolutionary Computation, 2000, 8(2), pp. 173-195.

[9] Ngatchou, P.; Zarei, A.; El-Sharkawi, A.; "Pareto Multi Objective Optimization" Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on 6-10 Nov. 2005 Page(s):84 - 91

[10] C. A. Coello Coello and M. S. Lechuga. "MOPSO: A proposal for multiple objective particle swarm optimization". In IEEE Proceedings World Congress on Computational Intelligence,pages1051–1056, 2003.