Volume: 10 Issue: V Month of publication: May 2022
DOI: https://doi.org/10.22214/ijraset.2022.43114

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

Comparative Analysis of PSO Algorithm in Cloud Computing

Palak Khanna[1], Mayank Agarwal[2], Khushi Rastogi[3]
[1, 2, 3]Meerut Institute of Engineering and Technology, Meerut

Abstract: Cloud computing makes it possible to access applications and data from anywhere so this has become new technology. The goals of the paper are to provide additional insights to suggest ways in which performance might be improved by incorporating features from one paradigm into the other. The Reasearch Paper focus on particle swarm optimization (PSO) heuristic-based algorithm based on Scheduling.
By scheduling, applications are scheduled to the cloud resources that are used for computation of cost and transmission cost of data in the cloud. by using PSO total cost of execution is minimized. Since the inception of Inertia Weight in PSO, a large number of variations of the Inertia Weight strategy have been proposed. In order to propose one or more than one Inertia Weight strategies that are efficient than others, this paper studies popular Inertia Weight strategies and compares their performance on optimization test problems.
Keywords: Particle Swarm Optimization, Inertia weight, Population, Cognitive component, Optimization Function.

I. INTRODUCTION

Particle swarm optimization (PSO) algorithm is a stochastic optimization technique based on the swarm, which was proposed by Eberhart and Kennedy (1995) and Kennedy and Eberhart (1995). PSO algorithm simulates animals' social behavior, including insects, herds, birds, and fishes.
These swarms conform to a cooperative way to find food, and each member in the swarms keeps changing the search pattern according to the learning experiences of its own and other members. The main design idea of the PSO algorithm is closely related to two types of research: One is an evolutionary algorithm, just like an evolutionary algorithm; PSO also uses a swarm mode which makes it simultaneously search large regions in the solution space of the optimized objective function. The other is artificial life, namely, it studies the artificial systems with life characteristics. It is different from other optimization algorithms in such a way that only the objective function is needed and it is not dependent on the gradient or any differential form of the objective. It also has very few hyperparameters.

A. Different Inertia Weight Strategies For Particle Swarm Optimization

Inertia Weight plays a key role in the process of providing a balance between exploration and exploitation process. The Inertia Weight determines the contribution rate of a particle's previous velocity to its velocity at the current time step. The basic PSO, presented by Eberhart and Kennedy in 1995, has no Inertia Weight. In 1998, first time Shi and Eberhart presented the concept of Inertia Weight by introducing Constant Inertia Weight. They stated that a large Inertia Weight facilitates a global search while a small Inertia Weight facilitates a local search. Further, dynamical adjusting of Inertia Weight was introduced by many researchers which can increase the capabilities of PSO. A review of Inertia Weight strategies in PSO is given chronologically in subsequent paragraphs. Eberhart and Shi proposed a Random Inertia Weight strategy and experimentally found that this strategy increases the convergence of PSO in early iterations of the algorithm. The Linearly Decreasing strategy [6] enhances the efficiency and performance of PSO. It is found experimentally that Inertia Weight from **0.9 to 0.4** provides excellent results. In spite of its ability to converge optimum, it gets into the local optimum solving the question of more apices function.

B. Parameter Setting

Swarm size is taken to be 50. The number of decision variables is fixed to be 10 for each experiment. The termination criterion is set to the "no improvement observed for 200 iterations (similar fitness value achieved for 200 consecutive iterations)". For those which require a maximum number of iterations, 1000 iterations are used.

Table 3.2.1. Different Inertia weights used in this paper.

Sr. No.	Name of Inertia Weight	Formula of Inertia Weight

1	Constant Inertia Weight	w = c
		c = 0.7(Considered for experiments)

2	Random Inertia Weight	w = 0.5 + rand()/2

3	Chaotic Inertia Weight	z = rand()
		z = 4*z*(1-z)
		w = (0.9-0.4)*((MaxIt - it)/MaxIt) + 0.4*z

4	Chaotic Random Inertia Weight	z = rand()
		z = 4*z*(1-z)
		w = 0.5 * rand()+0.5*z

Table 3.2.2. Different fitness functions.

Sr. No.	Name of Function	Objective Function	Search space
1	Sphere	$Min\ f(x) = \sum_{i=1}^{n} \blacksquare x_i^2$	$-5.12 <= x_i <= 5.12$
2	Rosen Brock	$Min\ f(x) = \sum_{i=1}^{n-1} \blacksquare \left[100\ (x_{i+1}^{\blacksquare} - x_i^2)2 + (x_i^{\blacksquare} - 1)2\right]$	$-5 <= x_i <= 10$
3	Ackley	$f(\mathbf{x}) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp($	$-30 <= x_i <= 30$

II. PROPOSED WORK

A. Approach

1) Setting Population

In PSO the higher the swarm size, the more scattered the search performed by the algorithm. With a higher population size each generation takes more function calls, and a larger part of the search space may be visited. However, so far there is no detailed study on the proper choice of PSO swarm size, although it is widely known that population size crucially affects the performance of metaheuristics. In most applications, authors follow the initial suggestion from 1995 and restrict the population size to 20–50 particles. In our Research work, we have set swarm size / population as 50 for better efficiency.

2) Setting Acceleration Coefficient

Acceleration coefficients controlled the impact of the particle's own experiences and the other particles' experiences on the trajectory of each particle. The setting of acceleration played a key role in the performance of particle swarm optimization. To efficiently control the local search and convergence to the global optimum solution, a good investigation to the key role of the setting of acceleration coefficients was made. In our Research work, we have set acceleration coefficients c1 and c2 as 0.9 and 0.4 respectively for better efficiency.

3) Setting Inertia Weight

Inertia Weight plays a key role in the process of providing balance between exploration and exploitation process. The Inertia Weight determines the contribution rate of a particle's previous velocity to its velocity at the current time step.

We have made research on the basis of 5 inertia weights as follows :

a) Constant Inertia Weight

The conventional PSO algorithm initially used a constant value for the inertia weight. We have set the value of constant Inertia Weight as 0.7 for our research work.

$$\mathbf{W = 0.7}$$

b) Random Inertia Weight

Random inertia weight improves algorithm's global optimization performance and an adaptive re-initialize mechanism is used when the global best particle is detected to be trapped. We have set the value of Random Inertia Weight as follows

$$W = 0.5 + \text{rand}()/2$$

c) Chaotic Inertia Weight

Chaotic Inertia Weight has been proposed by Feng et al. [7]. Comparison between CRIWPSO and RIW PSO has been done and found that CRIW PSO performs excellently. It has a rough search stage and minute search stage alternately in all its evolutionary processes.

$$z = \text{rand}()$$
$$z = 4*z*(1-z)$$
$$W = (0.9-0.4)*((MaxIt - it)/MaxIt) + 0.4*z$$

d) Chaotic Random Inertia Weight

We have set the value of Chaotic Random Inertia Weight as follows :

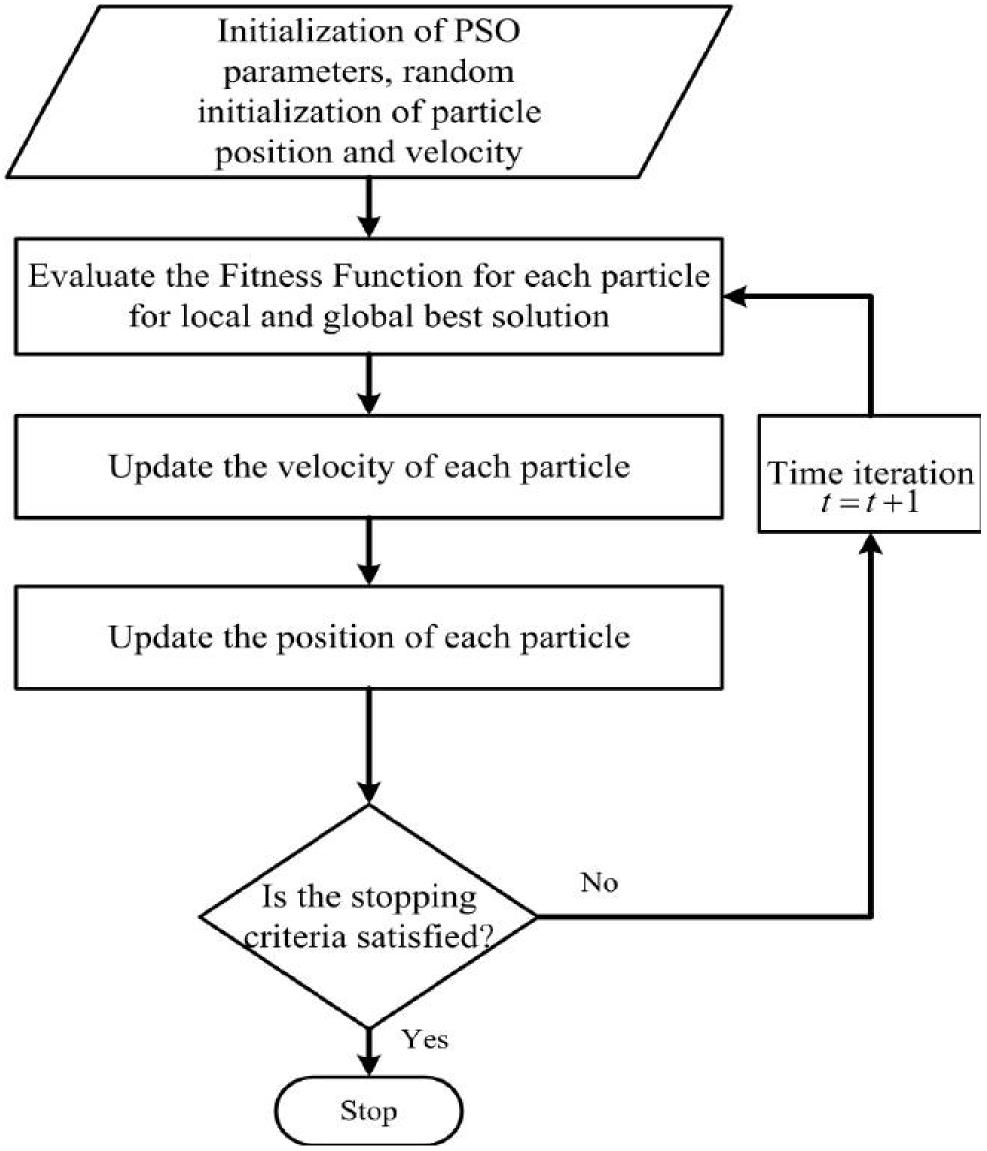
$$z = \text{rand}()$$
$$z = 4*z*(1-z)$$
$$W = 0.5 * \text{rand}()+0.5*z$$

4) *Setting Iterations:* Particle swarm optimization (PSO) is an iterative algorithm, where particle positions and best positions are updated per iteration. The order in which particle positions and best positions are updated is referred to in this paper as an iteration strategy. We have taken four iterations for analyzing better results whose values are 100, 500, 1000 and 5000.

B. Algorithm

1) Create a 'population' of agents (particles) uniformly distributed over X.
2) Evaluate each particle's position according to the objective function.
3) If a particle's current position is better than its previous best position, update it.
4) Determine the best particle (according to the particle's previous best positions).
5) Update particles' velocities.
6) Move particles to their new positions.
7) Go to step 2 until stopping criteria are satisfied.

III. FLOWCHART


Initialization of PSO parameters, random initialization of particle position and velocity
Evaluate the Fitness Function for each particle for local and global best solution
Update the velocity of each particle
Update the position of each particle
Is the stopping criteria satisfied?
No
Yes
Stop
Time iteration $t=t+1$

IV. RESULTS

A. *For Sphere Function*

Inertia Weight	For 100 iterations	For 500 iterations	For 1000 iterations	For 5000 iterations
Constant	0.0045952	4.2053 e^{-08}	1.7964 e^{-14}	2.1288 e^{-62}
Random	0.031662	0.0001926	2.902 e^{-07}	4.8814 e^{-17}
Chaotic	6.6952 e^{-06}	5.329 e^{-26}	7.7537 e^{-53}	0.655 e^{-267}
Chaotic Random	0.00052223	1.6490 e^{-16}	1.1258 e^{-34}	6.2453 e^{-159}

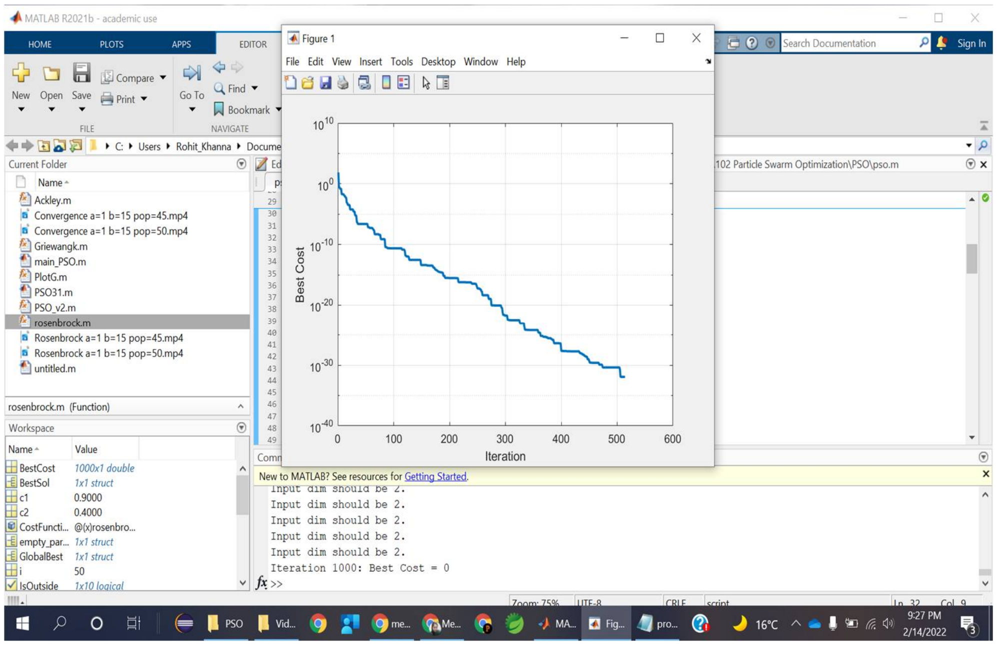
B. For Ackley Function

Inertia Weight	For 100 iterations	For 500 iterations	For 1000 iterations	For 5000 iterations
Constant	$4.26333\ e^{-05}$	$8.88178\ e^{-16}$	$1.0\ e^{-15}$	$1.0\ e^{-15}$
Random	0.0126532	$8.88178\ e^{-16}$	$8.88178\ e^{-16}$	$8.88178\ e^{-16}$
Chaotic	$6.6592\ e^{-05}$	$3.428\ e^{-12}$	$6.682\ e^{-20}$	$7.284\ e^{-26}$
Chaotic Random	$8.88178\ e^{-16}$	$8.88178\ e^{-16}$	0.000226543	$8.88178\ e^{-16}$

C. For Rosenbrock Function

Inertia Weight	For 100 iterations	For 500 iterations	For 1000 iterations	For 5000 iterations
Constant	$9.4249\ e^{-11}$	$2.2798\ e^{-28}$	0.00	0.00
Random	$3.7798\ e^{-08}$	$2.0588\ e^{-26}$	0.00	0.00
Chaotic	$6.3529\ e^{-07}$	0.00	0.00	0.00
Chaotic Random	$6.3901\ e^{-10}$	0.00	0.00	0.00

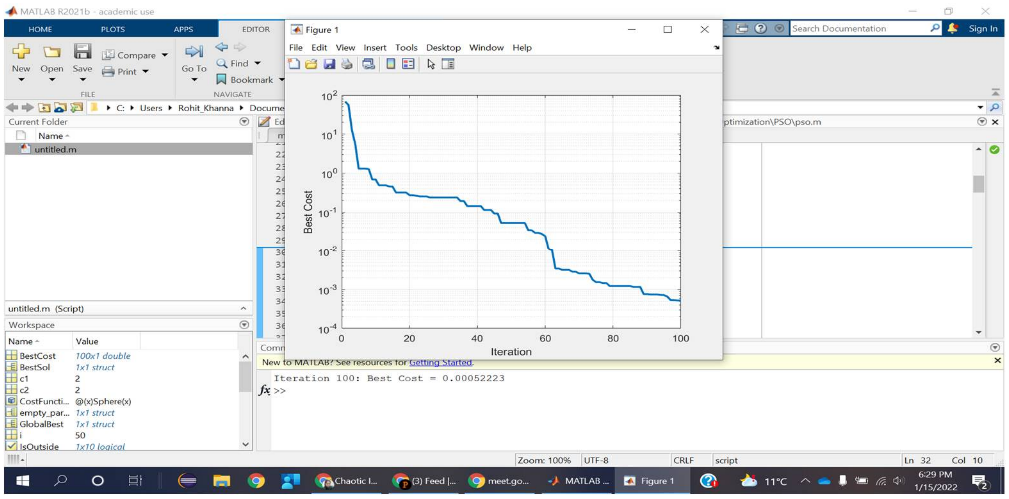
V. RESULTS IMAGES FOR FEW FITNESS FUNCTIONS IN RESEARCH


MATLAB R2021b - academic use
HOME
PLOTS
APPS
EDITOR
Figure 1
File Edit View Insert Tools Desktop Window Help
Current Folder
Name
Ackley.m
Convergence a=1 b=15 pop=45.mp4
Convergence a=1 b=15 pop=50.mp4
Griewangk.m
main_PSO.m
PlotG.m
PSO31.m
PSO_v2.m
rosenbrock.m
Rosenbrock a=1 b=15 pop=45.mp4
Rosenbrock a=1 b=15 pop=50.mp4
untitled.m
rosenbrock.m (Function)
Workspace
Best Cost
Iteration
10^{10}
10^{0}
10^{-10}
10^{-20}
10^{-30}
10^{-40}
0
100
200
300
400
500
600
102 Particle Swarm Optimization\PSO\pso.m
Search Documentation
Sign In
New to MATLAB? See resources for Getting Started.
9:27 PM
2/14/2022

Name	Value
BestCost	1000x1 double
BestSol	1x1 struct
c1	0.9000
c2	0.4000
CostFuncti...	@(x)rosenbro...
empty_par...	1x1 struct
GlobalBest	1x1 struct
i	50
IsOutside	1x10 logical

```
Input dim should be 2.
Input dim should be 2.
Input dim should be 2.
Input dim should be 2.
Input dim should be 2.
Iteration 1000: Best Cost = 0
fx >>
```

Fig 7.1 The resulting image is of Rosenbrock Function.



Name	Value
BestCost	100x1 double
BestSol	1x1 struct
c1	2
c2	2
CostFuncti...	@(x)Sphere(x)
empty_par...	1x1 struct
GlobalBest	1x1 struct
i	50
IsOutside	1x10 logical

```
Iteration 100: Best Cost = 0.00052223
>>
```

Fig 7.2 The resulting image is of Sphere function.

REFERENCES

[1] Comparative Analysis of PSO-derived Workflow Scheduling Algorithms in Cloud Computing
https://www.researchgate.net/publication/347973382_Comparative_Analysis_of_PSO-derived_Workflow_Scheduling_Algorithms_in_Cloud_Computing_based_on_QoS_Requirements

[2] A study on modified PSO algorithm in cloud computing
https://ieeexplore.ieee.org/document/8075341

[3] A Comparative Study of PSO, PSO Variants, and Random Scheduling in Solving Workflow Scheduling Problem in Cloud Computing Environment
https://link.springer.com/chapter/10.1007/978-981-16-7952-0_6

[4] Analysis of Particle Swarm Optimization and Genetic Algorithm based on Task Scheduling in Cloud Computing Environment
https://thesai.org/Downloads/Volume8No1/Paper_4-Analysis_of_Particle_Swarm_Optimization_and_Genetic_Algorithm.pdf

[5] A Review of PSO-Based Task and Workflow Scheduling Analysis in Cloud Environment
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3842717

[6] A Comprehensive Review of Swarm Optimization Algorithms
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0122827

[7] Comparative Analysis Of Pso-Derived Workflow Scheduling Algorithms In Cloud Computing Based On Qos Requirements
https://iaeme.com/MasterAdmin/Journal_uploads/IJARET/VOLUME_11_ISSUE_12/IJARET_11_12_131.pdf

[8] Comparison of global PSO for fixed and varying inertia
https://www.researchgate.net/figure/Comparison-of-global-PSO-for-fixed-and-varying-inertia_fig3_220531501

[9] Particle Swarm Optimization with Time Varying Parameters for Scheduling in Cloud Computing
https://www.matec-conferences.org/articles/matecconf/pdf/2015/09/matecconf_icame2015_06001.pdf

[10] A Uniform Initialized Particle Swarm Optimization Algorithm with Cosine Inertia Weight
https://www.hindawi.com/journals/cin/2021/8819333/