INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

Particle Swarm Optimization (PSO) Algorithm: Parameters Effect And Analysis

Vishal A. Rane
Asst. Prof., Mech. Dept. G.H. Raisoni Institute of Engg. & Management, Jalgaon, India

Abstract:
Optimization is a method of finding the optimum solution i.e. finding the maximum or minimum of a given objectives, subjected to various constraints. In the literature, various advanced optimization techniques are available out of which particle swarm optimization is one of the advanced optimization technique. Particle swarm optimization (PSO) is an efficient optimization method. Like other algorithm PSO is also population based algorithm. The PSO is inspired by the metaphor of social interaction observed between fishes or birds. In a PSO algorithm, each particle is a candidate solution and each particle "flies" through the search space, depending on two important factors; the best position the current particle have found so far and the global best position identified from the entire population. PSO has been used in many fields such as in aerospace design, manufacturing, heat transfer and automobile. In the present work PSO algorithm is applied on standard benchmark functions such as Rastrigin function, problem with equality and inequality constraints. The particle swarm optimization algorithm is applied on different Benchmark function for tunning various parameters like inertia weight, social parameter and cognitive parameter. Results obtained by particle swarm optimization algorithm are compared with results of previous work and it is observed that the results obtained by PSO algorithm are better than the previous result.
Key words*: Matlab, inertia weight, social parameter, cognitive parameter.*

1.Introduction
The goal of many engineers is to design system for automotive, aerospace, mechanical, civil, chemical, industrial, electrical, biomedical, agricultural, naval and nuclear engineering applications. In the highly competitive world of today, it is no longer sufficient to design a system that performs the required task satisfactorily. It is essential to design the best system. Best means an efficient, versatile, unique and cost effective system. To design such system proper analytical, experimental and numerical tools means optimal parameters are needed. Optimum design concepts and method provide some of the needed tools. When they are properly implemented into software, they give optimal parameters for designing best system.
Optimization is concerned with finding the maximum or minimum of functions, subjected to constraints. In most design activities, the design objective could be simply to minimise the cost of production or to maximise the efficiency of production. An optimisation algorithm is a procedure which is executed iteratively by comparing various solutions till the optimal or satisfactory solution is not found. In many industrial design activities, optimisation is achieved indirectly by comparing the few chosen optimisation design solutions and accepting the best solution. In the simplest case, an optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function.

2.Particle Swarm Optimization (PSO)
The Particle Swarm Optimization (PSO) is introduced by James Kennedy and Russell Eberhart in 1995 [1]. PSO is an evolutionary computation technique like genetic algorithms. Since PSO have many advantages such as comparative simplicity, rapid convergence and little parameters to be adjusted, it has been used in many fields such as mechanical, chemical, civil, aerospace design etc. The particle swarm algorithm is an optimization technique inspired by the metaphor of social interaction observed among insects or animals. The kind of social interaction modelled within a PSO is used to guide a population of individuals (particles) moving toward the most promising area of the search space. In a PSO algorithm, each particle is a candidate solution and each particle "flies" through the search space, depending on two important factors; the best position the current

particle have found so far and the global best position identified from the entire population. The rate of position change of particle is given by its velocity. Particles velocity and positions are updated related to the pbest and gbest values.

2.1.Literature Review On PSO

In the past various researchers had used PSO for different application. A brief review is carried out here by studying several research papers. The Particle swarm optimization (PSO) algorithm, originally developed by Kennedy and Eberhart [1995], they introduced concept for the optimization of nonlinear functions using particle swarm methodology.

Shi et al. [2] explained that a novel particle swarm optimization (PSO)-based algorithm for the travelling salesman problem (TSP). And this PSO-based algorithm was proposed and applied to solve the generalized travelling salesman problem by employing the generalized chromosome. PSO-based algorithm has been used to solve the generalized travelling salesman problem (GTSP) problems. Some benchmark problems are used to examine the effectiveness of the proposed algorithms. Rao and Patel [3] presented that the use of particle swarm optimization (PSO) algorithm for thermodynamic optimization of a cross flow plate-fin heat exchanger. Minimization of total number of entropy generation units for specific heat duty requirement under given space restrictions, minimization of total volume, and minimization of total annual cost were considered as objective functions and treated individually. Based on the applications, heat exchanger length, fin frequency, numbers of fin layers, lance length of fin, fin height and fin thickness or different flow length of the heat exchanger were considered for optimization.

Perez and Behdinan [4] presented in detail the background and implementation of a particle swarm optimization algorithm suitable for constraint structural optimization tasks. Improvements, effect of the different setting parameters, and functionality of the algorithm were shown in the scope of classical structural optimization problems. They optimised cross sectional area towards the minimization of total weight. The simplicity of implementation of the PSO along with the lower number of setting parameters makes it an ideal method for different structural optimization tasks. Savsani et al. [5] presented two advanced optimization algorithms known as particle swarm optimization (PSO) and simulated annealing (SA) to find the optimal combination of design parameters for minimum weight of a spur gear train. Because many high-performance power transmission applications (e.g., automotive, aerospace, machine tools, etc.) require low weight. They considered two cases of optimal design of a spur gear train and the objective considered in both the cases minimize total weight.

Carlos et al. [6] presented two examples of maintenance optimization using Particle Swarm as optimization technique and a tolerance interval based approach to address uncertainty, one was focused on a safety component and the other considers a nuclear power plant safety system. In nuclear power plants maintenance planning is a subject of importance from the safety point of view. Traditionally, the maintenance planning was formulated in terms of a constrained multi objective optimization problem involving different and usually conflicting criteria, such as unavailability and cost. Ho et al. [7] introduced to employ a particle swarm optimisation (PSO) approach to devise a railway timetable in an open market. The suitability and performance of PSO are studied on a multi agent based railway open-market negotiation simulation platform. Railway timetabling is an important process in train service provision as it matches the transportation demand with the infrastructure capacity while customer satisfaction was also considered. It was a multi objective optimisation problem, in which a feasible solution, rather than the optimal one, is usually taken in practice because of the time constraint.

Liu [8] described particle swarm optimization algorithm to estimate the unknown wall heat flux from transient temperature measurements at the boundary of a plane wall. An inverse computational method has been developed for the nonintrusive evaluation of the time-dependence of wall heat flux. The methodology is based on a particle swarm optimization algorithm with modifications. Two methods of modifications on the basic particle swarm optimization algorithm presented by him.

2.2.Methodology

PSO is similar to a genetic algorithm (GA) in that the system is initialized with a population of random velocity, and the potential solutions, called *particles*, are then “flown” through the problem solutions. It is unlike a GA, however, in that each potential solution is also assigned a randomized space.

Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. (The fitness value is also stored) This value is called pbest. Another “best” value that is tracked by the global version of the particle swarm optimizer is the overall best value, and its location, obtained so far by any particle in the population. This location is called gbest. The particle swarm optimization concept consists of, at each time step, changing the velocity (accelerating) each particle toward its pbest and gbest locations (global version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pbest and gbest locations.

El-Zonkoly [9] explained concept of modification of a searching point by PSO with Fig. 1.

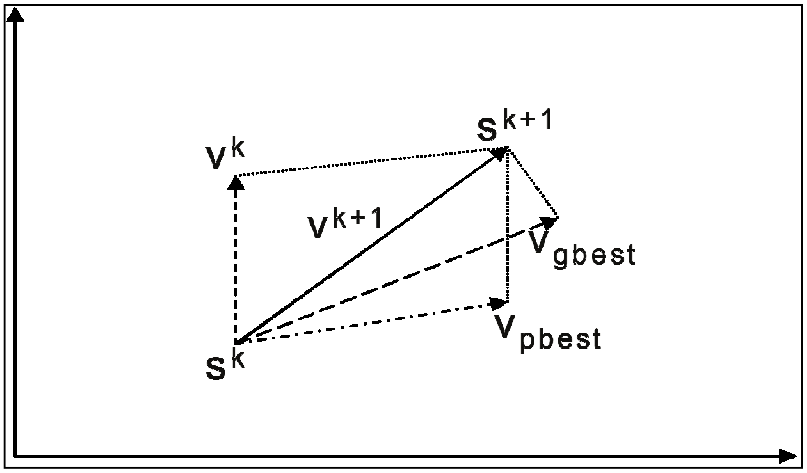

Figure 1: Concept of modification of a searching point by PSO

Where,

- s^k: current searching point.
- s^{k+1}: modified searching point.
- v^k: current velocity.
- v^{k+1}: modified velocity.
- v_{pbest} : velocity based on pbest.
- v_{gbest} : velocity based on gbest

2.3.Steps In PSO

- Initialize a population (array) of particles with random positions and velocities on'd' dimensions in the problem space.
- For each particle, evaluate the desired optimization fitness function in d variables.
- Compare particle's fitness evaluation with particle's pbest. If current value is better than pbest, then set pbest value equal to the current value, and the pbest location equal to the current location in d-dimensional space.
- Compare fitness evaluation with the population's overall previous best. If current value is better than gbest, then reset gbest to the current particle's array index and value.
- Change the velocity and position of the particle according to equations (1) and (2), respectively:

$Vnew_{i,j}=w*V_{i,j}+ C_1*r_1*(Pbest-X_{i,j})+C_2*r_2*(Gbest- X_{i,j})$ (1)

$Xnew=X_{i,j}+Vnew_{i,j}$ (2)

Where,

- Vnew = updated velocity
- w = inertia weight
- $V_{i,j}$=initial velocity
- C_1 = social parameter
- C_2 =cognitive parameter
- Pbest = personal or local best
- Gbest = global best
- $X_{i,j}$ = initial Particle position
- Xnew=updated particles position
- r_1, r_2 = Random numbers between 0 and 1

Loop to step (2) until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations (generations).

3.Case Studies

3.1.Case Study 1:Rastrigin Function

The Rastrigin function is a function used as a performance test problem for optimization algorithms. It is a typical example of non-linear multimodal function. It was first proposed by Rastrigin as a 2-dimensional function and has been generalized by Muhlenbein. This function is a fairly difficult problem due to its large search space and its large number of local minima. This function can be use for n number of variables. This case study is taken from research paper of Kita et al. [10].

It is defined by:

$$f(x)=An+\sum_{i=1}^{n}\left(x_i^2 - A\cos(2\pi x_i)\right) \qquad (3)$$

Where A = 10 and X_i search domain is (-5.12, 5.12)
The global optimal solution for this function is 0 at X_i is 0.

3.1.1 Effect Of Inertia Weight On Fitness Value

The PSO algorithm is applied on Rastrigin function. Various parameters setting are required for this algorithm. In most of the cases inertia weight plays an important role in this algorithm. Therefore particle swarm optimization algorithm is applied for various inertia weights (w) ranges from 0-1, the results are listed in Table 4.1. The number of particles used for setting the inertia weight are 10, number of generation (iteration) 100 at social parameter C_1 equal to 2 and cognitive parameter C_2 equal to 2.

W(inertia weight)	**X_1**	**X_2**	**Fitness values(f)**
1.0	-0.9791	-0.9360	2.79946
0.95	-1.0558	0.0594	2.42135
0.90	0.0607	-1.0105	1.76459
0.85	-0.9519	0.0247	1.47994
0.80	0.9973	-0.0180	1.03741
0.75	0.9839	-0.0047	1.02130
0.70	-0.0686	-0.0015	0.93208
0.65	-0.0520	0.0359	0.07709
0.60	0.0147	0.0421	0.44710
0.55	-0.0117	-0.0488	0.44835
0.50	0.0364	-0.0073	0.27245
0.45	0.0243	-0.0051	0.11434
0.40	0.0049	-0.0082	0.01431
0.35	$-2.56e^{-4}$	-0.0044	0.00382
0.30	$-2.0785e^{-7}$	0.0017	$5.9905e^{-4}$
0.25	$5.710e^{-5}$	$-3.01e^{-5}$	$3.2106e^{-7}$
0.20	-0.0014	$-6.5506e^{-4}$	0.000499
0.15	-0.0023	$7.4677e^{-7}$	0.00102
0.10	-0.0025	$-2.9020e^{-4}$	0.00128
0.05	0.995	$-7.0762e^{-10}$	0.99500
0.00	0.09394	-0.0427	1.98993

Table 1: Effect of Inertia weight on fitness value for Rastrigin function

From Table 1, it is observed that maximum fitness value 2.799 occurs at when the inertia weight is higher at its upper limit. The value of fitness function decreases as the inertia weight decreases up to w=0.25. Beyond which the fitness values gradually increases, as the inertia weight increases. The minimum fitness value is obtained when the inertia weight attains the value of 0.25.

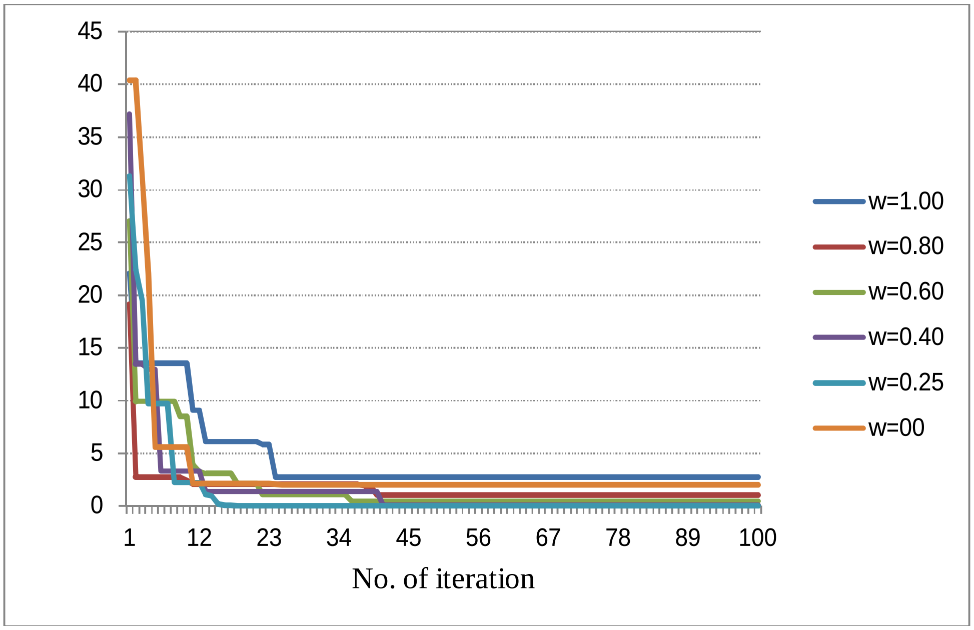

Figure 2: Effect of Inertia weight on fitness value for Rastrigin function

From Fig. 2 it is clear that for inertia weight w=1 the fitness value decreases significantly to the value of 2.799 up till 25th iteration, beyond which it is constant. Also for w=0.25 the fitness value decreases to zero in 15th iteration and therefore we are getting the lowest fitness value among all the cases of inertia weight.

3.1.2. Effect Of Acceleration Parameters On Fitness Value

C_1	C_2	X_1	X_2	F(x)
0	4	0.9943	-0.0010	0.9953
1	3	0.0195	-0.00213	0.1648
2	2	$7.1240e^{-5}$	$-2.837e^{-6}$	$1.0085e^{-6}$
3	1	-0.0064	0.9845	1.0247
4	0	1.0109	-0.0037	1.0480

Table 2: Effect of acceleration parameters on fitness value for Rastrigin function

Now in Table 2 different combination of acceleration parameters C_1 and C_2 are taken and fitness value is calculated for all cases. It is observed that the value of fitness function is not optimal when C_1 and C_2 are 0 and 4 respectively. When C_1 and C_2 both attain the same value equal to 2 we get optimal solution.

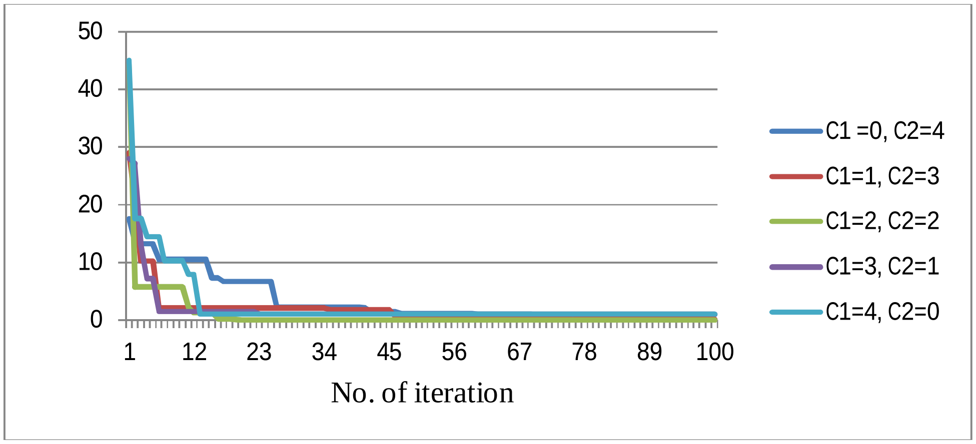

Figure 3: Effect of acceleration parameters on fitness value for Rastrigin function

At C_1=0 and C_2=4, the obtained solution is not optimal but when C_1=2 and C_2=2 optimal solution obtained in 15th iteration. (Fig. 3)

3.1.3.Effect Of Population Size On Fitness Value

Population size	X_1	X_2	F(x)
2	2.1125	2.0097	10.9139
5	-1.0147	-0.9105	3.4399
10	0.0023	0.0011	0.0012
15	$4.824e^{-4}$	$8.984e^{-4}$	$2.06e^{-4}$
20	$-3.865e^{-5}$	$-3.3664e^{-5}$	$5.213e^{-7}$
25	$1.897e^{-6}$	$5.56e^{-6}$	$6.860e^{-9}$

Table 3: Effect of population size on fitness value for Rastrigin function

Table 3 shows variation of population size on the fitness function. From the table it is clear that as the value of population is increases the solution moves towards optimal solution. At population size equal to 2 the value of fitness function is 10.913 and it is large deviation from optimal solution. Also, at population size equal to 25 the fitness value reaches optimal solution with its value near to zero.

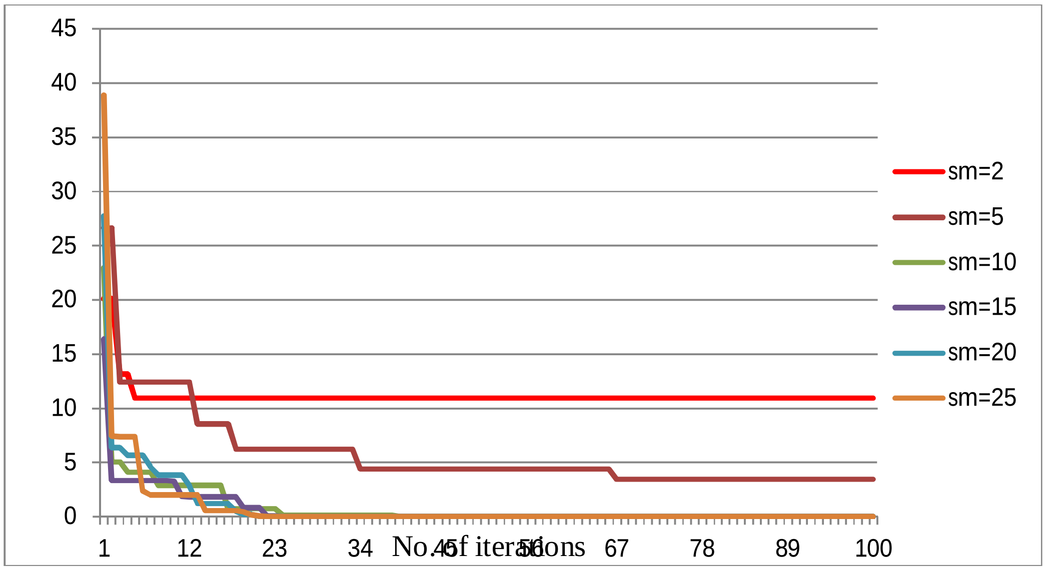

Figure 4: Effect of population size on fitness value for Rastrigin function

Fig. 4 shows that when number of population is 2 fitness value is constant after 5th iteration but the fitness value obtained is 10.91 which is very far away from optimal solution. When particle increases from 2 to 5 this fitness value decreases 3.44 at 68th iteration beyond which it is constant. As the particle size increases the solution moves towards more near to the optimal solution.

3.2.Case Study 2: Equality And Inequality Constraints

Many engineering optimization problems are constrained by equality and inequality constraints that can be linear or nonlinear. Therefore, the test case was chosen to evaluate the ability of the proposed method in solving such constrained optimization problems. This case study is taken from literature that presented by Zhao et al. [11].

- The test function g08 contain inequality constraints which is

Minimize

$$f(x) = (x_1-10)^3+(x_2-20)^3 \quad (4)$$

Subjected to constraints

$$g_1=100-(x_1-5)^2-(x_2-5)^2 \leq 0 \quad (5)$$

$$g_2=(x_1-6)^2+(x_2-5)^2-82.81 \leq 0 \quad (6)$$

And $13 \leq x_1 \leq 100$

$0 \leq x_2 \leq 100$

- The function g11 contain equality constraints which is

Minimize

$$f(x)=x_1^2+(x_2-1)^2 \quad (7)$$

Subjected to constraints

$$g_1 = x_2 - x_1^2 = 0 \quad (8)$$

And $-1 \leq x_1 \leq 1$

$-1 \leq x_2 \leq 1$

In the function g08 there are two inequality constraints with two variables with their bounds. In function g11 there is one inequality constraint with two variables. Both test function are example of minimization. Zhao et al. [11] obtained solution in 10000 generation using 200 number of population. After application of PSO algorithm, the optimal solution of single-objective test problems with equality, and inequality is obtained when the population is 100.

3.2.1 Effect Of Inertia Weight On Fitness Value For G08 Function

Inertia weight plays an important in PSO algorithm there effect on g08 function is given below.

w	X_1	X_2	F(x)
1.00	14.3430	1.3662	-6388.13
0.75	14.2391	1.1738	-6596.34
0.50	14.1880	1.0352	-6747.12
0.25	14.1074	0.8697	-6931.83
0.00	14.2464	1.1905	-6578.18

Table 4: Effect of Inertia weight on fitness value for g08 function

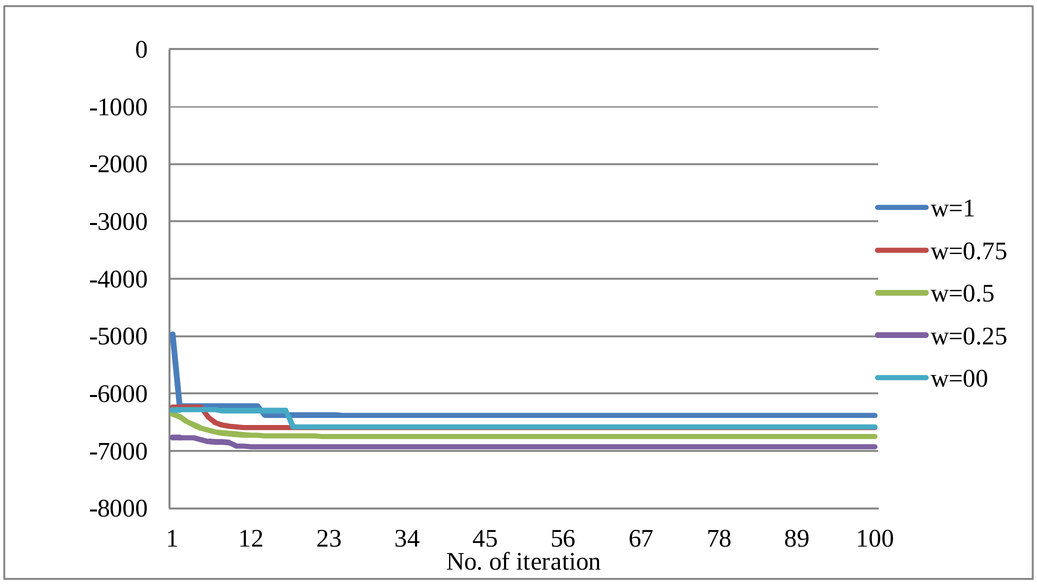

Figure 5: Effect of Inertia weight on fitness value for G08 function

From Table 4 it is observed that for inertia weight w=1 the fitness value decreases to the value of -6388.13 in 14^{th} iteration, after that it is constant and which far beyond the global optimal solution. As the inertia weight decreases from 1 to 0.25 obtained fitness values are moves toward the optimal solution. As w=0.25, the value of fitness function is much closer to the optimal solution. Fig. 5 shows that inertia weight 0.25 gives better result in minimum number of iteration and gives minimum fitness value. The number of particles used are 50 and number of iterations are 100 with C_1=2 and C_2=2.

3.2.2.Effect Of Acceleration Parameters On Fitness Value For G08 Function

C_1	C_2	X_1	X_2	F(x)
0	4	14.3054	1.3170	-6441.62
1	3	14.1812	1.0234	-6760.56
2	2	14.107	0.8697	-6931.83
3	1	14.322	1.0927	-6684.71
4	0	14.322	1.3210	-6436.42

Table 5

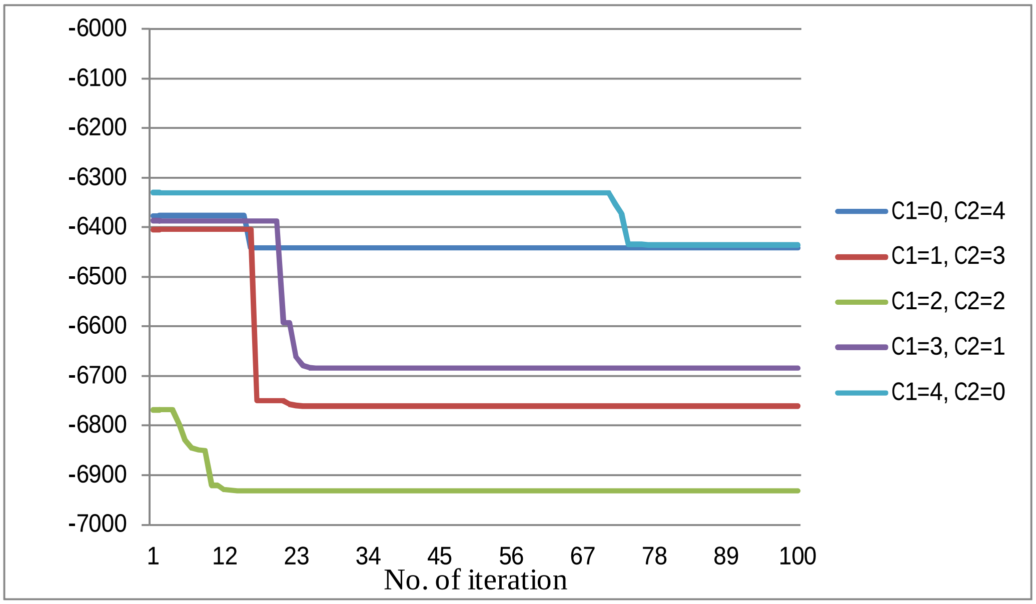

Figure 6: Effect of acceleration parameters on fitness value for g08 function

For setting the acceleration parameters C_1 and C_2 calculate fitness function values for different combination of C_1 and C_2 as shown in Table 5. And graph shows the effect of acceleration parameters on fitness values Fig. 6. With taking constant inertia weight 0.25, number of particles are 50 with 100 generation. It is observed that the minimum fitness value is occurring at C_1 and C_2 both attains the same value of 2.

3.2.3 Optimal Solution For g08 and g11 Test Function

Test function	No. of iteration	Optimal solution	PSO optimal solution
g08	200	-6961.81388 x_1= 14.095 x_2=0.84296	-6960.47655
g08	400		- 6961.76851
g08	600		-6961.80988
g08	800		-6961.81344
g08	1000		-6961.81381
g08	1200		-6961.81384
g08	1500		-6961.81388 x_1= 14.095 x_2=0.84296
g11	1500	0.75 x_1=± 0.7071068, x_2=0.5	0.75 x_1=0.7071068 x_2=0.5

Table 6: Optimal solution for g08 and g11 test function

From the Table 6 it is concluded that the optimal fitness value occurs at 1500 number of generation with number of particles are 100. For this the inertia weight was taken equal to 0.25, the value of social and cognitive parameters were 2.from the literature reviewed it is found that Zhao et al. [2007] obtained optimal solution in 10000 number of generation with 200 particles. Therefore the result obtain by PSO is better.

4.Conclusion

In the present work particle swarm optimization algorithm (PSO) is considered out of the various advanced optimization techniques available in the literature. It is found that particle swarm optimization algorithm provides much better solution as compared to the other optimization techniques. Literature review was carried out for PSO. It is concluded that the particle swarm optimization algorithm is used in many engineering applications for optimizing the various design variables and it gives better performance than many other optimization algorithms.

Various standard benchmark functions are selected for the present work such as Rastrigin function and equality and inequality constraints objective function. The particle swarm optimization algorithm was tested on these selected benchmark function. By varying parameters of particle swarm optimization algorithm such as inertia weight, social parameter C_1, cognitive parameter C_2 and number of particles case studies were done. It gives best performance at 0.25 inertia weight, 2 as social parameter and cognitive parameter is 2. It gives optimal result than the other optimization algorithm with minimum number of particles and minimum number of generations.

5.Referances

1. Kennedy J., Eberhart R. C. (1995), “Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 4”, IEEE Service Center, Piscataway, NJ, , 1942–1948.
2. Shi X.H., Liang Y.C., Lee H.P., Lu C., Wang Q.X. (2007), “Particle swarm optimization-based algorithms for TSP and generalized TSP”, Information Processing Letters 103, 169–176
3. Rao R. V., Savsani V. J., Vakharia D. P. (2011), “Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems” Computer-Aided Design 43, 303–315
4. Perez R. E., Behdinan K. (2007), “Particle swarm approach for structural design optimization”, Computers and Structures 85, 1579–1588
5. Savsani V., Rao R. V. , Vakharia D. P. (2010), “Optimal weight design of a gear train using particle swarm optimization and simulated annealing algorithms”, Mechanism and Machine Theory 45, 531–541
6. Carlos S., A. Sanchez A., Martorell S. (2012), Villanueva J. F., “Particle Swarm Optimization of safety components and systems of nuclear power plants under uncertain maintenance planning” Advances in Engineering Software
7. Ho T.K., Tsang C. W., Ip K. H., Kwan K. S. (2012), “Train service timetabling in railway open markets by particle swarm optimisation” Expert Systems with Applications 39, 861–868
8. Liu F. B. (2012), “Inverse estimation of wall heat flux by using particle swarm optimization algorithm with Gaussian mutation” International Journal of Thermal Sciences 54, 62-69
9. El-Zonkoly A.M. (2011), “Optimal placement of multi-distributed generation units including different load models using particle swarm optimization”, Swarm and Evolutionary Computation 1, 50–59
10. Kita E., Kan S., Fei Z. (2010), “Investigation of self-organizing map for genetic algorithm” Advances in Engineering Software 41, 148–153
11. Zhao Y., Xiong S., Li M. (2007), “Constrained single and multiple-objective optimization with differential evolution” Third International Conference on Natural Computation, 0-7695-2875