TSP OPTIMIZATION

(using Particle Swarm Optimization and Ant Colony Optimization)

Enrollment No. - 9913103672
Name - Gaurav Chauhan
Name of supervisor(s) - Dr. Mukesh Saraswat

May - 2017

Submitted in partial fulfillment of the Degree of
Bachelor of Technology B. Tech
in

Computer Science Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

The Travelling Salesman Problem describes a salesman who must travel between N cities. The order in which he does so is something he does not care about, as long as he visits each one during his trip, and finishes where he was at first. Each city is connected to other close by cities, or nodes, by airplane, or by road or railway. Each of those links between the cities has one or more weights (or the cost) attached. The cost describes how "difficult" it is to traverse this edge on the graph, and may be given, for example, by the cost of an airplane ticket or train ticket, or perhaps by the length of the edge, or time required to complete the traversal. The salesman wants to keep both the travel costs, as well as the distance he travels as low as possible.

Madeira Island (west of Morocco in Atlantic ocean)

Find a tour with minimum distance, visiting every city only once

Traveling Salesman Problem

- Theoretical interest
  - Basic NP-complete problem (➔ not easy)

  (only to traverse 10 cities , the solution space , i.e, the total number of possible solutions is 181440)
  - 1993-2001: +150 articles about TSP in INFORMS & Decision Sciences databases
- Practical interest
  - Vehicle Routing Problem
  - Genetic/Radiated Hybrid Mapping Problem
    - NCBI/Concorde, Carthagène, ...

Variants

- Euclidean Traveling Salesman Selection Problem
- Asymmetric Traveling Salesman Problem
- Symmetric Wandering Salesman Problem
- Selective Traveling Salesman Problem
- TSP with distances 1 and 2, TSP(1,2)
- K-template Traveling Salesman Problem
- Circulant Traveling Salesman Problem
- On-line Traveling Salesman Problem
- Maximum Latency TSP

- Minimum Latency Problem
- Max TSP
- Traveling Preacher Problem
- Bipartite TSP
- Remote TSP
- Precedence-Constrained TSP
- Exact TSP
- The Tour Cover problem
- Time-dependent TSP
- The Angular-Metric Traveling Salesman Problem

Introduction to the PSO: **Origins**

- Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling.

Introduction to the PSO: Concept

- Uses a number of agents (**particles**) that constitute a swarm moving around in the search space looking for the best solution

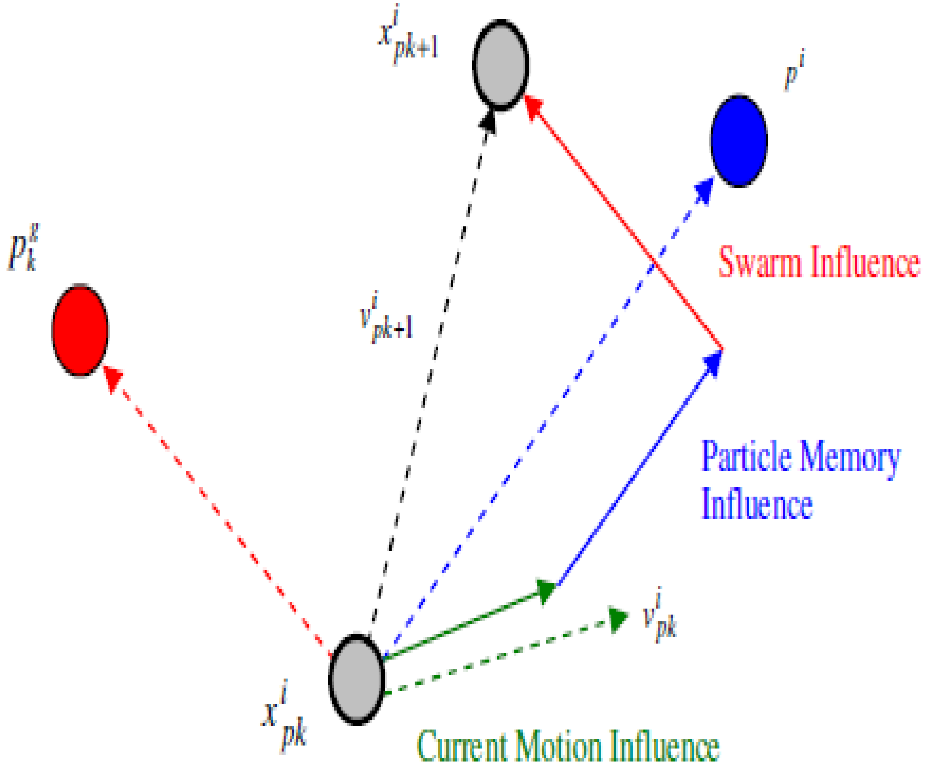
- Each particle in search space adjusts its “flying” according to its own flying experience as well as the flying experience of other particles

Introduction to the PSO: <u>**Concept**</u>

- Collection of flying particles (swarm) - Changing solutions
- Search area - Possible solutions
- Movement towards a promising area to get the global optimum
- Each particle keeps track:
  - its best solution, personal best, *<u>pbest</u>*
  - the best value of any particle, global best, *<u>gbest</u>*

Introduction to the PSO: Concept

- Each particle modifies its position according to:
  - its current position
  - its current velocity
  - the distance between its current position and *pbest*
  - the distance between its current position and *gbest*



Introduction to the PSO: Concept

Classical equation for Velocity is :

$$V_i = V_i \oplus \alpha^*(P_{besti} - X_i) \oplus \beta^*(P_{gbest} - X_i)$$

Velocity equation in my proposed approach:

Vi = V_MAX * (Gbest /Pbest)

The particles are updated as :

$$X_i = X_i + V_i.$$

Vi = is the velocity value

α , β is a uniform random value in the interval [0,1]

Pbesti = is the best postion of ith item

Gbest = is the best position of all items

Xi = is the current position of particle

V_MAX is max value of velocity

Output screen:

```
run:
Route 0:Distance: 189.6004043835619
Route 1:Distance: 174.09680303620968
Route 2:Distance: 143.19731967572307
Route 3:Distance: 163.47434080800855
Route 4:Distance: 155.18977281321816
Route 5:Distance: 171.9497079449014
Route 6:Distance: 136.80083060951245
Route 7:Distance: 142.66582314248245
Route 8:Distance: 179.439734861246
Route 9:Distance: 168.82036152889967
gbest:136.80083060951245
Changes for particle 6: 3
Changes for particle 4: 1
Changes for particle 7: 4
Changes for particle 3: 9
Changes for particle 2: 8
Changes for particle 5: 1
Changes for particle 7: 3
Changes for particle 0: 6
Changes for particle 4: 9
epoch number: 0
```

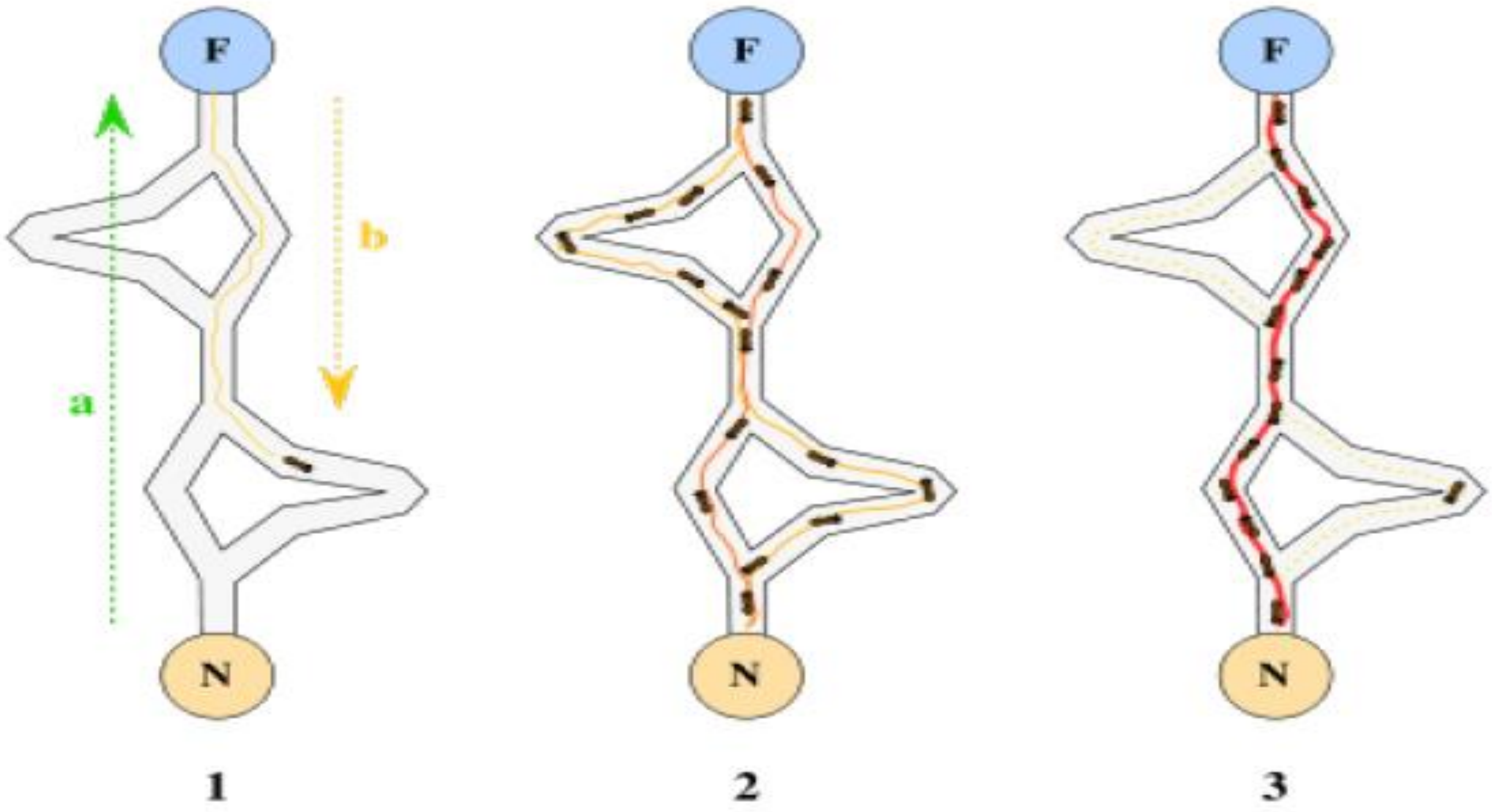
What is Ant Colony Optimization?

- Probabilistic technique.
- Searching for optimal path in the graph based on behaviour of ants seeking a path between their colony and source of food.
- Meta-heuristic optimization

ACO Concept:

Overview of the Concepts:

- Ants navigate from nest to food source. Ants are blind!
- Shortest path is discovered via pheromone trails.
- Each ant moves at random
- Pheromone is deposited on path
- More pheromone on path increases probability of path being followed


F
a
b
N
F
N
F
N
1
2
3

ACO System

Overview of the System:

- Virtual trail accumulated on path segments
- Path selected at random based on amount of "trail" present on possible paths from starting node
- Ant reaches next node, selects next path
- Continues until reaches starting node
- Finished tour is a solution.
- Tour is analyzed for optimality

Meta-heuristic

- Heuristic method for solving a very general class of computational problems by combining user-given heuristics in the hope of obtaining a more efficient procedure.
- ACO is meta-heuristic
- Soft computing technique for solving hard discrete optimization problems

ACO - Construct Ant Solutions

An ant will move from node i to node j with probability

$$p_{i,j} = \frac{(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}{\sum(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}$$

where

$\tau_{i,j}$ is the amount of pheromone on edge i, j

α is a parameter to control the influence of $\tau_{i,j}$

$\eta_{i,j}$ is the desirability of edge i, j (typically $1/d_{i,j}$)

β is a parameter to control the influence of $\eta_{i,j}$

ACO - Pheromone Update

Amount of pheromone is updated according to the equation

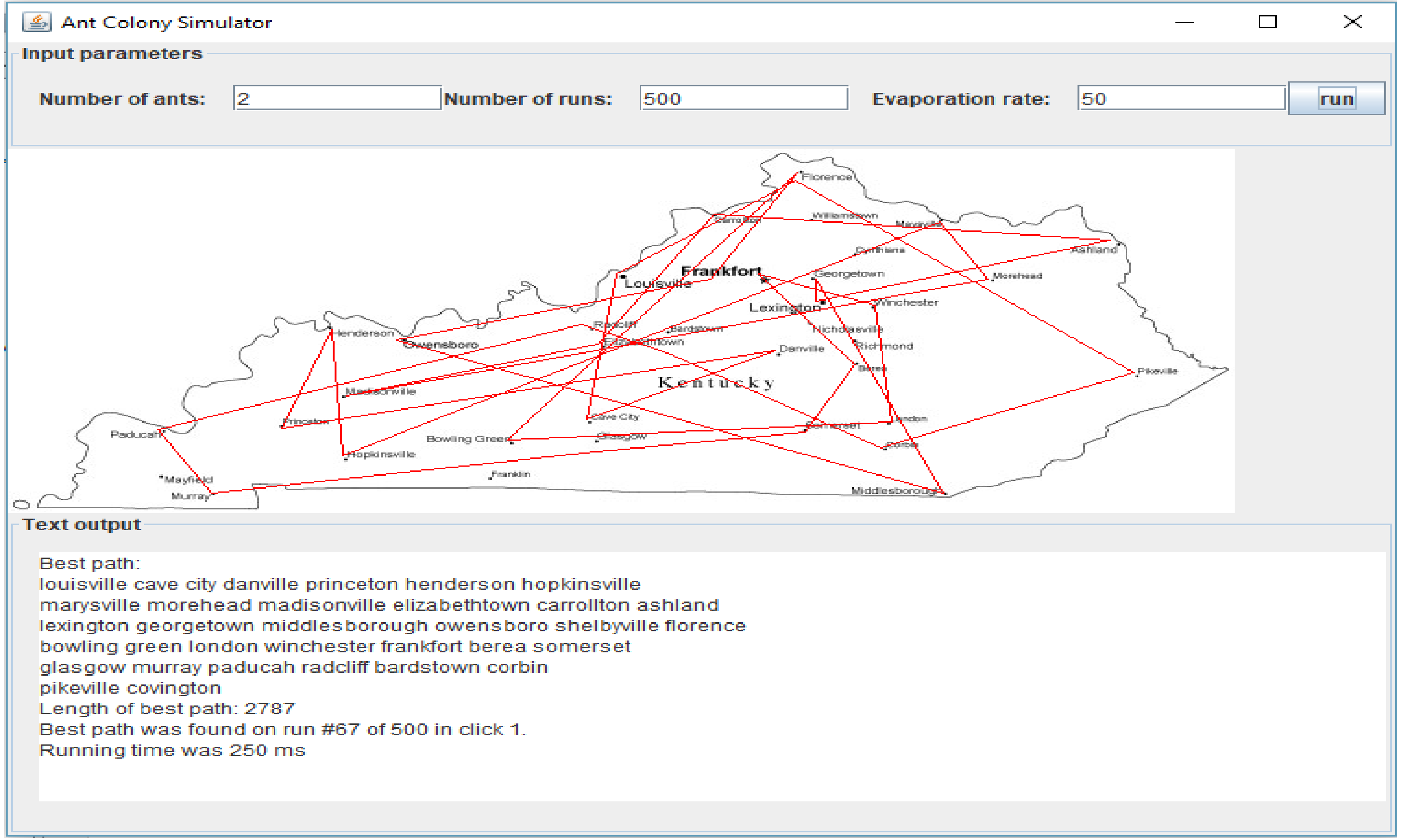
$$\tau_{i,j} = (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j}$$

where
$\tau_{i,j}$ is the amount of pheromone on a given edge i, j
ρ is the rate of pheromone evaporation
$\Delta\tau_{i,j}$ is the amount of pheromone deposited, typically given by

$$\Delta\tau_{i,j}^{k} = \begin{cases} 1/L_k & \text{if ant } k \text{ travels on edge } i, j \\ 0 & \text{otherwise} \end{cases}$$

where L_k is the cost of the k^{th} ant's tour (typically length).

Output screen:



Advantages of ACO

- Inherent parallelism
- Positive Feedback accounts for rapid discovery of good solutions
- Efficient for Traveling Salesman Problem and similar problems
- Can be used in dynamic applications (adapts to changes such as new distances, etc)

Disadvantages of ACO

- Theoretical analysis is difficult
- Sequences of random decisions (not independent)
- Probability distribution changes by iteration
- Research is experimental rather than theoretical
- Time to convergence uncertain (but convergence is gauranteed!)

Conclusion:

In the comparative study of implementation of both of these algorithms on travelling salesman problem , Ant Colony Optimization was found to give better result than the particle swarm optimization because when both algorithms were executed with same number of particles (10) , the running time of both algorithms to find the shortest route was almost identical , while ACO was working on dataset of 40 cities and PSO was working on dataset of only 8 cities , which concludes that Ant Colony Optimization(ACO) algorithm is way much faster that PSO and hence is a better approach for in the process of solving the Travelling Salesman Problem.