Visualising Ant Colony Optimisation

Major Project

Christopher Edwards

What is the Project?

- Provide a visual representation of an Ant Colony Optimisation algorithm
- To be used in an education environment
  - Teaching resource
- The algorithm's parameters can be changed by the user
  - Number of cities/ants
  - Alpha/beta values
  - Etc.

Ant Colony Optimisation

- Family of Swarm Intelligence methods
  - SI typically involves a population of simple agents
- Probabilistic path finding technique
  - Return the optimal path in a graph
- Based on real-world ant behaviors
  - Ants always find the shortest path between nest + food
  - Work together based on pheromone trails
- Currently this project focuses on the TSP
  - Travelling salesman problem

Probabilistic property

- An agents next location is calculated using the following probability:

$$p_{xy}^{k} = \frac{(\tau_{xy}^{\alpha})(\eta_{xy}^{\beta})}{\sum(\tau_{xy}^{\alpha})(\eta_{xy}^{\beta})}$$

  - α
    - Meta heuristic value relating to the favoring of pheromone
  - β
    - Meta heuristic value relating to the favoring of pheromone
  - η
    - 1/distance from current to node xy

Pheromone deposit

- The pheromone trails are updated as the agents traverse the graph
- The equation to model this behavior is:

$$\tau_{xy}^{k} = (1-\rho)\tau_{xy}^{k} + \Delta\tau_{xy}^{k}$$

$$\Delta\tau_{xy}^{k} = \begin{cases} Q/L_k & if\ Agent\ k\ uses\ curve\ xy\ in\ its\ tour \\ 0 & otherwise \end{cases}$$

- This correctly factors in decay and allows for any ant moving through node xy to add to the pheromone concentration at node xy

Technical issues

- Implemented in Java and its Swing packages
  - Cross-platform
  - JUnit tests
- Visualising the algorithm is difficult
  - Easy to implement, difficult to visualise
    - Showing pheromone levels to the user
    - Showing the agents moving
    - Show the best route
- Do all this in a reasonable time

Technical issues(2)

- Use of SwingWorker to execute the algorithm in its own thread
  - Allows updated to the view without it ‘freezing up’

- Simulate ants moving on the ‘paths’
  - Get the location the ant started and its destination
  - Use linear interpolation
    - Given the x /y of the start and finish
    - Linear interpolation can be used to get the value of any xy on the path between these two points
    - value1*(1-mu)+vlaue2*mu);
      - Value1/2 is the X or Y value (e.g value1 = start x, value2 = destination x)
      - Mu is how far on that path you want (e.g mu 0.5 = half way between the points)

Current algorithm

- To update the view to reflect the current state
- For every ant(agent)
  - Draw the ant at their current location
  - If they aren't finished, move them
    - During this move, get their current X and the destination
      - Linear interpolate these two locations to show the ant moving
      - Update pheromone
    - If the ant has the current best path, set its path as best
      - Loop through the cities visited in order and draw a line between them
      - paint the updated pheromone
  - Once all ants have finished
    - Remove all pheromone trails and only show the best path
- This is the general form there are other intricacies

Demos

- 3 demos (all 1 iteration):
- Demo 1: 10 cities 1 ant (slowed down)
- Demo 2: 10 cities 20 ants
- Demo 3: 20 cities 200 ants (sped up to show it solves)

- Iterations demo
- Demo 1: 10 cities, 5 ants, 1 iteration
- Demo 2: 10 cities, 5 ants, 10 iterations
- Demo 3: 10 cities, 5 ants, 25 iterations
- All of the iteration demos are sped up for demo purposes

Future work

- Incorporate the text boxes to allow user define values
- Display how many ants are at each city
- Allows loading/saving of cities configurations
  - Allows for the same problem to be demonstrated every time
- Display the best route and distance to the user in text form
- Experiment with correct limits to add to text fields
- Potentially add variations of the algorithm
  - Elitist ants, Min-Max system etc.
- Test the application thoroughly
- Potentially set up an nest-food style of problem
- Let the user define the speed which the algorithm runs

Questions?

Thank you for listening, if you have any questions I will gladly answer