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

Honghou Zhou

June 16, 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. 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.

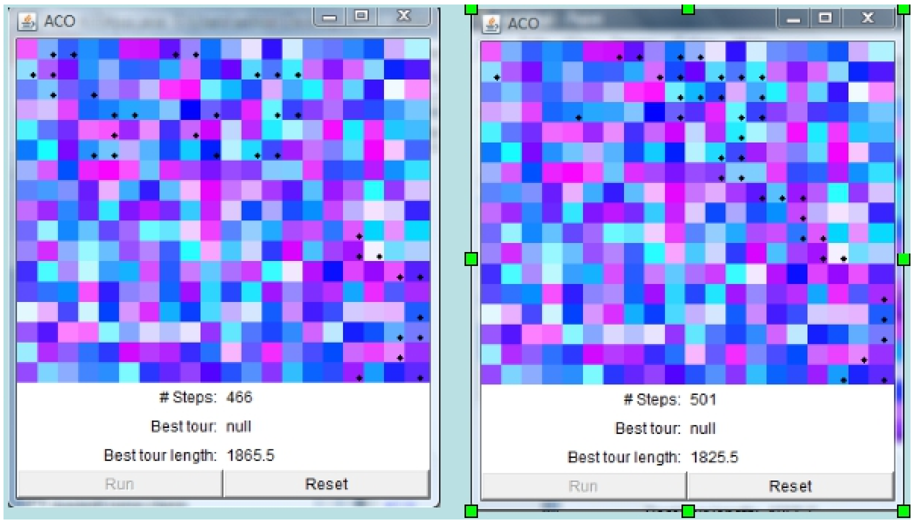

Figure 1: Current program run, note how by 500 steps the ants have congregated to a path

2 Background

2.1 Ant Colony Optimization

Ant Colony Optimization (ACO) was inspired by, and mimics, the process used by ants to find relatively 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. Multiple objectives ACO (MOACO) tries to be more realistic about modelling real life situations. An example would be for military pathfinding where both safety and distance need to be considered. Another example is for planning a delivery route where time and money spent are both important. When multiple variables are weighted, the path each ant chooses for its route is based on all of the variables.

3 Procedure

The project is done in java. There are four main classes: The network contains all the nodes, edges, and ants. Nodes have an ID and two weights. The nodes' color reflects the value of the weights. Edges have two 'lengths' that is the absolute value of the difference of nodes weights. Ants are released one at a time from the top left corner on the grid, until there are 100 on the net. Whenever an ant finishes another one comes out. The program does not display ants going back to the starting point, but pheromones are deposited at that time.

The destination is the bottom right node, and the ants make decisions on which node to go to next utilizing 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).

This general process is the same for all variants of ACO, including multiple objectives. The only difference really comes down to how phermones are processed. As the program proceeds it muptiplies the sum by the weight of the pheromone. This weight is a decimal between 0 and 1, inclusive. When at the two extremes (0 and 1) it is If an ant has reached the end it deposits on the edges between nodes on its path, updating both pheromone's counts. If an ant has reached the end it deposits on the edges between nodes on its path. The program prints out the tour lengths for both objectives. All the edges then update their weights based on new deposits. A variation of the Elitist Ant algorithm is used. This lets the ant that gives a best ever tour to deposit double the pheromone. The route is then reinforced each time until a better one is found.

After the system is constructed, the various variables controlling the movement of the ants and the pheromone are modified and tested.

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

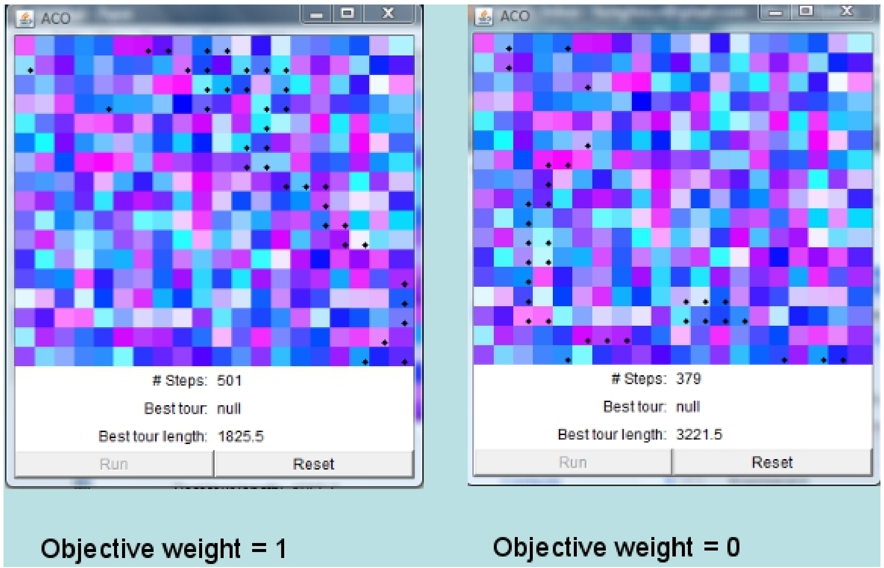

Figure 2: Results for two different weights

one it means only that one variable is taken into ac count. The results then currently do match a simple one variable program. 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

I am currently limiting each run to 500 steps. This way, the program simulates finding the best possible path with a limited amount of time. It appears that a paths can be found bewteen 450 and 500 steps. For each run, there is a best-ever-tour length and an average tour length. Depending on the objective weight, different lengths result. For weight of 1, the length average is around 2100 while for a weight of 0, the length average is around 3600. With a weight of 0.5, the average is actually the lowest at about 1600.

The system generally works as expected. When the objective weight is focused on one objective that objective reduces in length more and at a faster rate compared to the other. It is interesting that the optimal result overall

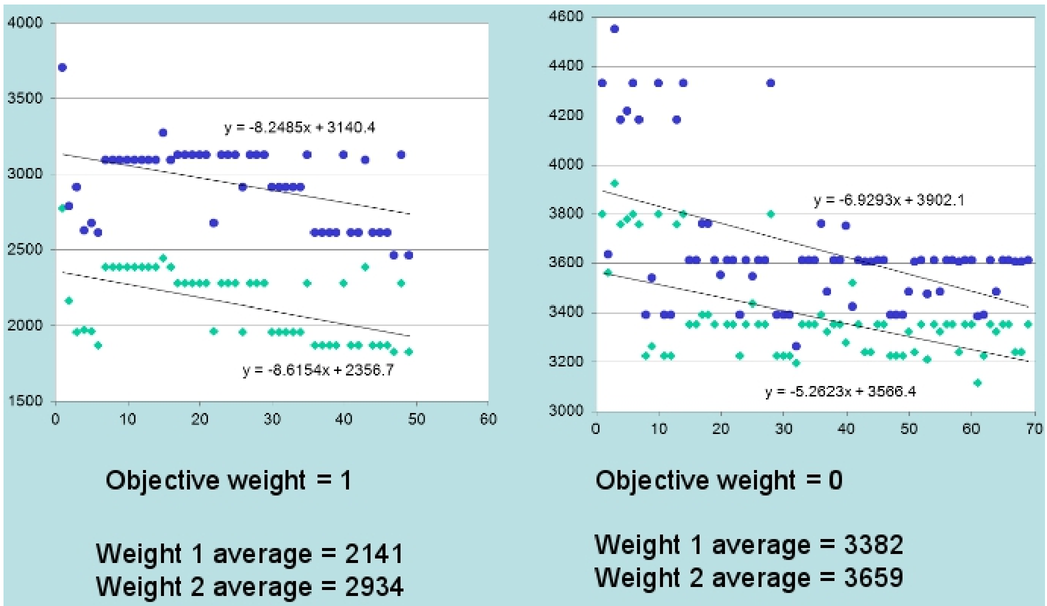

Figure 3: Graphs for results

and individually is achieved when both objectives are taken into account.

References

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

[2] Garcia-Martinez, C., Cordon, O., Herrera, F. *A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bicriteria TSP*. Granada, Spain 2006

[3] McMullen, P. R. *An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives*. Auburn, USA 2001

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

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