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

Honghou Zhou

April 8, 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

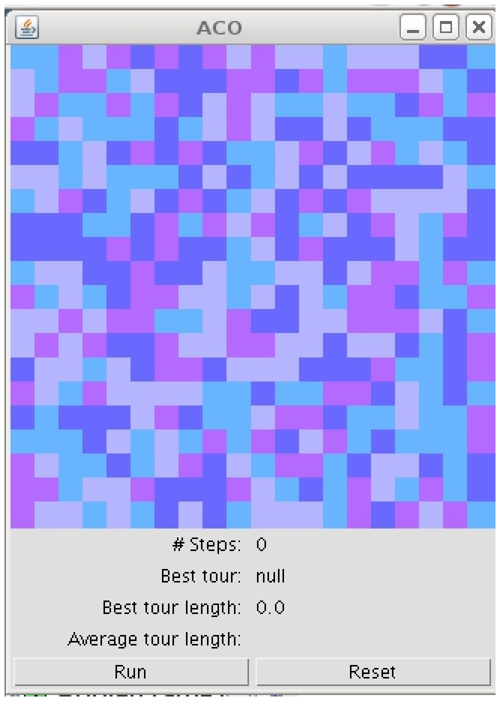
1 Introduction

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.

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 pheromon trail that other ants use. Other ants then base their decision for which route to use partly based on the amount of pheromones it detects. Over time, pheromones 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.



# Steps: 0
Best tour: null
Best tour length: 0.0
Average tour length:

Figure 1: Current program run

3 Procedure

The project is done in Java. There are ants until there are 100 on the net, released one at a time, 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 based on both pheromones 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). Currently, the ant goes through the same process as before twice, one for each phermone. As it does so it muptiplies the sum by the weight of the pheromone (currently 1/2) If an ant has reached the end it deposits on the edges between nodes on its path, updating both pheromone's counts. The program prints out the tour length and path (using node IDs) for that ant. All the edges then update their weights based on new deposits.

4 Expected Results

For each run, there is a best-ever-tour as well as the tour that ants congregate to. The closer that this path is to the true optimum 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 currently do match a simple one variable program. The program also now creatues a file with the results rather than simply printing them on screen. What needs to be recorded are the length in terms of both variables, the average length for both when the ant finishes, and the number of steps from start until that ant's finish.

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 1250. However, the path the ants congregate to has a length around 1550, which is not optimal. The fact that the multiple objective verson produces a 'worse' path is expected. The length are still based on one variable and consideration for the second variable isn't seen.

References

[1] Mora, A.M., et. al, *LaTeX: Balancing Safty and Speed in the Military Path Finding Problem: Analysis of Different ACO Algorithms example*. Longdon, England 2007