Problem Description

The overall goal of the problem is to deliver a set of packages (K) using a set of delivery vehicles (N) that exist in a map containing M locations. Each vehicle can only hold one package at a time, and each package has a pickup and drop off location. All will vehicles start their routes at the garage, also found at an arbitrary location on the map. The return trip to the garage for each truck after each delivery route was ignored, as it is an unavoidable overhead that every path will include.

For the first iteration of the problem, it was assumed that the map in which the solution was found was an environment where all locations were connected to their nearest neighbours in a square grid. All distances between neighbours throughout the map were uniform, and the map did not include bridges, parks, or highways. Speed limits, traffic, and one way streets were also ignored.

Solution Description

The problem given has a number of attributes that make it ideal for a solution based on a search algorithm. It has a well defined state space (the city map), a start state (packages have predefined pickup and dropoff locations within the map), an end state where all packages are delivered, and transition operators (a vehicle picking up and delivering a package). An AI search algorithm will find a combination of deliveries leading from the start state to the end goal.

Determining the optimal solution for any given set of vehicles and packages is an NP-Complete problem. This means that although any solution given can be verified in a polynomial time, there is no known method to find the optimal solution efficiently. This means that an exhaustive search of all possible paths, for each possible number of active vehicles, is only practical for a very small values of K, N, and M.

Rather than performing an exhaustive search, a heuristical and deterministic method was designed to find acceptably good solutions to the problem. Known as an A* search, search paths that are deemed too “expensive” will not be explored, and the first path to the goal state found will be the “best” solution.

Implementation Description

The implementation of the solution contained three important aspects: The active search process, the current state of the search, and the state space. These were represented as interdependent classes, and each had responsibilities related to finding the end goal state.

State Space: This contained all of the information and methods used to represent the problem definition. It featured containers for vehicles, packages, a representation of the garage, as well as the cartesian coordinates for all of the above. It also contained public methods to determine the distance between two points within the state space, as well as a variety of methods to determine closest dropoff and pickup locations from any given position. Closest dropoff and pickup locations were simple algorithms that searched the pickup and dropoff data structures. Distance between two points was calculated using their cartesian coordinates, and ignored the actual path a vehicle would take. This reduced the time-space complexity of the calculation to a constant.

State: A representation of the current state of the problem. This class kept track of the active vehicles, the inactive vehicles (those waiting at the garage and not yet needed for the solution), the packages still waiting on delivery, and the total distance traveled by all vehicles. It also contained methods associated with calculating the heuristics needed to determine the next state, as well as methods needed to update the current state after a delivery was made.

Search: This class initializes the state space of the problem, and contains methods to find the end state. The search is as follows: while the current state is not “finished”, assign the next package to be delivered, update the current state to reflect the delivery of that package, and if necessary (and possible) activate a vehicle waiting at the garage so that it may be considered in the next state of operations.

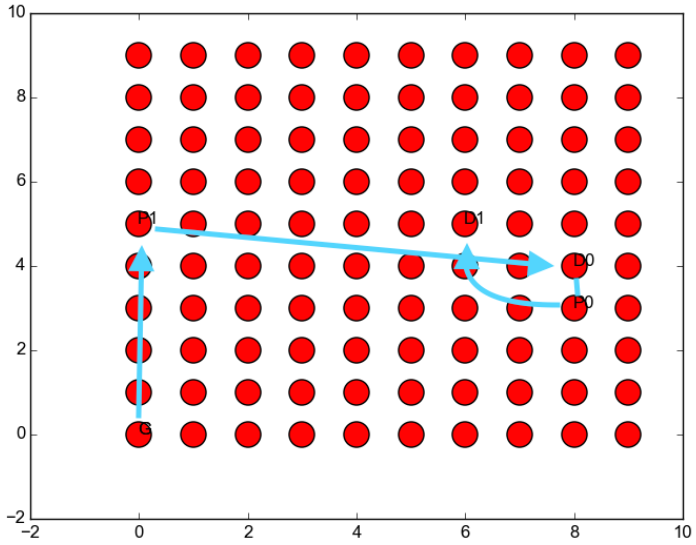
For each update to the current state of the problem, the search for the next package to be picked up and delivered was aided by the following heuristic:

distance to closest active vehicle + distance from dropoff to closest pickup location.

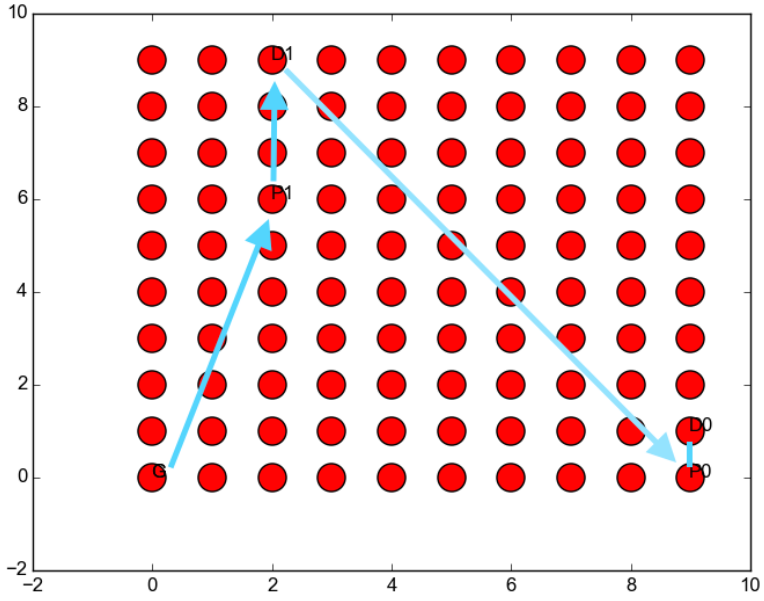
The package with the lowest overall cost was chosen as the next delivered, and was done so by it’s closest delivery vehicle. It was theorized that this would minimize distances driven by individual vehicle while also ensuring that packages in close proximity to each other would be picked up and delivered in sequential order. This approach also only considers vehicles already in the field, and a single active vehicle at the garage if there is a non-empty list of vehicles waiting at the starting point.

Results

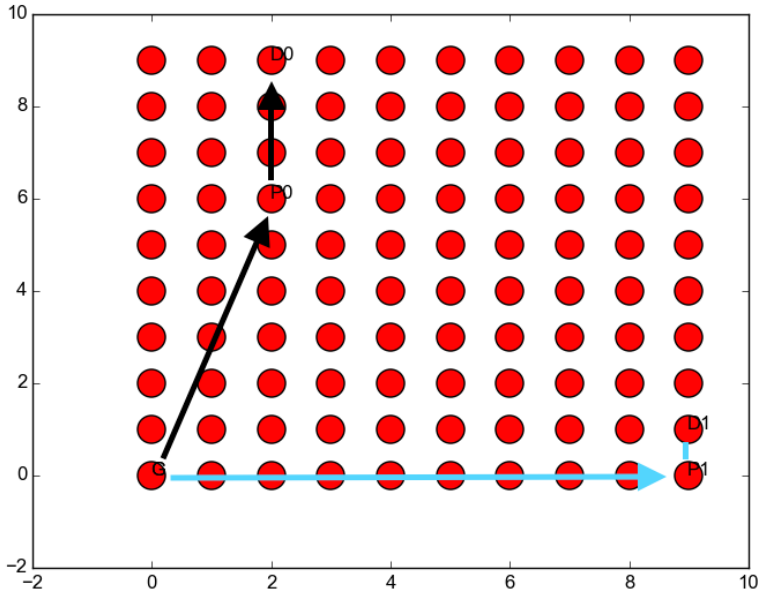
Proof of concept test. 1 vehicle, 2 packages, and a total distance of 15 units.



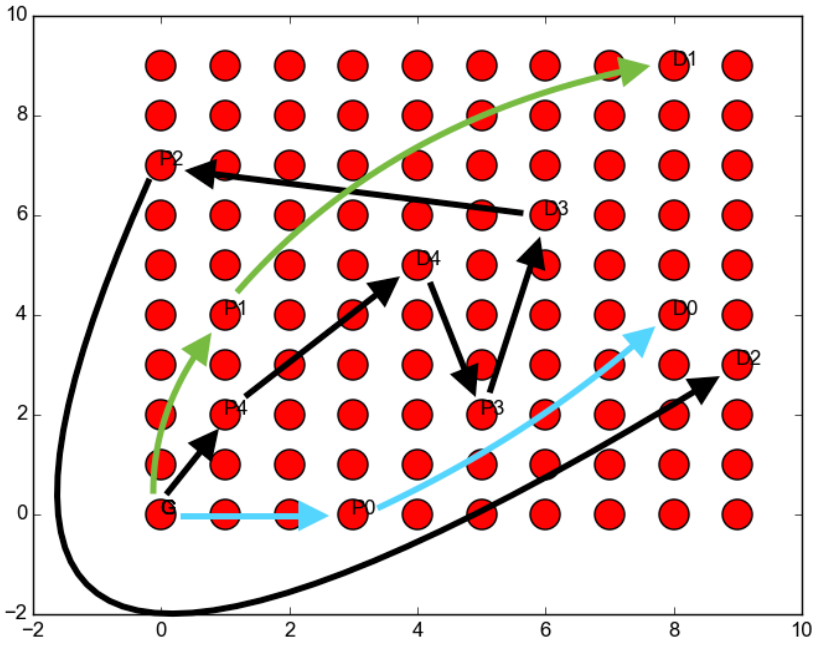
In this case the algorithm appears to have chosen an optimal route with only one vehicle to work with. The following graph is of yet another simple example, however it was found that using two vehicles instead of one lead to a shorter overall route. With one vehicle:



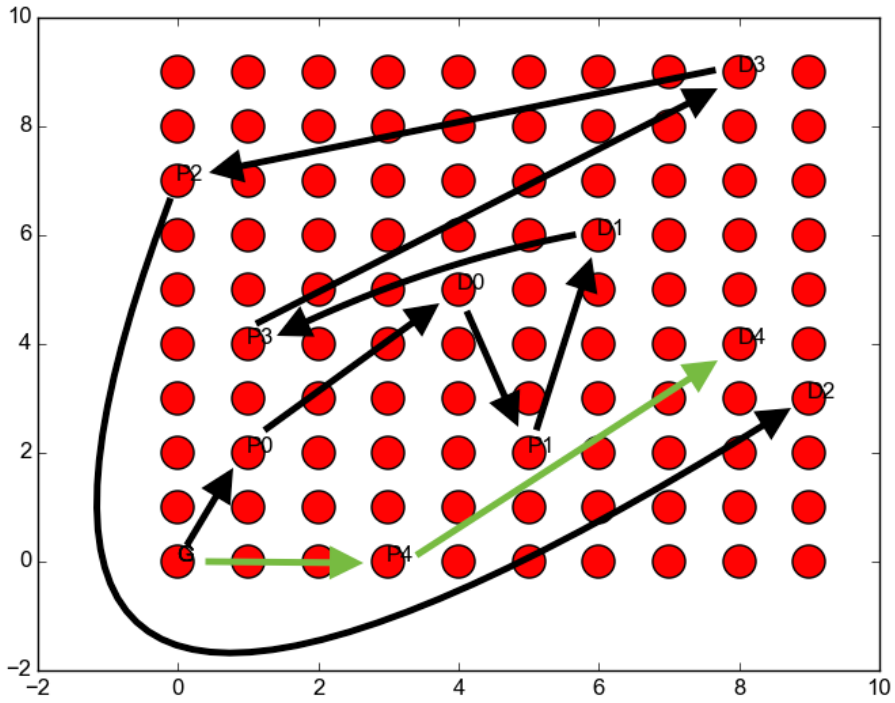
The same problem space with two vehicles available:



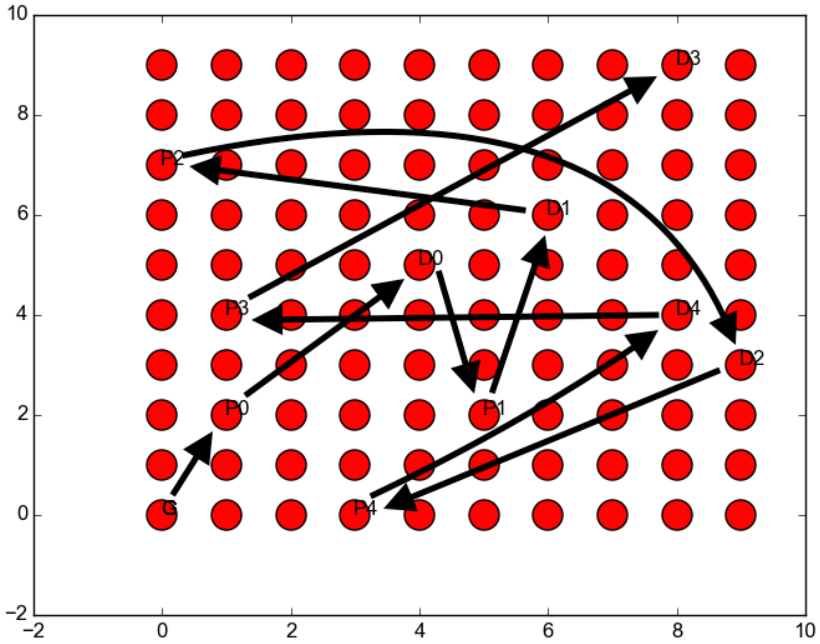
The distance traveled with one vehicle was 22 units versus 19 units with two. More complex scenarios were also examined, and in each case the AI program found a solution using several vehicles that was more optimal that solutions when it was restricted to a smaller group of delivery trucks. An example with 5 vehicles available, 3 used. Total distance of 51 units:



The following is the same problem space with only two vehicles available. Total distance of 55 units:



And finally the path taken with only one vehicle. Total distance traveled was 58 units:



Tests on larger spaces were also performed:

Table 1. Summary of Test in Large State Space

Packages	Vehicles Used	Size of Space	Total Distance
50	3	10,000 locations	1,591 Units
50	2	10,000 locations	1,612 Units
50	1	10,000 locations	1,645 Units

Table 2. Summary of Test in Large State Space

Packages	Vehicles Used	Size of Space	Total Distance
20	3	10,000 locations	1,535 Units
20	2	10,000 locations	1,562 Units
20	1	10,000 locations	1,679 Units

Computation time with this particular implementation appeared to be linear with respect to the size of the state space. This is due to the simplistic nature of the heuristic and search computations.

Conclusions

Preliminary results from testing the AI program seem to indicate that it is capable of finding efficient routes for a fleet of delivery vehicles operating in a uniform grid-like map. It's performance in small state spaces was observed to be similar to what a human would deem optimal.

In larger state spaces where the validity of the solution could not be confirmed by visual inspection or compared to the results of an exhaustive search, solutions where the AI opted to use more than one vehicle were analysed. In these state spaces, the number of vehicles were restricted to a lower number than what the AI had determined was optimal, and the solutions of those searches were compared to each other. It was found that as the number of available vehicles decreased, the total distance traveled by the fleet increased. This correlation does not prove that the best solution has been found. It does however indicate that a practical solution was found.

Future iterations of this solution will need to operate in more complex state spaces that better simulate real-world situations. It is likely that the heuristic used in this situation will still be applicable to the updated state spaces, however it's computation will need to take into

account the actual paths taken by vehicles. This will likely increase the space-time complexity of the overall program into logarithmic or exponential scales depending on the implementation.