Multiple Amazon Truck Routing

The Travelling Salesman Problem Simulated in C++

By Trey Gower

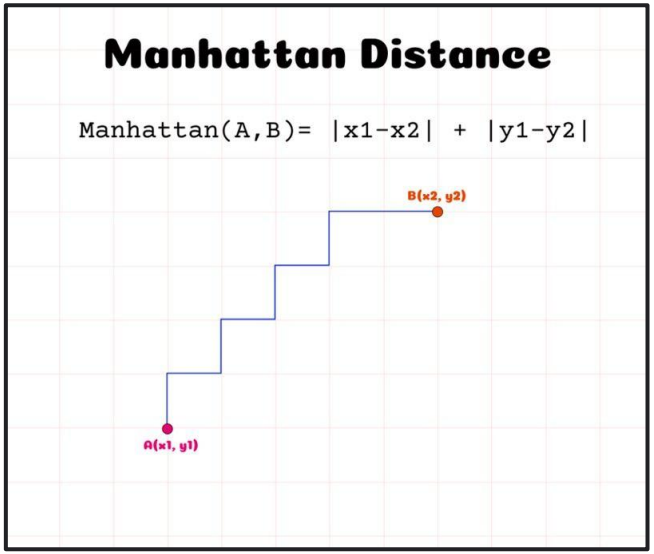
Objective and The TSP

- The Traveling Salesman Problem says, “Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?"
- The first task was to minimize the total distance that two delivery trucks have to travel corresponding to the Traveling Salesman Problem.
- The second task, the program must optimize and include those with Amazon prime and those whose delivery date is approaching.
- Using C++ vectors we stored addresses and some other info from a customer such as date to be delivered
- Using C++ Vector of Vectors to create a matrix to store addresses into a larger address list.

Manhattan Distance

- Having Established the use of matrices and vectors in our code, we can use the L1 Norm to compute distances.

- This method is known as The Manhattan Distance and allows us to compute distances on a grid of any size.



Greedy Algorithm

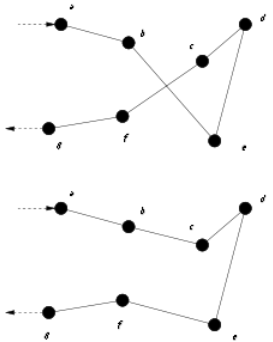
- The simplest optimization method, allowing a bases for future implementation of other, quicker, algorithms.

- Algorithm is designed to continually search for the next locally optimal address based on The Manhattan Distance calculation.

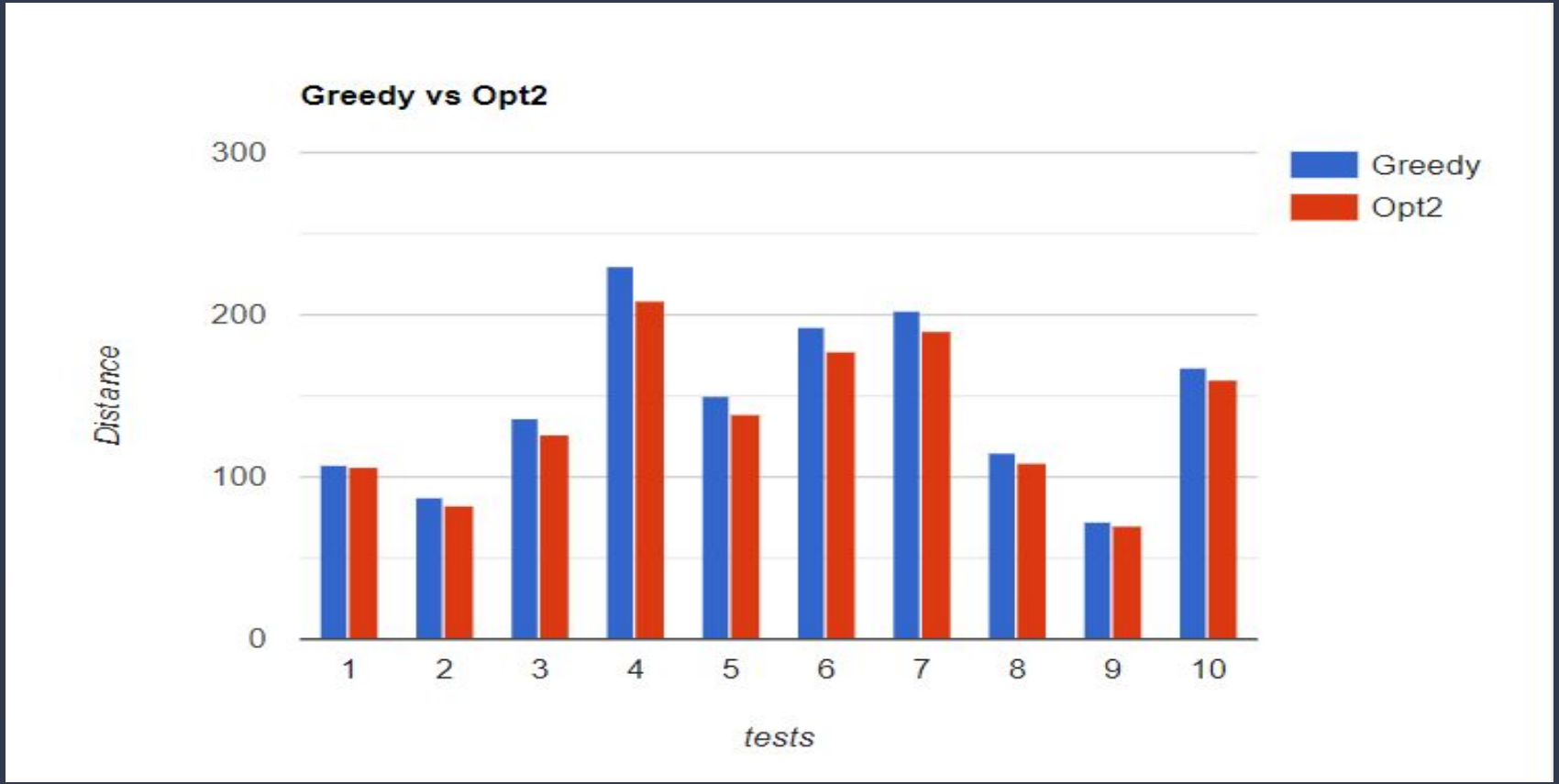
```
int index_closest_to(Address otheraddress){
  //finds the next index of closest address to previous address to sort addresses
  double mindist, currentdist;
  int closestindex = 0;
  for (int i=0; i < alladdresses.size(); i++){
    if (i == 0){
      mindist = otheraddress.ManDistance(alladdresses[i]);
    }else{
      currentdist = otheraddress.ManDistance(alladdresses[i]);
      if (currentdist < mindist){
        mindist = currentdist;
        closestindex = i;
      }
    }
  }
  return closestindex;
}
```

Opt-2 Heuristic Algorithm

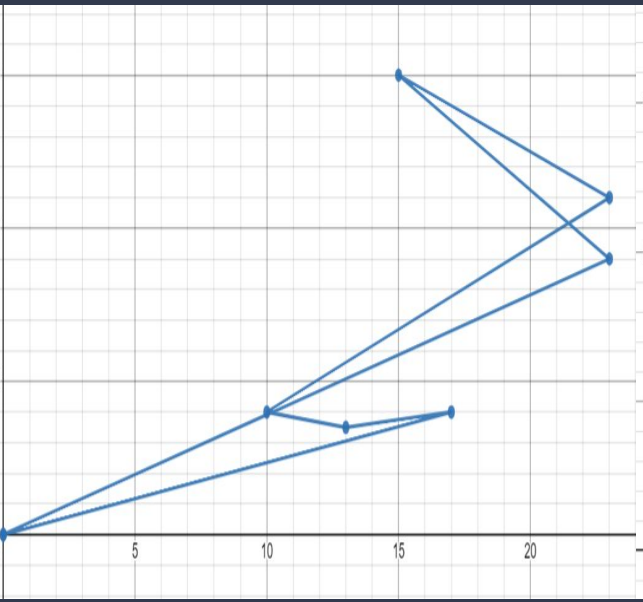
- Opt-2 takes a route that crosses over itself and performs a few tasks to untangle the path (Global Scope)
- Opposed to Greedy Algorithm Opt-2 is intended to decrease time complexity.
- Opt-2 is more computationally expensive, usually $O(n^3)$ where n is the number of vertices in route.
- We further optimized Opt-2 by performing $O(n)$ operations, removing previous edges and adding two different edges, thus only adding and subtracting those new different edges.



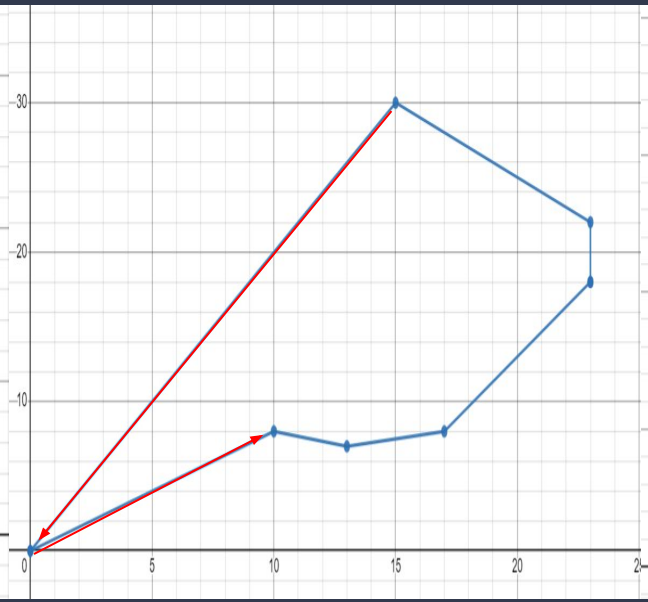
Distance vs. # of Tests



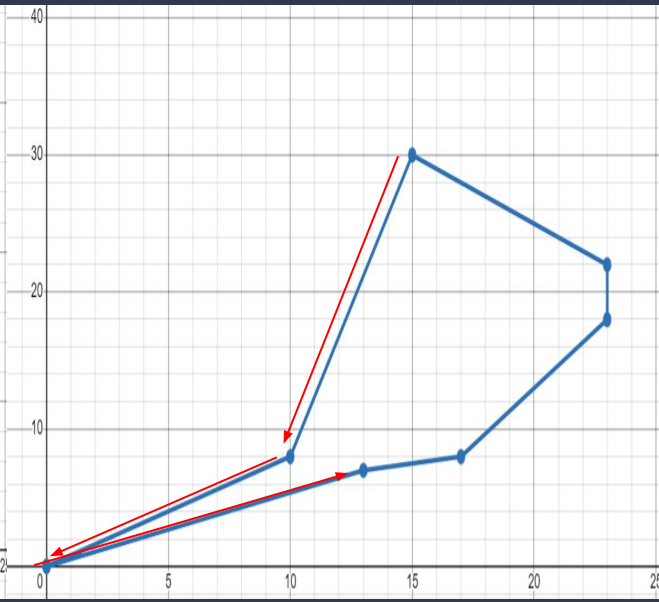
Original Route
Total Distance: 138 Units



Greedy Route
Total Distance: 108 Units



Opt-2 Route
Total Distance: 106 Units



Multiple Truck Comparison of Prime vs. Non-Prime

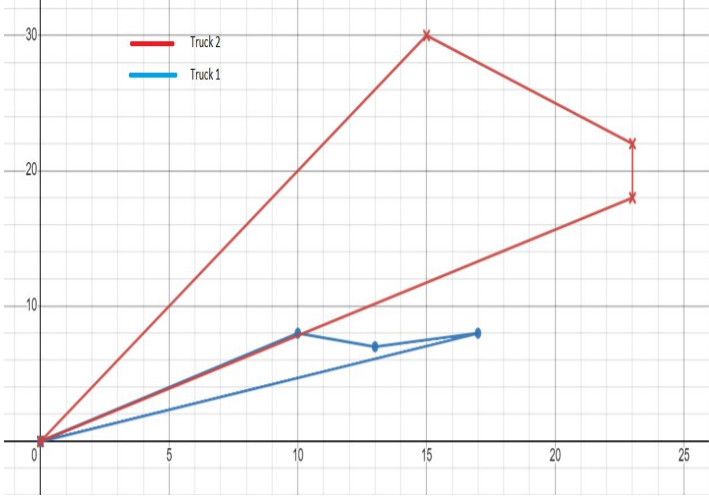
Case 1:
Addresses within an address list were able to be switched and optimized between both trucks.

Case 2:
Trucks could not swap addresses with each other since one address list was prime and one non-prime.

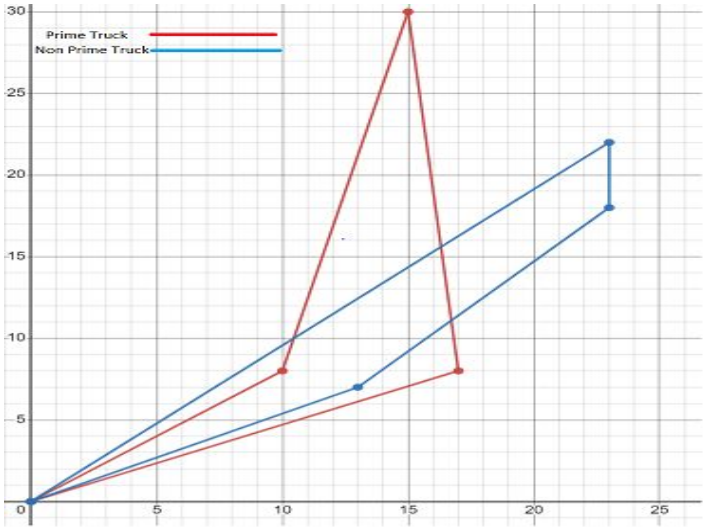
Non-Prime (Blue line): Distance = 52 units

Prime (Red line): Distance = 106 units

Total Distance = 158 units



Prime (Red line) : Total distance = 94
Non-Prime (Blue line) : Total Distance = 91
Total Distance = 179



Conclusion and Findings

- Opt-2 Heuristic Algorithm consistently calculated routes of lesser distance in every case due to scope
- Once Multiple Trucks were added we intended to simulate real life scenarios (Prime vs. Non-Prime shipping)
- With random assignment of addresses a significant amount of distance was added when Amazon Prime was a factor with any algorithm
- Our findings raised ethical questions regarding Amazon's labor policies and hours of work
- Possible next steps would be to optimize addresses before being placed onto trucks or incorporate a network of trucks

Thank You!

Opt-2 optimization with 40+ data points: