Ant Colony Optimization

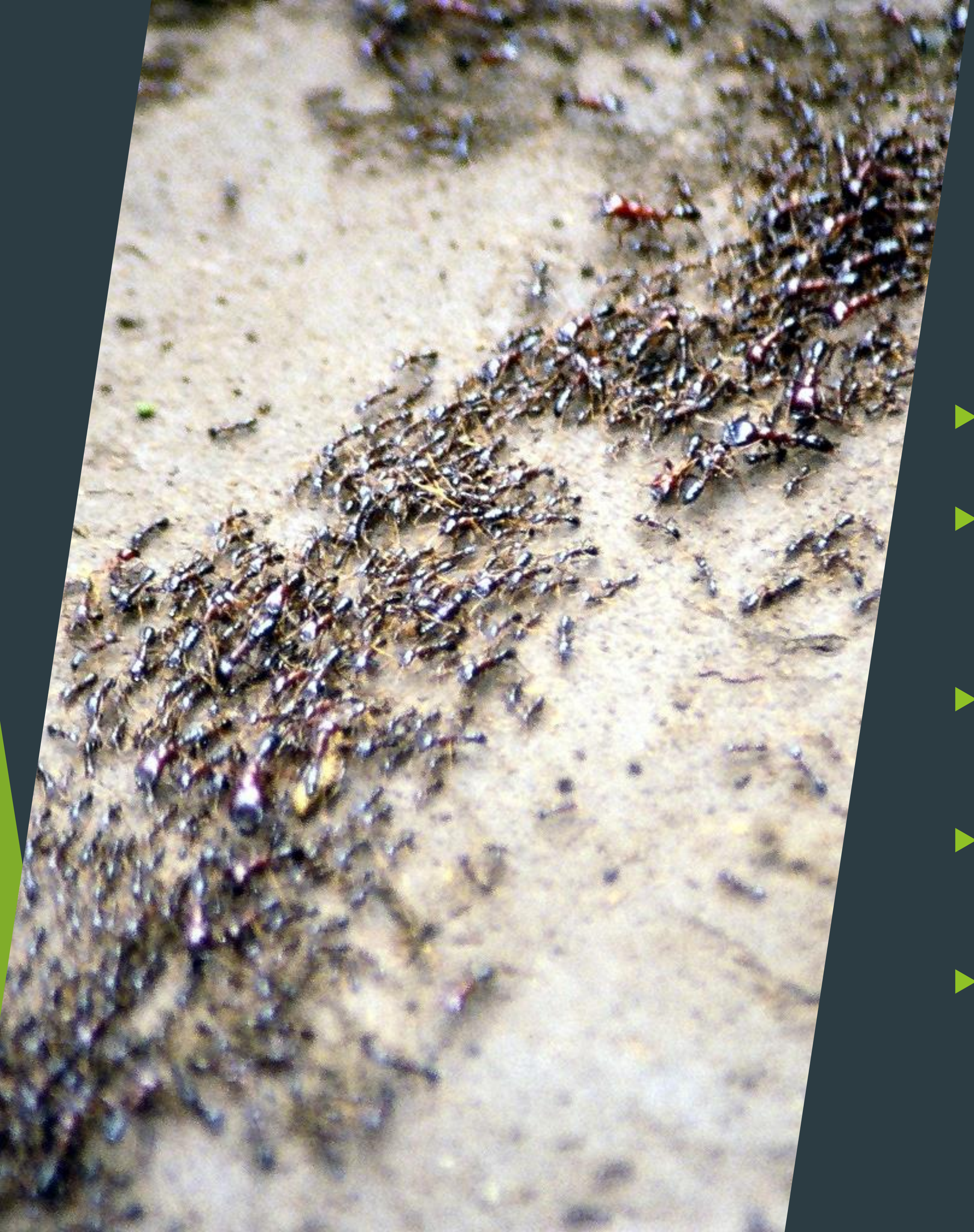
Introduction

Anurag Malyala:	2K15/CO/035
Aparna Jain:	2K15/CO/037
Ayush Gupta:	2K15/CO/040
Ayush Gupta:	2K15/CO/041

Introduction

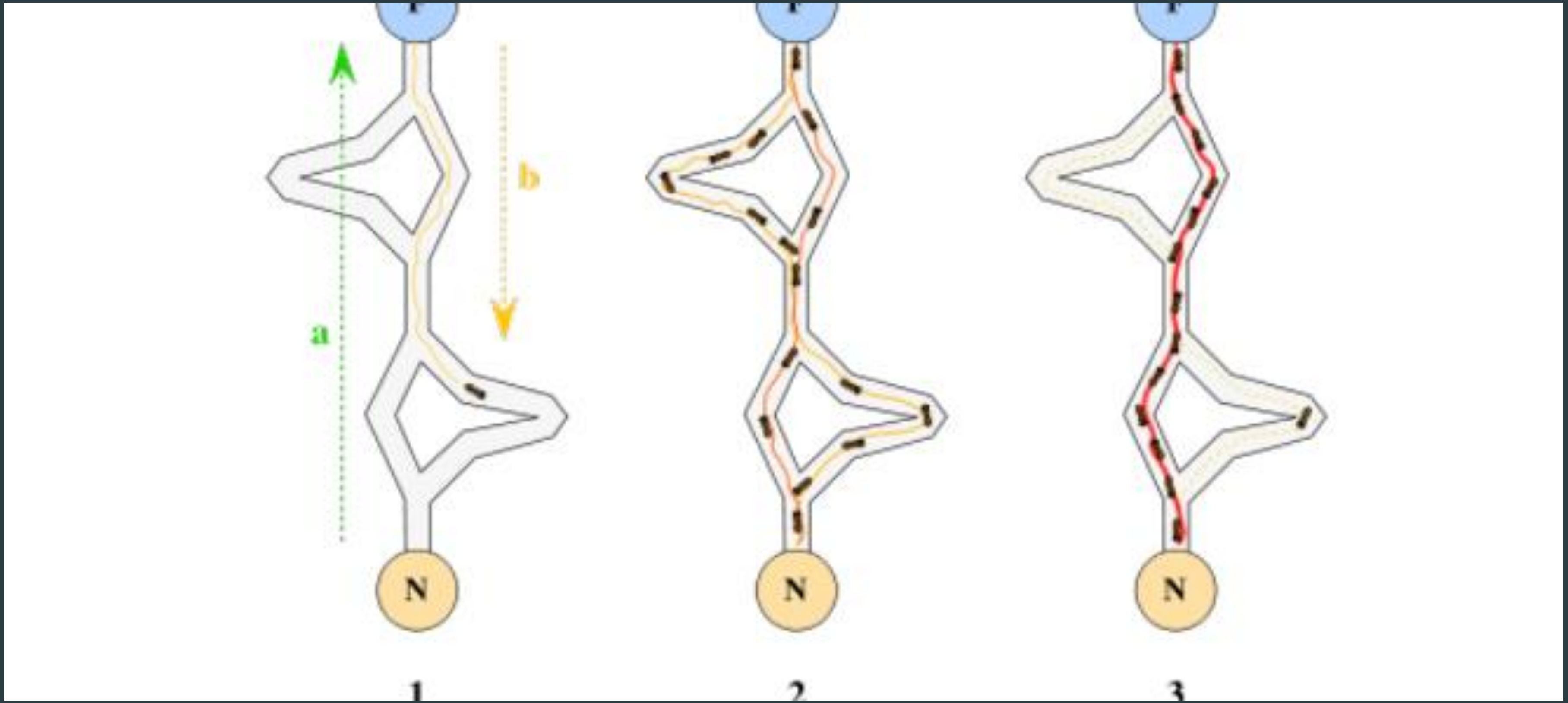
Ant Colony Optimization

- ACO is a population-based search technique for solving combinatorial optimization problems.
- ACO Is a Probabilistic technique which is used for finding optimal paths in a graph based on the behavior of ants trying to find a path between their colony and a source of food.
- This is a meta-heuristic optimization.



Real Ants

- Real ants find shortest routes between food and nest.
- They mainly rely on pheromone trails, chemicals left on the ground to signal other ants. This is called Stigmergy. The use of vision is minimal or not at all.
- Whenever an ant decides to follow a trail, it itself lays more pheromones thus fortifying that path.
- The more ants follow a trail, the stronger the signal gets, and it becomes more likely for an ant to follow it.
- The pheromones laid down decay over time.



Stigmergy in action

Artificial Ants

- Each Ant has some memory.
- They use virtual trails, selections a path based on the amount of “pheromone” present on the paths from any given node.
- This happens in discrete time but has since been adapted for continuous problems.
- The virtual pheromone levels accumulate on the path segments.
- At each node, the ant selects a path and moves to the next node.
- This is repeated until the ant reaches the goal node.

ACO Meta-Heuristic

Ant System was developed by Marco Dorigo (Italy) in his PhD thesis in 1992

The ACO Meta-heuristic

- Set parameters and Initialize pheromone trails.
  - Schedule activities
    - Construct the Ant Solutions
    - Daemon Actions
    - Update pheromone
- Virtual trails accumulate on path segments.

Procedure Construct Ant Solution

- An Ant moves from a 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 $\eta_{i,j}$

Sub-Procedure Pheromone Update

- The pheromone on a segment between i, j is updated according to:

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

- 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

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

  - Where L_k is the cost of the Kth ant's tour.

The Three main ACO Algorithms

Ant system, Ant Colony System, MAX-MIN Ant system. And their application to the TSP Problem

Ant System

- This was the first algorithm proposed in 1992
- The pheromone values are updated by all ants that have completed the tour by:

$$\tau_{ij} \leftarrow (1-\rho)\,.\,\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

- Where
  - ρ is the evaporation rate.
  - m is the number of ants
  - $\Delta\tau_{ij}^{k}$ is pheromone quality laid on edge (i,j) by the K[th] ant

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

  - Where L_k is the length of the tour of ant K.

Ant Colony System

- This improved upon the Ant System Algorithm by
  - Including a Decision Rule, a pseudorandom, proportional rule
  - Local Pheromone Update
  - Best only offline Pheromone Update.
- Ants move from i to j depending on a random variable q uniformly distributed over $[0, 1]$ and a parameter q_0.
- If $q \leq q_0$ then among the feasible components, the component that maximizes the product $\tau_{il}\eta_{il}^{\beta}$ is chosen, otherwise the same equation as in Ant System is used.
- This favours exploration of the pheromone information

Ant Colony System

- To prevent over-exploration and counter term is present, local pheromone update.
- This update is done by all ants after each step.
- Each ant applies it only to the last edge traversed.

$$\tau_{ij} = (1 - \varphi).\tau_{ij} + \varphi.\tau_0$$

- Where:
  - $\varphi \in (0,1]$ is pheromone decay coefficient.
  - τ_0 is the initial value of the pheromone.

Ant Colony System

- Best only offline pheromone update after construction follow the equation

$$\tau_{ij} = (1 \; - \rho).\tau_{ij} + \rho.\Delta\tau_{ij}^{best}$$

- Where

$$\tau_{ij}^{best} = \begin{cases} \dfrac{1}{L_{best}}, & if\ best\ ant\ k\ travels\ on\ edge\ i,j \\ 0, & otherwise \end{cases}$$

- L_{best} can be set to the length of the best tour found in the current iteration or the best solution found since the start of the algorithm.

MAX-MIN Ant System

- This retains the Best only offline Pheromone update rule.
- The min and max values are explicitly limited by:
  - τ_{ij} is constrained between τ_{min} and τ_{max}
  - After the update, τ_{ij} is set to τ_{max} if $\tau_{ij} > \tau_{\mathrm{max.}}$ and to τ_{min} if $\tau_{ij} < \tau_{min}$

Advantages and Drawbacks

Application Domains

- Routing in telecommunication networks
- Traveling Salesman
- Graph Coloring
- Scheduling
- Constraint Satisfaction

Advantages

- 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

- 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.

Summary and Closing statements

Summary

- Artificial Intelligence technique used to develop a new method to solve problems unsolvable since last many years
- ACO is a recently proposed metaheuristic approach for solving hard combinatorial optimization problems.
- Artificial ants implement a randomized construction heuristic which makes probabilistic decisions
- ACO shows great performance with the “ill-structured” problems like network routing

The end