Particle Swarm Optimization Performance for Unconstrained Optimization Problems

Snehal Kamalapur[1], Varsha Patil[2], Shirish Sane[1]
[1]K.K. Wagh Institute of Engineering
Education and Research. India
Snehal_kamalapur@yahoo.com, SSSane65@yahoo.com

[2]Matoshri College of Engineering
Nashik, India
varsha.patil@gmail.com

ABSTRACT: *Particle swarm Optimization (PSO) is mainly inspired by social behavior patterns of organisms that live and interact within large groups. The term PSO refers to a relatively new family of algorithms that is used to find optimal or near to optimal solutions to numerical and qualitative problems. It is an optimization paradigm that simulates the ability of human to process knowledge. The capability of PSO method to address the maximization and minimization unconstrained problems is investigated through numerous experiments on different test problems. Results obtained are reported. The two variants PSO-IW (Inertia Weight) and PSO-IC (Inertia weight and Constriction factor) are used for the experiments. Conclusions are derived. These variants exhibit different performance for different test problems.*

Keywords: Maximization, Minimization, Particle swarm optimization, Unconstrained problem

Received: 11 March 2010, Revised 2 April 2010, Accepted 10 April 2010

1. Introduction

Finding a maximum or a minimum has its application in pure mathematics like finding the sides to get maximum area of rectangle. It also has its application to commercial problems, such as finding the least material used of a tank that has given volume. For all these problems it is necessary to write the problem in the form of mathematical function.

For solving max/min problems manually following steps are followed.

a) Preparation
To know exactly what the problem is asking read each problem carefully. You may restate the problem accordingly.

b) Translation
If appropriate, draw a sketch or diagram of the problem to be solved.

Define variables to be used and carefully label the diagram with these variables. This step is very important as it guide directly or indirectly to write the mathematical equations.

c) Optimization formulation
Write down all the equations that are related to the problem or diagram that you have drawn. Most of the optimization problems will begin with two equations. One equation is a “constraint” equation and the other is the “optimization” equation. Clearly indicate that equation which is to maximize or minimize. The “constraint” equation is used to solve for one of the variables. This is then substituted into the “optimization” equation. Some problems may not have constraint equation while other problems may have two or more constraint equations.

d) Applying
This depends on the problem. You may use graphical approach, systematic numerical search, calculus etc.

e) Verification

Verify that your result is a maximum or minimum value as per problem.

Depending on the complexity of optimization problem the manual work becomes tedious. Here we have used Particle swarm optimization (PSO) technique to solve three problems out of which two problems can be converted to one variable and another problem of two variables. PSO exhibits good performance in finding solutions to static optimization problems [1, 2, 3] In PSO all the particles have fitness values. These values are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space. The particles fly by following the current optimum particles.

In this paper the maximization and minimization unconstrained problems is explored using PSO. The two variants of PSO are considered here. First uses only inertia weight which is denoted as PSO-IW, and second utilizes both inertia weight as well as constriction factor and denoted as PSO-IC. In the next section PSO approach is presented. In section 3, the test problems as well as the experimental results are reported. The paper ends with conclusions.

2. Particle Swarm Optimization Method

Particle swarm optimization (PSO) is optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995[4]. PSO is mainly inspired by social behavior patterns of organisms that live and interact within large groups. In particular, PSO incorporates swarming behaviors observed in insects, birds and fish. PSO technique provides an evolutionary based search. The term PSO refers to a relatively new family of algorithms that is used to find optimal or near to optimal solutions to numerical and qualitative problems. PSO optimizes an objective function by undertaking a population based search. PSO is Global gradient-less stochastic search method and is well suited to continuous variable problems.

In PSO all the particles have fitness values. These values are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space. The particles fly by following the current optimum particles.

2.1 The PSO algorithm is revisited here-

a. Initialize the population - locations and velocities

b. Evaluate the fitness of the individual particle (local Best or pbest). Each particle keeps track of its coordinates in the problem space which are associated with the best solution it has reached so far.

c. Keep track of all individuals' highest fitness (global Best or gbest)

d. Modify velocities depending on local best and global best position

The velocity of all particles is updated according to:

$$v[] = w * v[] + c1 * r1 * (pbest[] - present[]) + c2 * r2 * (gbest[] - present[]) \tag{1}$$

v[] is the particle velocity, w is inertia weight persent[] is the current particle (solution), pbest[] and gbest[] are defined as stated before, rand () is pseudorandom, scalar value drawn from a uniform distribution on the unit interval and c1, c2 are learning factors. Typically c1 = c2 = 1.4.

e. Update the particles position

$$present[] = present[] + v[] \tag{2}$$

f. Terminate if the condition (depending on application) is met else go to step 2

3. Experimental Results

Two variants of PSO are considered on the problems of varying difficulty from simple to hard. One variant uses only inertia weight which is denoted as PSO-IW, and second utilizes both inertia weight as well as constriction factor and denoted as PSO-IC. The parameter settings for the PSO implementation are as follows-

- The swarm size or number of particles is 20.
- The Search domain: $1 \leq xi \leq 20$, i = 1, 2. . . . n that is, the particle cannot move out of this range in each dimension. Here two dimensions are considered.

- The parameters r1 and r2 are uniformly distributed in the range [0, 1].
- Vmax = 5
- The Acceleration Coefficients $c1 = c2 = 1.4$
- The stop criterion used is the maximum number of iterations allowed is 1000 or the generated input values remains same for 10 consecutive iteration
- 10 runs were performed for each variant

PSO-IW

The term an inertia weight in the particle swarm optimization algorithm was first reported in the literature in 1998 [10, 11]. The motivation was to be able to eliminate the need for Vmax. The use of the inertia weight w has provided improved performance. As initially developed, w often is decreased linearly from about 0.9 to 0.4 during a run. But it was found that the maximum velocity factor that is Vmax couldn't always be eliminated.

- The inertia weight(w) is decreased gradually for all iterations in between 0.9 and 0.4

PSO-IC

An adaptive PSO model [9] proposed by Clerc and Kennedy uses a parameter 'χ' called the constriction factor but also excluded the inertia weight w and the maximum velocity parameter Vmax. The constriction factor χ controls on the magnitude of the velocities χ results in the quick convergence of the particles over time. The constriction coefficient method balances the need for local and global search depending on what social conditions are in place.

- Constriction Factor = 0.79
- The inertia weight(w) is decreased gradually for all iterations in between 0.9 and 0.4

$$vi[] = \chi * [w* vi[] + c1 * rand() * (pbest[] - xi[]) + c2 * rand() * (lbest[] - xi [])] \quad (5)$$

First three steps that are discussed in manual guidelines are done manually. After that the particles are generated randomly for PSO. With constraint equation the particles are restricted.

Test problem 1

A child's rectangular play ground is to be built next to the house. To make the three sides of the playpen, twenty-four feet of fencing are available. What should be the dimensions of the sides to make a maximum area?

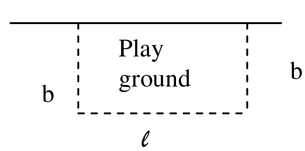
We will follow the steps discussed in section I

a. Preparation

Play yard is rectangular with 3 sided fencing, 48 feet of fencing is available and area is as large as possible

In words, what is to be found is Dimensions of play area for maximum area

b. Translation –



Other variables required are –

A = total area of play ground = ℓ *b (both in sq. feet)

L = total length of fence = 48 (feet)

c. Optimization formulation

(a) Design variables: ℓ, b

(b) Optimization equation/function: Area = A = ℓ * b

(c) Constraint function: $2*b + \ell = 48$ or $y = 48 - 2*b$

d. Apply

Manual Method:

b (Breadth)	ℓ (Length)	A(Area)
2	44	88
4	40	160
6	36	216
8	32	256
10	28	280
12	24	288
14	20	280
16	16	256
18	12	216
20	8	160

Table 1. A table 1 shows the manual calculation of area for b and ℓ values

b (Breadth)	**ℓ (Length)**	**A(Area)**
10	28	280
11	26	286
12	24	288
13	22	286
14	20	280

Table 2. We can observe that the maximum is between b = 10 and b = 14, so we try some more values between 10 and 14 as shown in table 2

Construct a graph of the criterion function, A(area), and the (now) single design variable, b. The symmetry around b = 12 leads us to suspect that that the optimum is at b = 12, where A = 288 ft. The figure 1 shows the 3D graph of breadth VS length VS area.

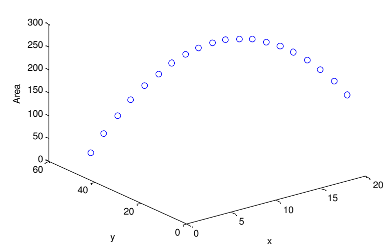


Figure 1. breadth VS length VS area

By PSO

The manual calculation can be carried out either for each input value. This can also be done as discussed above first with some step values and then figuring out to exact value. This trial and error depends on the problem definition. To avoid this, experiments are carried out using PSO. The initial input values are generated randomly. Table 3 shows the random generated sequence of length and breadth. Figure 2 shows the graph plotted for breadth VS length VS area for this random generated sequence. 10 runs are performed and almost after 20 iterations all the particles have input values b=12, l=24. The output of last iteration is shown in figure 3.

breadth	length	breadth	length
16	16	5	38
16	16	4	40
17	14	9	30
2	44	3	42
9	30	20	8
3	42	12	24
2	44	2	44
12	24	15	18
4	40	17	14
13	22	18	12

Table 3. Random generated sequence of length and breadth

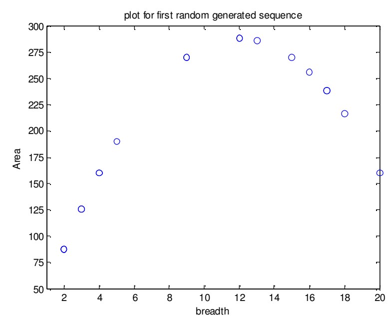


Figure 2. breadth VS length VS area

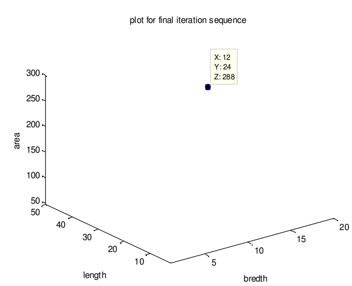


Figure 3. output of last iteration

e. Verification

We could verify that assumption with trials at b = 11.9 and 12.1. The maximum is at b = 12 and ℓ= 24 ft.

Test problem 2

An open-top cylindrical tank with a volume of ten cubic feet is to be made from a sheet of steel. Find the dimensions of the tank that will require as little material used in the tank as possible.

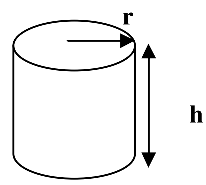
a. Preparation

A cylindrical tank is open at the top It is made from as little sheet material as possible. Volume is 12 ft^3 In words, what is to be found is Dimension of the tank that minimizes the sheet material needed?

b. Translation

r = radius of tank (ft)

h = height of tank (ft)



Other variables required are

Volume = V ft^3 = $\prod r^2$ h

Surface Area = A ft^2 = $\prod r^2 + 2 \prod r$ h

c. Optimization problem Formulation

(a) Design variables: r, h

(b) Optimization equation/function: A = $\prod r^2 + 2 \prod r$ h

(c) Constraint function: V = $\prod r^2$ h = 12 Or h = 12 / $\prod r^2$

d. Apply

Now we have the optimization problem in mathematical form. In words, we have to find the values of r and h that give a tank of minimum surface area with a volume of 10 ft^3

Manual Method:

We will proceed by systematic numerical search. Table 4 shows manual calculation for radius value from 1 to 4

Trial	R Radius	h(Height)=12/ $\prod r^2$	A(Area)= $\prod r^2$ + 2 $\prod r$ h
1	1	3.821656	Decrement 27.14
2	2	0.955414	Increment 24.56
3	3	0.424628	36.26
4	4	0.238854	56.24

Table 4. Manual calculation for radius value from 1 to 4

After the second trial, it is apparent that the minimum A is between r=1 and r=2. Table 5 shows manual calculation for radius value from 1 to 2

Trial	r(Radius)	h(Height)=12 / $\prod r^2$	A(Area)= $\prod r^2$ + 2 $\prod r$ h
1	1.1	3.158393	25.61
2	1.2	2.653928	24.52
3	1.3	2.261335	23.76
4	1.4	1.949825	23.29
5	1.5	1.698514	Decrement 23.06
6	1.6	1.492834	Increment 23.03
7	1.7	1.322372	23.19

Table 5. Manual calculation for radius value from 1 to 2

It is apparent that the minimum A is between r=1.5 and r=1.6 as between r=2 and r =3 area increases, anual calculation carried out for radius value from 1.5 to 1.6 are shown in table 6.

Trial	r(Radius)	h =12 / Πr²	A = Πr² + 2 Πr h
1	1.5	1.698514	23.07
2	1.51	1.676091	23.05
3	1.52	1.65411	23.04
4	1.53	1.632558	23.04
5	1.54	1.611425	23.03
6	1.55	1.5907	23.03
7	1.56	1.570371	23.03
8	1.57	1.55043	23.03
9	1.58	1.530867	23.03
10	1.59	1.511671	23.03
11	1.6	1.492834	23.04

Table 6. Calculation for radius value

From table 6 the optimum appears to be 23.03 ft2 and it is found at a value of r approximately equal to 1.54 to 1.59 ft. To get more accuracy, we could refine the search.

By PSO –

As per manual method systematic search is carried out which may become tedious depending on the problem and accuracy required. The Particles are generated randomly for PSO. Table 7 shows the random generated sequence of length and breadth and its corresponding area.

This random generated sequence is then used and using velocity, inertia weight and constriction factor the value of each particle is calculated in iteration. The particles positions are updated accordingly. As per PSO all the particles should converge towards the global optimum.

Figure 4 shows the graph plotted for breadth VS length VS area for initial random generated values. For 10 runs the mean iterations required are 75 to get all the particles at one position.

r(Radius)	h(Height	A(Area)
1.773659	.214818	23.40937
1.485955	1.730774	23.08455
1.827298	1.144545	23.61866
1.992032	0.963072	24.50812
1.41381	1.911919	23.25182
1.517088	1.660466	23.04667
1.424061	1.884491	23.22097
1.471648	1.76459	23.1087
1.552137	1.586323	23.02722
1.926645	1.029552	24.11244
1.835253	1.134644	23.65322
1.864767	1.099013	23.78914
1.47818	1.749028	23.09713
1.391932	1.972492	23.32589
1.873475	1.088819	23.83153
1.819564	1.154296	23.58592
1.775166	1.212757	23.41467
1.790334	1.192294	23.46995
1.683223	1.348865	23.15473
1.771235	1.218146	23.40091

Table 7. Rrandom generated sequence of length and breadth

After 75 iteration the output of last iteration is shown in figure 5.

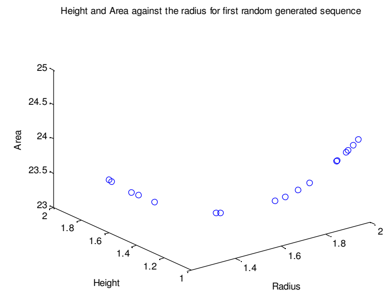


Figure 4. the output of first iteration

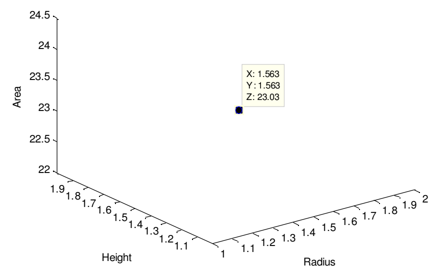


Figure 5. The output after 75 iteration

Test problem 3

Find the values of x and y (both > 0) that maximize

$f = - x^2 + 10 x + xy - y^2 + 8y + 2$

a. Preparation

Find the values of x and y (both > 0) that maximize

$f = - x^2 + 10 x + xy - y^2 + 8y + 2$

Note the two design variables x and y also there are no constraints that can be used to eliminate one of them.

b. Translation

x, y are variables f is a function of two variables

c. Optimization problem:

(a) Design variables: x, y

(b) Optimization equation/function:

$f = - x^2 + 10 x + xy - y^2 + 8y + 2$

(c) Constraint: x, y positive

d. Apply

Manually this is solved using calculus.

Solving the above two equations for x and y gives:

$\frac{\partial f}{\partial x} = -2x + 10 + y$ $\qquad$ $-2x + y = -10$

$\frac{\partial f}{\partial y} = x - 2y + 8$ x–2y = -8(multiply by 2)

-2x + y = -10

+2x-4y = -16

-3y = -26

y = 26/3 = 8.667

x – 2 * 26/3 = -8

x = -8+52/3

x = 9.333

With these values as x = 9.333 and y = 8.667, f = 83.33.

By PSO

The manual calculations are using first derivative. Then the values of each variable are derived by solving the two equations. Using PSO does not require finding the derivative. Initial population of particles is generated randomly for PSO. Table 8 shows the random generated sequence of x and y. By updating the velocities and particle position the function value is calculated.

x	Y	X	y
8.285331	7.163474	3.704786	4.071289
7.091644	9.978887	9.059636	4.63793
2.256647	6.380547	5.284849	8.953692
8.332036	9.805734	9.823386	9.663485
9.701875	4.188764	3.180595	9.961247
5.469749	9.803283	4.504434	8.875267
4.873988	7.128654	1.017486	9.208141
8.842883	7.55882	2.389808	4.409015
4.870914	4.622215	6.994766	9.238846
7.393375	0.081196	4.983816	5.767114

Table 8. Random generated sequence of x and y

Algorithm is then used to find out the optimum value. The 10 runs are taken. Depending on the input generated values the algorithm is terminated. For PSO-IW the number iteration required are 1000 while PSO-IC requires only 500 iterations. These are the mean values. Figure 6 shows the graph plotted for breadth VS length VS area. After 1000 iteration the output of last iteration is shown in figure.

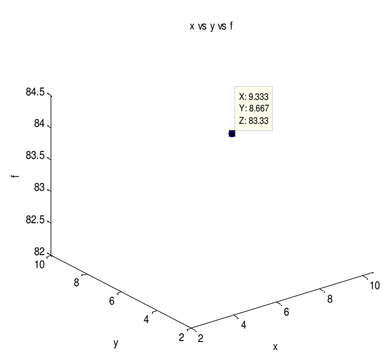


Figure 6. breadth VS length VS area output of last iteration

The experiments are carried out for test problems.

Test Problem	Method	MeanSolution	Std. Deviation
1	PSO-IW	72	0.0
	PSO-IC	72	0.0
2	PSO-IW	23.026	0.0
	PSO-IC	23.026	0.0
3	PSO-IW	83.31	0.02
	PSO-IC	83.33333	0.0

Table 9. Results obtained for each test problem

Table 9 shows the results obtained for each test problem. The number of iterations required for each test problem differs. The Mean output value for each test problem is calculated for each variation of PSO. The PSO-IW and PSO-IC performance is same for first two test problems. PSO-IC is better for third problem.

4. Conclusions

The capability of PSO method to address the maximization and minimization unconstrained problems is investigated through the performance of numerous experiments on different test problems. Results obtained are reported. The two variants used exhibits similar performance for first two test problems while PSO-IC gives better performance for third test problem. For PSO-IW the number iteration required are 1000 while PSO-IC requires only 500 iterations. The constriction factor χ controls on the magnitude of the velocities χ results in the quick convergence of the particles over time.

5. Acknowledgment

This work is supported by Board of College and University Development, University of Pune under research grant.

References

[1] Parsopoulos, K.E., Plagianakos, V.P., Magoulas, G.D., Vrahatis, M.N. (2001). Objective function "stretching" to alleviate convergence to local minima, Nonlinear Analysis TMA 47, 3419-3424.

[2] Parsopoulos, K.E., Plagianakos, V.P., Magoulas, G.D., Vrahatis, M.N (2001). Stretching technique for obtaining global minimizers through Particle Swarm Optimization, *In*: Proceedings of the PSO Workshop, Indianapolis, USA, 22-29.

[3] Parsopoulos, K.E., Vrahatis, M.N (2001). Modification of the Particle Swarm Optimizer for locating all the global minima, Artificial Neural Networks and Genetic Algorithms, V. Kurkova et al. (Eds.), Springer, 324-327.

[4] Kennedy, J., Eberhart, R. C. (1995). Particle Swarm Optimization, Proc. IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV: 1942-1948.

[5] Angeline, P. J. (1998). Using Selection to Improve Particle Swarm Optimization, IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9.

[6] Shi, Y., Eberhart RC (1998). Parameter Selection in Particle Swarm Optimization, Evolutionary Programming VII, Springer, Lecture Notes in Computer Science 1447, 591-600.

[7] Eberhart RC and Shi Y (2000) Comparing inertia weights and constriction factorsin particle swarm optimization, Proceedings of IEEE International Congress onEvolutionary Computation, vol. 1, 84-88.16.

[8] Van den Bergh F, Engelbrecht PA (2001) Effects of swarm size on cooperative particle swarm optimizers, In Proceedings of GECCO-2001, San Francisco, CA,892-899.

[8] Angeline, P. (1998). Evolutionary Optimization versus Particle Swarm Optimization: Philosophy and Performance Difference, *In:* Proceedings of the 7th Annual Conference on Evolutionary Programming, San Diego, USA.

[9] Clerc, M., Kennedy J (2002). The Particle Swarm -Explosion, Stability, and Convergence in a Multidimensional Complex Space, IEEE Transactions onEvolutionary Computation, 6 (1) 58-73.

[10] Hackwood, S., Wang, J. (1988). The engineering of cellular robotic systems. *In*: IEEE International Symposium on Intelligent Control, p. 70-75.

[11] Eberhart, R. C., Shi, Y. (2001). Tracking and optimizing dynamic systems with particle swarms. *In*: Proceedings of the 2001 Congress on Evolutionary Computation.