

Particle Swarm Optimization (PSO)*

1st Hansraj, 2nd Bijesh Yadav, 3rd Anjana Gupta

Department of Applied Mathematics, Delhi Technological University, Delhi-110042, India
hansul0801@gmail.com

Abstract—Particle Swarm Optimization (PSO) is a type of optimization algorithm that is based on the behavior of social animals. It represents a collection of possible solutions to an optimization problem as a swarm of particles that move through the parameter space. The movement of the particles is guided by their own performance and the performance of their neighbors, leading to an optimized solution. In this paper we are solved linear programming problems, transportation problem using Particle Swarm Optimization and applying on a Data Set.

Index Terms—Particle Swarm Optimization (PSO), Swarm Intelligence, Current Position, Current Velocity, Individual Perfect or (Pbest), Global Best (Gbest).

I. INTRODUCTION

Dr. Kennedy and Dr. Eberhart. first proposed the based on populations probabilistic search approach known as Particle Swarm Optimization (PSO) in 1995. PSO's fundamental concept was influenced by how creatures interact with one another, such as when flocks of birds or schools of fish, and it provides an alternative approach to solving non-linear optimization problems. PSO depends on a group interaction method observed in animals such as birds and insects when they search for food or migrate. The algorithm simulates the sharing of individual information among group members to identify the positive direction in a search space. If one member of the group finds a positive direction, the others will quickly follow, reflecting the behavior seen in natural social systems.

The PSO algorithm uses the concept of a swarm and particles to solve optimization problems by simulating the behavior of animals. Every particle in the population represents a potential solution and traverses the search space, starting from a random location and moving in random directions. The particles remember their best past locations and those of their neighbors, and constantly modified their location and velocity based on the best locations found by the entire population. The particles communicate and transmit favorable positions to each other. The search process continues until the swarm converges towards the maximum of the fitness function $g : \mathcal{R}^n \rightarrow \mathcal{R}$.

The PSO algorithm is growing in popularity due to its ease of implementation and ability to converge quickly on a practical solution to optimization problems. In comparison to other optimization techniques, it is quicker, less expensive, and more effective. Additionally, PSO only has a few parameters that may be changed. PSO is a great tool for solving optimization problems because of this. PSO is a good choice for problems of the non-convex, continuous, discrete, integer variable type.

II. THEORY

A. PSO with optimization problems

Optimization is the process of selecting the best solution from a set of alternatives based on one or more criteria specified by the user. This is typically done mathematically by representing the objective as a parameterized function f that depends on D parameters. The optimization problem is to determine the parameter values that maximize the objective function f . The objective function is also known as the “fitness function” and the optimization process involves finding the values that lead to the maximum of the fitness function. The focus will be on maximizing the function in the following:

$$\begin{aligned} &\text{Given } g : \mathbb{R}^D \rightarrow \mathbb{R} \\ &\text{Find } \mathbf{z}_{opt} | g(\mathbf{x}_{opt}) \geq g(z) \quad \forall z \in \mathbb{R}^D \end{aligned} \quad (1)$$

The search (or parameter) space is the D -dimensional domain of the function $\mathbb{R}^D$ and each of its points, denoted by the vector of coordinates $\mathbf{x}$ represents possible solutions to the problems, with $\mathbf{z}_{opt}$ being the best option i.e., the one that maximizes f .

In the context of optimization, Particle Swarm Optimization (PSO) was inspired by the social behavior of fish schools and bird flocks. Each particle in PSO is viewed as a point in an N -dimensional space and its position is adjusted based on its own velocity and the information from other particles in the swarm. The information used to update the particle's position includes:

- The present location of the particle
- The particle speed at the moment
- The difference between a particle's present location and its most well-known location (Pbest)
- The distance between the present location and the swarm's overall best-known position (Gbest).

B. PSO Algorithm

In PSO, Consider a population (swarm) size of N with position vector $\mathbf{X}_i^t = [x_1, x_2, x_3, \dots, x_n]^T$ where T is transpose, and velocity vector $\mathbf{V}_i^t = [v_1, v_2, v_3, \dots, v_n]^T$ at t iteration for each one of the i particle that composes it. These vectors are updated through the dimension j according to following equation:

$$\mathbf{X}_{i,j}^{t+1} = \mathbf{X}_{i,j}^t + \mathbf{V}_{i,j}^{t+1} \quad (2)$$

where v_i is the vector containing the velocity components of the i -th particle and t and $t+1$ denote two further iterations of the algorithms. The three terms that make up the velocity vectors, which control how particles move about the search space,

are as follows: the first term, defined as inertia or momentum, keeps track of the prior flow direction to stop the particle from quickly changing direction; the second term, referred to as the cognitive component, explains particles tendencies to return to their previously identified optimal locations; the last term, known as the social component, indicates a particle's tendency to migrate to the optimal location for the entire swarm (or of a local neighborhood of the particle, depending on whether a global or partial PSO is implemented). Based on these considerations, the velocity of the i -th particle is defined as:

$$V_{i,j}^{t+1} = V_{i,j}^t + c_1 \cdot r_1^t (pbest_{i,j}^t - X_{i,j}^t) + c_2 \cdot r_2^t (gbest_j^t - X_{i,j}^t) \quad (3)$$

In particle swarm optimization, the “personal best” (pbest) and “global best” (gbest) are terms used to refer to the best position a particle has achieved so far and the best position achieved by the entire swarm, respectively. The “cognitive coefficient” (c_1) and “social coefficient” (c_2) are constants that determine the size of the steps the particle takes towards its personal and global best positions. R_1 and R_2 are random matrices used to introduce a stochastic effect on the velocity update in the optimization process. These coefficient and matrices are used to update the velocity of a particle, which determines its next movement in the search space.

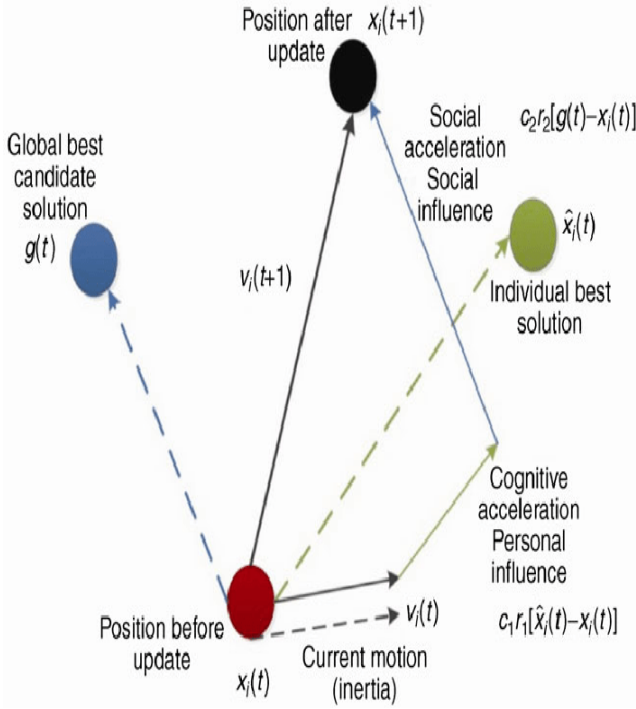


Fig. 1: Implementing the PSO

The modification of the particle's velocity can be mathematically modeled according the following equation:

$$V_{i,j}^{t+1} = \omega \cdot V_{i,j}^t + c_1 \cdot r_1^t (pbest_{i,j}^t - X_{i,j}^t) + c_2 \cdot r_2^t (gbest_j^t - X_{i,j}^t) \quad (4)$$

C. Steps of Algorithm

1) Initialization

- a) For each particle i in a swarm population size P .
 - i) Initialize X_i randomly.
 - ii) Initialize V_i randomly.
 - iii) Evaluate fitness value $f(X_i)$.
 - iv) Initialize $pbest_i$ with the help of X_i .
- b) Initialize $gbest$ with the help of X_i with the best fitness

2) Repeat until the stopping criteria satisfied.

- a) For each particle i :
 - i) Update X_i^t and V_i^t according to (2) and (4)
 - ii) Evaluate fitness $f(X_i^t)$.
 - iii) $pbest_i \leftarrow X_i^t$ if $f(pbest_i) < f(X_i^t)$
 - iv) $gbest_i \leftarrow X_i^t$ if $f(gbest_i) < f(X_i^t)$

D. Flowchart

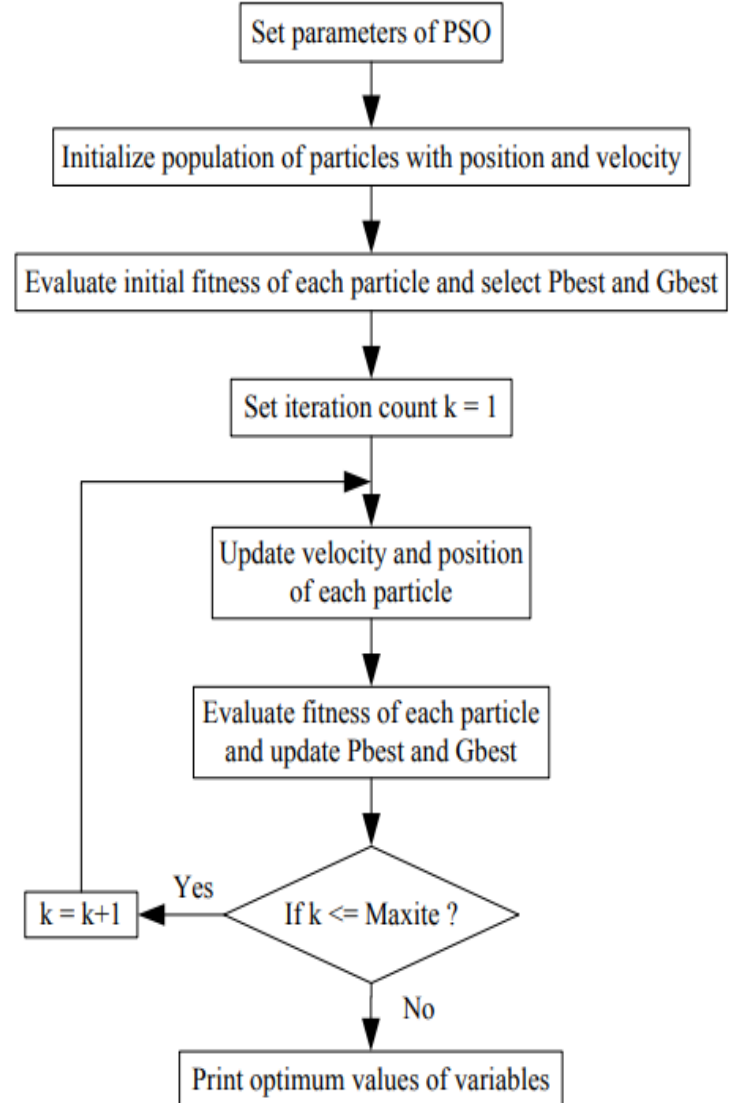


Fig. 2: Flowchart of PSO

E. Acceleration Constant C_1 and C_2

As from Eq 3, The amounts by which the particles travel in the direction of the individual and global best particle are determined by the acceleration constants c_1 and c_2 , adjusting the relative contributions of the social and cognitive aspects. A number of authors have examined how these coefficients affect the trajectory of the particles and the algorithm's convergence properties, and their findings demonstrate that as the acceleration constants are increased, the frequency of the particle's oscillation around the optimum increases while smaller values produce sinusoidal patterns. In general, it has been shown that conditions:

$$c_1 = c_2 = 2$$

F. Inertial Weight Factor

Some authors advise using a combination of $\omega_{max} = 0.9$ and $\omega_{min} = 0.4$ for the best performance. Implementations of linearly reduced inertial weight have demonstrated that it provides very excellent results in many real-world applications. Overall, Bansal et al.'s comparison of a set of common optimisation functions demonstrates that chaotic reduced inertia weights are the best fit (resulting in the lowest error mean in a set of 30 repeated simulations) while stochastic inertial weights are better if faster convergence is desired. However, the methods that result in the lowest error are linear and constant decreasing inertial weighting.

TABLE I: Inertia weight dynamic adjustment methods

Strategy	Inertia weight
Constant weight of inertia	$\omega(t) = \omega = \text{const}$
Random weight of inertia	$\omega(t) = 0.5 + \frac{r}{2} \quad r \sim (0, 1)$
Reducing inertia weight linearly	$\omega(t) = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{t_{max}} t$
Chaotic random inertia weight	$\omega(t) = 0.5r_1 + 0.5z$ $z = 4r_2(1 - r_2) \text{ with } r_1, r_2 \sim U(0, 1)$

III. LINEAR PROGRAMMING PROBLEM

$$\text{Max } Z = X_1 + 2X_2$$

Subject to:

$$X_1 + X_2 \leq 8$$

$$2X_1 + X_2 \leq 10$$

$$X_1, X_2 \geq 0$$

Parameter are set to be

- Population size = 3
- c_1 and $c_2 = 2$
- dimension of problem = 2

- r_1 and r_2 are random number between 0 and 1

The movement of particles are

$$V_{i,j}^{t+1} = \omega.V_{i,j}^t + c_1.r_1^t (pbest_{i,j}^t - X_{i,j}^t) + c_2.r_2^t (gbest_j^t - X_{i,j}^t)$$

$$X_{i,j}^{t+1} = X_{i,j}^t + V_{i,j}^{t+1}$$

TABLE II: Solution of linear programming problem

Iteration	Particle	X_1	X_2	Function Value	Pbest	Gbest
1	1	0.6	6.4	13.40	13.40	14.90
	2	0.5	7.2	14.90	14.90	
	3	1	5.6	12.20	12.20	
2	1	0.52	7.04	14.60	14.60	14.90
	2	0.5	7.2	14.90	14.90	
	3	0.6	6.88	14.36	14.36	
3	1	0.432	7.74	15.91	15.91	16.16
	2	0.5	7.2	14.90	14.90	
	3	0.16	8	16.16	16.16	
4	1	0.136	8	16.136	16.136	16.136
	2	0.228	7.8	15.82	15.82	
	3	0	8	16	16.16	
5	1	0	8	16	16	16
	2	0	8	16	16	
	3	0	8	16	16	

then,

$$|f(x)_{prev} - f(x)| = 0$$

$$|16.13 - 16| = 0$$

$$0.13 \geq 0$$

then by particle swarm optimization

$$X_1 = 0, \quad X_2 = 8, \quad Z = 16$$

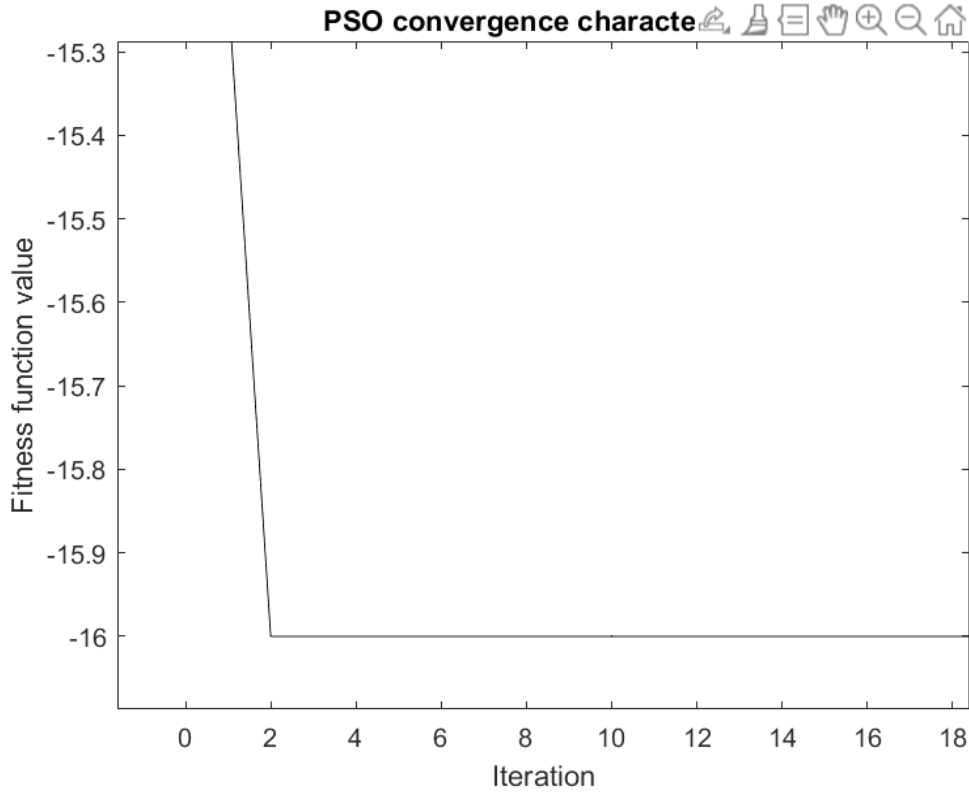
A. Computational Result:

```
*****
Final Results-----
bestfun =
-16|

bestrun =
1

best_variables =
0      8

*****
Elapsed time is 0.249822 seconds.
```



IV. TRANSPORTATION PROBLEM

			a_i
	1	1	4
	2	1	6
b_j	5	5	

$$\sum a_i = \sum b_j = 10$$

Maximize profit

$$\text{Max } Z = C_{11}X_{11} + C_{12}X_{12} + C_{21}X_{21} + C_{22}X_{22}$$

Subject to

$$X_{11} + X_{12} = 4$$

$$X_{21} + X_{22} = 6$$

$$X_{11} + X_{21} = 5$$

$$X_{12} + X_{22} = 5$$

then

$$\text{Max } Z = X_1 + X_2 + 2X_3 + X_4$$

Subject to

$$X_1 + X_2 = 4$$

$$X_3 + X_4 = 6$$

$$X_{11} + X_3 = 5$$

$$X_2 + X_4 = 5$$

$$X_1, X_2, X_3, X_4 \geq 0$$

Parameter are set to be

- Population size = 3
- c_1 and $c_2 = 2$
- dimension of problem = 4
- r_1 and r_2 are random number between 0 and 1

The movement of particles are

$$V_{i,j}^{t+1} = \omega \cdot V_{i,j}^t + c_1 \cdot r_1^t (pbest_{i,j}^t - X_{i,j}^t) + c_2 \cdot r_2^t (gbest_j^t - X_{i,j}^t)$$

$$X_{i,j}^{t+1} = X_{i,j}^t + V_{i,j}^{t+1}$$

TABLE III: Solution of Transportation problem

Iteration	Particle	X_1	X_2	X_3	X_4	Function Value	Pbest	Gbest
1	1	1.2	3.6	4	0.92	13.72	13.72	13.75
	2	0.75	3.2	4.5	0.8	13.75	13.75	
	3	1.6	3.4	3.5	0.7	9.2	9.2	
2	1	0.601	3.844	5	0.978	15.42	15.42	15.42
	2	0.817	3.488	4.905	0.872	14.98	14.98	
	3	0.729	3.425	5	0.871	15.02	15.02	
3	1	0.183	4	5	1	15.183	15.42	15.46
	2	0.649	3.819	5	1	15.46	15.46	
	3	0.12	3.617	5	1	14.73	15.02	
4	1	0	3.709	5	1	14.709	15.183	15.54
	2	0.548	4	5	1	15.54	15.54	
	3	0.496	3.745	5	1	15.24	15.24	
5	1	0	4	5	1	15	15	15.54
	2	0.497	4	5	1	15.49	15.54	
	3	0.435	4	5	1	15.43	15.43	
6	1	0	4	5	1	15	15	15.53
	2	0.53	4	5	1	15.53	15.54	
	3	0.413	4	5	1	15.41	15.43	

Final Results-----

$$Gbest = [0.53 \ 4 \ 5 \ 1]$$

then,

$$|f(x)_{prev} - f(x)| = 0$$

$$|15.54 - 15.53| = 0$$

$$0.01 \geq 0$$

then by particle swarm optimization

$$X_1 = 0.53, \ X_2 = 4, \ X_3 = 5, \ X_4 = 1, \ Z = 15.53$$

bestfun =

-15

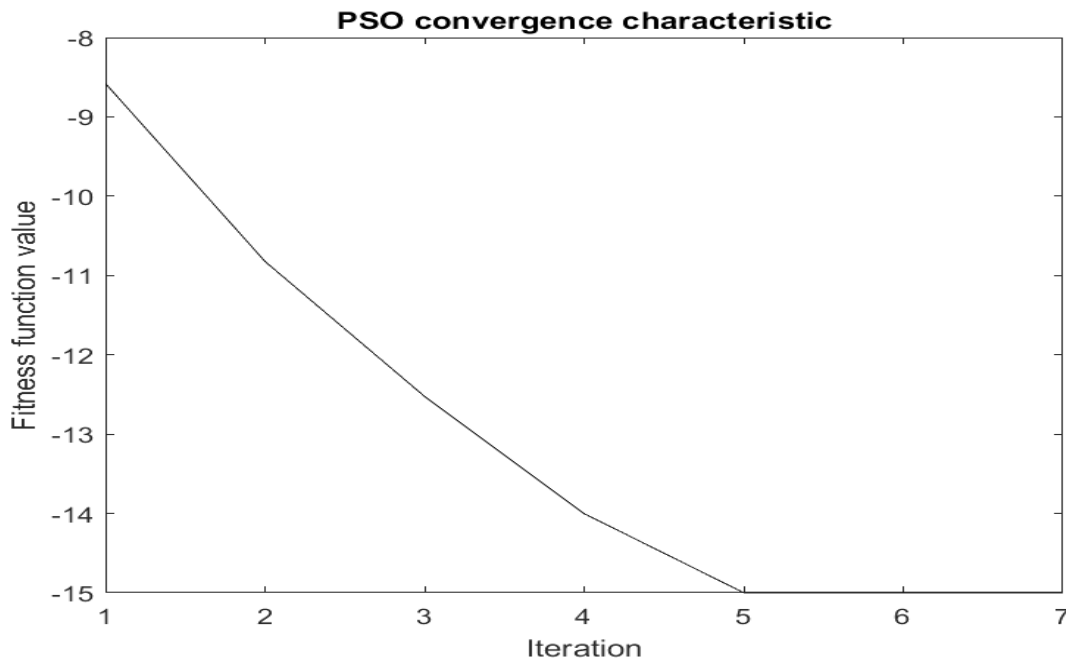
bestrun =

6

best_variables =

0 4 5 1

A. Computational Result:



V. BREAST CANCER DIAGNOSTIC DATA SET

TABLE IV: Data Set

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave pt mn
0	842302	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710
1	842517	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017
2	84300903	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790
3	84348301	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520
4	84358402	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430
...	...	...	...	...	...	...	...	...	...
564	926424	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890
565	926682	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791
566	926954	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302
567	927241	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200
568	92751	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000

	symtry_mean	...	radius_worst	text_worst	perimeter_worst	area_worst	...	concavity_worst	c pts_worst	symt_worst	fractal_dim_wrt
0	0.2419	...	25.380	17.33	184.60	2019.0	...	0.7119	0.2654	0.4601	0.11890
1	0.1812	...	24.990	23.41	158.80	1956.0	...	0.2416	0.1860	0.2750	0.08902
2	0.2069	...	23.570	25.53	152.50	1709.0	...	0.4504	0.2430	0.3613	0.08758
3	0.2597	...	14.910	26.50	98.87	567.7	...	0.6869	0.2675	0.6638	0.17300
4	0.1809	...	22.540	16.67	152.20	1575.0	...	0.4000	0.1625	0.2364	0.07678
...	...	...	...	...	...	...	...	...	...	...	...
564	0.1726	...	25.450	26.40	166.10	2027.0	...	0.4107	0.2216	0.2060	0.07115
565	0.1752	...	23.690	38.25	155.00	1731.0	...	0.3215	0.1628	0.2572	0.06637
566	0.1590	...	18.980	34.12	126.70	1124.0	...	0.3403	0.1418	0.2218	0.07820
567	0.2397	...	25.740	39.42	184.60	1821.0	...	0.9387	0.2650	0.4087	0.12400
568	0.1587	...	9.456	30.37	59.16	268.6	...	0.0000	0.0000	0.2871	0.07039

The attributes of the digital picture of a fine needle aspirate (FNA) of a breast mass include the

- ID Number
- Diagnosis, which is either Malignant (M) or Benign (B)

The picture also has ten real-valued features which describe the characteristics of the visible cell nuclei. These features include the

- radius
- texture
- perimeter
- area
- smoothness
- compactness
- concavity
- concave points
- symmetry
- fractal dimension, Length of Data set = 569

These features can help in the diagnosis of the breast mass by providing information about the characteristics of the cell nuclei in the picture.

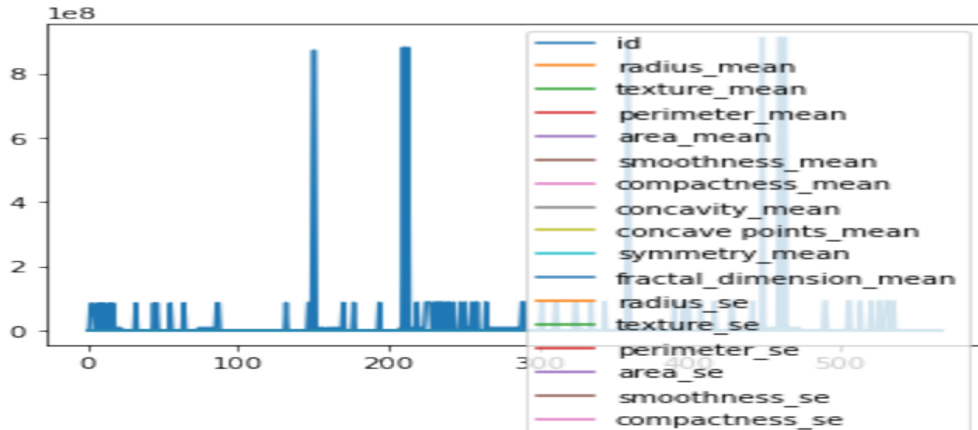


Fig. 3: Data Set Plot

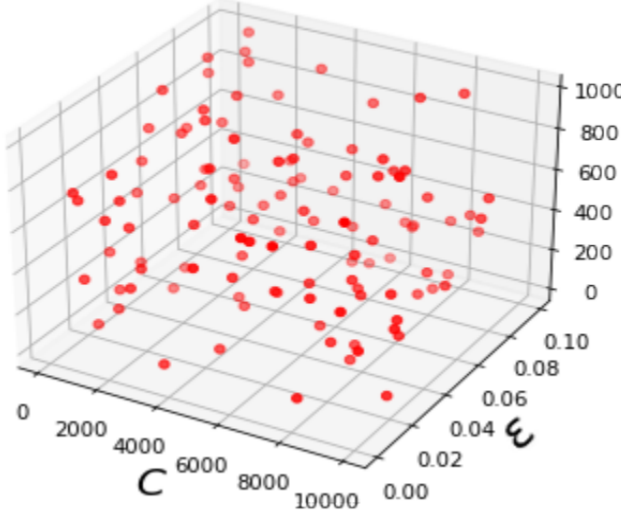


Fig. 4: DataSet Plot with Particles

- Group best configuration found: [33.44793 0.0355742 0.001]
- Regressor: $C = 33.44793$ $\epsilon = 0.0355742$ $\gamma = 0.001$

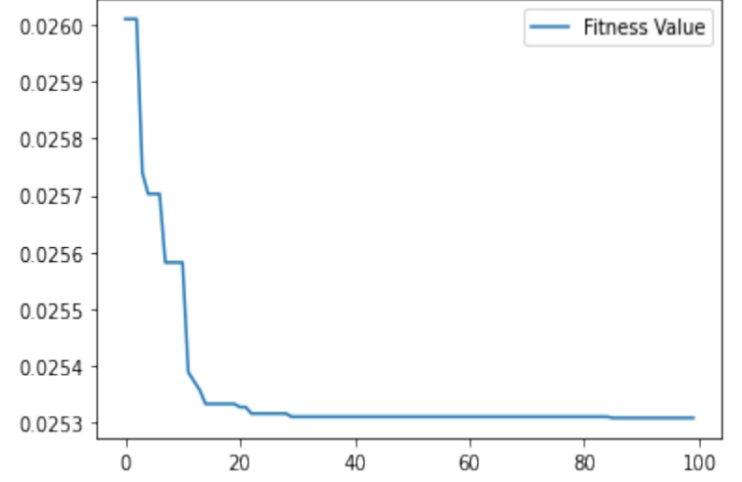


Fig. 5: Best Regressor fitness value

A. Result:

TABLE V: Function Value with inertia 1

Iter	Funct Value	Iter	Funct Value	Iter	Funct Value
0	0.026010	33	0.025310	66	0.025310
1	0.026010	34	0.025310	67	0.025310
2	0.026010	35	0.025310	68	0.025310
3	0.025740	36	0.025310	69	0.025310
4	0.025702	37	0.025310	70	0.025310
5	0.025702	38	0.025310	71	0.025310
6	0.025702	39	0.025310	72	0.025310
7	0.025582	40	0.025310	73	0.025310
8	0.025582	41	0.025310	74	0.025310
9	0.025582	42	0.025310	75	0.025310
10	0.025582	43	0.025310	76	0.025310
11	0.025389	44	0.025310	77	0.025310
12	0.025373	45	0.025310	78	0.025310
13	0.025357	46	0.025310	79	0.025310
14	0.025333	47	0.025310	80	0.025310
15	0.025333	48	0.025310	81	0.025310
16	0.025333	49	0.025310	82	0.025310
17	0.025333	50	0.025310	83	0.025310
18	0.025333	51	0.025310	84	0.025310
19	0.025333	52	0.025310	85	0.025308
20	0.025327	53	0.025310	86	0.025308
21	0.025327	54	0.025310	87	0.025308
22	0.025316	55	0.025310	88	0.025308
23	0.025316	56	0.025310	90	0.025308
24	0.025316	57	0.025310	91	0.025308
25	0.025316	58	0.025310	92	0.025308
26	0.025316	59	0.025310	93	0.025308
27	0.025316	60	0.025310	94	0.025308
28	0.025316	61	0.025310	95	0.025308
29	0.025310	62	0.025310	96	0.025308
30	0.025310	63	0.025310	97	0.025308
31	0.025310	64	0.025310	98	0.025308
32	0.025310	65	0.025310	99	0.025308

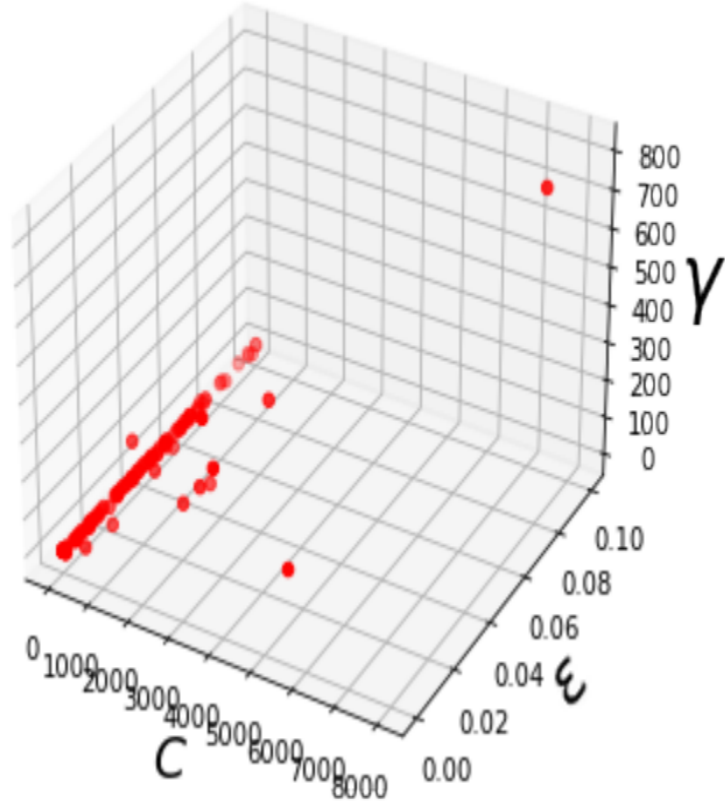


Fig. 6: Prediction

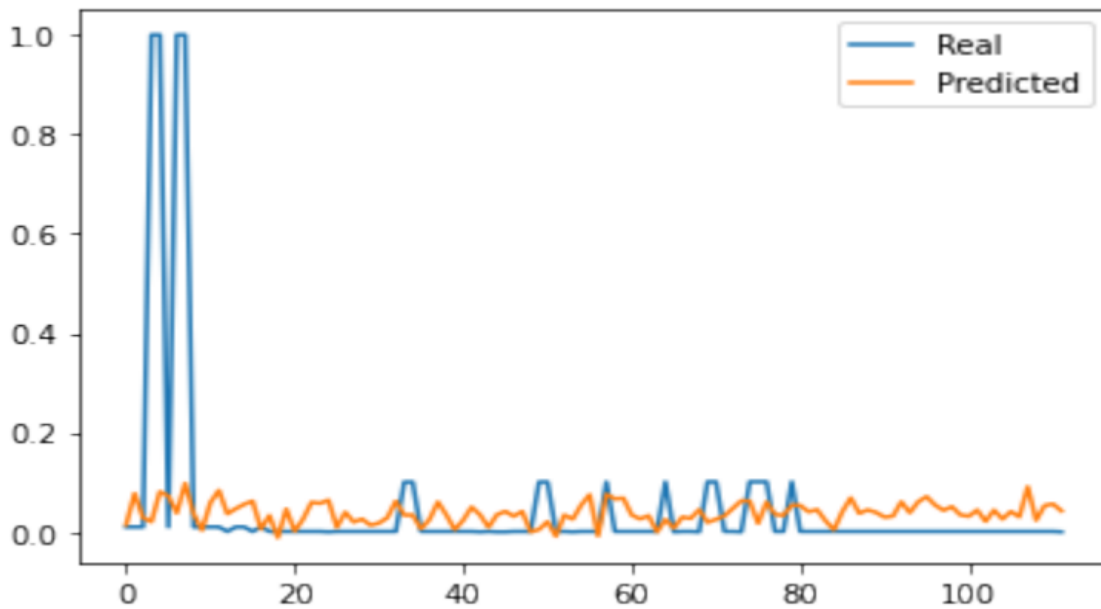


Fig. 7: Prediction with population best value found

- Mean Squared error for the test set: 0.033581
- Predictions Average: 0.037820
- Predictions Median: 0.034948

- [8] Garg, H. (2016) "A hybrid PSO-GA algorithm for constrained optimization problems," *Applied Mathematics and Computation*, 274, pp. 292–305. Available at: <https://doi.org/10.1016/j.amc.2015.11.001>.
- [9] Hu, X. (no date) Pso Tutorial, Particle Swarm Optimization: Tutorial. Available at: <http://www.swarmintelligence.org/tutorials.php> (Accessed: January 29, 2023).
- [10] Cuevas, E. and Rodríguez, A. (2020) "Particle Swarm Optimization (PSO) algorithm," *Metaheuristic Computation with MATLAB®*, pp. 159–181. Available at: <https://doi.org/10.1201/9781003006312-6>.

REFERENCES

- [1] James Kennedy, Russell Eberhart, and Yuhui Shi. *Swarm Intelligence*. Morgan Kaufmann, 2001
- [2] Clerc, M. (2010) *Particle swarm optimization*. London: ISTE.
- [3] Rao, S. S. (2009). *Engineering Optimization: Theory and Practice* (4th ed.). Wiley.
- [4] Particle Swarm Optimization (PSO). A Tutorial. Particle Swarm Optimization (PSO). A Tutorial - ScienceDirect, 12 Sept. 2015
- [5] Learning, U.C.I.M. (2016) Breast cancer wisconsin (diagnostic) data set, Kaggle. Available at: <https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data> (Accessed: January 29, 2023).
- [6] Prajapati, R., Dubey, O.P. and Kumar, R. (2017) "Improved particle swarm optimization for non-linear programming problem with barrier method," *International Journal of Students' Research in Technology and Management*, 5(4), pp. 72–80. Available at: <https://doi.org/10.18510/ijstrtm.2017.5410>.
- [7] Huang, H. and Hao, Z. (2009) "Particle swarm optimization algorithm for transportation problems," *Particle Swarm Optimization* [Preprint]. Available at: <https://doi.org/10.5772/6754>.