**MealPal Particle Swarm Optimization**

**Application of Particle Swarm Optimization to solve Transportation problem of the business MEALPAL**

**Problem Statement:** In a real-world scenario, it is a cumbersome task to manage the delivery system efficiently. A good transportation system is the one which establishes the least cost, most efficient transportation connection network while honoring all customer’s service requirements. Achieving such efficiency in a transportation system is always a problem.

For example, for my business MEALPAL which prepares food for the customers and that food is being delivered by a driver to various customers located at different location. The goal is to find the shortest and fastest path for the driver to deliver the food to the corresponding resulting in maximizing the profit for the organization.

**My Solution:**

For effective transportation involves an efficient and faster connection to the destination which will be achieved by implementing particle swarm optimization. This algorithm will give the shortest path for a given transportation system. It is a single source and multiple destination problem.

**I have refereed the following paper written on Particle Swarm Optimization.**

**Link -** <http://digitalcommons.uri.edu/cgi/viewcontent.cgi?article=1264&context=oa_diss>

Since particle swarm optimization is a np hard problem therefore there is no fixed solution for this. Its just how optimized is your solution. To build my algorithm, the core logic resides in particle and swarm.

* **Swarm**
* **Particle**

In my case, I have fixed the number of customers to be seven and number of vehicle 1.

Swarm consists of gbestValue, gBestArray and the list of Particle. And each Particle will have a PBestArray, PBestValue, x(current solution), fitnessValue and velocity in which the particle is travelling in the swarm.

I make one array of the size of the number of customer.

NoOfCustomer 🡪 7

NoOfCustomer 🡪 3

Say, array= {1,2,3,4,5,6,7}; then for the first iteration my particles and their velocity are:

F(x1) – shuffleArray(array) 🡪 {2,4,5,6,1,7,3}

F(x2) 🡪 shuffleArray(array) 🡪 {3,1,2,4,5,6,7}

F(x3) 🡪 shuffleArray(array) 🡪 {7,6,5,4,1,2,3}

And their respective velocities will be random generated list of size 7 from (0-7).

V1 🡪 {0.34,0.56,1.4,5.6,6.6,3.2}

V2 🡪 {0.24,0.16,2.4,3.6,7,3.2}

V3 🡪 {4.34,2.56,1.4,5.6,6.6,3.2}

For the first iteration, fitness value will be maximum value of the Integer all these particle with the best solution will be set and random velocities will be set.

Iteration=10

For each **iteration** I calculated {

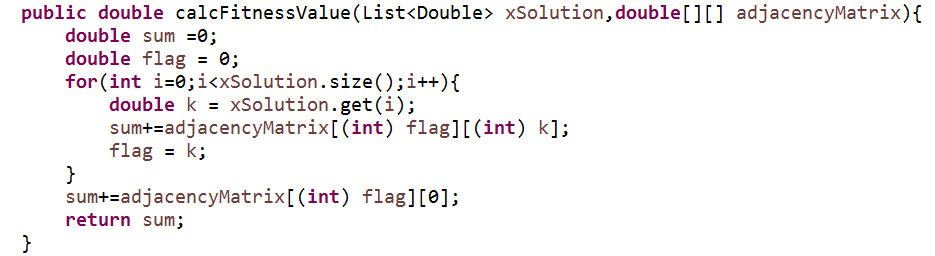
FitnessValue,

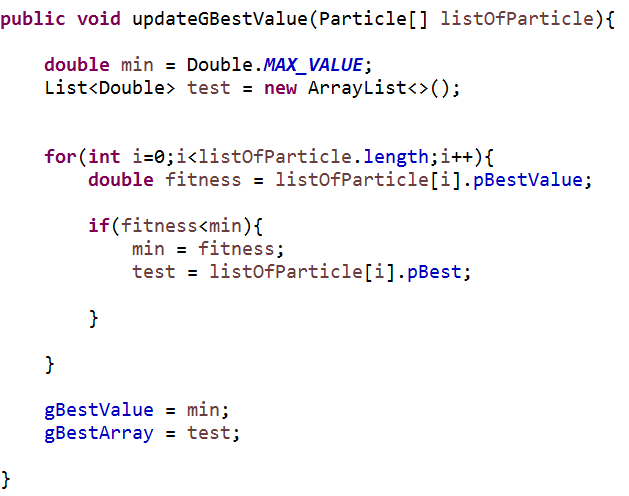
UpdatedGBest and UdpatedGBestArray

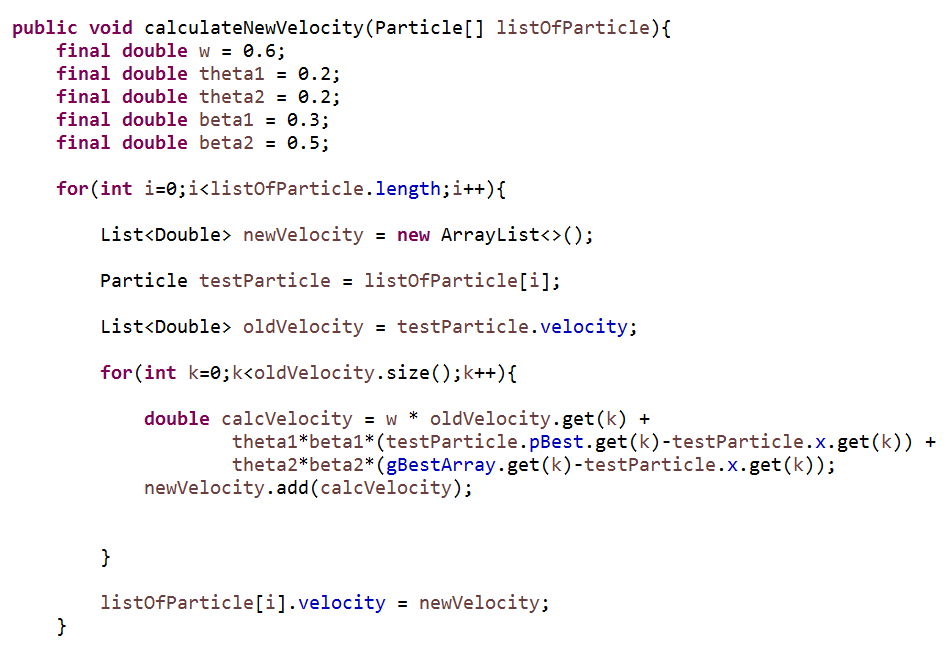
Calculate new Velocity

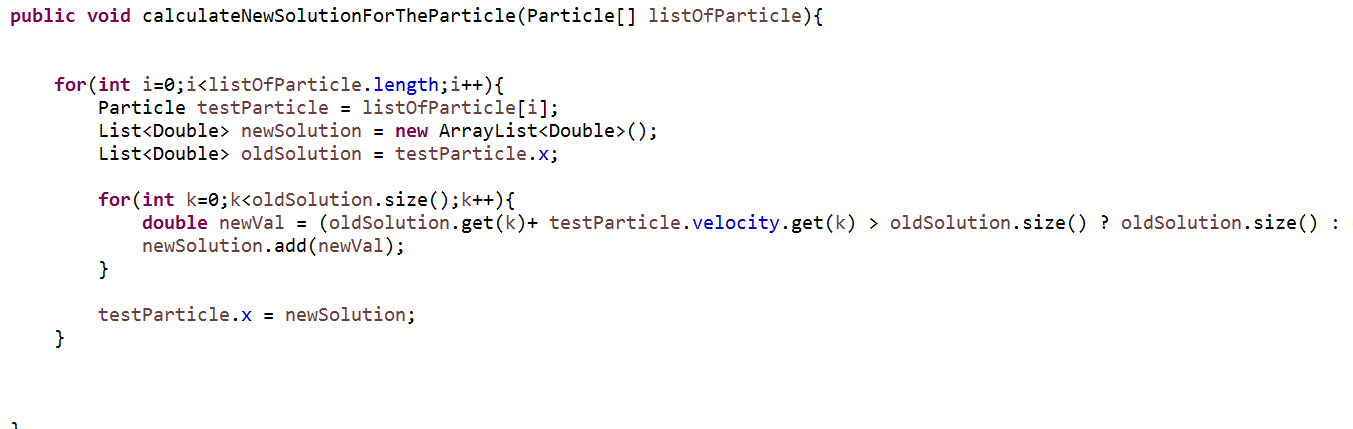
MakeNewXSolution of the particle.

}









After 10 iteration, the value of the gBestArray will the solution array.

Say gbestArray after 10 iteration is – { 2.4,5.6,4.4,1.3,6.8,7,7}-🡪 the lowest value index will be served first. That is to say 3 index customer will be served first, 0th index customer and so on so forth.

