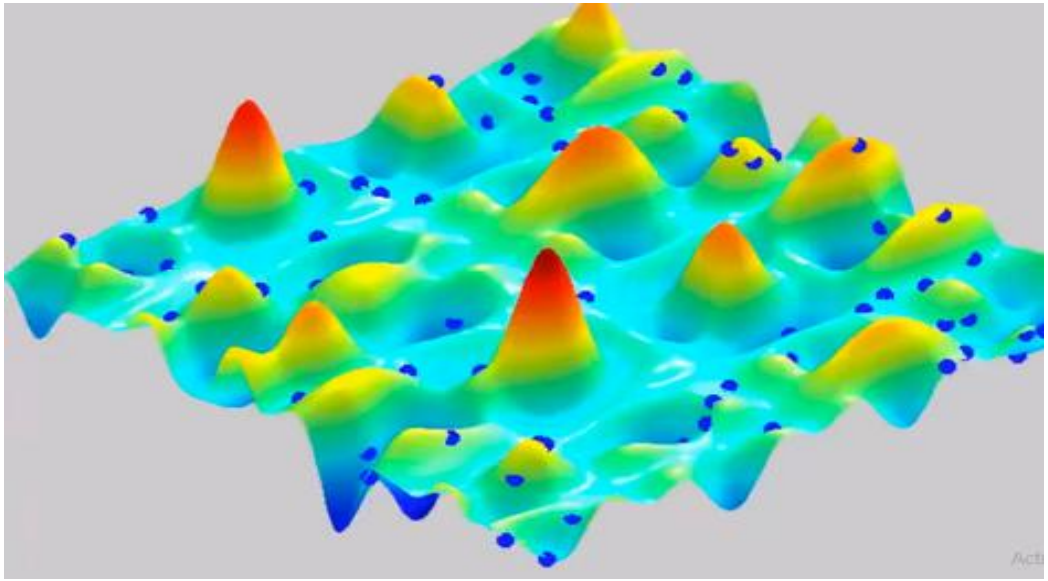
Introduction to PSO

- SA: Single solution based; GA: Population based Evolutionary computation type metaheuristic.
- Particle Swarm Optimization (PSO) is one of the most efficient optimization strategies for continuous nonlinear optimization problems.
- PSO is a **population based Swarm Intelligent (SI)** type metaheuristic.



PSO

- J. Kennedy *et al.* first proposed PSO that is inspired by the **collective social behaviors of swarm like movements of flocks of birds** or **schools of fish in search of food**s.
- In case of PSO, a particle (i.e. bird or fish) denotes a potential solution for the optimization problem.
- A set of particles is known as a **swarm**, where particles are initially distributed or positioned in random manner in *d*-dimensional search space of the problem.

Continue..

- Swarm is flown through the search space and the position of each particle is updated based on the experiences (fitness value at that point) of all neighbors particle including itself.
- Every particle is considered as **intelligent** as knows its own current fitness value, its own best value so far (locally best solution).
- They also knew the best fitness value of the whole swarm (globally best solution).
- Particles share information about Global fitness: Intelligent

PSO Algorithm

For, d-dimensional optimization problem, the position of i-th particle of a swarm (consist of N particles) at t-th iteration is given as $X_{i,d}^{t} = (x_{i1}, x_{i2}, \ldots., x_{id})$ and the velocity is represented by $V_{i,d}^{t} = (v_{i1}, v_{i2}, \ldots., v_{id})$.

Locally best solution by i-th particle at current iteration is given as $P_{best,i,d}^{t} = (P_{i1}, P_{i2}, \ldots., P_{id})$ and global best solution is denoted by $G_{best,d}^{t} = (G_1, G_2, \ldots., G_d)$. As iteration proceeds, the velocity and position of the particles are updated according to following rules.

$$V_{i,d}^{t+1} = V_{i,d}^{t} + C_1 * rand(1,d) \odot \left(P_{best,i,d}^{t} - X_{i,d}^{t}\right) + C_2 * rand(1,d) \odot \left(G_{best,d}^{t} - X_{i,d}^{t}\right) \quad (1)$$

$$X_{i,d}^{t+1} = V_{i,d}^{t+1} + X_{i,d}^{t} \quad (2)$$

Where C_1 and C_2 are called as acceleration constants, also named as cognitive learning rate and social learning rate respectively. $rand(1,d)$ is generate a d-dimensional array of random values within [0,1]. $\odot$ denotes element wise multiplication.

Memory Influence of PSO

In case of PSO, the velocity update formula i.e. new search direction is always effected by three components namely: current velocity $\left(v_{i,d}^{t}\right)$, current particle memory influences $(P_{best,i,d}^{t})$ and swarm memory influences$(G_{best,d}^{t})$. Particle memory influences is associated with current best position of a particle and swarm memory influences is associated with the global best position among all particles.

So, it may be considered that the particles memory influences is responsible for local search and swarm memory influences is responsible for global search of the optimization. It is interesting to note that random parameters are incorporated along with the global search or local search terms so that chance of sticking at local optima for the metaheuristics can be reduced.

Inertia Weight

However, one of the main reasons behind the success of a metaheuristic is a delicate balance between exploration and exploitation capability of the algorithm. To get a better control between the global and local search characteristics of PSO, Shi and Eberhart proposed a modified PSO where the velocity of each particle is updated based on inertia weight (ω). So, the velocity update rule is modified according to following equation.

$$V_{i,d}^{t+1} = V_{i,d}^{t} * \omega^{t} + C_1 * rand(1,d) \odot \left(P_{best,i,d}^{t} - X_{i,d}^{t}\right) + C_2 * rand(1,d) \odot \left(G_{best,d}^{t} - X_{i,d}^{t}\right) \quad (3)$$

The inertia weight (ω) of PSO is normally used for maintaining balance between exploration and exploitation capability. Recently, researchers give lots of attentions to the inertia weight parameter for improving the performance of original PSO. Lots of strategies were already proposed for updating inertia weight during the course of iteration.

Inertia Weight

The inertia weight (ω) of PSO is normally used for maintaining balance between exploration and exploitation capability. Recently, researchers give lots of attentions to the inertia weight parameter for improving the performance of original PSO. Lots of strategies were already proposed for updating inertia weight during the course of iteration.

In 1998, Shi and Eberhart proposed Constant Inertia Weight (CIW) technique where they claimed that large constant value of Inertia Weight is suitable for exploration while a small constant value of Inertia Weight is suitable a exploitation. So, CIW can be described using following equation.

$$\omega^t = Constant \qquad (4)$$

Further, in case of Random Inertia Weight (RIW), the value of inertia weight is selected in random manner and it is very efficient to find out the optima in a dynamic system. For RIW, the value of inertia weight is assigned using following equation

$$\omega^t = 0.5 + rand/2 \qquad (5)$$

where $rand$ is a function that generates random number within [0, 1]. Therefore, value of inertia weight is uniformly distributed over [0.5, 1] and this technique partially solve the problem of selection for constant of CIW.

Continue…

Linear Decreasing Inertia Weight (LDIW) is very popular and efficient technique in improving the fine-tuning characteristics of the PSO where the value of inertia weight is linearly depend on the iteration number. In case of LDIW, the value of ω is linearly decreased from an initial large value (ω_{max}) to a final small value (ω_{min}) according to the following equation:

$$\omega^t = \omega_{max} - \left\{\frac{\omega_{max} - \omega_{min}}{t_{max}}\right\} \times t \qquad (6)$$

Where t is iteration index and t_{max} denotes maximum number of iteration. LDIW has better efficiency over the others technique due to smooth transition from initial global search to local search during iteration process.