Parameters determining Optimal Solution of Particle Swarm Optimization (PSO) algorithm - Role of Population size and number of Iterations

Abstract - Particle Swarm Optimization is an optimization algorithm that finds the convergence point at some particular position in the entire swarm. Some of the parameters on which the optimal solution depends are studied. Out of them some parameters are made to be static and some of them are varied to study their effect on the optimal solutions found. The effect on the computation time is also checked to see the variations by altering number of particles of the swarm i.e. population size and maximum number of iterations. This paper presents the simulation results of particle swarm optimization algorithm with varying the number of particles and number of iterations.

Keywords - Particles, optimal solution, convergence, gbest value, parameters, population, iteration.

1. Introduction
Particle Swarm Optimization simulates the behavior of bird flocks when they scatter for searching food [1]. It was originally proposed in 1995 [2]. Particle Swarm Optimization is an optimization technique which is used to find the optimal solutions for any given problem. It could find the maximum or the minimum value from the many available feasible solutions. The algorithm is based on the number of particles, which denotes the many available candidate solutions that are feasible for the result of the problem. These particles are collectively known as the swarm. The particles fly [3] in their surroundings, within the swarm and tries to find the best solution by visiting the different positions in the swarm. When a particle finds the best solution, it lets all other particles accumulate at that position, by exchanging information with them and thus, these particles converge at that point. The PSO algorithm has a fast converging feature.

2. PSO Algorithm
The real-world parts are listed up to get an abstracted version. The world is taken to be the solution space. It consists of all the candidate solutions. For instance, for solving some equation, the solution space is an n-dimensional vector space where the variable n equals the number of unknowns in the equation to optimize. In effect, the animals are abstracted into software agents which reside at a position in this n-dimensional space, having a velocity and a fitness at their position regarding the problem.

\[
\begin{align*}
\text{First part represents momentum term} \\
\text{Second represents cognitive component that is responsible for attraction of particles on current positive direction of pbest.} \\
\text{Third part represents social component that is responsible for attraction of particles at current position towards positive direction of global best.}
\end{align*}
\]

Main steps for PSO algorithm is as follows:
1. Initialize number of particles with random position and velocity.
2. Evaluate the fitness value for each particle.
3. Evaluate gbest.
4. Evaluate pbest.
5. Update velocity & position.
6. Evaluate the fitness value for new position.
7. If condition is fulfilled gbest is the solution else repeat above steps [5].

**A. Pseudo code of original PSO**

```
Initialize the population randomly
While (Population Size)
    Loop
        Calculate the fitness
        If the fitness value is better from the best fitness value (pbest) found so far then,
            Update the pbest with new pbest
        End loop
        Select particle with the best fitness value from all particles as the gbest
    While maximum iterations or the minimum error criteria is not attained
        For each particle
            Calculate the particle velocity
            Update particle position according to the equation
        Next
    }

The following criterion may be used for termination of the PSO algorithms.
(a) PSO algorithm is executed for a fixed number of iterations.
(b) PSO algorithm can be terminated if the velocity changes are close to zero for all particles, in which case there will be no further changes in particle positions.

3. **Parameters of Particle Swarm Optimization algorithm**

   Standard PSO algorithm is influenced by some system parameters. These parameters include:
   (a) dimension of the problem- The problem is mostly fixed to be 2 dimensional.
   (b) number of particles in each iteration- The number of particles considered for every iteration.
   (c) upper limit on the maximum velocity- The maximum velocity value set for a problem.
   (d) Number of iterations
   (e) Inertia weights[6]

   Here, two of the parameter variations is studied i.e. Number of particles in each iteration and maximum number of iterations. The other parameters are taken to be fixed i.e. upper limit of velocity is set to be 10, dimension of problem be 2-dimensional, inertia weight is taken to be 0.73.

4. **Related Work**

   *Yuhui Shi and Russell Eberhart, “A Modified Particle Swarm Optimizer”*- A new parameter, called inertia weight was introduced into the original particle swarm optimizer to enhance its performance. Simulations were done to present the effective impact of this new parameter (inertia weight) on the particle swarm optimizer.
Yamille del Valle, Ganesh Kumar Venayagamoorthy, Salman Mohagheghi, Jean-Carlos Hernandez, and Ronald G. Harley, “Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems”. It presents some basic concepts of PSO and some of its variants. A survey of power system applications including the discussions on solution representation and most efficient fitness functions for PSO were presented.

Muhammad Imran, Rathiah Hashima and Noor Elaiza Abd Khalid “An Overview of Particle Swarm Optimization Variants” Particle swarm optimization (PSO) is a stochastic algorithm used for the optimization problems proposed by Kennedy. It is a very good technique for the optimization problems. But still there is a drawback in the PSO is that it stuck in the local minima. To improve the performance of PSO, the researchers proposed the different variants of PSO. Some researchers try to improve it by improving initialization of the swarm. Some of them introduce the new parameters like constriction coefficient and inertia weight. Some researchers define the different method of inertia weight to improve the performance of PSO. Some researchers work on the global and local best particles by introducing the mutation operators in the PSO. In this paper, we will see the different variants of PSO with respect to initialization, inertia weight and mutation operators.

Prabha Umapathy, C. Venkataseshaiah, and M. Senthil Arumugam, “Particle Swarm Optimization with Various Inertia Weight Variants for Optimal Power Flow Solution” Particle swarm optimization is a heuristic global optimization method and also an optimization algorithm, which is based on swarm intelligence. It comes from the research on the bird and fish flock movement behavior. The algorithm is widely used and rapidly developed for its easy implementation and few particles required to be tuned. It presents some kinds of improved versions of PSO and research situation, and the future research issues are also given.

Shailendra S. Aote, Dr M M Raghuwanshi, Dr. Latesh Malik “A Brief Review on Particle Swarm Optimization: Limitations & Future Directions” - Various efforts have been made for solving the unimodal and multimodal problems as well as two dimensional to multidimensional problems. Efforts were made towards topology of the communication, initial distribution of particles, parameter adjustment and the efficient problem solving capabilities. In this paper, a detailed study of PSO and limitation in present work was presented and based on the limitations, future directions were proposed.

Daniel Bratton, James Kennedy “Defining a Standard for Particle Swarm Optimization” - Though many researchers have worked in the field of pso, yet there is no standard definition representing the modern implementations of the technique of Particle swarm optimization. A standard is defined here which is an extension of the original algorithm to improve performance on standard measures.

5. Experimental Setup and Results
The Particle Swarm Optimization (PSO) algorithm finds the best solution among the whole swarm. PSO algorithm is implemented on MATLAB (R2011b). The best position found is known as the global best i.e. gbest position and the resultant value calculated for a problem is called its global best solution i.e. gbest value. The parameters for their implementation are set in advance.

A. Parameter Setting for PSO
Maximum number of iterations of process- 20
Cognitive factor c1- 2.05
Social Factor c2- 2.05
Peak velocity-10 which means maximum velocity is 10 and minimum velocity is -10
Peak position=100 which means maximum position is 100 and minimum position is -100
Random variables r1 and r2 are found within range[0,1].
Dimensions=2
Inertia weight=0.73
Iterations = the number of times the process is run
The error between the actual value and the desired value is found in order to compare the gbest values.
Error= difference between actual value and the desired value
The PSO algorithm has been run using Weirstrass test function to see the effect of increasing population size on the values of gbest and Time taken for computation of solution.

B. Varying the number of particles in the swarm
The population of the swarm influences the speed of finding the optimal solution, more the number of particles taken into the swarm, larger the set of solutions from which optimal solution is to be found, more time will be taken for the optimization method to find the best solution. The results of varying the population of the swarm can be seen by comparing the gbest value( global best value) for a set of number of population.

C. Results of increasing population size for gbest values and its time of computation
The value of gbest varies when the number of particles are varied. The results of varying the population are simulated using the weirstrass test function for PSO. The effect of varying the population size of the swarm can be seen as below:

Figure- The graph plots for Error and Iteration for the Weirstrass test function used for PSO, showing the error plots for 10 particles in the swarm and 20 iterations for the process.
Figure - The graph plots for Error and Iteration for the Weirstrass test function used for PSO, showing the error plots for 20 particles in the swarm and 20 iterations for the process.

Figure - The graph plots for Error and Iteration for the Weirstrass test function used for PSO, showing the error plots for 30 particles in the swarm and 20 iterations for the process.

Figure - The graph plots for Error and Iteration for the Weirstrass test function used for PSO, showing the error plots for 40 particles in the swarm and 20 iterations for the process.
The graph plots for Error and Iteration for the Weirstrass test function used for PSO, showing the error plots for 50 particles in the swarm and 20 iterations for the process.

<table>
<thead>
<tr>
<th>Population</th>
<th>Gbest</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2.2318</td>
<td>0.0170</td>
</tr>
<tr>
<td>20</td>
<td>0.0152</td>
<td>0.0192</td>
</tr>
<tr>
<td>30</td>
<td>0.0540</td>
<td>0.0276</td>
</tr>
<tr>
<td>40</td>
<td>0.0482</td>
<td>0.0517</td>
</tr>
<tr>
<td>50</td>
<td>0.0507</td>
<td>0.0570</td>
</tr>
</tbody>
</table>

Table- gbest values and time of computation for gbest solution for the weirstrass test function in PSO algorithm

**Discussion of Results**- The gbest values and the time taken for finding the gbest values for the PSO algorithm has been tabulated. It can be clearly seen that the time taken for the computation of result of best solution i.e. gbest value increases with the increase in population size i.e. the number of particles taken into the swarm. It is so because as the number of particles increase, (which depict the number of candidate solutions for a given problem ), the time for comparing results for every particle increases, leading to increase in time for computation of final result of global best value.

**Conclusion**- With the increase in number of particles, the time of computation for calculating the global best solution increases.

**D. Varying the number of iterations for the PSO algorithm**

The algorithm is run for a predefined number of turns, called iterations. The effect of varying number of iterations can be seen as below:

The Population size is fixed to be 50 and iterations are varied.
Figure - The graph plots for Error and Iteration for the Weirstrass test function used for PSO, showing the error plots for 50 particles in the swarm and 10 iterations for the process.

Figure - The graph plots for Error and Iteration for the Weirstrass test function used for PSO, showing the error plots for 50 particles in the swarm and 20 iterations for the process.

Figure - The graph plots for Error and Iteration for the Weirstrass test function used for PSO, showing the error plots for 50 particles in the swarm and 30 iterations for the process.
Figure- The graph plots for Error and Iteration for the Weirstrass test function used for PSO, showing the error plots for 50 particles in the swarm and 40 iterations for the process.

Figure- The graph plots for Error and Iteration for the Weirstrass test function used for PSO, showing the error plots for 50 particles in the swarm and 50 iterations for the process.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Gbest</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0336</td>
<td>0.0331</td>
</tr>
<tr>
<td>20</td>
<td>0.0759</td>
<td>0.0615</td>
</tr>
<tr>
<td>30</td>
<td>3.1215</td>
<td>0.0785</td>
</tr>
<tr>
<td>40</td>
<td>0.0122</td>
<td>0.0988</td>
</tr>
<tr>
<td>50</td>
<td>3.0987</td>
<td>0.1197</td>
</tr>
</tbody>
</table>

Table- gbest values and time of computation for different iterations

**Discussion of Results**- The error plots for different iterations is shown and the values of global best positions, time for computation is tabulated. It can be seen that the computation time for increased number of iterations increases as the loop for finding solutions is run for increased number of times.

**Conclusion**- With the increase in number of iterations, the computation time for calculating the gbest value increases.
6. **Conclusion**

   The global best position is found at the point where all the particles of the swarm converge and this point is said to be the convergence point. The time taken for the computation of the global solution is known as time of computation. The effect of population size of the swarm and the number of iterations for the process of particle swarm optimization algorithm is studied. The time of computation for the global best (gbest) value increases with the increase in number of particles i.e. population size and increase in the number of iterations taken for the process.

7. **References**


---

**Authors**

Er. Avneet Kaur