CIS 421 Artificial Intelligence

Fall 2019 **Dr. Laura Grabowski**

Assignment 4: Ant Algorithms – Traveling in Middle-earth, The Sequel

"All good stories deserve embellishment," or a sequel. For this program, you will solve the same problem as in Program 2 using ant colony optimization. Please refer to the handout from the A* search assignment for the details of the problem, including the values to use for the heuristic measure.

This assignment has two parts, the program itself (run 5 different times with specified parameters) and a discussion that looks at your results and compares/contrasts the two different approaches we used to solve the problem (A*, ACO).

Part 1: PROGRAM: 70 points

Use the ant colony optimization algorithm to find the best path from the Blue Mountains to the Iron Hills.

Overview

1. NOTE: This problem is NOT a TSP problem. For this problem, you have a specific start node and end node. Your task is to find the path of lowest cost between those two nodes. Because of this, your ants will always start at the same node (the Blue Mountains), and end at the same node (the Iron Hills). You will not visit each node on the map in your solution.
2. You will run experiments with your program using 5 different sets of parameters (see Table 1 below for the parameter values). One set of values will give more weight to the heuristic, the 2nd set favors the pheromone level, and the 3rd eliminates the heuristic entirely, using only pheromones. The 4th and 5th experiments test for the influence of different values of RHO. Keep your results for each of the parameter sets (and make sure you know which results go with which parameters) so that you can use the results in the discussion.
3. Use a population size of 10 ants, and a maximum of 25 cycles. These numbers are somewhat arbitrary, so they can be increased if you think adjustments are needed (*e.g.,* you can increase the number of cycles if your solution does not converge).
4. For the heuristic measure used in the movement probability equation (equation 7.1, p. 145 in textbook), **use only the point-to-point (next-hop) information** (column 3, "Distance (miles)" in Table 2 in the A* handout); **do not use the information for the node to goal distance** (Table 1 in the A* handout). PLEASE NOTE THAT THE INFORMATION YOU NEED IS ALREADY GIVEN IN TABLE 2.

Program output will be a list of the locations in the path that your implementation selects, in the order the nodes are traversed.

A few considerations

1. I already pointed out that your program is NOT a TSP problem. This is different from the code from the book.
2. Your graph is NOT fully connected. You need to have a way to backtrack your ant agent if she gets into a node where she is "trapped."
3. You may do pheromone evaporation *either* before *or* after depositing new pheromone. You may want to try both and see if it makes a difference in your program's result.

Table 1. Parameter values for experiments.

Parameter Name	Parameter Meaning	Run 1	Run 2	Run 3	Run 4	Run 5
ALPHA	Weight of pheromone	0.4	1.0	1.0	1.0	1.0
BETA	Weight of heuristic measure	1.0	0.4	0.0 *	1.0	1.0
RHO	Trail persistence; 1 - RHO is evaporation rate	0.65	0.65	0.65	0.4	0.95
Q	Quantity of pheromone deposited on trail	100	100	100	100	100

***Note:** In Run 3, you are eliminating the heuristic measure. In other words, the ant agents will use only the pheromone for path finding. You will still use the point-to-point distances when calculating the tour length.

Part 2: DISCUSSION: 30 points
Report your results (the paths found for each set of parameters from Table 1, time to convergence, and any other interesting data that you may have to include). Reflect on and discuss your results, using the following questions to guide your discussion.

1. What effect (if any) did changing the values of the parameters (ALPHA, BETA, RHO) have on the solution returned or the functioning of the algorithm? For example, did different parameters change the path? Did the parameter values make the algorithm converge faster or slower? Did any parameter values produce poor results?
2. Compare/contrast the operation of the two different methods (A*, ACO) on the problem. Did the algorithms find the same solutions or were they different? Why do you think they were different (or the same, as the case may be)? Did one algorithm seem more efficient than the other for this problem? Be sure to clearly explain how you are using the term "efficient" in this context.
3. What factors would you think about if you had to choose between these two methods (A*, ACO) to solve a particular problem? What sorts of problems does A* lend itself to solving? For what sorts of problems would ACO be the better choice?

GENERAL REQUIREMENTS
Programs must compile and run correctly on the CS lab server.

WHAT YOU WILL TURN IN (through Moodle)
You will turn in all of the following files**:**
1. Program code.
2. Input file.
3. README file with instructions on how to compile and run your program.
4. Discussion (.pdf, .doc, docx, .rtf formats are all acceptable, but .pdf is preferred).