Particle Swan Optimization

**Introduction:**

Particle Swarm Optimization(PSO) Algorithm is based on a theme of ‘Work as a Team’. In 1995, Kennedy and Eberhart wrote a research paper based on the social behavior of animal groups, where they had stated that sharing information among the group increases survival advantage. Like while a bird searching for food randomly can optimize her searching if she works with the flock. The advantage of working is mutual sharing of the best information, which can help a flock to discover the best place to hunt.

**Group optimization and Ensemble Learning:**

NFL(No Free Lunch) in machine learning speaks that no single model works best for all possible situations. We can also say that all optimization algorithms perform equally well when averaged across all potential problems. The last statement that I have written isn’t self-explanatory with the example of flock of bird. Why do we need optimization in machine learning or deep learning? To train a model, we must define a loss function to measure the difference between our model predictions. Our objective is to minimize or optimize this loss function so that it will be closer to 0. Maybe you have heard about a term called ‘Ensemble Learning.’ If you have not, then let me explain you. ‘Ensemble’ is a French word—meaning ‘Assembly.’ It speaks about learning in a group or crowd. It is like you are trying to train a model with the help of multiple algorithms. So, what type of benefit are we going to get here? A single base learner is a weak learner. But, when we combine all these vulnerable learners, they become strong learners. They become strong learners because their predictive power, accuracy, precision are high. And the error rate is less. We call this type of combined model ‘Meta-learning’ in machine learning. It refers to learning algorithms that can learn from other learning algorithms. It decreases variance, decreases bias, and improves prediction. Now, when you achieve that, that’s your ultimate ‘Nirvana’ moment as a data analyst.

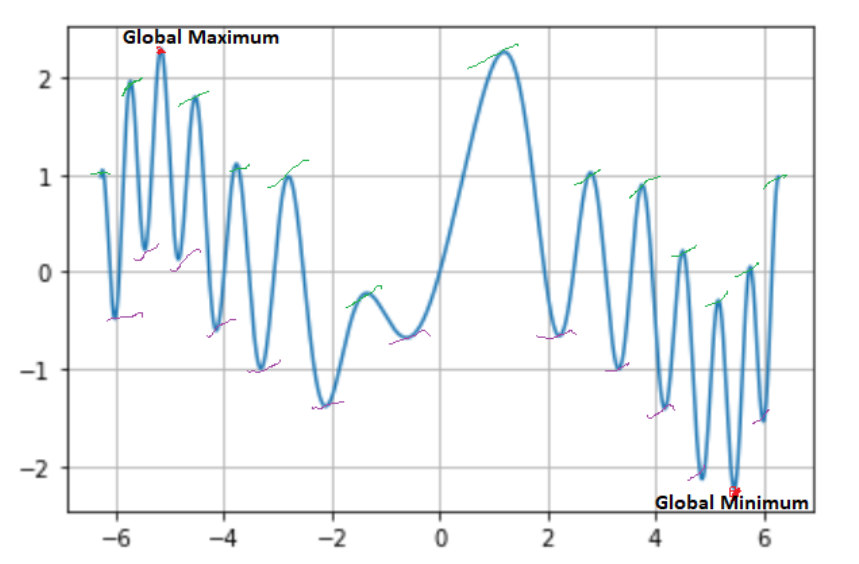
**The problem of optimization:**

The concept of swarm intelligence inspired the POS. Here we are speaking about finding the optimal solution in a high-dimensional solution space. It talks about Maximizing earns or minimizing losses. So, we are looking to maximize or minimize a function to find the optimum solution. A function can have multiple local maximum and minimum. But, there can be only one global maximum as well as a minimum. If your function is very complex, then finding the global maximum can be a very daunting task. PSO tries to capture the global maximum or minimum. Even though it cannot capture the exact global maximum/minimum, it goes very close to it. It is the reason we called PSO a heuristic model.

Let me give you an example of why the finding of global maximum/minimum is problematic. Check the below function :

                   y=f(x)=sinx+sinx^2+sinxcosx

If we plot this function, it looks like

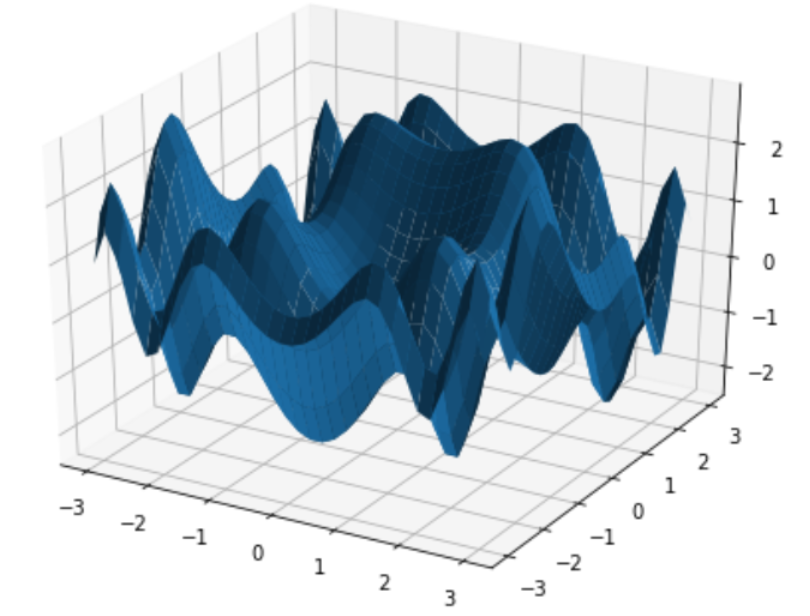


We can see that we have one global maximum and one global minimum. If we consider the function based on an interval in X-axis value from -4 to 6, we will have a maximum that will not be our global maximum. It is a local maximum. So we can say that finding out the global maximum may depend upon the interval. It is something like we try to observe a portion of a continuous function. Also, one thing to note while describing a dynamic system or entity, you can not have a static function. The function that I have defined here is fixed. Data analytics is data-hungry. To train a model or to find a suitable mathematical function, you must have enormous data. It is impossible to

have all the data. Meaning it’s challenging to get the exact global minimum or maximum. Well, for me, it’s a limitation of Mathematics. Fortunately, we have Statistics that advocate sampling, and from there, it can optimize some value like global maximum or minimum concerning the original function. But again, you won’t get the exact global maximum or minimum. You will get some values that will be closer to the actual global maximum or minimum.

Also, when we describe a mathematical function based on some real-life scenario, we must explain it with multiple variables or higher-dimensional vector space. The growth of bacteria in a jar may depend upon temperature, humidity, the container, the solvent, etc. For this type of function, it’s more challenging to get the exact global maximum and minimum. Check the below function. And see if we add more variables than how difficult it becomes to get global maximum and minimum.

z=f(x, y)=sin x^2+siny^2+sinxsiny



## The mathematical formulation of an Optimization Problem :

In the optimization problem, we have a variable represented by a vector *X=[x1x2x3…xn]* that minimizes or maximizes cost function depending on the proposed optimization formulation of the function f(X). X is known as **position vector**; it represents a variable model. It is an n dimensions vector, where n represents the number of variables determined in a problem. We can call it **latitude and the longitude**in the problem of choosing a point to land by a flock of birds. The function f(X) is called the **fitness function or objective function**. The job of f(X) is to assess how good or bad a position X is; that is, how perfect a certain landing point a bird thinks after finding a suitable place. Here, the evaluation, in this case, is performed through several survival criteria.

**An Intuition of Particle Swarm Optimization:**

The movement towards a promising area to get the global optimum.

* Each particle adjusts its traveling velocity dynamically, according to the flying experiences it has and its colleagues in the group.
* Each particle tries to keep track of :

1. It’s best result for him/her, known as personal best or **pbest**.
2. The best value of any particle is the global best or **gbest**.

* · Each particle modifies its position according to:

1. its current position
2. its current velocity
3. the distance between its current position and pbest.
4. The distance between its current position and gbest.

**Particle Swarm Optimization Algorithm:**

Lets us assume a few parameters first. You will find some new parameters, which I will describe later.

f: Objective function, Vi: Velocity of the particle or agent, A: Population of agents, W: Inertia weight, C1: cognitive constant, U1, U2: random numbers, C2: social constant, Xi: Position of the particle or agent, Pb: Personal Best, gb: global Best

The actual algorithm goes as below :

1. Create a ‘population’ of agents (particles) which is uniformly distributed over X.

2. Evaluate each particle’s position considering the objective function( say the below function).

z=f(x, y)=sin x^2+siny^2+sinxsiny

3. If a particle’s present position is better than its previous best position, update it.

4. Find the best particle (according to the particle’s last best places).

5. Update particles’ velocities.

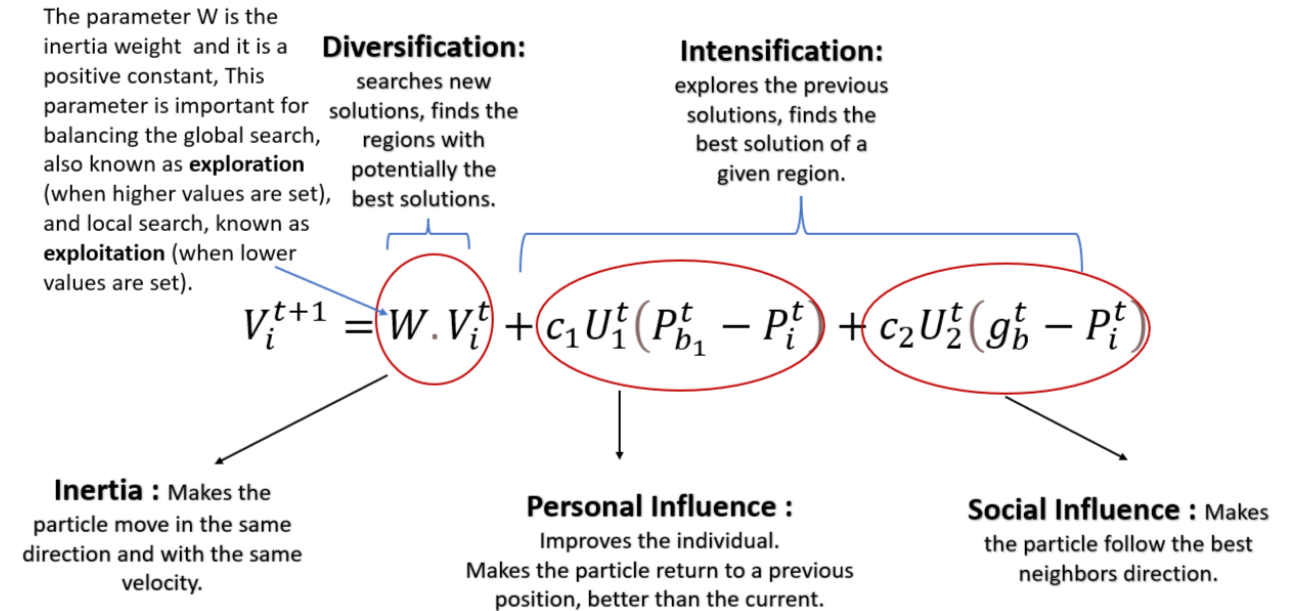
Image

6. Move particles to their new positions.

Image

7. Go to step 2 until the stopping criteria are satisfied.

## Analysis of the Particle Swarm Optimization Algorithm:



If W=1, the particle’s motion is entirely influenced by the previous motion, so the particle may keep going in the same direction. On the other hand, if 0≤W<1, such influence is reduced, which means that a particle instead goes to other regions in the search domain.

Pb1t And its current position Pit. It has been noticed that the idea behind this term is that as the particle gets more distant from the Pb1t (Personal Best) position, the difference (Pb1t-Pit ) Must increase; hence, this term increases, attracting the particle to its best own position. The parameter C1 existing as a product is a positive constant, and it is an individual-cognition parameter. It weighs the importance of the particle’s own previous experiences.

The other hyper-parameter which composes the product of the second term is U1t. It is a random value parameter with [0,1] range. This random parameter plays an essential role in avoiding premature convergences, increasing the most likely global optima.

The difference (gbt-Pit) Works as an attraction for the particles towards the best point until it’s found at t iteration. Likewise, C2 is also a social learning parameter, and it weighs the importance of the global learning of the swarm. And U2t plays precisely the same role as U1t.

In the case of C1=C2=0, all particles continue flying at their current speed until they hit the search space’s boundary.

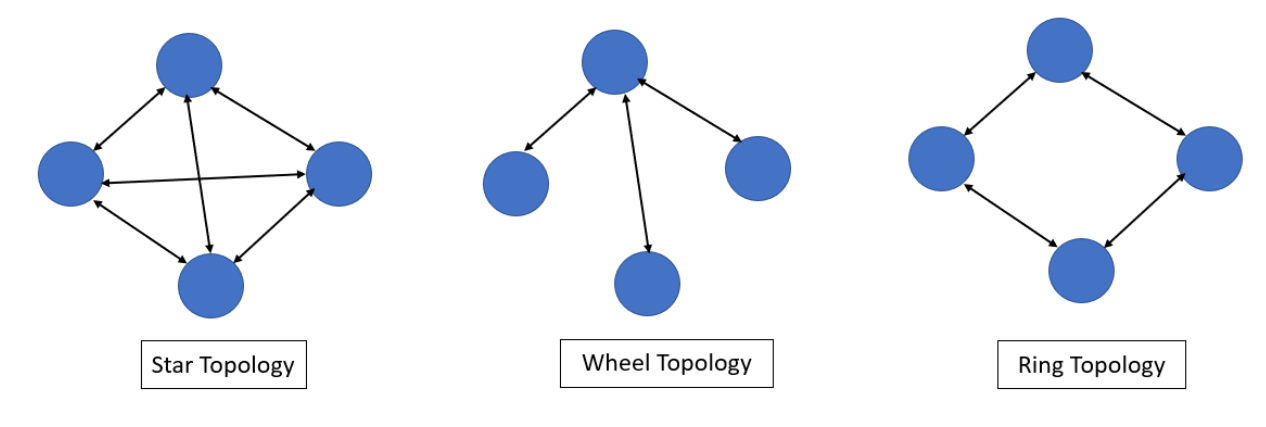
In cases C1>0 and C2=0, all particles are independent.

In cases C1>0 and C2=0, all particles are attracted to a single point in the entire swarm.

In case C1=C2≠0, all particles are attracted towards the average of pbest and gbest.

## Neighborhood Topologies:

A neighborhood must be defined for each particle. This neighborhood determines the extent of social interaction within the swarm and influences a particular particle’s movement. Less interaction occurs when the neighborhoods in the swarm are small. For small neighborhoods, the convergence will be slower, but it may improve the quality of solutions. The convergence will be faster for more prominent neighborhoods, but the risk that sometimes convergence occurs earlier.



For Star topology, each particle is connected with other particles. It leads to faster convergence than other topologies, Easy to find out gbest. But it can be biased to the pbest.

For wheel topology, only one particle connects to the others, and all information is communicated through this particle. This focal particle compares the best performance of all particles in the swarm, and adjusts its position towards the best performance particle. Then the new position of the focal particle is informed to all the particles.

For Ring Topology, when one particle finds the best result, it will make pass it to its immediate neighbors, and these two immediate neighbors pass it to their immediate neighbors until it reaches the last particle. Here the best result found is spread very slowly.

## Types of Particle Swarm Optimization:

Gradient PSO:

Efficiently exploration many local minimums can be combined with the ability of gradient-based local search algorithms to effectively calculate an accurate local minimum.

Hybrid PSO:

Combines PSO with other optimizers.

Newton’s method:

Netwon’s method also uses second derivative information to accerlate the convergence of the iterative process.

## Hybrid methods: coupling PSO deterministic methods

In general, optimization methods are divided into deterministic and heuristic. Deterministic methods aim to establish an iterative process involving a gradient, which, after a certain number of iterations, will converge to the minimum of the objective function. The iterative procedure of this type of method can be written as follows:

x ^(k+1) = x^ k + α^k(d^k)

where is the variable vector, is the step size, is the descent direction, and is the iteration number. The best that can be expected from any deterministic gradient method is its convergence to a stationary point, usually a local minimum. Heuristic methods, in contrast to deterministic methods, do not use the objective function gradient as a downward direction. Its goal is to mimic nature in order to find the minimum or maximum of the objective function by selecting, in an elegant and organized manner, the points where such a function will be calculated . Hybrid methods represent a combination of deterministic and heuristic methods in order to take advantage of both approaches. Hybrid methods typically use a heuristic method to locate the most likely region where the global minimum is. Once this region is determined, the hybrid formulation algorithm switches to a deterministic method to get closer and faster to the minimum point. Usually, the most common approach used for this formulation is using the heuristic method to generate good candidates for an optimal solution and then using the best point found as a start point for the deterministic methods in order to converge to local minimums. Numerous papers have been published over the last few years showing the efficiency and effectiveness of hybrid formulations. There are also a growing number of publications over the last decade regarding hybrid formulations for optimization . In this context, PSO algorithm can be combined with deterministic methods, increasing the chance of finding the function’s most likely global optimal. This chapter presents the three deterministic methods in which the PSO was coupled: conjugate gradient method, Newton’s method, and quasi-Newton method (BFGS). The formulation of each one of those is briefly presented in the following sections.

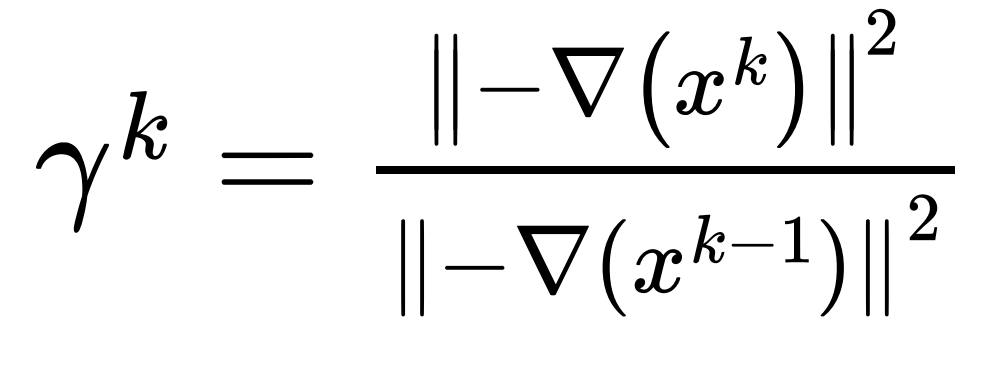
Conjugate gradient

The conjugate gradient method improves the convergence rate of the steepest descent method by choosing descending directions that are a linear combination of the gradient direction with the descending directions of previous iterations. Therefore, their equations are:

X^(k+1)=x^k+α^k(d^k)

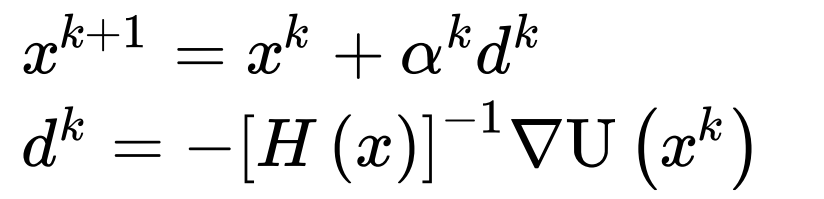
d^k=−∇(x^k)+γ^kd^(k−1)

where is the conjugation coefficient that acts by adjusting the size of the vectors. In the Fletcher-Reeves version, the conjugation coefficient is given by:



Netwon’s method:

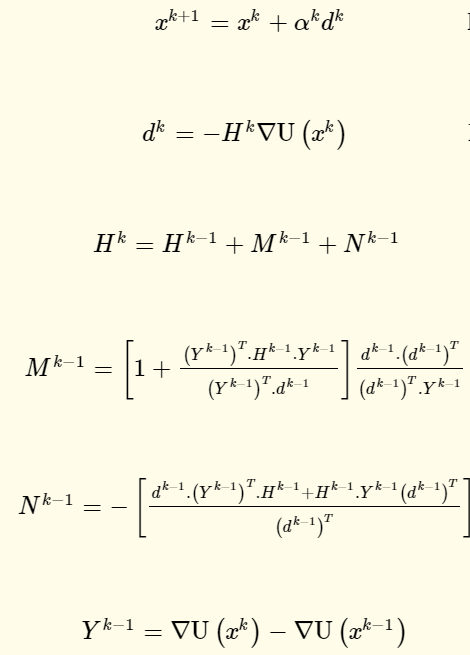
While the steepest descent and conjugate gradient methods use first derivative information, Newton’s method also uses second derivative information to accelerate the convergence of the iterative process. The algorithm used in this method is presented below:



where is the Hessian of the function. In general, this method requires few iterations to converge; however, it requires a matrix that grows with the size of the problem. If the estimate is far from the minimum, the Hessian matrix may be poorly conditioned. In addition, it involves inverting a matrix, which makes the method even more computationally expensive.

Quasi-Newton (BFGS):

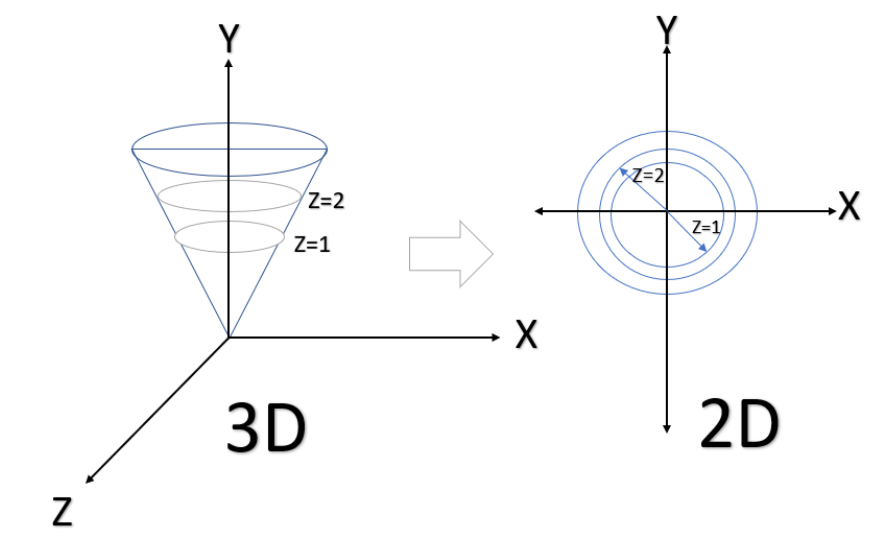
BFGS is a type of quasi-Newton method. It seeks to approximate the inverse of the Hessian using the function’s gradient information. This approximation is such that it does not involve second derivatives. Thus, this method has a slower convergence rate than Newton’s methods, although it is computationally faster. The algorithm is presented below:

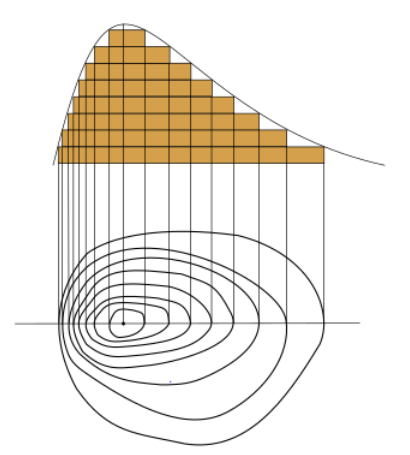


## Contour plot:

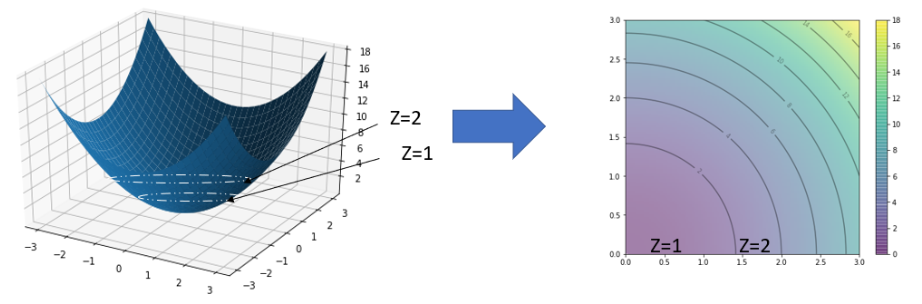
It is a graphical technique to represent 3 -Dimensional surface in 2- dimensional Plot using variable Z in the form of slices known as contours. I hope the below example can give you the intuition.

Let’s draw a graph of circle z=x2+y2 at fixed heights ‘z’ , z =1,2,3 etc.

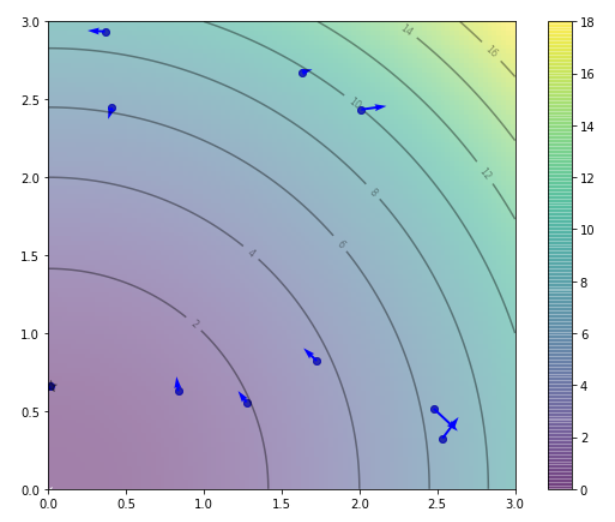




z=x2+y2 its actual plotting and the contour plotting will look like below:



Here we can see the function in the region of f(x,y). We can create ten particles at random locations in this region, together with a random velocity which is sampled over a normal distribution with mean 0 and standard deviation 0.1, as follows:



The actual outcome will be like :

PSO found best solution at f([0.01415657,0.65909248])=0.4346033028251361

Global optimal at f([0.0, 0.0])=0.0

Recent applications and challenges:

PSO can be applied to many types of problems in the most diverse areas of science. As an example, PSO has been used in healthcare in diagnosing problems of a type of leukemia through microscopic imaging . In the economic sciences, PSO has been used to test restricted and unrestricted risk investment portfolios to achieve optimal risk portfolios. In the engineering field, the applications are as diverse as possible. Optimization problems involving PSO can be found in the literature in order to increase the heat transfer of systems or even in algorithms to predict the heat transfer coefficient. In the field of thermodynamics, one can find papers involving the optimization of thermal systems such as diesel engine–organic Rankine cycle, hybrid diesel-ORC/photovoltaic system ,and integrated solar combined cycle power plants (ISCCs). PSO has also been used for geometric optimization problems in order to find the best system configurations that best fit the design constraints. In this context, we can mention studies involving optical-geometric optimization of solar concentrators and geometric optimization of radiative enclosures that satisfy temperature distribution and heat flow. After having numerous versions of PSO algorithm such as those mentioned in the first section, PSO is able to deal with a broad range of problems, from problems with a few numbers of goals and continuum variables to others with challenging multipurpose problems with many discreet and/or continuum variables. Besides its potential, the user must be aware that the PSO will only achieve appreciated results if one implements an objective function capable of reflecting all goals at once. To derive such a function may be a challenging task that should require a good understanding of the physical problem to be solved and the ability to abstract ideas into a mathematical equation as well. The problems presented in the fourth section of this work provide examples of objective functions capable of playing this role. Another challenge for one using PSO is how to handle the bounds of the search domain whenever a particle moves beyond it. Many popular strategies that had already been proposed are reviewed and compared for PSO classical version in. Those strategies may be reviewed and understood by PSO users so this person can pick up the one that best fits the optimization problem features.

## Advantages and disadvantages of Particle Swarm Optimization:

***Advantages :***

1. **Insensitive to scaling of design variables.**
2. **Easily parallelized for concurrent processing.**
3. **Derivative free.**
4. **Very few algorithm parameters.**
5. **A very efficient global search algorithm**.

***Disadvantages :***

**1.PSO’s optimum local searchability is weak**

## Conclusion & Overview:

1. PSO is a stochastic optimization technique based on the movement and intelligence of swarms.
2. In PSO, the concept of social interaction is used for solving a problem.
3. It uses a number of particles (agents) that constitute a swarm moving around in the search space, looking for the best solution.
4. Each particle in the swarm looks for its positional coordinates in the solution space, which are associated with the best solution that has been achieved so far by that particle. It is known as pbest or personal best.
5. Another best value known as gbest or global best is tracked by the PSO. This is the best possible value obtained so far by any particle in the neighborhood of that particle.

The most exciting part of PSO is there is a stable topology where particles are able to communicate with each other and increase the learning rate to achieve global optimum. The metaheuristic nature of this optimization algorithm gives us lots of opportunities as it optimizes a problem by iteratively trying to improve a candidate solution.