Ant Colony Optimization with Multiple Objectives
TJHSST Senior Research Project
Computer Systems Lab 2009-2010

Honghou Zhou

January 21, 2010

Abstract

Ant Colony Optimization (ACO) is a useful method to find near optimal paths. The most common algorithms only have ants that choose their individual paths based on pheromones left by other ants. This serves to optimize the distance between the starting location and a certain goal. However, it is often more realistic and useful to factoring other variables.

Keywords: Ant Colony Optimization

1 introduction

Ant Colony Optimization is a process used to find a near optimal path that satisfies certain restrictions. It There is a often need to find the optimal path while satisfying several variables. An example would be to minimizing the time of a certain route while keeping costs low. Often, one or few of the variables are more important or more strict compared to the others. In this case, it would be more important for those select variables to be optimized even if others are not.

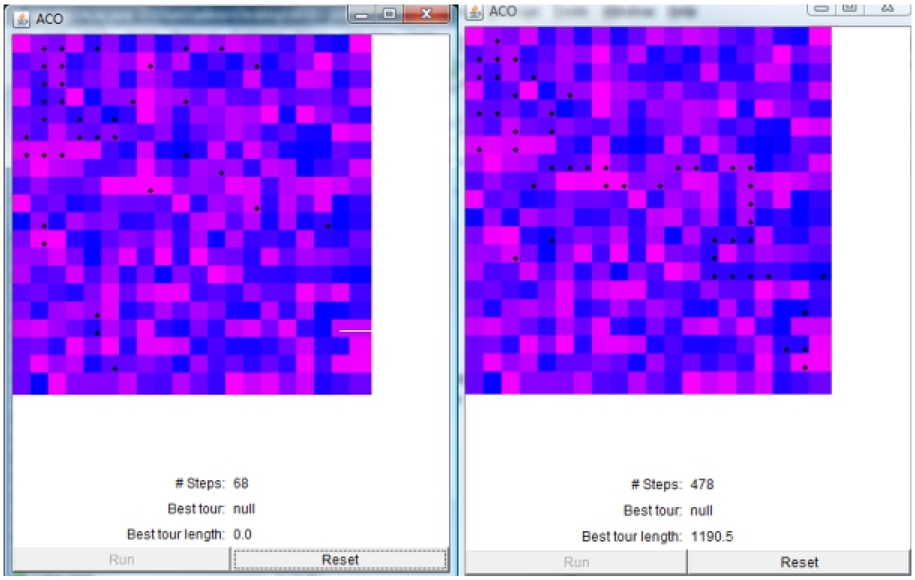

Figure 1: Current program run

2 Background

2.1 Ant Colony Optimization

Ant Colony Optimization (ACO) was inspired by, and mimics, the process used by ants to find a short path from their colony to a food source. As an ant travels back to its colony, it leaves behind a pheromone trail that other ants use. Other ants then base their decision for which route to use partly based on the amount of pheromone it detects. Over time, pheromone builds up faster on shorter paths and evaporates from the rest. This allows the population of ants to weed out different paths until only a near optimal one remains. The process does not guarantee an optimal solution, but the one it finds should be close. When multiple variables are weighted, the path each ant chooses for its route is based on all of the variables.

3 Procedure

First a more basic Ant Colony Optimization program is created. For this program, there are four main classes dealing with the ACO, and several others for graphics output. The network contains all the nodes, edges, and ants. Nodes have an ID, starting with 0 and increasing by one going down the column, and a weight (potentially multiple weights). Edges are created with a length that is the absolute value of the difference of nodes weights. Ants are released from the starting location, the top left corner on the grid, until there are 100 on the net. They are released one at a time, and whenever an ant finishes another one comes out (or think of it as just not showing the ants traveling back). The ants travel from the top left node to bottom right node, making decisions on which node to go to next utilizing pheromone deposited by ants that have completed the tour. As the program runs, it goes through each node, counts all the ants on that node, and has each ant calculate its next node (each ant also has a vector of all the nodes it has been to). If an ant has reached the end it deposits on the edges between nodes on its path. The program prints out the tour length and path for that ant. All the edges then update their weights based on new deposits.

After the system is constructed, the various variables controlling the movement of the ants and the pheromone deposition and evaporation are modified and tested. Other variations on the basic ACO such as the Elitist Ant are tried. These modify the basic procedure in an attempt to create more optimal tours using less time.

Currently, I plan to implement a new pheromone for the second variable. Each node will have two weights, one for each variable. The ants will be depositing two different pheromones then. If that can work, I can begin to test and record the results for different combinations

4 Expected Results

For each run, there is a best-ever-tour as well as the tour that ants congregate to. The length of each will be recorded (as best tour length and average tour length). The resulting path is an approximation of the optimal. The closer the average tour length is to the actual optimal tour length the better. The data for variables that are used in decision making will be recorded. The results for different weights can be compared to find a near optimal combination. For multiple objective, when the all weights for variables set to zero except one it means only that one variable is taken into account. The results then are expected to match a simple one variable run.

5 Results and Conclusion

The goodness of the program is based on how many steps before the ants focus on a path and how close to optimal the path is. Right now the best ever tour had a length of about 1050. Each run can usually produce a tour at least around 1190. However, the path the ants congregate to has a length around 1350 (perhaps I should keep of an average tour length as well), which is not optimal.

References

[1] Blum, C. Ant colony optimization: Introduction and recent trends. Barcelona, Spain 2005

[2] Mora, A.M. *Balancing Safety and Speed in the Military Path Finding Problem: Analysis of Different ACO Algorithms.* Granada, Spain 2007

[3] Mora, A.M. *CHAC. A MOACO Algorithm for Computation of Bi-Criteria Military Unit Path in the Battlefield.* Granada, Spain 2006