Assignment SCOA-8

Title: Apply the Particle swarm optimization for Travelling Salesman Problem

Aim: Implementation of Particle Swarm Optimization for Travelling Salesman Problem

Objective:

Student will learn:

i) Understand the particle swarm optimization.

ii) Implementation of travelling salesman problem with particle swarm optimization.

Software and Hardware Requirements:

4GB RAM, Processor i3 and above, 500GB HDD, OS Fedora / Ubuntu, Text Editor, Jupiter Notebook, Python 2.7 or 3.6

Theory:

Travelling Salesman Problem:

This is a rendition of the classic Traveling Salesman Problem, where the shortest tour needs to be found among all cites without visiting the same one twice. The algorithm works out the minimum Cartesian distance through eight cities named 0, 1, 2, 3, 4, 5, 6 and 7. There are 8^8 (or, 16,777,216) possible combinations, but this algorithm can find them in less than 8^3, and sometimes less than 8^2!

The cities' locations are (30, 5), (40, 10), (40, 20), (29, 25), (19, 25), (9, 19), (9, 9), (20, 5). This is roughly a circle; something easy to see on a graph and check the algorithm's solutions with. All manner of combinations using eight digit strings of 0 through 7 are explored to find the one that represents the shortest distance (i.e.; 01234567). By hand, the solution to be about 86.6299, which is the target value of the algorithm. To simplify this example, there is no starting or ending point, and it doesn't matter which direction the tour travels in (i.e.; forward or backward). For instance, a solution that looks like 34567012 is valid because it travels forward from 3 and around to 2. The solution 76543210 is equally valid; it just goes backward from 7 to 0.

Steps in PSO-TSP

Step 1: First adds global worst to the global variables.

Step 2: The velocity score is calculated using the global worst, defining velocity as the measure of how bad each particle is doing (as opposed to how good).

Step 3: Modifying data is done by swapping digits within each particle own data set. The amount of swapping depends on how bad it's doing (i.e.; velocity).

Step 4: Particles are slowly pushed towards the global best by copying pieces of the next best particle's data (single-sighted topology).

Step 5: Swapping and copying are done to all particles except the global best.

Pseudo code:

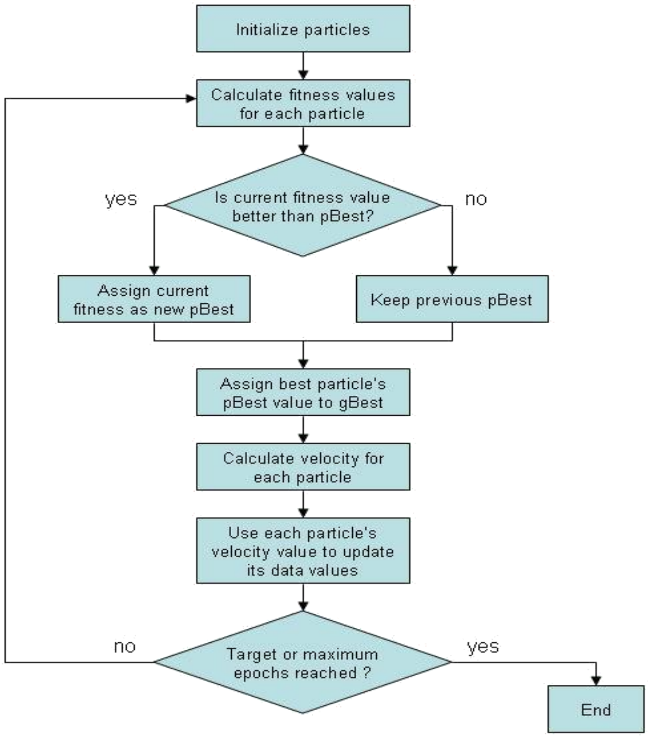
```
For each particle
{
    Initialize particle
}

Do until maximum iterations or minimum error criteria
{
    For each particle
    {
        Calculate Data fitness value
        If the fitness value is better than pBest
        {
            Set pBest = current fitness value
        }
        If pBest is better than gBest
        {
            Set gBest = pBest
        }
    }

    For each particle
    {
        Calculate particle Velocity
        Use gBest and Velocity to update particle Data
```

```
}
```

Flowchart:



Conclusion:

Compared with genetic algorithms (GAs), the information sharing mechanism in PSO is significantly different. However, PSO does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. They also have memory, which is important to the algorithm.