



Particle Swarm Optimization Algorithm – A review

Supervised By

Dr: Ahmed Elngar

Assistant Professor at Faculty of Computers & Artificial Intelligence - Beni Suef University

Email: elngar_7@yahoo.co.uk

Submitted by

- 1- Mohamed Abdelrahman Aboelkassem
Computer Science Student at Beni Suef University, E-mail: mohamedabdelrahman345_sd@fcis.bsu.edu.eg
- 2- Mohamed Gamal Yaseen
Computer Science Student at Beni Suef University, E-mail: mohamedgamal336_sd@fcis.bsu.edu.eg
- 3- Mohamed Essam Abdelmoneem
Computer Science Student at Beni Suef University, E-mail: mohamedessam346_sd@fcis.bsu.edu.eg
- 4- Nourhan Mahmoud Hussein
Computer Science Student at Beni Suef University, E-mail: nourhanmahmoud382_sd@fcis.bsu.edu.eg
- 5- Ola Abdallah Mogawer
Computer Science Student at Beni Suef University, E-mail: olaabdallah319_sd@fcis.bsu.edu.eg

Abstract: Among the many methods used for optimization, Particle Swarm Optimization (PSO) is a global heuristic method of optimization, originally introduced by Researchers Kennedy and Eberhart developed a particle swarming optimization (PSO) algorithm by simulating the prey behavior of birds or fish. In this algorithm, every possible solution in the search area is viewed as a "bird", which is known as a "particle". All particles have specific fitness values. It is evaluated by the fitness function to be improved, and it has particle flight velocities. It flies through the problem space by following the current optimum particles. It is now one of the most popular and popular optimization techniques it provides because of its amazing results and its ease of understanding, working with and real implementation. In this paper, we present an explanation of the concept of Particle Swarm Optimization (PSO) , its method of operation, the idea of work, its algorithm, as well as its mathematical equation, and it includes an example for clarification, the types into which this method is divided and a comparison between these types and the advantages and disadvantages of this type of improvement.

Keywords: *PSO, Swarm Intelligence (SI), evolutionary computation (EC), Global best (Gbest), Personal best (Pbest), population, Fitness value, objective function, Searching for Optimum Solution.*

1. Introduction

Artificial Intelligence (AI) is the intelligence displayed by machines, where the intelligent agent represents a system that is aware of its environment and takes actions that increase its chances of success. Artificial intelligence research is very high-tech, very specialized, and divided into subfields that often fail to communicate with one another.[9] Popular methods of AI today include traditional statistical methods, traditional symbolic artificial intelligence, and computational intelligence (CI). CI is a fairly new field of research. It is a set of nature-inspired computational methodologies and approaches to address complex real-world problems for which traditional approaches are ineffective or inapplicable.[16] CI includes the artificial neural network (ANN), fuzzy logic, and evolutionary computation (EC). Swarm Intelligence (SI) is part of the EC. It investigates the collective behavior of self-organizing, natural or artificial decentralized systems. [16]

Typical SI systems consist of a set of simple agents that interact locally with each other and with their environment. Inspiration often comes from the nature surrounding humans, especially biological systems and the behavior of other living things.[5] Although in the SI system, agents follow very simple rules. There is no specific central control structure that clarifies how individuals should interact. Real customer behaviors at the local level, and to some extent randomly occurring; However, the interactions that occur between these factors give rise to "intelligent" global behavior. [16]

The most well-known examples of SI include ant colonies, bird gathering, animal herding, and fish education. Dorigo proposed an ant colony improvement (ACO) method based on simulating an ant colony. Kennedy and Eberhart proposed a Particle Swarm Improvement (PSO) method based on bird flow simulations. These two are the most popular optimization algorithms affiliated with SI. Additionally, scholars have demonstrated a keen interest in proposing smart new approaches and ideas. Researchers Storn and Price proposed a differential evolution (DE). Karaboga and Basturk proposed an artificial bee colony (ABC), which mimics the foraging behavior of honey bees. Yang proposed a bat algorithm (BA), inspired by the echolocation behavior of microbes.[5]

Reports and numbers prove that the total posts related to PSO are significantly higher than other algorithms, and the number of posts per year related to PSO is the highest among all the seven SI-based algorithms. This indicates that PSO is the most popular SI-based optimization algorithm. Therefore, we focus this review on PSO.[5]

Basic Idea of PSO Each particle is searching for the optimum. Each particle is moving and hence has a velocity. Each particle remembers the position it was in where it had its best result so far (pbest). The particles in the swarm co-operate. They exchange information about what they've discovered. To get the global optimum (gbest), so each particle keeps track: its best solution, personal best, pbest. the best value of any particle, global best, gbest.[9]

In practice, the algorithm works as follows. Initialize the swarm (population) from the solution space. Evaluate the fitness of each particle according to the objective function. If a particle's current position is better than its previous best position, update it. Determine the best particle according to the swarm. Update particle's velocity. Update particle's position. Go to step2 and repeat until termination condition.[14][15]

2. Overview

Particle swarm optimization (PSO) is defined as computational procedure to recruit or select the most effective element from a collection of accessible alternatives. Associated optimization drawback either deal with increment or minimization in the true operation for simplest state of affairs while constantly screening the input elements at intervals associated with allowed set of accessible alternatives.[2]

Bird flocks, fish schools, and animal herds constitute representative examples of natural systems where aggregated behaviors are met, producing impressive, collision-free, synchronized moves. In such systems, the behavior of each group member is based on simple inherent responses, although their outcome is rather complex from a macroscopic point of view. For example, the flight of a bird flock can be simulated with relative accuracy by simply maintaining a target distance between each bird and its immediate neighbors. This distance may depend on its size and desirable behavior. For instance, fish retain a greater mutual

distance when swimming carefree, while they concentrate in very dense groups in the presence of predators. The groups can also react to external threats by rapidly changing their form, breaking in smaller parts and re-uniting, demonstrating a remarkable ability to respond collectively to external stimuli in order to preserve personal integrity.[7]

Similar phenomena are observed in physical systems. A typical example is the particle aggregation caused by direct attraction between particles due to Brownian motion or fluid shear. Humans too are characterized by agnate behaviors, especially at the level of social organization and belief formulation. However, these interactions can become very complex, especially in the belief space, where, in contrast to the physical space, the same point (a belief or an idea) can be occupied concurrently by large groups of people without collisions. The aforementioned aggregating behaviors, characterized by the simplicity of animal and physical systems or the abstractness of human social behavior, intrigued researchers and motivated their further investigation through extensive experimentation and simulations (Heppner & Grenander, 1990; Reynolds, 1987; Wilson, 1975).

Intense research in systems where collective phenomena is met prepared the ground for the development of swarm intelligence. Notwithstanding their physical or structural differences, such systems share common properties, recognized as the five basic principles of swarm intelligence (Millonas, 1994):

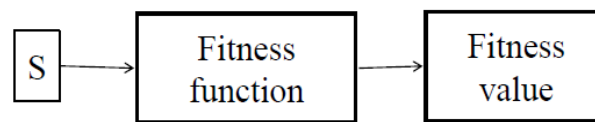
- 1) Proximity: Ability to perform space and time computations.
- 2) Quality: Ability to respond to environmental quality factors.
- 3) Diverse response: Ability to produce a plurality of different responses.
- 4) Stability: Ability to retain robust behaviors under mild environmental changes.
- 5) Adaptability: Ability to change behavior when it is dictated by external factors.

Moreover, the social sharing of information among individuals in a population can provide an evolutionary advantage.[15] This general belief, which was suggested in several studies and supported by numerous examples from nature, constituted the core idea behind the development of PSO.

3. Mechanism

PSO simulates the behaviors of bird flocking. Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration. So what's the best strategy to find the food? The effective one is to follow the bird which is nearest to the food.

PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All of particles have fitness values which 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 by following the current optimum particles.



PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is 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 particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest.[1]

After finding the two best values, the particle updates its velocity and positions with following equations

$$V_{i+1} = \omega V_i + c_1 r_1 (p_i - x_i) + c_2 r_2 (p_g - x_i) \quad (1)$$

$$x_{i+1} = x_i + V_{i+1} \quad (2)$$

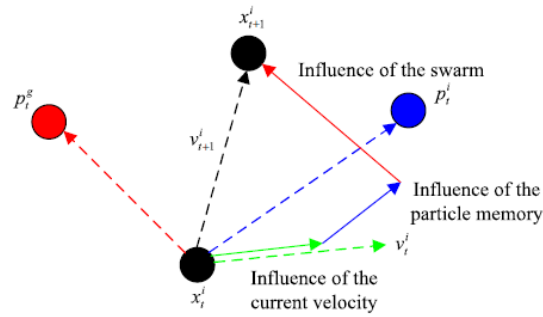


Fig 1: Iteration scheme of the particles

Where x_i is the current position of particle i . V_i is the particle's velocity at time i . ω is positive constants, called inertia factor which is linearly reducing with iteration. c_1 and c_2 are non-negative constants, called cognitive learning rate. r_1 and r_2 are random numbers in range $[0,1]$. p_i is the particle's individual best solution as of time i . p_g is the swarm's best solution.

4. Flowchart

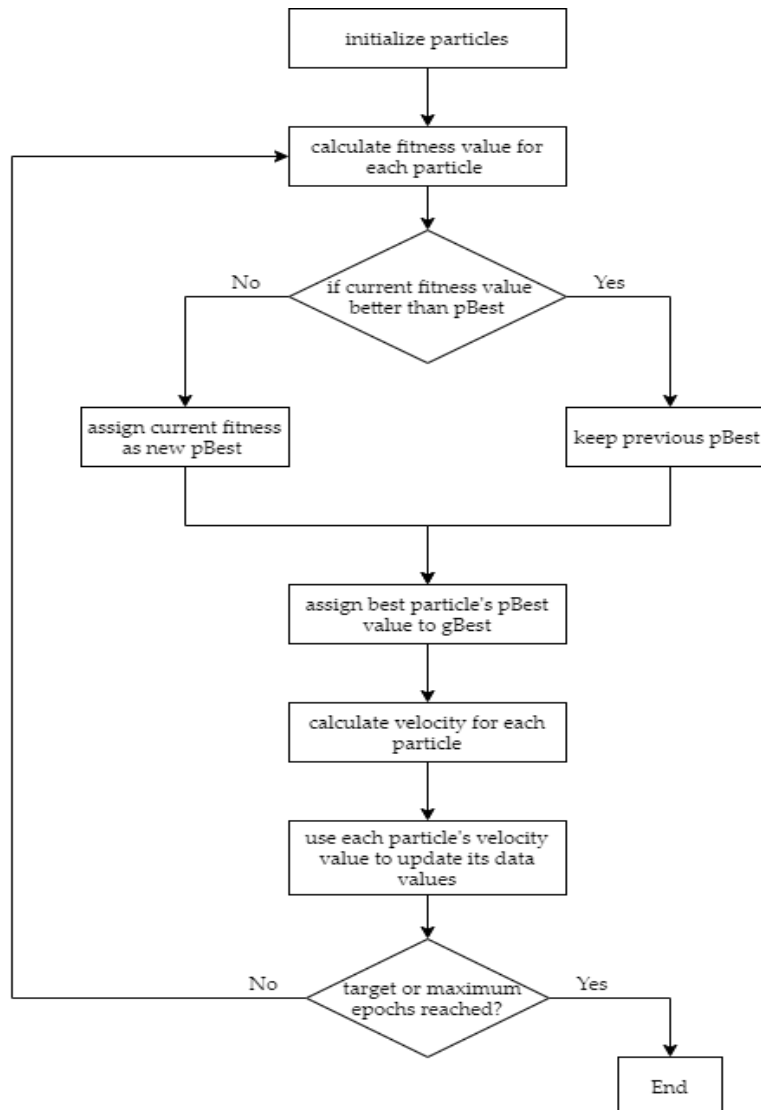


Fig 2: Flowchart of Particle Swarm Optimization Algorithm

5. Pseudo code

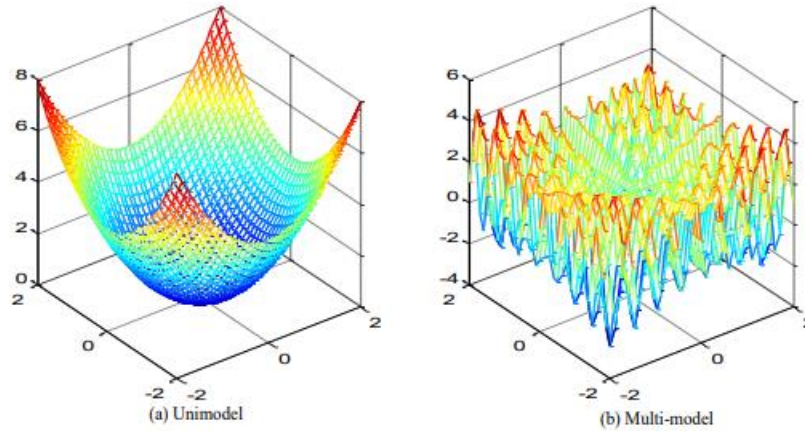
```

P=Particle_Initialization();
for i=1 to it-max
  for each particle p in P do
    fp = f(p)
    if fp is better than f(pBest)
      pBest = p;
    end
  end

  gBest=best p in P;
  for each particle p in P do
    v = v + c1*r1*(pBest - p) + c2*r2*(gBest - p)
    p =p + v
  end
end

```

6. The Model of PSO algorithm



From figure (a), it is clear that the global minimum of the function f_1 is at $(x_1, x_2) = (0, 0)$, i.e. at the origin of function in the search space. This certainly means that it is a single-model function, with only one minimum. However, finding the optimum global level is never easy for multi-model functions, which have multiple local minimums.

Figure (b) shows the function f_2 which has a rough search space with multiple peaks, so many agents have to start from different initial locations and continue exploring the search space until at least one agent reach the global optimal position. During this process all agents (elements) can communicate and share their information among themselves.

The Particle Swarm Optimization (PSO) algorithms are a multi-agent parallel search method which preserves a swarm of particles and every particle denotes a possible solution in the swarm. All particles fly during a multi-dimensional search space where every particle is modifying its position agree to its own experience and the neighbors. Suppose x_i^t represent the position vector of particle i in the multi-dimensional search space (i.e. R^n) at time step then the position of every particle is updated in the search space by:[6]

$$x_i^{t+1} = x_i^t + v_i^{t+1} \text{ with } x_i^0 \sim U(x_{min}, x_{max})$$

Where

v_i^t is the velocity vector of particle i that drives the optimization process and reflects both the own experience knowledge and the social experience knowledge from the all particles.

$U(x_{min}, x_{max})$ is the uniform distribution where min x and max x are its minimum and maximum values respectively.

Hence, in a Particle Swarm Optimization technique, all particles are originated unsystematically and estimated to calculate suitability together with finding the best value of each particle and best value of particle in the entire swarm. After that a loop starts to find an optimum solution. In the loop, first the particles' velocity is updated by the personal and global bests, and then each particle's positions are updated by the current velocity. The loop is ended with a stopping criterion predetermined in advance.[8]

6.1 The global best PSO algorithm

The global best PSO (or gbest PSO) is a method where the position of each particle is influenced by the best-fit particle in the entire swarm. the social information obtained from all particles in the entire swarm, In this method each individual particle, has a current position in search space, a current velocity, and a personal best position in search space.

The personal best position corresponds to the position in search space where particle had the smallest value as determined by the objective function, considering a minimization problem. In addition, the position yielding the lowest value amongst all the personal best is called the global best position which is denoted by G_{best}

the personal best position $P_{best,i}$ at the next time step, $t + 1$ where $t \in [0, \dots, N]$, is calculated as

$$P_{best,i}^{t+1} = \begin{cases} P_{best,i}^t & \text{if } f(x_i^{t+1}) > P_{best,i}^t \\ x_i^{t+1} & \text{if } f(x_i^{t+1}) \leq P_{best,i}^t \end{cases}$$

Where $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is the fitness function. The global best position G_{best} at time step is calculated as

$$G_{best} = \min\{P_{best,i}^t \text{ where } i \in [0, \dots, n] \text{ and } n > 1\}$$

For gbest PSO method, the velocity of particle is calculated by

$$V^{j+1} = v^j + c_1 r_1^j [P_{best,j}^r - x_j^r] + c_2 r_2^j [G_{best,j}^r - x_j^r]$$

where

f : The Function Being Minimized or Maximized. It Takes A Vector Input and Returns A Scalar Value. v_{ij}^t : particle velocity vector i in dimension j at time t . x_{ij}^t : particle position vector i in dimension j at time t . G_{best} : particle global best position i in dimension j found from initialization through time t . $P_{best,i}^t$: Particle Personal Best Position i in dimension j found from initialization through time t . c_1, c_2 : Positive acceleration coefficients which are used to level the contribution of social components and the cognitive respectively. r_{1j}^t, r_{2j}^t : Unsystematic quantities from uniform distribution $U(0,1)$ at time t . n : The swarm size or number of particles. t : Denotes time or time steps. [6] n : The swarm size or number of particles. t : Denotes time or time steps.

7. Mathematical Formulation and Example

Problem: Find the maximum of the function

$f(x) = -x^2 + 5x + 20$ with $-10 \leq x \leq 10$ using the PSO algorithm. Use 9 particles with the initial positions

$x_1 = -9.6, x_2 = -6, x_3 = -2.6, x_4 = -1.1, x_5 = 0.6, x_6 = 2.3, x_7 = 2.8, x_8 = 8.3, x_9 = 10.$

Show the detailed computations for iterations 1, 2 and 3.

Solution:

Step 1: Choose the number of particles

$$x_1 = -9.6, x_2 = -6, x_3 = -2.6, x_4 = -1.1, x_5 = 0.6, x_6 = 2.3, x_7 = 2.8, x_8 = 8.3, x_9 = 10.$$

The initial population (i.e. the iteration number $t = 0$) can be represented as x^0 , where

$$i = 1, 2, 3, 4, 5, 6, 7, 8, 9$$

$$x_1^0 = -9.6, x_2^0 = -6, x_3^0 = -2.6, x_4^0 = -1.1, x_5^0 = 0.6, x_6^0 = 2.3, x_7^0 = 2.8, x_8^0 = 8.3, x_9^0 = 10.$$

Evaluate the objective function values as

$$f_1^0 = -120.16, f_2^0 = -46, f_3^0 = 0.24, f_4^0 = 13.29, f_5^0 = 22.64, f_6^0 = 26.21, f_7^0 = 26.16, f_8^0 = -7.39, f_9^0 = -30$$

Set the initial velocities of each particle to zero:

$$v_1^0, v_2^0, v_3^0, v_4^0, v_5^0, v_6^0, v_7^0, v_8^0, v_9^0 = 0$$

Step 2: Set the iteration number as $t = 0 + 1 = 1$ and go to step 3.

Step 3: Find the personal best for each particle by

$$P_{best,i}^{t+1} = \begin{cases} P_{best,i}^t \rightarrow \text{if } f(x_i^{t+1}) > P_{best,i}^t \\ x_i^{t+1} \rightarrow \text{if } f(x_i^{t+1}) \leq P_{best,i}^t \end{cases}$$

so

$$P_{best,1}^1 = -9.6, P_{best,2}^1 = -6, P_{best,3}^1 = -2.6, P_{best,4}^1 = -1.1, P_{best,5}^1 = 0.6, P_{best,6}^1 = 2.3, P_{best,7}^1 = 2.8, P_{best,8}^1 = 8.3, P_{best,9}^1 = 10$$

Step 4: Find the global best by

$$G_{best} = \min\{P_{best,i}^t \text{ where } i = 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

Since, the maximum personal best is:

$$P_{best,6}^1 = 2.3, G_{best} = 2.3$$

Step 5: Considering the random numbers in the range (0,1) As $r_1^1 = 0.213$ and $r_2^1 = 0.876$ and find the velocities of the particles by $V^{j+1} = v^j + cr^j [P_{best,j}^r - x_j^r] + c_2 r_2^1 [G_{best,j}^r - x_j^r]$ where $i = 1, 2, 3, 4, 5, 6, 7, 8, 9$

So

$$v_1^3 = 4.4052, v_2^3 = 3.0862, v_3^3 = 1.8405, v_4^3 = 1.2909, v_5^3 = 0.6681, v_6^3 = 0.053, v_7^3 = -0.1380, v_8^3 = -2.1531, v_9^3 = -2.7759$$

Step 6: Find the new values of $x_i^1, i = 1, \dots, 9$ by $x_i^{t+1} = x_i^t + v_i^{t+1}$

So

$$x_1^1 = 0.8244, x_2^1 = 1.2708, x_3^1 = 1.6924, x_4^1 = 1.8784, x_5^1 = 2.0892, x_6^1 = 2.3, x_7^1 = 2.362, x_8^1 = 3.044, x_9^1 = 3.2548.$$

Step 7: Find the objective function values of $x_i^1, i = 1, \dots, 9$:

$$f_1^1 = 23.4424, f_2^1 = 24.7391, f_3^1 = 25.5978, f_4^1 = 25.8636, f_5^1 = 26.0812, f_6^1 = 26.21, f_7^1 = 26.231, \\ f_8^1 = 25.9541, f_9^1 = 25.6803$$

Step 8: Stopping criterion:

If the terminal rule is satisfied, go to step 2, Otherwise stop the iteration and output the results.

Step 2: Set the iteration number as $t = 1 + 1 = 2$, and go to step 3.

Step 3: Find the personal best for each particle.

$$P_{best,1}^2 = 0.8244, P_{best,2}^2 = 1.2708, P_{best,3}^2 = 1.6924, P_{best,4}^2 = 1.8784, P_{best,5}^2 = 2.0892, \\ P_{best,6}^2 = 2.3438, P_{best,7}^2 = 2.362, P_{best,8}^2 = 3.044, P_{best,9}^2 = 3.2548.$$

Step 4: Find the global best.

$$G_{best} = 2.362$$

Step 5: By considering the random numbers in the range (0, 1) as $r_1^3 = 0.178$ and $r_2^3 = 0.507$, find the velocities of the particles by $v_i^{t+1} = v_i^t + c_1 r_1^t [P_{best,i}^t - x_i^t] + c_2 r_2^t [G_{best} - x_i^t]$ Where $i = 1, \dots, 9$.

So

$$v_1^2 = 11.5099, v_2^2 = 8.0412, v_3^2 = 4.7651, v_4^2 = 5.1982, v_5^2 = 1.6818, \\ v_6^2 = 0.0438, v_7^2 = -0.04380, v_8^2 = -5.7375, v_9^2 = -7.3755.$$

Step 6: Find the new values of x_i^2 where $i = 1, \dots, 9$ by $x_i^{t+1} = x_i^t + v_i^{t+1}$,

So

$$x_1^2 = 12.3343, x_2^2 = 9.312, x_3^2 = 6.4575, x_4^2 = 5.1982, x_5^2 = 3.7710, \\ x_6^2 = 2.3438, x_7^2 = 1.9240, x_8^2 = -2.6935, x_9^2 = -4.1207.$$

Step 7: Find the objective function values of f_i^2 Where $i = 1, \dots, 9$;

$$f_1^2 = -70.4644, f_2^2 = -20.1532, f_3^2 = 10.5882, f_4^2 = 18.9696, \\ f_5^2 = 24.6346, f_6^2 = 26.2256, f_7^2 = 25.9182, f_8^2 = -0.7224, f_9^2 = -17.5839.$$

Step 8: Stopping criterion:

If the terminal rule is fulfilled, go to step 2, Else stop the iteration and output the results.

Step 2: Set the iteration number as $t = 2 + 1 = 3$, and go to step 3

Step 3: Find the personal best for each particle.

$$P_{best,1}^3 = 0.8244, P_{best,2}^3 = 1.2708, P_{best,3}^3 = 1.6924, P_{best,4}^3 = 1.8784, \\ P_{best,5}^3 = 2.0892, P_{best,6}^3 = 2.3438, P_{best,7}^3 = 2.362, P_{best,8}^3 = 3.044, P_{best,9}^3 = 3.2548.$$

Step 4: Find the global best.

$$G_{best} = 2.362.$$

Step 5: since the random numbers in the *range* (0,1) as $r_1^3 = 0.178$ and $r_2^3 = 0.507$ find the velocities of the particles by : $V^{j+1} = v^j + c_1 r_1^j [P_{best,j}^r - x_j^r] + c_2 r_2^j [G_{best,j}^r - x_j^r]$ Where $i = 1, \dots, 9$. So

$$\begin{aligned} v_1^3 &= 4.4052, v_2^3 = 3.0862, v_3^3 = 1.8405, v_4^3 = 1.2909, v_5^3 = 0.6681, v_6^3 = 0.053, v_7^3 = -0.1380, \\ v_8^3 &= -2.1531, v_9^3 = -2.7759 \end{aligned}$$

Step 6: Find the new values of x_i^3 , $i = 1 \dots, 9$ by $x_i^{t+1} = x_i^t + v_i^{t+1}$ So

$$\begin{aligned} x_1^3 &= 16.7395, x_2^3 = 12.3982, x_3^3 = 8.298, x_4^3 = 6.4892, x_5^3 = 4.4391, x_6^3 = 2.3968, x_7^3 = 1.786, \\ x_8^3 &= -4.8466, x_9^3 = -6.8967. \end{aligned}$$

Step 7: Find the objective function values of f_i^3 Where $i = 1, \dots, 9$:

$$\begin{aligned} f_1^3 &= -176.5145, f_2^3 = -71.7244, f_3^3 = -7.3673, f_4^3 = 10.3367, f_5^3 = -22.49, f_6^3 = 26.2393, f_7^3 = 25.7402, \\ f_8^3 &= -27.7222, f_9^3 = -62.0471. \end{aligned}$$

Step 8: Stopping criterion:

If the terminal rule is fulfilled, go to step 2, Else stop the iteration and output the results.
Lastly, the values of x_i^t , $i = 1, 2, 3, 4, 5, 6, 7, 8, 9$. Did not converge, so we increment the iteration number as $t = 4$ and go to step 2. When the positions of all particles converge to similar values, then the method has converged and the corresponding value of x_i^t is the optimum solution. Therefore the iterative process is continued until all particles meet a single value.

8. Related Works

There are different types of PSO methods which help to solve different types of optimization problems such as Multi-start (or restart) PSO for when and how to reinitialize particles, binary PSO (BPSO) method for solving discrete-valued problems, Multi-phase PSO (MPPSO) method for partition the main swarm of particles into sub-swarms or subgroups, Multi-objective PSO for solving multiple objective problems.

TABLE 1: Types of Particle Swarm Optimization and its related works

	Algorithm's name	Description
1	Modified Particle Swarm Optimization (MPSO)	In this algorithm, the birds have a memory about the previous best and worst positions so that particles have 2 experiences, a bad experience helps each particle to remember its previous worst position. To calculate the new velocity, the bad experience of each particle is considered with the next equation:[10] $V_{i+1} = \omega V_i + C_1 r_1 (P_{best_i} - S_i) + C_1 b r_2 (P_{worst_i} - S_i) + C_2 r_3 (g_{best_i} - S_i)$
2	Weight Improved Particle Swarm Optimization (WIPSO)	the WIPSO algorithm finds the answer with less iteration and with higher speed convergence among the proposed methods. WIPSO algorithms have lower iteration numbers than PSO algorithm. In the WIPSO algorithm, in order to improve the global search quality of standard PSO, the inertia weight factor and the cognitive and social components (C_1 , C_2) have been configured with the next equation: $V_{i+1} = W_{new} V_i + C_1 r_1 (P_{best_i} - S_i) + C_2 r_2 (g_{best_i} - S_i)$ Where $W_{new} = W_{min} + w r_1$

3	Multi-Start PSO (MSPSO)	<p>In the basic PSO, when particles start to converge to the same place, one of the major problems is lack of diversity. Several methods have been developed to constantly introduce randomness, or chaos, into the swarm to avoid this issue of the fundamental PSO. These types of methods are called the Particle Swarm Optimizer Multi-start (or restart) (MSPSO). The multi-start approach is a global search algorithm and has the primary objective of increasing diversity in order to explore greater portions of the search space.</p> <p>In SOCPSO model the velocity of each particle is updated by</p> $v_{ij}^{t+1} = X[\omega v_{ij}^t + \phi_{1j}(P_{best_i} - x_{ij}^t) + \phi_{2j}(G_{best} - x_{ij}^t)]$
4	Multi-phase PSO (MPPSO)	<p>Multi-phase PSO (MPPSO) method partitions the main swarm of particles into sub-swarms or subgroups, where each sub-swarm performs a different task, exhibits a different behavior and so on. This task or behavior performed by a sub-swarm usually changes over time and information are passed among sub-swarms in this process.</p> <p>• Attraction phase • Repulsion phase</p> <p>In MPPSO algorithm, the particle velocity updating equation is presented as follows:</p> $v_{ij}^{t+1} = \omega v_{ij}^t + c_1 x_{ij}^t + c_1 G_{best}$
5	Perturbed PSO (PPSO)	<p>There are several drawbacks to the simple particle swarm optimization (PSO). for example, high speed of convergence also leads to a rapid loss of diversity during the optimization process. Then, the mechanism leads to unnecessary premature convergence. Zhao Xinchao identified a disturbed particle swarm algorithm to resolve this disadvantage, which is based on a new strategy for particle updating and the principle of disturbed global best (p-gbest) within the swarm. The disturbed global best (p-gbest) updating strategy is based on the principle of measure of possibility to model the lack of data on the gbest's true optimality. The particle velocity in PPSO is rewritten by:</p> $v_{ij}^{t+1} = \omega v_{ij}^t + c_1 r_{1j}^t [P_{best,i}^t - x_{ij}^t] + c_2 r_{2j}^t [G'_{best} - x_{ij}^t]$ <p>Where $G'_{best} = N(G_{best}, \sigma)$</p>
6	Multi-Objective PSO (MOPSO)	<p>Multi-objective optimization problems have several objective functions that need to be optimized simultaneously. In multiple-objectives cases, due to lack of common measure and confliction among objective functions, there does not necessarily exist a solution that is best with respect to all objectives. There exist a set of solutions for the multi-objective problem which cannot normally be compared with each other. Such solutions are called non-dominated solutions (or Pareto optimal solutions) only when no improvement is possible in any objective function without sacrificing at least one of the other objective functions.</p> <p>In multi-objective optimization algorithms, these cases are considered the most difficult.</p> $v_{ij}^{[S]}(t+1) = \omega v_{ij}^{[S]}(t) + c_1 r_1 [P_{best,i}^{[S]}(t) - x_{ij}^{[S]}(t)] + c_2 r_2 [G_{best}^{[a]} - x_{ij}^{[S]}(t)]$

7	Binary PSO (BPSO)	<p>The PSO algorithm was developed and most of its updated versions operated in continuous spaces that could not be used to optimize discrete-valued search spaces. In order to function on binary search spaces, they created the PSO, since real-value domains can be converted into binary-value domains. The proposed algorithm is called the binary PSO (BPSO) algorithm, in which the particles represent position in binary space and the position vectors of the particles can take the binary value 0 or 1, i.e. In this case, it maps the n-dimensional binary space (i.e. n-length bit strings) to actual numbers. (where is a fitness function and a real number set respectively). That means a particle's positions must belong to, in order to be calculated by. In BPSO, a particle's velocity is connected to the possibility that the particle's position takes a value of 0 or 1. The update equation for the velocity does not change from that used in the original PSO and the equation.</p> $v_{ij}^{t+1} = v_{ij}^t + c_1 r_{1j}^t [P_{best_i} - x_{ij}^t] + c_2 r_{2j}^t [G_{best} - x_{ij}^t]$
---	--------------------------	--

9. Advantages and Disadvantages

It is said that PSO algorithm is the one of the most powerful methods for solving the non-smooth global optimization problems while there are some disadvantages of the PSO algorithm. The advantages and disadvantages of PSO are discussed below:

9.1 Advantages of the PSO algorithm:

The main advantage of using the PSO is its simple concept and ability to be implemented in a few lines of code. Furthermore, PSO also has a short-term memory, which helps the particles to fly over the local best and the global best positions. Alternatives, such genetic algorithms (GA), are more complex and, most of the time, they do not consider the previous iteration or the collective emergent performance. For instance, in GA, if a chromosome is not selected, the information contained by that individual is lost.

9.2 Disadvantages of the PSO algorithm:

Despite its features, a general problem with the PSO, similarly to other optimization algorithms that are not exhaustive methods, such as the brute-force search (Schaeffer et al. 1993), is that of becoming trapped in a local optimum, or sub optimal solution, such that it may work well on one problem but yet fail on another problem.

In general, the main drawbacks of PSO can be summarized as follows.

- 1) Premature convergence of a swarm: Particles try to converge to a single point. located on a line between the global best and the personal best positions (local best). This point is not guaranteed for a local optimum (Van den Bergh and Engelbrecht 2004). Another reason could be the fast rate of information flow between particles, which leads to the creation of similar particles. This results in a loss in diversity and the possibility of being trapped in local optima is increased (Premalatha and Natarajan 2009).
- 2) Parameter settings dependence. This leads to the high-performance variances for a stochastic search algorithm (Premalatha and Natarajan 2009). In general, there is not any specific set of parameters for different problems. As an example, and by simple observation of $v_n[t+1] = wv_n[t] + \sum_{i=1}^2 p_i r_i(x_{in}[t] - x_n[t])$, increasing the inertia weight w will increase the speed of the particles v_{i+1} and cause more exploration (global search) and less exploitation (local search). As

a result, finding the best set of parameters is not a trivial task, and it might be different from one problem to another (Premalatha and Natarajan 2009)[6]

10. Difference between Genetic Algorithm and PSO

10.1 Commonalities

Both algorithms start with a group of a randomly generated population. Both have fitness values to evaluate the population. Both update the population and search for the optimum with random techniques. Both systems do not guarantee success.

10.2 Differences

PSO does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. In GAs, chromosomes share information with each other, So the whole population moves like a one group towards an optimal area. In PSO, only gBest gives out the information to others. It is a one - way information sharing mechanism. In GAs, the evolution only looks for the best solution. In PSO, all the particles tend to converge to the best solution quickly.[1]

11 Some applications that use PSO.

PSO have been successfully used across a wide range of applications, for instance, telecommunications, system control, data mining, power systems, design, combinatorial optimization, signal processing, network training, and many other areas. Nowadays, PSO algorithms have also been developed to solve constrained problems, multi-objective optimization problems, problems with dynamically changing landscapes, and to find multiple solutions, while the original PSO algorithm was used mainly to solve unconstrained, single-objective optimization problems. Various areas where PSO is applied like in[12]:

11.1 Antennas Design

The optimal control and design of phased arrays, broadband antenna design and modeling, reflector antennas, design of Yagi-Uda arrays, array failure correction, optimization of a reflect array antenna, far-field radiation pattern reconstruction, antenna modeling, design of planar antennas, conformal antenna array design, design of patch antennas, design of a periodic antenna arrays, near-field antenna measurements, optimization of profiled corrugated horn antennas, synthesis of antenna arrays, adaptive array antennas, design of implantable antennas.

11.2 Signal Processing

Pattern recognition of flatness signal, design of IIR filters, 2D IIR filters, speech coding, analogue filter tuning, particle filter optimization, nonlinear adaptive filters, Costas arrays, wavelets, blind detection, blind source separation, localization of acoustic sources, distributed odour source localization, and so on.

11.3 Networking

Radar networks, Bluetooth networks, auto tuning for universal mobile telecommunication system networks, optimal equipment placement in mobile communication, TCP network control, routing, wavelength division-multiplexed network, peer-to-peer networks, bandwidth and channel allocation, WDM telecommunication networks, wireless networks, grouped and delayed broadcasting, bandwidth reservation, transmission network planning, voltage regulation, network reconfiguration and expansion, economic dispatch problem, distributed generation, microgrids, congestion management, cellular neural networks, design of radial basis function networks, feed forward neural network training, product unit networks, neural gas networks, design of recurrent neural networks, wavelet neural networks, neuron controllers, wireless sensor network design, estimation of target position in wireless sensor networks, wireless video sensor networks optimization.

11.4 Biomedical

Human tremor analysis for the diagnosis of Parkinson's disease, inference of gene regulatory networks, human movement biomechanics optimization, RNA secondary structure determination, phylogenetic tree reconstruction, cancer classification, and survival prediction, DNA motif detection, biomarker selection,

protein structure prediction and docking, drug design, radiotherapy planning, analysis of brain magnetoencephalography data, electroencephalogram analysis, biometrics and so on.

11.5 Electronics and electromagnetic

On-chip inductors, configuration of FPGAs and parallel processor arrays, fuel cells, circuit synthesis, FPGA-based temperature control, AC transmission system control, electromagnetic shape design, microwave filters, generic electromagnetic design and optimization applications, CMOS RF wideband amplifier design, linear array antenna synthesis, conductors, RF IC design and optimization, semiconductor optimization, high-speed CMOS, frequency selective surface and absorber design, voltage flicker measurement, shielding, digital circuit design.

11.6 Robotics

Control of robotic manipulators and arms, motion planning and control, odour source localization, soccer playing, robot running, robot vision, collective robotic search, transport robots, unsupervised robotic learning, path planning, obstacle avoidance, swarm robotics, unmanned vehicle navigation, environment mapping, voice control of robots, and so forth.

11.7 Design and Modelling

Conceptual design, electromagnetics case, induction heating cooker design, VLSI design, power systems, RF circuit synthesis, worst case electronic design, motor design, filter design, antenna design, CMOS wideband amplifier design, logic circuits design, transmission lines, mechanical design, library search, inversion of underwater acoustic models, modeling MIDI music, customer satisfaction models, thermal process system identification, friction models, model selection, ultrawideband channel modeling, identifying ARMAX models, power plants and systems, chaotic time series modeling, model order reduction.[6]

11.8 Image and Graphics

Image segmentation, autocropping for digital photographs, synthetic aperture radar imaging, locating treatment planning landmarks in orthodontic x-ray images, image classification, inversion of ocean color reflectance measurements, image fusion, photo time-stamp recognition, traffic stop-sign detection, defect detection, image registration, microwave imaging, pixel classification, detection of objects, pedestrian detection and tracking, texture synthesis, scene matching, contrast enhancement, 3D recovery with structured beam matrix, character recognition, image noise cancellation.

11.9 Power generation and Controlling

Automatic generation control, power transformer protection, power loss minimization, load forecasting, STATCOM power system, fault-tolerant control of compensators, hybrid power generation systems, optimal power dispatch, power system performance optimization, secondary voltage control, power control and optimization, design of power system stabilizers, operational planning for cogeneration systems, control of photovoltaic systems, large-scale power plant control, analysis of power quality signals, generation planning and restructuring, optimal strategies for electricity production, production costing, operations planning.

11.10 Fuzzy systems, Clustering, data mining

Design of neurofuzzy networks, fuzzy rule extraction, fuzzy control, membership functions optimization, fuzzy modeling, fuzzy classification, design of hierarchical fuzzy systems, fuzzy queue management, clustering, clustering in large spatial databases, document and information clustering, dynamic clustering, cascading classifiers, classification of hierarchical biological data, dimensionality reduction, genetic-programming-based classification, fuzzy clustering, classification threshold optimization, electrical wader sort classification, data mining, feature selection.[6]

11.11 Optimization

Electrical motors optimization, optimization of internal combustion engines, optimization of nuclear electric propulsion systems, floor planning, travelling-sales man problems, n-queens problem, packing and knapsack, minimum spanning trees, satisfiability, knights cover problem, layout optimization, path optimization, urban planning, FPGA placement and routing.

11.12 Prediction and forecasting

Water quality prediction and classification, prediction of chaotic systems, streamflow forecast, ecological models, meteorological predictions, prediction of the floe stress in steel, time series prediction, electric load forecasting, battery pack state of charge estimation, predictions of elephant migrations, prediction of surface roughness in end milling, urban traffic flow forecasting, and so on.

12. Experimental Results and Discussion

This is an experiment work done to remove Backpropagation and in-turn Gradient Descent and use Particle Swarm Optimization technique for Neural Network Training for optimizing the network's weights and biases. The MNIST dataset used [19], it contains 60,000 training images and 10,000 testing images. Half of the training set and half of the test set were taken from NIST's training dataset, while the other half of the training set and the other half of the test set were taken from NIST's testing dataset. The original creators of the database keep a list of some of the methods tested on it.

TABLE 2: Datasets description.

Dataset	Dimension	Samples	Classes
MNIST	784	1347	10

After doing the experiment for training Neural Networking for training MNIST dataset using Particle Swarm Optimization with only 500 iterations. The final result of training NN get Bbest at iteration 499 with mean error: 0.0533665225281364, and accuracy: 0.5777777777777777.

The following table show the report of training test dataset of MNIST with main measures.

TABLE 3: Report of Testing Dataset

	Precision	Recall	F1-score	Support
0	0.92	0.83	0.87	41
1	0.00	0.00	0.00	42
2	0.71	0.57	0.63	42
3	0.00	0.00	0.00	52
4	0.97	0.71	0.83	45
5	0.82	0.65	0.73	43
6	0.00	0.00	0.00	49
7	0.77	0.51	0.62	47
8	0.00	0.00	0.00	44
9	0.66	0.51	0.57	45
Micro avg	0.84	0.46	0.60	450
Macro avg	0.58	0.47	0.52	450
Weighted avg	0.58	0.46	0.51	450
Samples avg	0.46	0.46	0.46	450

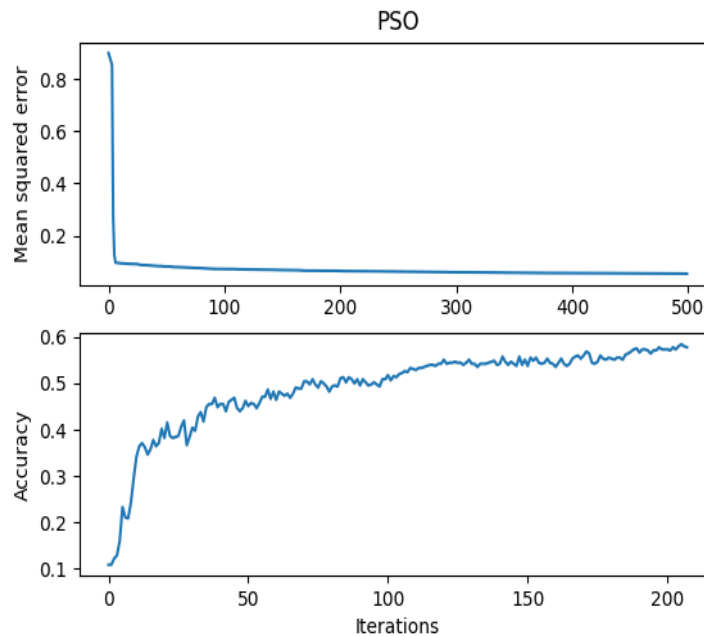


Fig 3: Performance and behavior of the Neural Network over 500 iterations with MNIST Dataset

13. Conclusion

In this paper we discussed some topics related to the basic algorithm for optimizing particle swarming, the mathematical interpretation of PSO, particle motion during work and updating the velocity of each particle in the search space, different types of PSO algorithms that solve different types of optimization problems. The multi-stage PSO algorithm (MPPSO) divides the main swarm into sub-swarms or sub-groups, with each sub-swarm performing a different task, exhibiting different behavior etc. The swarms then collaborate to solve the problem by sharing the best solutions they have discovered in their sub-flocks. During the optimization process, high convergence velocity sometimes causes rapid loss of diversity resulting in undesirable early convergence. To solve this problem, we went to explain some basic concepts in the method of Particle Swarm Optimization (PSO).

References

- [1] Eberhart RC, Shi Y and Kennedy J. (2001). *Swarm Intelligence*, Morgan Kaufmann, Burlington, 300-340.
- [2] Fractional Order Darwinian Particle Swarm Optimization: Applications and Evaluation of an Evolutionary Algorithm By Michael Couceiro, Pedram Ghamisi.
- [3] Shuai Li, Zhicong Zhang, Xiaohui Yan, Liangwei Zhang, "Probability Mechanism Based Particle Swarm Optimization Algorithm and Its Application in Resource-Constrained Project Scheduling Problems", *Discrete Dynamics in Nature and Society*, vol. 2019, 11 pages <https://doi.org/10.1155/2019/9085320>.
- [4] Kennedy, J. and Eberhart, R. C. Particle swarm optimization. *Proc. IEEE int'l conf. on neural networks* Vol. IV, pp. 1942-1948. IEEE service center, Piscataway, NJ, 1995.
- [5] Eberhart, R. C. and Kennedy, J. A new optimizer using particle swarm theory. *Proceedings of the sixth international symposium on micro machine and human science* pp. 39-43. IEEE service center, Piscataway, NJ, Nagoya, Japan, 1995.

- [6] Mathematical Modelling and Applications of Particle Swarm Optimization - by Satyobroto Talukder - Master's Thesis - Mathematical Modelling and Simulation.
- [7] Li, X.: Better spread and convergence: Particle swarm multi-objective optimization using the maximin fitness function. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3103, pp. 117–128. Springer, Heidelberg (2004).
- [8] Corne, D.W., Knowles, J.D.: Techniques for highly multi-objective optimization: Some nondominated points are better than others. In: Proceedings of Genetic and Evolutionary Computation Conference 2007 (GECCO 2007), pp. 773–780 (2007).
- [9] Marini F and Walczak B. (2015). Particle swarm optimization (PSO). A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 149, 153–165.
- [10] Junliang Li and Xinping Xiao. (2008). Multi- Swarm and Multi- Best particle swarm optimization algorithm. 7th World Congress on Intelligent Control and Automation.
- [11] Wang L, Cui Z and Zeng J. (2009). Particle Swarm Optimization with Group Decision Making. Ninth International Conference on Hybrid Intelligent Systems.
- [12] Kennedy, J. (2010). Particle swarm optimization. In *Encyclopedia of Machine Learning* (pp. 760_766). Springer [1] Intelligent System (edited by Chatchawal Wongchoosuk).
- [13] Hassan, R., Cohanin, B., De Weck, O., & Venter, G. (2005). A comparison of particle swarm optimization and the genetic algorithm. In *Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference*, 370 Honolulu, Hawaii, April 23 - 26 (pp. 1-13).
- [14] He G and Huang N. (2014). A new particle swarm optimization algorithm with an application. *Applied Mathematics and Computation*, 232(2), 521–528.
- [15] Junjun Li and Xihuai Wang. (2004). A modified particle swarm optimization algorithm. Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No.04EX788).
- [16] Wang L, Cui Z and Zeng J. (2009). Particle Swarm Optimization with Group Decision Making. Ninth International Conference on Hybrid Intelligent Systems.
- [17] Rao SS. (2009). *Engineering optimization: theory and practice*. John Wiley and Sons, Inc. New York, USA. 290- 350
- [18] Hanafi I, Cabrera FM, Dimane F and Manzanares JT. (2016). Application of Particle Swarm Optimization for Optimizing the Process Parameters in Turning of PEEK CF30 Composites. *Procedia Technology*, 22(2), 195–202.
- [19] Assessing Four Neural Networks on Handwritten Digit Recognition Dataset (MNIST) by Feiyang Chen, Nan Chen, Hanyang Mao, Hanlin Hu.