Ant Colony Optimization

Ant Colony Optimization (ACO) is an optimization algorithm where the algorithm imitates the behaviour of the ant colony for a given optimization problem. Given an unequal weighted graph, the ants take a route. The lower the cost of the route is, higher the chance that the ant will take the route. As ants move around the points in the graph, they leave a trail of pheromone. As more ants take the route, the stronger the pheromone is, resulting in more probability of ants taking the route.

# Implementation Description

There are three phases with the ACO: initialization, running, and return phase. The function takes inputs of the matrix of the room, number of ants, number of cameras, and the max iteration.

In the initialization phase, the variables are set. nRow and nColumn are set according to the size of the room, Then the number of variation of the degrees are set. Right now, the code is set to utilize 0, 45, 90, 135, 180, 225, 270, and 315 degrees. Then it initializes pheromone deposit matrix and the node value matrix. Pheromone deposit matrix is matrix that stores the pheromone level. Node value matrix is the matrix that stores the value of a node. [Row, column, varying degree] matrix represents both the pheromone deposit matrix and node value matrix. The pheromone matrix is initialized with value of 1 as usual. The node value matrix is initialized with 0. The normal algorithm uses somewhat greedy algorithm and calculates the difference between a node to the another. However, it is near impossible due to the sheer size of the matrix. (50x50x8 matrix for 50x50 room.) So the adaptive algorithm is used: the value of going to the node is differentiated and adapted by adding up the resulting values to it.

The running phase is where the ants are controlled. The definition of the node in the problem sense is the location of the camera. The problem is understood as the multilayered graph, with each layer as a camera placement. Delta value matrix is for adaptation, and it is used to update the value matrix (which heavily influences the value matrix). Movement of an ant is also record at the delta value matrix, and the matrix is used for the pheromone update too. For each ant, the empty camera list is generated. Based on the pheromone value and value of the node value, next camera placement is chosen. The formula for choosing the next node is the following:

where:

The process of choosing nodes is repeated according to the number of the camera for each ant. After all the ants are done, half of the pheromone is evaporated. Then the pheromones are added according to the number of ants visited the node and the value of the road. The following equation is the formula for updating the pheromone matrix.

where:

At the same time, the value matrix is adapted. It is because the node in a possible solution is not in a linear relationship with the score of solution. Also, due to prolonged running time, the value of the camera position is not assessed in the beginning. Therefore, the value of the position is adjusted every iteration. The formula for the adjustment is the following.

The idea is to update the value of the position in each iteration. Initially, each position is treated equally. The equal value is unfair result; therefore, bias should be added. For the first iteration, given enough ants, it is likely that the ants will visit every node. Every node is then assessed and the bias is added. If the node provides greater benefit, it will be more biased to take the node.

The returning phase is a simple phase. In this phase, the camera positions in accordance with best scores are picked and put into the camera list. The number of cameras picked depends on the number of the cameras the user desires. Then the camera list is returned with the score of the camera list and the pheromone deposit matrix.

# Hand Iteration

To do the hand iteration, the problem has been simplified. Instead of a room 4 by 4, room of 2 by 2 will be used with 4 variations of the degrees. 2 cameras will be placed. 3 ants will be used. Alpha beta values are set to 1 for convenience.

## Initialization Phase

Pheromone deposit matrix and value matrix is initialized.

Pheromone deposit matrix (:,;,0)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 1 |

Pheromone deposit matrix (:,:,-90)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 1 |

Pheromone deposit matrix (:,:,-180)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 1 |

Pheromone deposit matrix (:,:,-270)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 1 |

Value deposit matrix (:,;,0)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 1 |

Value deposit matrix (:,:,-90)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 1 |

Value deposit matrix (:,:,-180)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 1 |

Value deposit matrix (:,:,-270)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 1 |

## Iteration 1

Total Probability matrix (:,;,0)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 1 |

Total Probability matrix (:,:,-90)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 1 |

Total Probability matrix (:,:,-180)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 1 |

Total Probability matrix (:,:,-270)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 1 |

It is equal distribution in the first iteration. 2 points are randomly chosen. Each ant chose:

Ant 1 CamList:

[1,1,-90]

[2,2,-270]

Ant 2 CamList

[1,1,-90]

[1,2,0]

Ant 3 CamList

[2,1,-180]

[1,1,-90]

Then the choices are evaluated. The scores are

Ant 1: 0.404

Ant 2: 0.202

Ant 3: 0.202

The Value Matrix is updated:

Value deposit matrix (:,;,0)

|  |  |
| --- | --- |
| 1 | 1.202 |
| 1 | 1 |

Value deposit matrix (:,:,-90)

|  |  |
| --- | --- |
| 1.808 | 1 |
| 1 | 1 |

Value deposit matrix (:,:,-180)

|  |  |
| --- | --- |
| 1 | 1 |
| 1.202 | 1 |

Value deposit matrix (:,:,-270)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 1.404 |

Then the pheromone deposit matrix is updated.

Pheromone deposit matrix (:,;,0)

|  |  |
| --- | --- |
| 0.5 | 1.702 |
| 0.5 | 0.5 |

Pheromone deposit matrix (:,:,-90)

|  |  |
| --- | --- |
| 2.308 | 0.5 |
| 0.5 | 0.5 |

Pheromone deposit matrix (:,:,-180)

|  |  |
| --- | --- |
| 0.5 | 0.5 |
| 1.702 | 0.5 |

Pheromone deposit matrix (:,:,-270)

|  |  |
| --- | --- |
| 0.5 | 0.5 |
| 0.5 | 1.904 |

## Iteration 2

Total Probability matrix (:,;,0)

|  |  |
| --- | --- |
| 0.5 | 2.046 |
| 0.5 | 0.5 |

Total Probability matrix (:,:,-90)

|  |  |
| --- | --- |
| 4.173 | 0.5 |
| 0.5 | 0.5 |

Total Probability matrix (:,:,-180)

|  |  |
| --- | --- |
| 0.5 | 0.5 |
| 2.046 | 0.5 |

Total Probability matrix (:,:,-270)

|  |  |
| --- | --- |
| 0.5 | 0.5 |
| 0.5 | 2.673 |

Each ant chose the place with highest probability:

Ant 1 CamList:

[1,1,-90]

[2,2,-270]

Ant 2 CamList:

[1,1,-90]

[1,1,-270]

Ant 3 CamList:

[1,1,-90]

[1,1,-270]

Then the choices are evaluated. The scores are

Ant 1: 0.404

Ant 2: 0.404

Ant 3: 0.404

The Value Matrix is updated:

Value deposit matrix (:,;,0)

|  |  |
| --- | --- |
| 1 | 1.202 |
| 1 | 1 |

Value deposit matrix (:,:,-90)

|  |  |
| --- | --- |
| 3.020 | 1 |
| 1 | 1 |

Value deposit matrix (:,:,-180)

|  |  |
| --- | --- |
| 1 | 1 |
| 1.202 | 1 |

Value deposit matrix (:,:,-270)

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 2.616 |

Then the pheromone deposit matrix is updated.

Pheromone deposit matrix (:,;,0)

|  |  |
| --- | --- |
| 0.25 | 0.851 |
| 0.25 | 0.25 |

Pheromone deposit matrix (:,:,-90)

|  |  |
| --- | --- |
| 2.366 | 0.25 |
| 0.25 | 0.25 |

Pheromone deposit matrix (:,:,-180)

|  |  |
| --- | --- |
| 0.25 | 0.25 |
| 0.851 | 0.25 |

Pheromone deposit matrix (:,:,-270)

|  |  |
| --- | --- |
| 0.25 | 0.25 |
| 0.25 | 2.174 |