ACO Algorithm Discussion and Analysis

Braydon Spaulding
Rigby, ID United States
spaubray@isu.edu

Benjamin Keninger
Pocatello, ID United States
kenibenj@isu.edu

Rowan Murphy
Pocatello, ID United States
murprowa@isu.edu

***Abstract*—Ant Colony Optimizations are a family of evolutionary algorithms that work by simulating the behavior of ants to search a given path and find a good solution. This paper adapts of the work by Yang et al[1]. in order to implement an ACO algorithm for the travelling salesmen problem. The algorithm works by repeatedly producing generations of ants who are randomly placed among nodes on a table before probabilistically deciding which city to visit next based on the "pheromone" value of the path - which represents roughly how often a given path is traveled - and a distance value. Analysis of the algorithm reveal that it typically produces a path better than the greedy algorithm if given enough time however as it is generally outperformed by the branch and bound algorithm.**

I. Greedy Algorithm

The greedy approach to the TSP is to simply visit the nearest unvisited city until all cities have been visited. As each city makes a check to an ever-shrinking list of unvisited cities, the greedy algorithm has a time complexity of O($n^2 \cdot log_2 n$) where n is the number of cities. While not guaranteed to find an optimal solution, the greedy algorithm is generally able to produce an efficient path in very little time. As such, its results are what those of this paper's implementation of ACO are being compared to.

II. Introduction

Ant Colony Optimizations or ACOs are a family of evolutionary algorithms first proposed by Marco Dorigo in 1992. These algorithms work by recreating the path finding capabilities of real world ant colonies, hence the name. While ACOs can be used to solve a variety of path finding problems, this paper is concerned with the implementation and efficiency of ACOs in the context of the travelling salesmen problem. The implementation of the algorithm in this paper is an adaption of one proposed by Yang et al[1].

III. Algorithm Explanation

The ACO starts by iteratively creating and computing the path of a given number of ants. The probability of an ant picking a particular path is given by:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in allowed} [\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}$$

An excerpt from Yang et. al describes the parameters as τ_{ij} "where τ_{ij} is the intensity of pheromone trail between cities i and j, α the parameter to regulate the influence of τ_{ij}, η the visibility of city j from city i, which is always set as $1/d_{ij}$ (d_{ij} is the distance between city i and j), β the parameter to regulate the influence of η_{ij} and $allowed_k$ the set of cities that have not been visited yet, respectively."

On the completion of a tour, each ant updates the pheromone matrix by computing Q/L_k. Q is a constant, and L_k is the length of the tour. Once each ant in a given generation size completes a tour, the pheromone matrix is evaporated by a multiplier ρ as given by:

$$\tau_{t+1} = \rho \cdot \tau_{ij}(t)$$

IV. Complexity

The ACO algorithm is probabilistic and doesn't return a best path until it times out it or runs through n iterations. In other words, the algorithm will run forever if it is left to do so. This results in an undefined time complexity. If given v iterations we believe algorithm will run in O($v \cdot m \cdot n^2$) where m is the number of ants and n is the number of cities. The $m \cdot n^2$ is coming from each the ant computing its path.

We determined our space complexity to be O($m \cdot n^2$) where m is the number of ants and n is the number of cities. Each ant is its own class that keeps track of its path, however this is discarded when determining the final space complexity.

V. Analysis of Results

	Random		Greedy	
Cities	T(s)	Cost	Cost	% /Random
15	.0038	19532	11568	0.051
30	.0512	36331	14877	0.409
60	9.280	85641	29388	0.343
100	TB	inf	36482	NA
200	TB	inf	54410	NA

B&B			ACO		
T(s)	Cost	% /Greedy	T(s)	Cost	% /Greedy
1.60	10281	0.888	TB	11309	0.977
60	16595	1.115	TB	16205	1.089
500	25086	0.853	TB	27997	0.952
600	36427	0.998	TB	39974	1.095
TB	55536	1.020	TB	42516	0.781

VI. Future Work

For our future work in ACO in relation to TSP, we would like to explore a few possibilities. We have an interest in using machine learning algorithms to modify α and β values which determine the influence of both our pheromones and distances in relation to our probability to determine if we can better optimize ACO for a specified number of cities. We would also like to modify the number of ants per iteration to see what kind of impact it may have on our efficiency and the overall time and space complexity of our algorithm. Additional GUI aspects such as ant visualizations, and the ability to see how ants are traveling across iterations were also discussed but we did not have time to implement them in this project scope.

References

[1] J. Yang, X. Shi, M. Marchese, and Y. Liang, "An ant colony optimization method for generalized TSP problem," Progress in Natural Science, vol. 18, no. 11, pp. 1417–1422, Nov. 2008, doi: 10.1016/j.pnsc.2008.03.028.