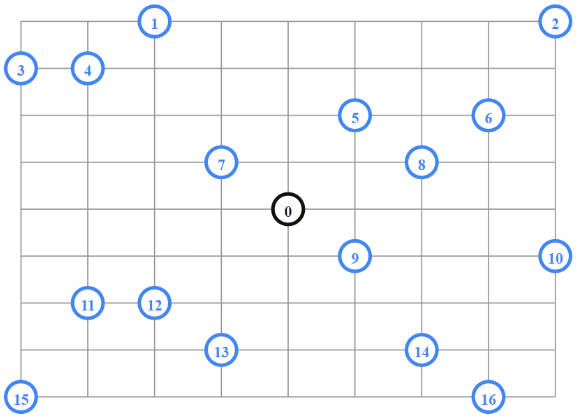
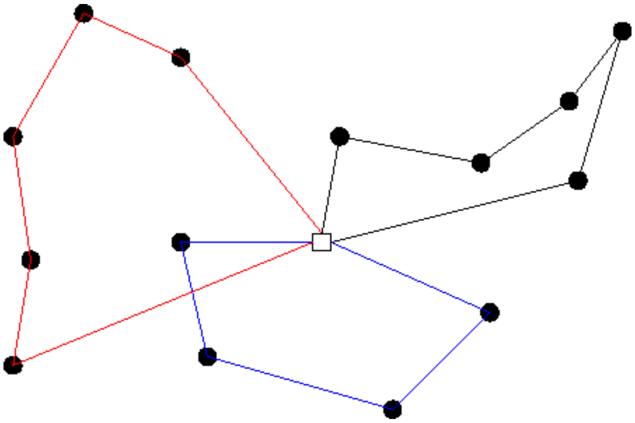
VRP: Vehicle Routing Problem using ACO

Vincent Geiger, vhg2106

Intro:

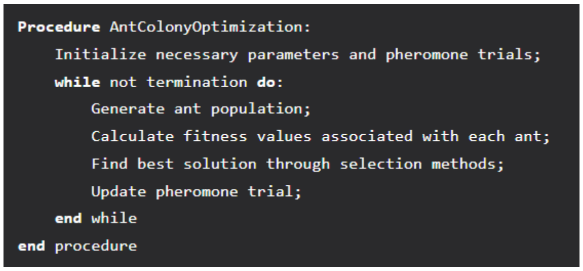
This project would deal with VRP, or the Vehicle Routing Problem. In this problem a given set of vehicles is defined. There are items which need to be delivered to various customers. There is also a company location defined at which all vehicles begin at. The problem stems from finding the best choices of routes for each vehicle to take such that all deliveries are fulfilled and the travel cost of all vehicles is minimized. From Wikipedia, “The vehicle routing problem (VRP) is a combinatorial optimization and integer programming problem which asks ‘What is the optimal set of routes for a fleet of vehicles to traverse in order to deliver to a given set of customers?’”



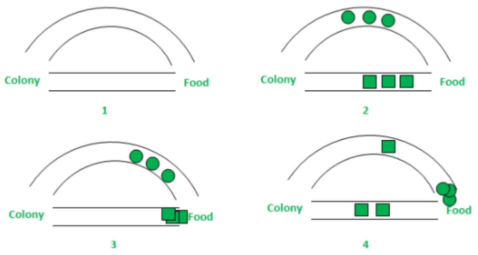


Details:

ACO, or Ant Colony Optimization is an algorithm which can be used for combinatorial optimization problems, such as the TSP problem, VRP, and so on. The general idea is to use artificial “ants” alongside pheromones to determine optimal paths. Put more succinctly from Wikipedia, “the ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs.” From geeksforgeeks, pseudocode of ACO alongside a visualization:



```
Procedure AntColonyOptimization:
    Initialize necessary parameters and pheromone trials;
    while not termination do:
        Generate ant population;
        Calculate fitness values associated with each ant;
        Find best solution through selection methods;
        Update pheromone trial;
    end while
end procedure
```



The Vehicle Routing Problem can then be reduced to such artificial ants and then can be quite easily built up using modifications of this general algorithm. In order to better optimize some other heuristics may have to be added.

Parallelization:

The ACO portion of the program can be parallelized in a pretty straightforward manner. The artificial ants will be split up into different subsets for each core. We then allow the pathing to occur for each of these and then collect the returned pathings to generate our new pheromone matrix. The actual computation of this matrix can also be parallelized if found to be a bottleneck. The speeds for various cores will then be recorded. Also, if more complexity is needed, CVRP can be solved instead.

References:

https://en.wikipedia.org/wiki/Vehicle_routing_problem
https://developers.google.com/optimization/routing/vrp
https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms
https://www.geeksforgeeks.org/introduction-to-ant-colony-optimization/