Algorithms for Vehicle Routing

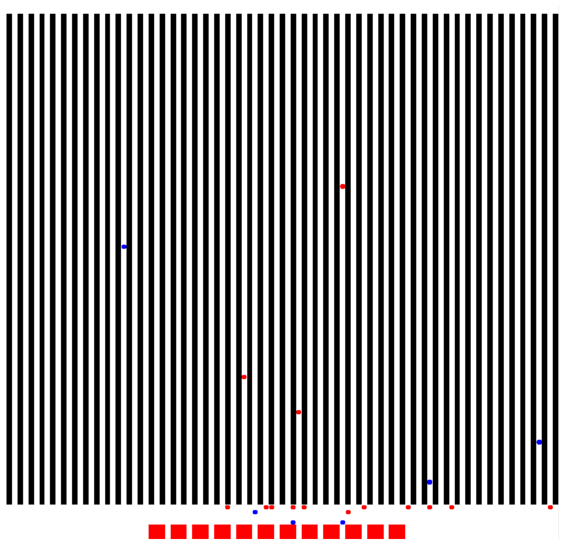
Kyle Davis* and Daniel Hull†
Mentors: Henrik I. Christensen‡ and Prasad Tetali§

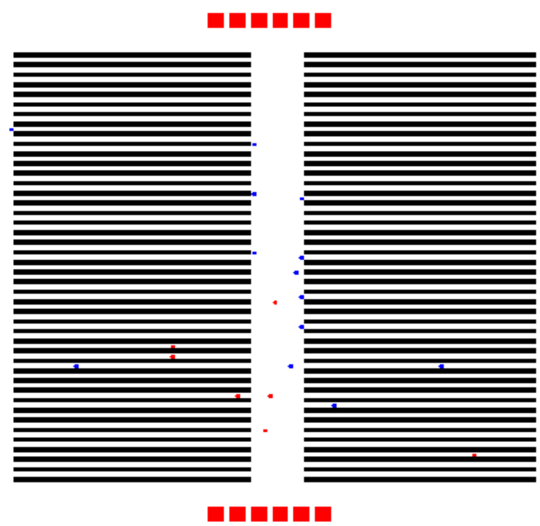
1 Introduction

Vehicle routing has been a challenging problem since the advent of self driving vehicles and robots. This problem is interesting due to its wide range of applications. Vehicle routing algorithms can be used to route packages through large networks, analyze and help improve traffic patterns, and be used to improve the efficiency of shipping warehouses. Working with several shipping companies we have tailored our algorithms to achieve the best performance under the last application. Everything from topology of the warehouse, distribution of the jobs, ordering of the jobs, and the vehicle routes influence the efficiency a warehouse can achieve.

2 Problem Statement

Given a topology and a list of packages, move the packages from their source to destination while minimizing the total duration. A topology is a 2-D grid containing sources, destinations, and vehicles. Vehicles are not allowed to collide with other vehicles, sources or destinations. Sources are black, destinations are red, and vehicles are red or blue circles depending on if they are currently carrying a package. The figures below represent two different topologies with the vehicles in the process of running their respective routes.





Topology 1 and 2

The list of packages is given in a specific order and describes the source each package is originally located at, and the destination it must travel to. Usually there is a partial ordering that must be maintained when delivering packages, the packages must arrive in the order given as input to each destination. The problem has been relaxed to ignore the ordering restriction as well as allowing the vehicles to have infinite acceleration. This means that vehicles can accelerate to their maximum (unit) velocity and deccelerate to zero velocity instantaneously.

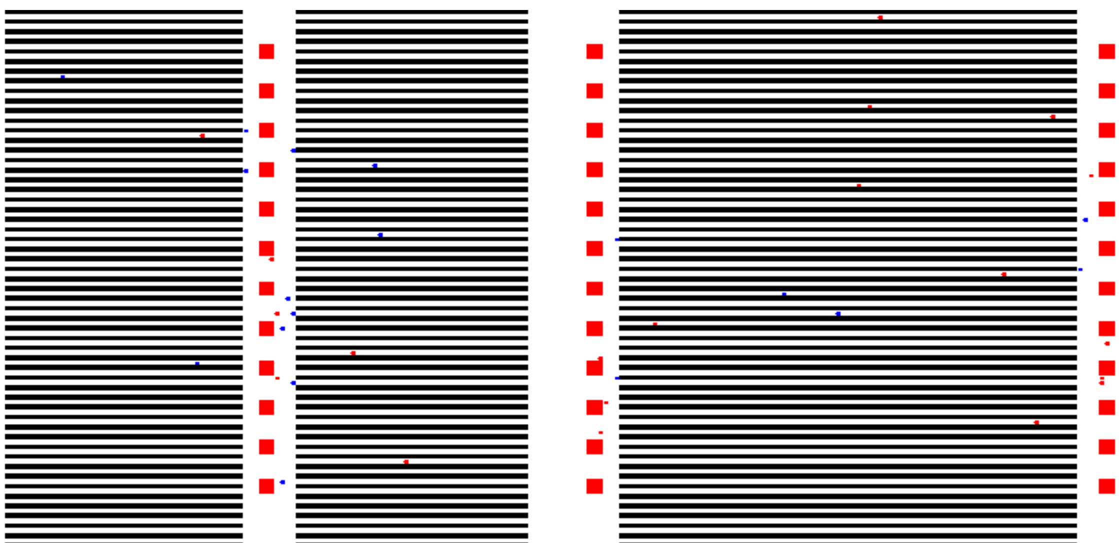
*CS Undergraduate Student, College of Computing, Georgia Institute of Technology. Email : *kdavis@gatech.edu*
†CS Undergraduate Student, College of Computing, Georgia Institute of Technology. Email : *dhull6@gatech.edu*
‡School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA-30332.
§School of Mathematics and School of Computer Science, Georgia Institute of Technology, Atlanta, GA-30332.

3 Solution Strategies

There were many aspects to this problem. Our initial research was focused on determining which parts of the problem had the greatest impact on overall effeciency. This will pave the way to more focused research related to the most important parts of vehicle routing. With a goal of implementing these solutions and improving the performance of warehouses around the world.

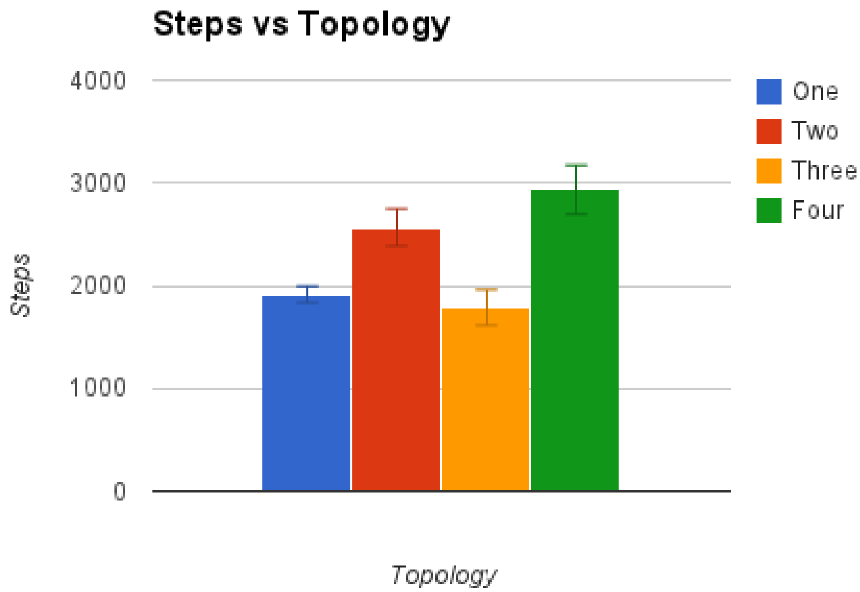
3.1 Topology

When desigining a toplogy there is a tradeoff beween space and speed. With more space vehicles have more room and avoiding collisions becomes easier allowing for the faster transporting of packages; however, space is expensive and can not be sacrificed without significant increases in performance. We tested four different topologies that were built with approximately the same space effeciency but the sources and destinations rearranged, number of vehicles was held constant throughout our testing. The first two topologies can be seen above; the remaining two are pictured below.



Topology 3 and 4

When holding other aspects of the problem constant there was a wide variance in performance when changing topologies. The table below contains statistics about the simulated performance of each topology.

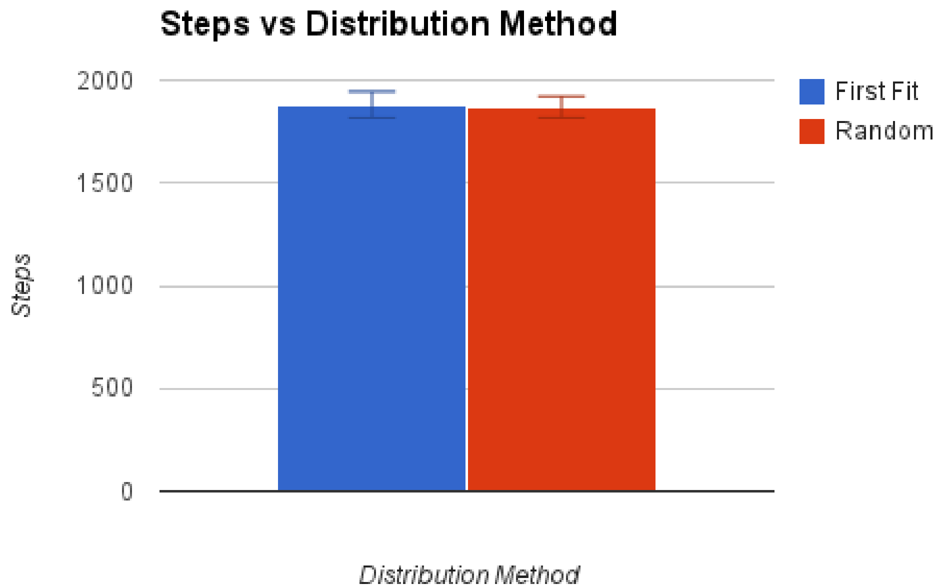


3.2 Job Distribution

A job does not have a specific vehicle it must be assigned to. Distributing the jobs to the vehicles is another part of the problem that must be addressed. If each job's completion time can be estimated then to minimize total time the jobs should be evenly distributed among all robots. This minimizes any given robots total completion time. This can be modeled as a 1-D bin packing problem. Each robot is a bin with a maximum capacity. Pack the bins with jobs using their estimated completion time, adding more robots as necessary.

In order to conform to our original problem some additional work is required. A binary search is performed to find the minimum, maximum capacity that does not require more than the predetermined number of robots. The packing using this maximum capacity is then given as the job distribution.

A simple first-fit solution was used as the bin packing algorithm. It was compared against randomly assigning jobs to robots without attempting to acheive an even distribution. The resulting performance difference is presented below.



3.3 Job Ordering

Once the jobs are fixed to a robot, the order in which each robot completes its jobs can be altered. Intuitively having all of the robots work on an order in the same part of the warehouse at the same time will causes more congestion than if the robots' current jobs are spread out evenly through the warehouse for a given point in time. The algorithm we use to evenly distribute the jobs through the warehouse uses a simulated annealing approach [2].

1. **Cache paths:** The first step is to compute and store the paths between every source and destination, taking into account the statinoary obstacles of the topology. Let $P_{i,j}(t)$ be the point on the path between source S_i and destination D_j at time t. We will be using these values as estimates since we cannot take into account robots colliding yet, estimates are fine for our purposes.

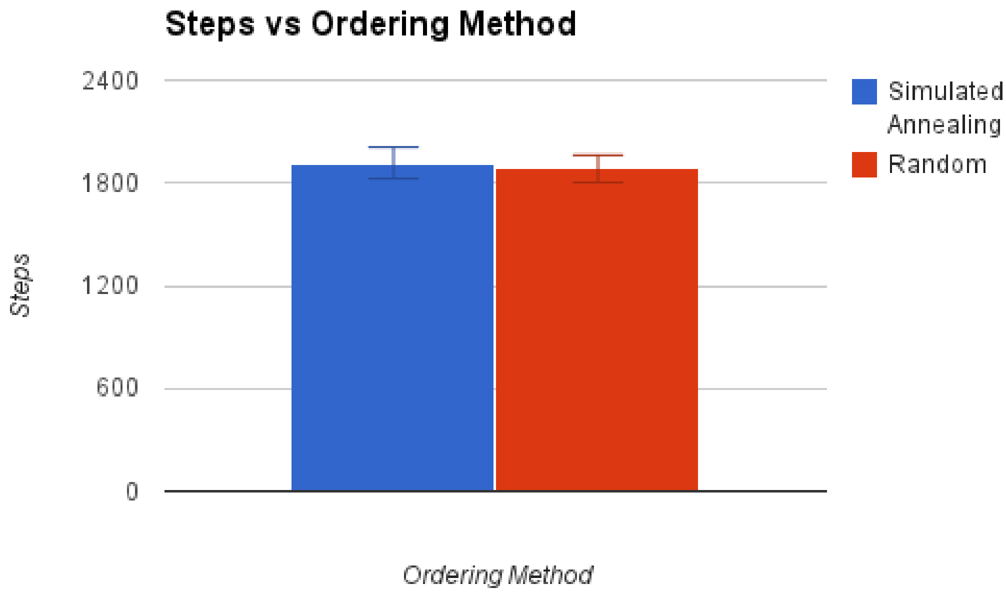
2. **Randomly pick a robot to optimize:** At random decide on a robot. For every pair of jobs assigned to the robot swap them, and if they increase the fitness function keep the change with a probability that increases to 1 over time.

3. **Compute interesting points:** We will define interesting points to be the 3 points location at the start of the time interval, the middle of the time interval, and the end of the time interval. These points in time must be recomputed each time the fitness function updates the arrangement.

4. **Fitness function:** The fitness function can be computed by going through every pair of jobs J_a, J_b with sources and destinations S_a, D_a, S_b, D_b, and summing over every interesting point in time relative to the start of the jobs, t. $\sum \text{distance}(P_{a,a}(t), P_{b,b}(t))^2$. Maximizing this fitness function will evenly distribute the paths.

5. **Repeat and cool:** repeat starting at step 2 and cool down the annealing until a maximum is found.

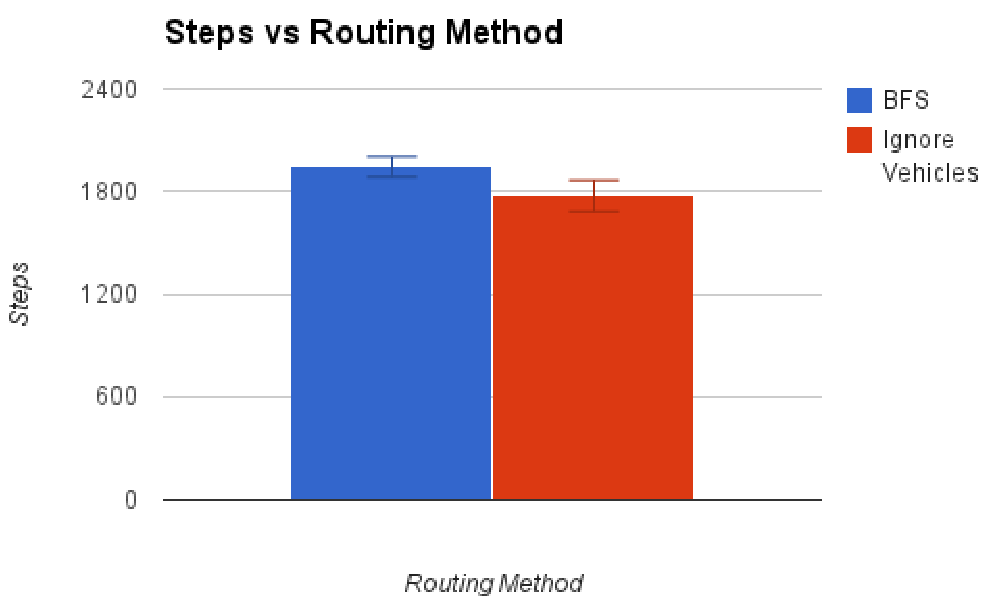
We compared this simulated annealing approach to randomly ordering the jobs within robots. The results comparing the two different strategies are below.



3.4 Vehicle Routes

To find vehicle routes a simple algorithm that performs a breadth first search (BFS) [1] at every time step is used. For each robot, determine where it needs to move towards next, perform a BFS using the robot as a starting point. Move along the path if one exists, otherwise do nothing. This very simple algorithm treats robots as stationary objects for each time step, so if a robot is in an aisle blocked off by two other robots it will not move until the other robots are out of the way.

We compared this algorithm to an optimal solution where robots were not considered obstacles. This allowed robots to always take the shortest path. The results can be found below.



4 Conclusions

After running many experiments two parts of the problem that were explored had a significant impact on the number of steps to deliver all packages. The most significant factor was the topology. Topologies 2 and 4 required many more steps than topologies 1 and 2. The construction of topologies 2 and 4 included two separate central delivery locations. This appeared to slow everything down compared to the topologies with one cental delivery location.

The different strategies of routing vehicles also led to significant variance in the number of steps required. When collisions between vehicles were ignored the optimal routing was achieved for the fixed distribution and ordering of jobs. Using BFS and treating vehicles as obstacles resulted in an average number of steps that were more than 1 standard deviation away from the optimal. There is a lot of room for improvement in how the routes are found.

Distributing and ordering the jobs did not significantly impact the number of steps required. The solution strategies proposed for each part of the problem did not achieve results better than a randomized strategy. This could mean that our strategies are flawed, or these parts of the problem will not be a large contributing factor to the overall performance.

References

[1] Charles E. Leiserson Ronald L. Rivest Cormen, Thomas H. *Introduction to Algorithms*. 1990. 5

[2] John Tsitsiklis Dimitris Bertsimas. Simulated annealing. *Statistical Science*, 8(1):10–15, 1998. 4