**SCOA MINI PROJECT REPORT**

**Title**: Travelling Salesman Problem using Particle Swarm Optimization

**Domain**: Particle Swarm Optimization (PSO)

**Project Partners**:

Rakshit Pensalwar(403059), Chakor Pandkar(403053), Shubham Patil(403058)

**Brief Explanation of Project:**

This project demonstrates how to use Particle Swarm Optimization (PSO) algorithm to solve travelling salesman problem (TSP) for Expert System.

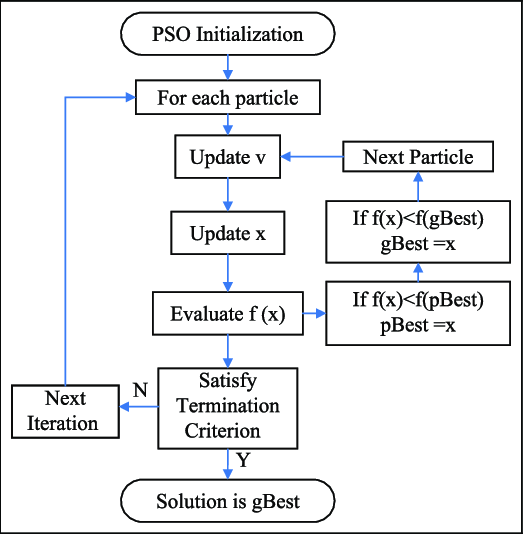
To solve the TSP using PSO we need to:

(1) Create a ‘population’ of agents (called particles) uniformly distributed over X (feasible region) and evaluate each particles’ position according to the objective function,

(2) Update particles’ velocities,

(3) Move particles to their new positions,

(4) If a particles’ current position is better than its previous best position, then update it. By doing this, you can be sure to find the best strategy for solution of TSP.

**PSO Algorithm Working Flowchart**:

**Brief Explanation of Algorithms**:

1. **PARTICLE SWARM OPTIMIZATION ALGORITHM**: Particle swarm optimization, PSO, is an evolutionary computation technique inspired in the behavior of bird flocks. PSO is population-based algorithm that performs a parallel search on a space of solutions. In the optimization context, several solutions of a given problem constitute a population (the swarm). Each solution is seen as a social organism, also called as particle. The method attempts to imitate the behavior of real creatures making the particles “fly” over a solution space.

In the classical PSO algorithm, each particle

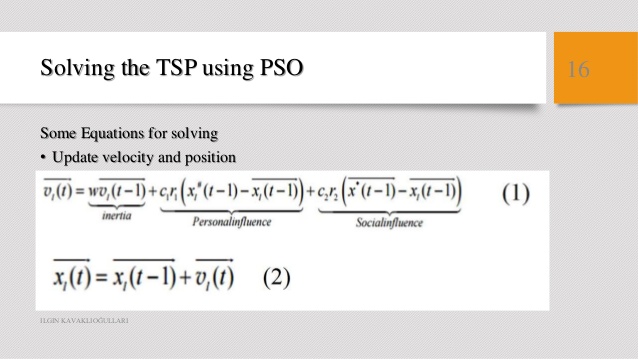
• has a position and a velocity

• knows its own position and the value associated with it

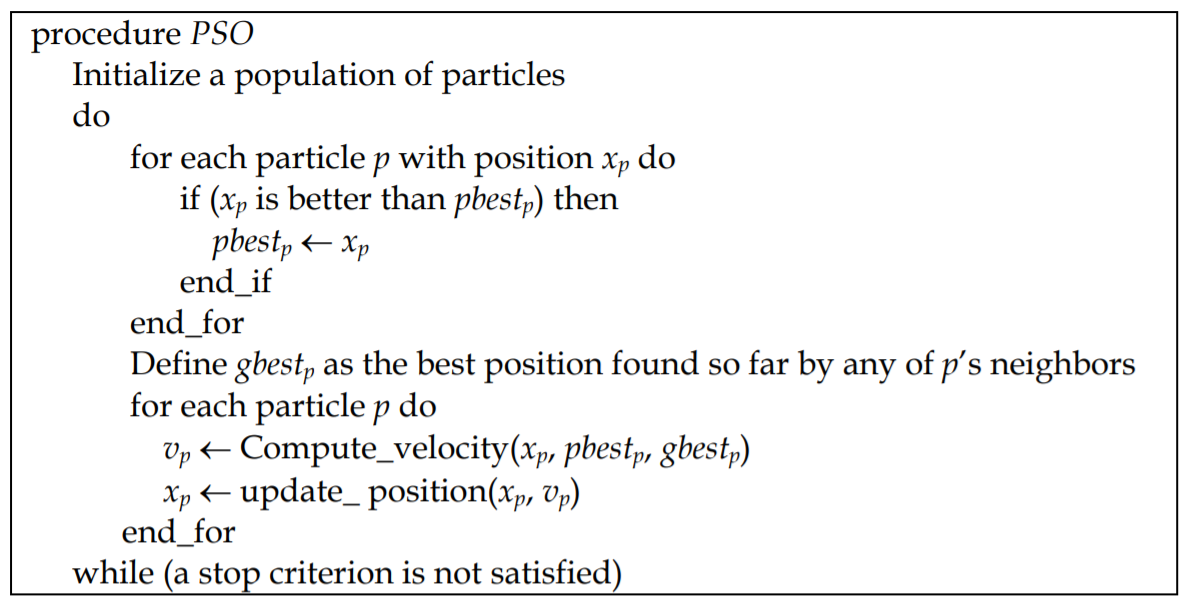
• knows the best position it has ever achieved, and the value associated with it

• knows its neighbors, their best positions and their values

The best position a given particle has ever achieved is called **pbest**. In some versions of particle swarm algorithm the particles also track the best position achieved so far by any particle of the swarm. This position is called **gbest**. By changing their velocities with individualistic moves or toward **pbest** and **gbest**, the particles change their positions.



**PSO Algorithm**



**PLOTS GRAPHS AND EXPLAINATION:**

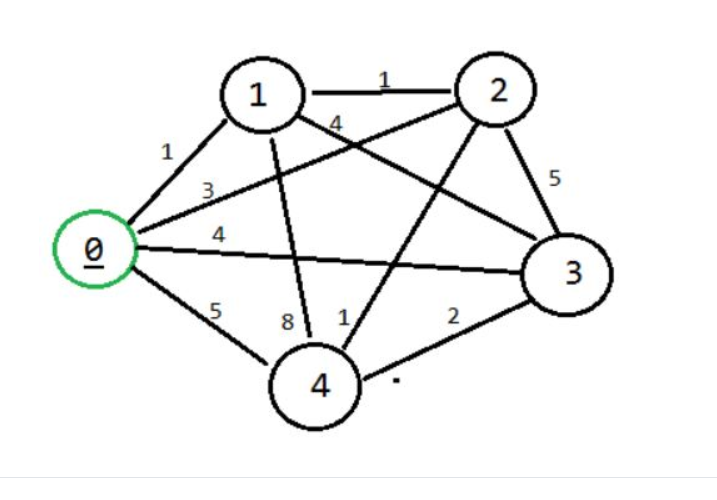
Libraries used are attrgetter, random and copy

Number of vertices: 5

maximum number of iterations: 100

Population size: 10 : = N\*(N-1)/2

(pbest - x(t-1)): 0.9

(gbest - x(t-1)): 1

Initial solution each particle:

[0, 1, 3, 4, 2] = 11 [0, 4, 2, 3, 1] = 16 [0, 2, 4, 1, 3] = 20

[0, 2, 3, 1, 4] = 25 [0, 3, 4, 2, 1] = 9 [0, 2, 3, 4, 1] = 19

[0, 1, 4, 3, 2] = 19 [0, 2, 1, 3, 4] = 15 [0, 3, 1, 2, 4] = 15

Particles:

pbest: [0, 1, 2, 4, 3] | cost pbest: 9 | current solution: [0, 1, 2, 4, 3] | cost current solution: 9

pbest: [0, 3, 4, 2, 1] | cost pbest: 9 | current solution: [0, 3, 4, 2, 1] | cost current solution: 9

pbest: [0, 1, 2, 4, 3] | cost pbest: 9 | current solution: [0, 1, 2, 4, 3] | cost current solution: 9

pbest: [0, 3, 4, 2, 1] | cost pbest: 9 | current solution: [0, 1, 2, 4, 3] | cost current solution: 9

gbest: [0, 1, 2, 4, 3] | cost: 9

Hence by using PSO we can find out optimal solution for Traveling salesman problem.

**SOFTWARE AND HARDWARE USED:**

Software: Jupyter Notebook (IDE), Python 3(Programming Language)

Hardware: 64-bit Operating System ( Windows 10 )

**APPLICATION:**

**-**Find the shortest possible route that visits each city exactly once and returns to the origin city => Hamiltonian cycle

**-**Posed such computational complexity that any programmable efforts to solve such problems would grow super-polynomially with the problem size

Can be used in:

• Transportation: school bus routes, service calls, delivering meals

• Manufacturing: an industrial robot that drills holes in printed circuit boards

• VLSI (microchip) layout

• Communication: planning new telecommunication networks

**FUTURE SCOPE:**

In future, other methods to compose velocities and heuristics hybridization under the PSO framework will be investigated. Another idea to be explored in future researches is variable velocities. The proposed approach will be applied to the Generalized TSP and to the Bi-objective TSP.

**CONCLUSION:**

This project summarized to develop PSO algorithms for the TSP. Many of the PSO algorithms presented previously for the investigated problem do not tackle large instances and present results far from the best known heuristic solutions obtained by effective algorithms.

The new approach, first introduced by Goldbarg et al. (2006a), differentiates velocity operators according to the type of move the particle does. Additionally, methods to compose the velocity operators were proposed.

Computational experiments with instances up to 7397 cities were presented. The results of those experiments show that the proposed method produces high quality solutions, when compared with four effective heuristics designed specifically for the investigated problem.