Ant Colony Optimization with Multiple Objectives

Honghou Zhou

TJHSST Computer Systems Lab 2009-2010

Abstract

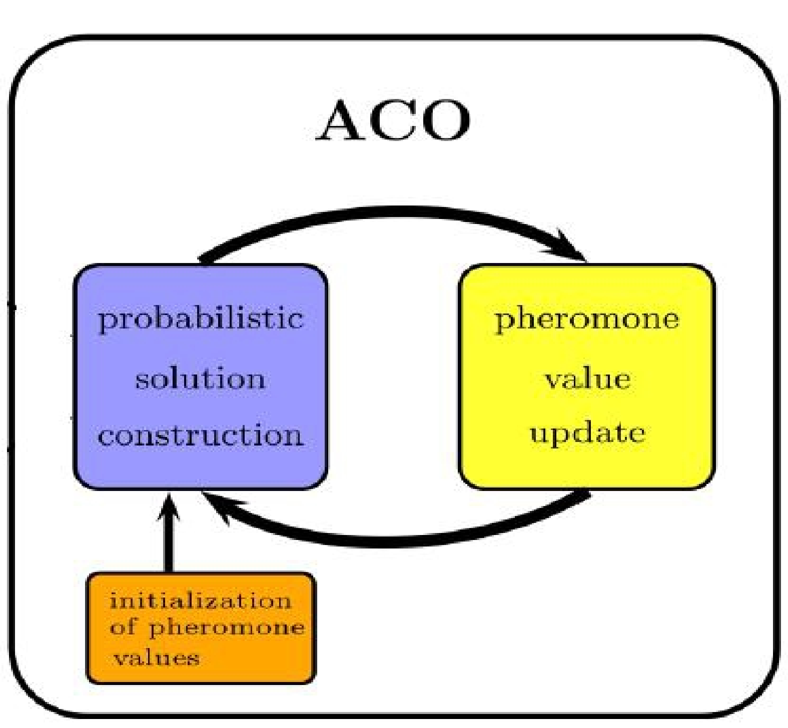
Ant Colony Optimization (ACO) is a useful method to find near optimal paths. The most common algorithms only works toward finding the best path with regard to one condition. However, it is often more realistic and useful to factoring other variables and constraints as well. This project will be to develop a way to consider two objectives and weigh their relative importance.

Introduction/Background

Ant Colony Optimization models how ants 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 chose their next node partly using the pheromone levels.

Pheromone builds up faster on shorter path than longer ones and evaporates over time. This allows the colony to weed out paths until only a near optimal one remains. The process does not guarantee an optimal solution, but the one it finds should be close.

More than one factor generally affect which path is actually the best and focusing on only one means sacrificing others. The idea then is to implement a way to take multiple objectives into account.



Procedures and Methods

For the basic ACO, there are four main classes and several others for graphics output. 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 (amount of blue and red). 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.

Procedures and Methods cont.

The destination is the bottom right node, and the ants make 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. **Currently I am using a variation of the Elitist Ant algorithm. This lets the ant that gives a best ever tour to deposit double the pheromone of a normal ant. The route is then reinforced each time until a better one is fount**

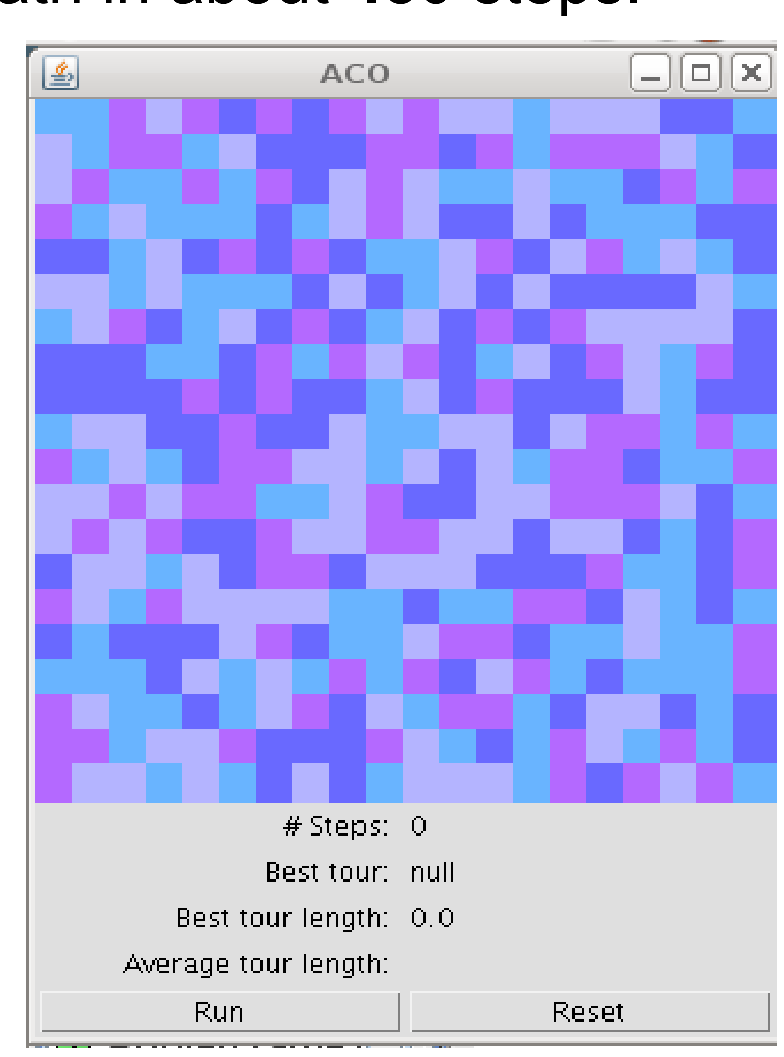
With the implementation of a second objective each node now has two weights, one for each variable. The ants deposit two different pheromones then. Almost all the basic structure for single objective ACO is kept.

Results

For each run, there is a best-ever-tour length and an average tour length.

The best tour length is usually **1252.5 (multiple objective).** The average tour length is approximately **1550 in multiple objective and 1300 with a single objective**

The ants do behave as expected, and congregate to a path in about **450** steps.



Current program run

Conclusion

There is a difference between best tour length and average tour length, which is not desirable. **The fact that multiple objective runs are slower are to be expected**

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