Solving TSP with Ant Colony Optimization Algorithms

Billy Cox, Erin Masatsugu

Harvard College CS 182

Problem - TSP

Our goal was to find near-optimal solutions to instances of the Traveling Salesman Problem (TSP), the problem of finding the least costly tour on a graph, where a tour is a path from some start point that visits each node exactly once and back at the start point. This problem is NP-Hard.

Solution - ACO

Our approach is to solve TSP with Ant Colony Optimization (ACO) algorithms. ACO is a biologically inspired reinforcement learning algorithm based off of how ant colonies forage for food.

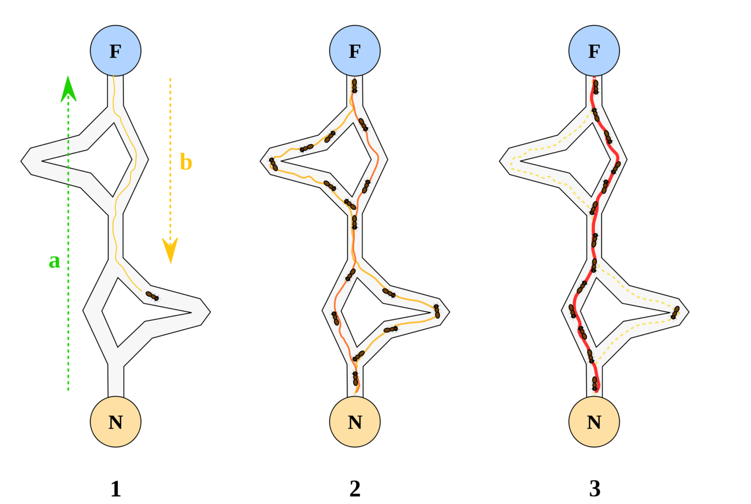


Figure 1: Reinforcement Learning in Ants

Each ant wanders randomly, returning to its nest after finding food. An ant leaves pheromones wherever it steps, and so as it wanders, it leaves behind a "pheromone trail" which other ants can detect. Other ants follow an encountered pheromone trail with a certain probability, else they continue to wander randomly. This probability determines an ant's tendency to explore new terrain vs. exploring the learning that its fellow ants have done. The fact that this exploration/exploitation tradeoff is handled implicitly in the algorithm is a huge benefit to ACO algorithms.

Additionally, pheromones constantly evaporate, so the strength of a pheromone trail decreases over time. Since shorter paths can be traversed more times, shorter paths are more likely to be reinforced than longer paths, which is the goal of the algorithm.

Ant System - $O(t \cdot m \cdot n^2)$

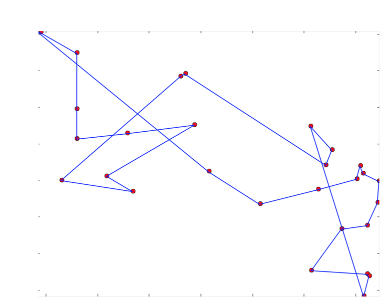


Figure 2: With AS, cost/optimal cost = 1.36

n is the number of nodes
m is the number of ants
t is the number of iterations to run

Each ant moves from i to j with probability:

$$p_{ij} = \frac{{\tau_{ij}}^a \cdot {v_{ij}}^b}{\Sigma_j {\tau_{ij}}^a \cdot {v_{ij}}^b}$$

Each edge of tour is updated as follows:

$$\tau_{ij} = (1-d) \cdot \tau_{ij} + \Delta_{ij}$$

After t iterations, we deterministically find the best path given the updated pheromone strengths.

τ_{ij} is the pheromone strength on edge ij
$v_{ij} = \frac{1}{w_{ij}}$ is the visibility of edge ij
w_{ij} is the weight of edge ij
d is the rate of decay
$\Delta_{ij} = \frac{C}{l}$ is the amount of pheromone to add to edge
C is the cost of a reasonably good tour
l is the cost of the tour found

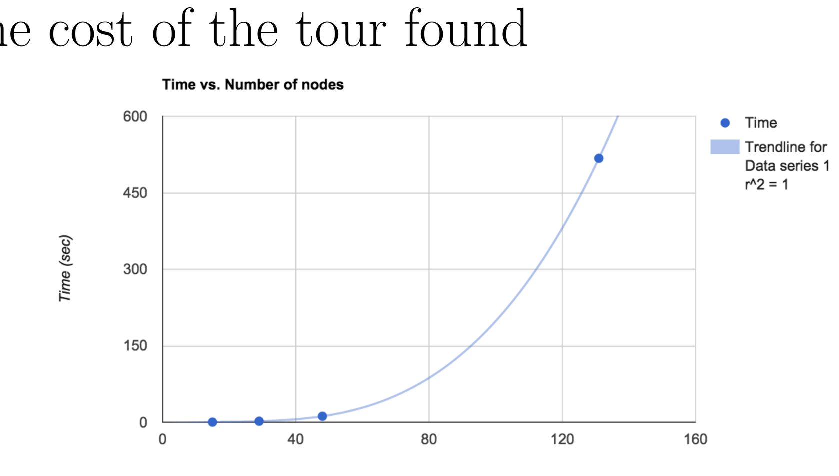


Figure 3: AS, 10 iterations

AS with Elitist Ants - $O(t \cdot m \cdot n^2)$

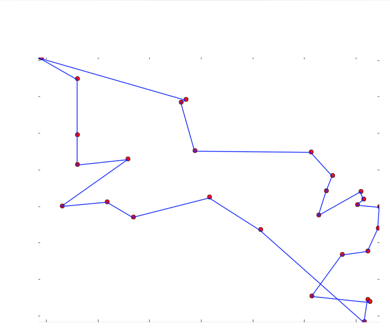


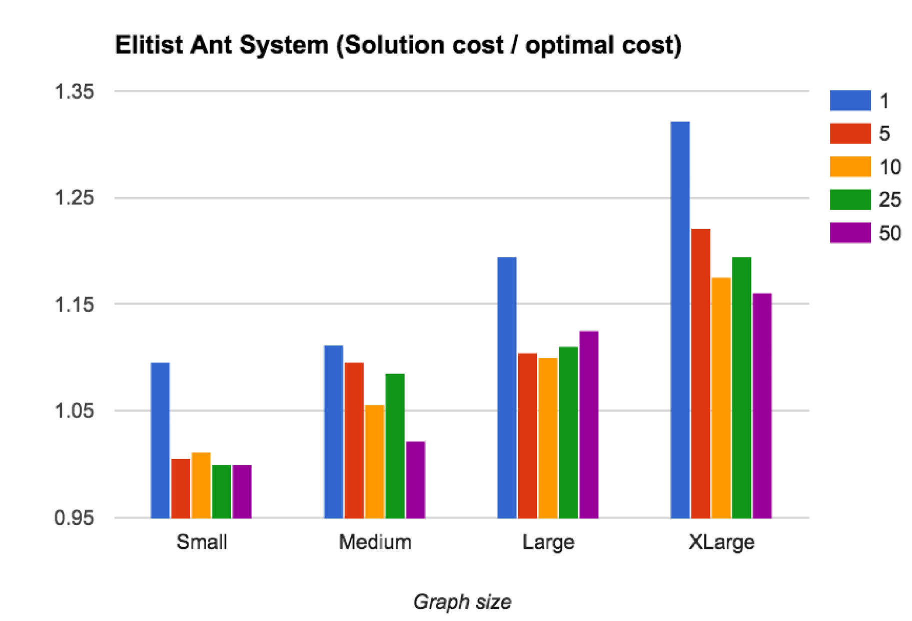
Figure 4: With ASE, cost/optimal cost = 1.06

ASE, Cont.

"Elitist ants" only reinforce edges in the best path found so far, directing other ants towards that path. Thus, we add an extra term when updating the pheromone strengths along an edge:

$$\tau_{ij} = (1-d) \cdot \tau_{ij} + \Delta_{ij} + e \cdot \frac{C}{l*}$$

e is the number of elitist ants
$l*$ is the cost of the best tour so far



Ant Colony System - $O(t \cdot m \cdot n^2)$

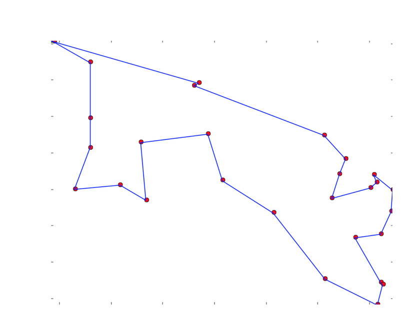


Figure 5: With ACS, cost/optimal cost = 1.03

With probability r, the ant takes the most desirable edge (exploitation). Else, it explores like in Ant System, except it only considers the candidate list, the k closest nodes. If there are no unvisited candidates, the ant takes the closest unvisited node.

Let $q \sim \text{Unif}(0,1)$. Each ant moves from i to j where:

$$j = \begin{cases} \text{argmax}(\tau_{ij} \cdot {v_{ij}}^b) & q < r \\ \text{sample from}(p_{ij}) & q > r \end{cases}$$

Pheromone strengths along an edge get weaker when an ant traverses it, promoting exploration by other ants. This local update occurs as follows:

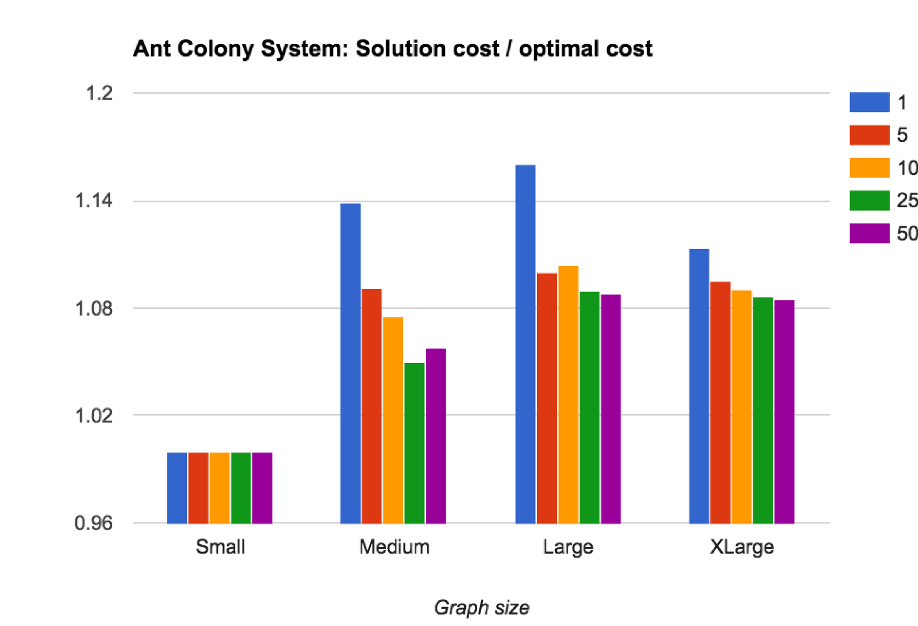
$$\tau_{ij} = (1-d) \cdot \tau_{ij} + d \cdot \tau_0$$

ACS, Cont.

Pheromone strengths only increase after finding a new best tour. This global update occurs as follows:

$$\tau_{ij} = (1-d) \cdot \tau_{ij} + \Delta$$

$\tau_0 = \frac{1}{n \cdot C}$ is the initial τ given to all edges
$\Delta = \frac{1}{l*}$ is the new update factor



Challenges

There were many constants involved, like number of ants, number of iterations, decay rate, scaling factors for pheromone strength and visibility, and initial pheromone strength. With 1 machine, it was difficult to know if we were using constants that optimized our performance. We plan to run more experiments to find the best combination of values.

Conclusion

Our results show that on smaller graphs, Ant System (even with Elitist Ants) performs well, but Ant Colony System performs much better at scale. For further study, we would explore different variations such as Rank-based Ant System or Recursive Ant Colony System, examining which circumstances are best solved by which variations.

References

1. Bonabeau, Eric, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* New York: Oxford UP, 1999.