Shikhar Mahajan

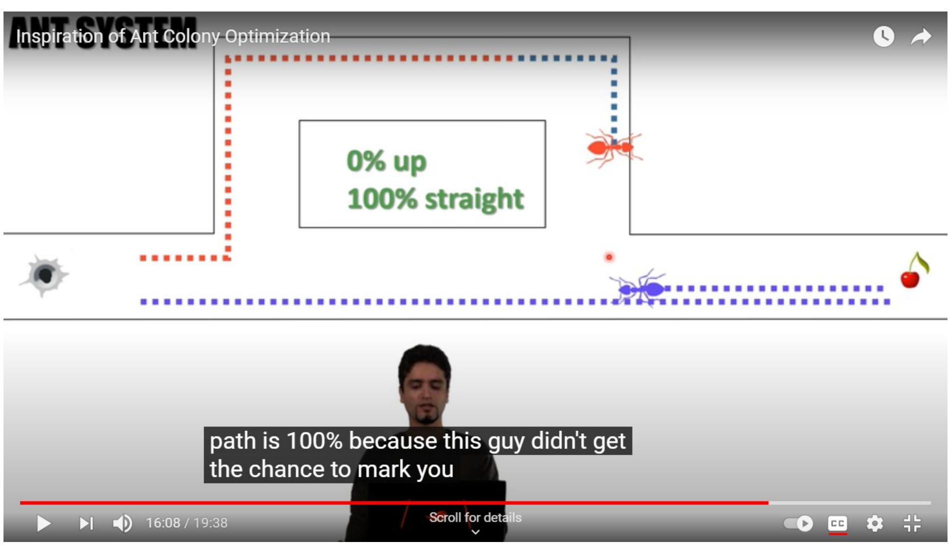
ACO - ant colony optimization

Inspiration comes from Stigmergy
Stigmergy is a mechanism of indirect coordination, through the environment, between agents or actions. The principle is that the trace left in the environment by an individual action stimulates the performance of a succeeding action by the same or different agent.

A pheromone is a secreted or excreted chemical factor that triggers a social response in members of the same species. Pheromones are chemicals capable of acting like hormones outside the body of the secreting individual, to impact the behavior of the receiving individuals.

Ant releases pheromone and the path which has higher content of pheromone is chosen by the ant.by this way they know about the shortest path. If there is the same level of pheromone then probability of choosing a path is ½.

Description ACO:
Ant Colony Optimization (ACO) was first used by M. Dorigo et al. to solve discrete optimization problems, in the late 1980s. ACO is based on the behavior of ants for finding food. Ants deposit pheromone as they walk and find their route by walking along the pheromone deposition. Density of pheromone deposition increases as ants walk back to the source with food. Pheromone deposition on the way back is dependent on quality and quantity of food taken to the source point. Pheromone deposition /evaporation is directly related to the number of ants traveling on that path. Ants find the optimal path by following maximum pheromone deposition



$$\Delta\tau_{i,j}^{k} = \begin{cases} \dfrac{1}{L_k} & k^{th} \textit{ant travels on the edge } i,j \\ 0 & \textit{otherwise} \end{cases}$$

$$\tau_{i,j}^{k} = \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} \quad \textbf{Without vaporization}$$

$$\tau_{i,j}^{k} = (1-\rho)\,\tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} \quad \textbf{With vaporization}$$

CALCULATING THE PROBABILITIES

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

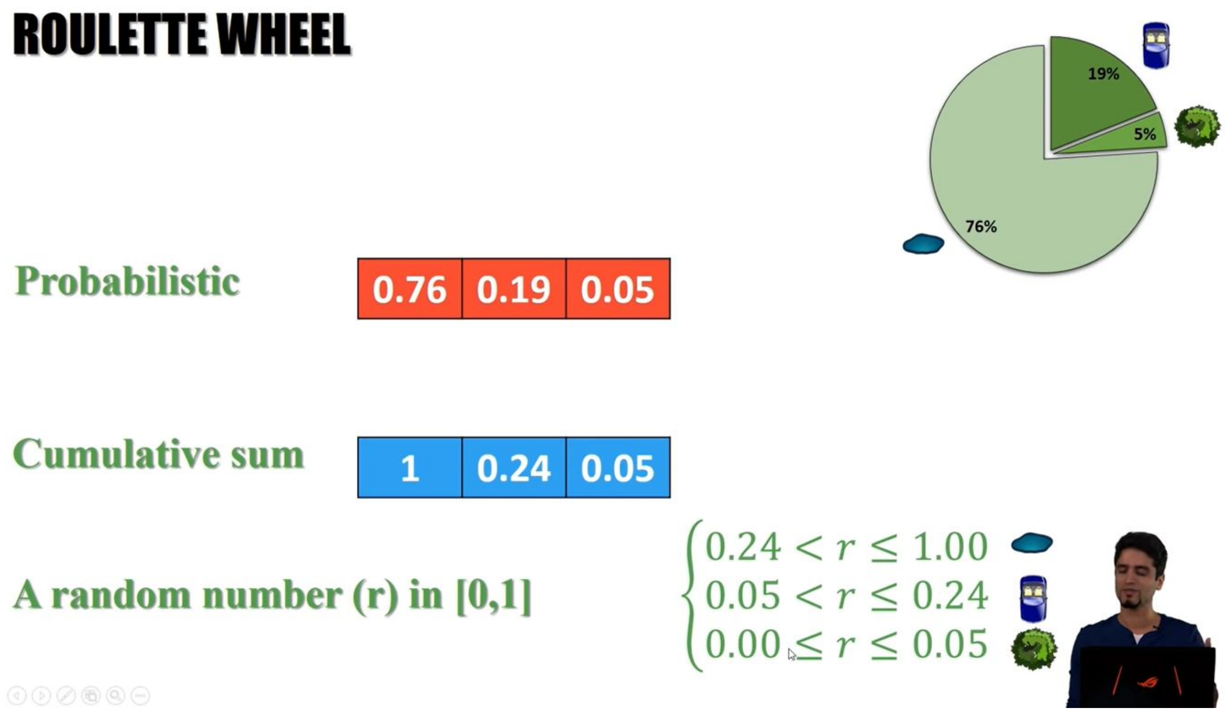
$$where\colon\ \eta_{i,j} = \frac{1}{\mathrm{L}_{i,j}}$$

- Tau represent pheromone level.
- Pi,j is probability to be calculated
- Eta is the quality/cost of i,j path(L is the cost)
- Alpha and beta are the impact of pheromone and quality of path respectively.
- We will be given 2 matrices

cost matrix and pheromone matrix

- Roh is the amount of evaporation of pheromone,

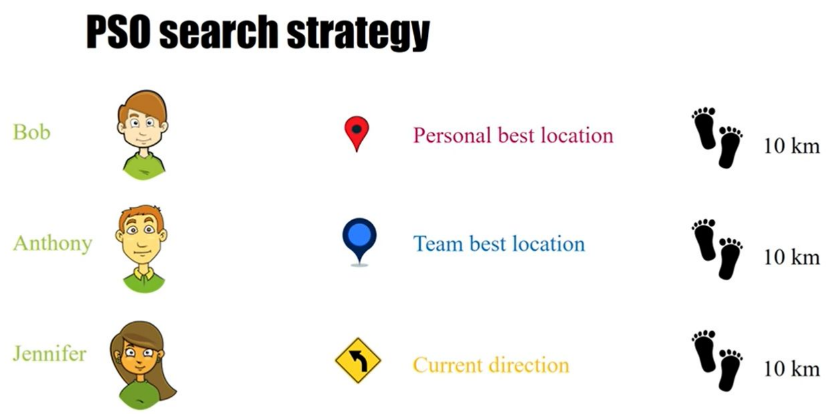
=0 means no evaporation =1 means full evaporation



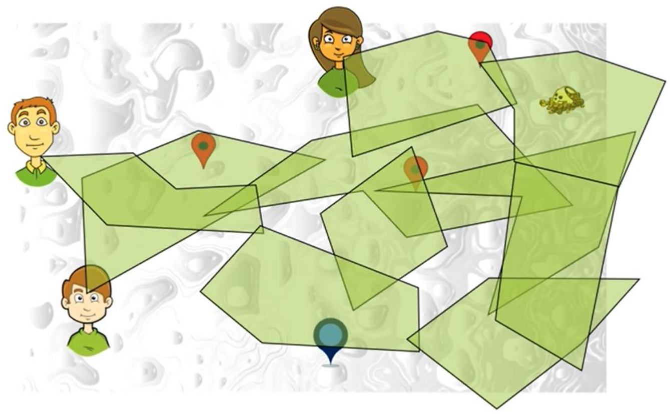
Roulette wheel is used after computing the probability for which path to be chosen. Probability vector is converted to a cumulative vector by adding current value to the value on it's right side.

PSO- particle swarm optimization.

Description PSO:
PSO is a heuristic global optimization technique. Particle swarm optimization (PSO), originally developed by Kennedy and Eberhart. In PSO each possible solution is called particle and the group of these possible solutions is called population. PSO is based on how a group of birds will randomly search for food. Birds don't know where exactly the food is, so they follow the bird which is nearest to the food. Each bird is known as a particle and each particle has its fitness function (here it is square of error). A group of particles is known as swarm. In each iteration, first the best solution found in the swarm is stored and is known as pbest. Another best solution is also stored, which is the best solution found so far. Values of pbest and gbest are updated.



PSO search strategy



PSO search strategy

$$\overrightarrow{V_i^{d+1}} = 2r_1 \boxed{\overrightarrow{V_i^{d}}} + 2r_2 \left(\boxed{\overrightarrow{P_i^{d}}} - \overrightarrow{X_i^{d}} \right) + 2r_3 \left(\boxed{\overrightarrow{G^{d}}} - \overrightarrow{X_i^{d}} \right)$$

Next velocity (tomorrow)

Current velocity (today)

Personal best solution

Global best solution

Distance to the personal best

Distance to the global best

PSO search strategy

$$\overrightarrow{X_i^{d+1}} = \overrightarrow{X_i^{d}} + \overrightarrow{V_i^{d+1}}$$

Position in day *d+1*

Position in day *d*

Velocity in day *d+1*

Exploration and exploitation in PSO

$$\overrightarrow{X_i^{t+1}} = \overrightarrow{X_i^t} + \overrightarrow{V_i^{t+1}}$$

$$\overrightarrow{V_i^{t+1}} = \underbrace{w\overrightarrow{V_i^t}}_{\text{Inertia}} + \underbrace{c_1 r_1 \left(\overrightarrow{P_i^t} - \overrightarrow{X_i^t}\right)}_{\text{Cognitive component}} + \underbrace{c_2 r_2 \left(\overrightarrow{G^t} - \overrightarrow{X_i^t}\right)}_{\text{Social component}}$$

Pseudo code of PSO

```
Initialize the controlling parameters (N, c1, c2, Wmin, Wmax, Vmax, and MaxIter)

Initialize the population of N particles

do
    for each particle
        calculate the objective of the particle
        Update PBEST if required
        Update GBEST if required
    end for

    Update the inertia weight
    for each particle
        Update the velocity (V)
        Update the position (X)
    end for

white the end condition is not satisfied

Return GBEST as the best estimation of the global optimum
```

Comparisons:

Simulation results show that ACO attains global minima in lesser number of iterations and to a lower value when compared with PSO. Also there is a lesser number of transients in the graph for square of error for ACO. Thus it can be concluded that **ACO gives better performance** than PSO for distance optimization problems.