Project Proposal

Project Title: Optimizing the route for large, frequently stopping trucks to reduce their impact on traffic and the environment.

Author: Kazuya Erdos

Date: 11/02/20

Project Definition:

The overall aim of this project is to model an improved system of garbage collection for small, suburban areas. The models will all be made virtually, and any physical designs mentioned will only be hypothetical and abstract. The models should run on any personal computer, and if possible the project will be adapted to run as a mobile application. The program should run in a timely manner, and should not be too much of a burden to use (rendering it useless for the real world). It is expected that traffic caused by garbage trucks can theoretically be improved by the model created, and hopefully the information from the results can be used to help improve the conditions in the real world.

Background:

In suburban areas, garbage trucks often cause traffic jams by blocking large portions of the road and making frequent, sudden stops. How do the routes the truck takes and the size of the truck affect traffic in the local area? The goal of this project is to engineer a new system for garbage collection, which would be applied to the current routes of trucks used as well as smaller, less obtrusive vehicles. The new system's cost will be evaluated, and the cost of the system cannot be too much greater than any current methods. A computer algorithm of some sorts will be made, and such an algorithm must run in polynomial time and must be usable on a standard desktop computer or laptop with proper software installed. If time allows, a dedicated application will be made, and information about the road networks surrounding the user should be able to be imported from Google Maps, or similar map software. For testing and designing purposes, a randomly-generated environment for testing will be generated using an adaptation of the A-Star algorithm which factors in effect on traffic as a heuristic cost value.

On narrow suburban roads, large, frequently stopping garbage trucks obstruct other drivers' view of the road in front of them. Additionally, these large vehicles get only 3 miles per gallon of diesel they consume, which is only 10% of what a standard truck can get (Coren, 2016). Because of this hazard, other drivers are forced to slow their vehicles down and follow the truck from a distance, creating large amounts of traffic. More aggressive drivers will erratically brake behind the trucks, creating a ripple effect of brake use down the stretch of the road behind the truck. Thus a new system of garbage collection is proposed in this project.

Currently, most suburban garbage systems work as follows: A large truck with 2 workers will circulate the town for 5 days a week. Garbage is collected as the truck passes by a house, and pre-paid stickers are needed on the garbage bags in order for them to be collected. All garbage collected is centralized in the truck, hence requiring a large and heavy truck.

Though this project looks to change some of these procedures, some of them must remain the same. The 5-day route must be maintained in order to give the workers a full working week. Additionally, a Saturday route will be added with smaller trucks being used to collect garbage. Ideally, the large trucks would be replaced outright, but this project will engineer a solution that involves keeping the large trucks, since they would be very difficult to replace.

Search and routing algorithms are discussed extensively in this project, with the main algorithm used being A-Star. On a basic level, the algorithm works to connect nodes from start to finish in the most efficient way possible. In terms of navigating roads, this would mean the A-Star algorithm would find the shortest path from a start node to an end node. The figures to the right show improvement made on a wireless routing network with A-Star, with each iteration utilizing a more thorough version of the algorithm(Septiania, 2016). It should be observed how the path between the start and finish node gets more and more direct.

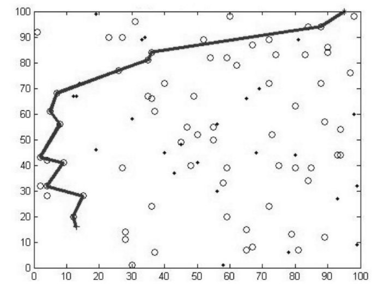


Fig. 1. Path formed using the initial A-star

In this project, 5 pairs of start and end nodes will be chosen semi-randomly to represent each of the 5 working days and an adapted version of A-Star will be used to find the best path between each pair of nodes. The algorithm will then record the number of leftover roads and re-try the algorithm for a set amount of tries, until the combination of paths with the least leftover roads is selected, and the routes are then adapted to include more of the roads, with the final Saturday route covering the remaining leftover roads. At this point it has been decided that the problem turns into the “Travelling Salesman” problem (Klarreich, 2020).

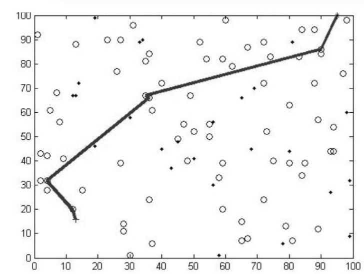


Fig. 2. Path formed using two heuristic functions

Experimental Design/Research Plan Goals:

Major Parts of the Project (rough outline) will continue to evolve over time and should be updated frequently. Make sure the goals are SMART oriented.
List IDV, DV, standardized variable/controls, experimental/control groups, iterations, etc., process of product design.
Materials List
Procedure

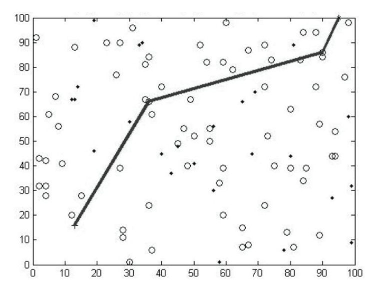


Fig. 3. Path formed using three heuristic functions

The first part of the project would be to create a modeling environment for the program to run. Initial ideas are JAVA, Net2Pan, and MatLab. Roads will be simplified down to straight lines on a grid to avoid overcomplication. The next step is to create realistic randomly generated roads using an algorithm. Once roads can reliably be generated, the program must be made to pick its start and end points for all 5 days. The next step is writing and implementing my version of the A-Star algorithm for each of the 5 pairs of points. The parameters of the program can be adjusted to create better results, such as how heavily the A-Star algorithm favors distance versus how it favors traffic impact. The next goal is for the program to be able to try many attempts of this routing process and find the best starting and ending points for each day after a chosen number of attempts. It could be possible for the program to try *every* possible combination of points, and if such a design runs in linear or polynomial time then it will be implemented.

Some initial models will be developed and they will be tested and evaluated based on cost models and their overall impact on traffic, as well as environmental impact.

Independent Variables: Algorithms and their implementations / approaches

- A-Star valuing distance over traffic or vice-versa

Dependent Variables: Estimated time taken to complete full route, number of turns taken, trucks required, estimated cost of implementation, distance traveled each day.
Standardized variable / controls: An intuitive route (done by driver or me?)
Control Group: Current methods used by towns or cities (Call town and ask how they do it?)
Iterations:

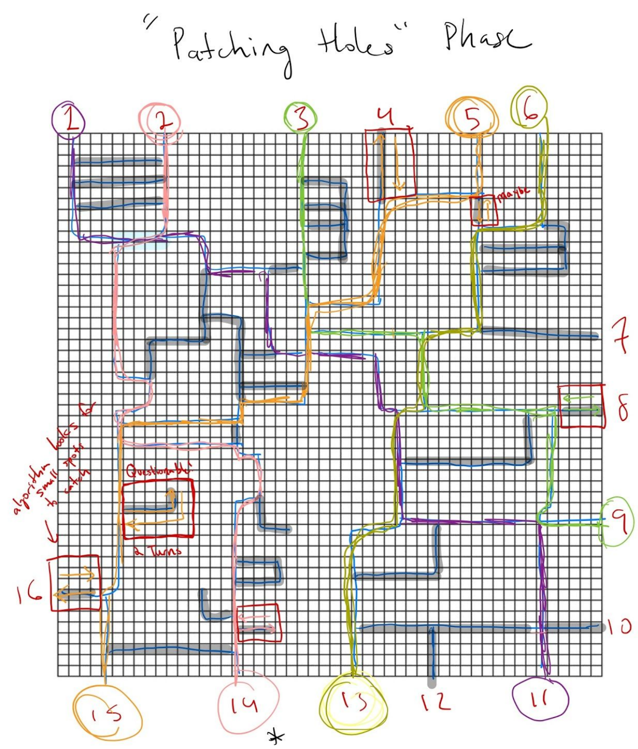
- 5 route A-Star method with Saturday cleanup
- 5 day “catch-all” routes
    - In place of the Saturday cleanup approach, the leftover roads will be added to the nearest routes to complete a full sweep in 5 days.
- Ant colony optimization approach
    - This potential approach implements a biomimicry algorithm that would be used in place of A-Star. More research needs to be done.

Materials

-Computer and software for designing the program are the only materials needed.

Procedure

-With the JAVA DrawingPanel class, a grid similar to the one shown to the right can be created, along with a 2-dimensional array that works behind the scenes. Both components will represent the exact same information, with the former displaying it visually and the latter remaining hidden.



-All routes can be drawn with color coding similar to the drawing on the right. Leftover roads will be displayed in gray as shown. A route or routes for the Saturday crew can then be traced out, and the program will then output its final metrics, which can include distance travelled per day, estimated time taken for travel, and number of turns taken.

-Throughout the process of running, the program should be updating the visual component in real-time, to help the user understand what is happening.

-During the testing phases, the first priority of the program will be to have it fill up the entire map every time, regardless of any of the metrics. Then if the program consistently fills all the space and the bugs are sorted out, iterations of the program can be tested against all the aforementioned metrics.

Risk/Safety Concerns:

Potential Safety Concerns and how they will be addressed.

There are no safety concerns for the development of the modeling environment, as it is done completely virtually. As this project is conceptual, no safety concerns are possible.

Data Analysis:

How the Data will be Analyzed.

Data will be objectively analyzed for many quantitative metrics that result from an algorithm. These include time taken to run, estimated time of route, estimated impact on traffic, employee satisfaction, turns taken per day, etc. The algorithm with objectively the best statistics will be chosen as the optimal method, as measured through a decision matrix.

Potential Roadblocks: (with action steps identified of how you might solve these):

Gathering road maps from the internet will be a potential roadblock. To solve this, I will not only consult the Computer Science Instructor at Mass Academy, but I will also reach out to experts in the field and consult online resources.

Keeping the estimated cost low of this new system is also a potential roadblock. If this challenge is insurmountable, a new approach will have to be taken, or this model can be seen as a hypothetical for the future, when potentially the budget can expand. (Cost affects mainly the timeline)

While optimizing the routes, it can potentially be difficult to catch every single road (on both sides if not a small road) during the course of the routes. Perhaps the algorithm can start by mapping out 5 optimal paths from one perimeter point to the next, then the routes with the least uncovered roads will be chosen and adapted upon to see the most optimal path.

If at any point the algorithm fails to run, the SCALE of the project can be reduced and the modeling environment can be simplified, leaving the project open for improvement in the future.

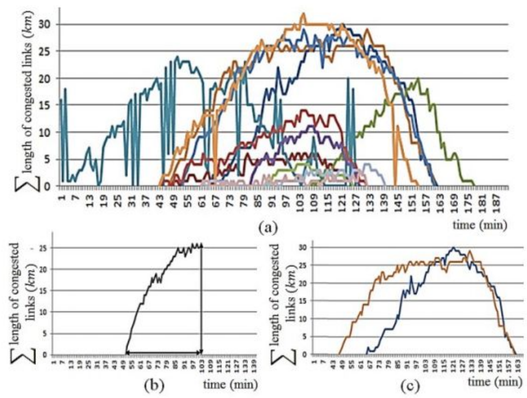


Figure 5. (a) Historical congestion pattern, (b) Current congestion pattern, (c) Similar congestion patterns

Determining the traffic for each road may also be a roadblock, so for the initial iteration of the project the factor of traffic will be decided by how many roads connect to a given road, along with length and relative location on the grid. If successful, it is possible to implement models that can predict traffic based on historical data for a given town, as seen in the figure to the right (Lee et al, 2014). Such a method would have to involve collecting that historical data, which could be done by accessing public databases or calling town officials. The only other feasible method of analyzing real-life traffic data involves pattern observation through use of special tools that are too costly for this project(D. Ma et al, 2018).

References: (In APA Format with in-text citations):

R. Septiana, I. Soesanti and N. A. Setiawan, "Evaluation function effectiveness in Wireless Sensor Network routing using A-star algorithm," *2016 4th International Conference on Cyber and IT Service Management*, Bandung, 2016, pp. 1-5.
doi: 10.1109/CITSM.2016.7577519

Coren, M. (2016, August 04). The economics of electric garbage trucks are awesome. Retrieved November 02, 2020, from https://qz.com/749622/the-economics-of-electric-garbage-trucks-are-awesome/

Wired. (2020, October 10). *Computer Scientists Break the 'Traveling Salesperson" Record*. Wired. Retrieved October 11, 2020, from

https://www.wired.com/story/computer-scientists-break-the-traveling-salesperson-record/
D Gunawan *et al* 2018 *J. Phys.: Conf. Ser.* 978 012122

K. Lee, B. Hong, D. Jeong and J. Lee, "Congestion pattern model for predicting short-term traffic decongestion times," 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, 2014, pp. 2828-2833, doi: 10.1109/ITSC.2014.6958143.

Munguía, R. (2014). A GPS-aided Inertial Navigation System in Direct Configuration. *Journal of Applied Research and Technology*, *12*(4). https://doi.org/10.1016/S1665-6423(14)70096-3

B. Sheng, D. Ma, P. Gao, S. Jin, and X. Ma, "Short-Term Traffic Flow Forecasting by Selecting Appropriate Predictions Based on Pattern Matching," in IEEE Access, vol. 6, pp. 75629-75638, 2018, doi:10.1109/ACCESS.2018.2879055.

Timeline: (with action steps identified- sub-deadlines will continue to evolve):

Rough timeline of major phases. As these phases get established, specific tasks under these phases will be defined further.

Brainstorming - June - September 2020

During brainstorming a large pool of diverse ideas were gathered based on my hobbies and interest, as well as problems that needed to be solved. I jumped around from idea to idea, and I noticed that a lot of my potential project ideas involved Computer Science. That led me to narrow down my project ideas to those specifically dealing with the subject, and I sought to find problems that could be solved.

Garbage trucks and school busses have always caused a disturbance in my town, since they obstruct large portions of the road and stop frequency, wasting time while also being dangerous to drive around. Thus I wished to somehow alleviate that problem, and I decided that I would use computer science to solve the problem.

Research / Planning - September - October 2020

This consisted of compiling scientific articles and patents to develop a strong understanding of the field I was working in. From this information I was able to develop possible strategies and methods to accomplish my goal. It was important to base my work off of proven and tested methods and building off from there.

The next step was to design a rough blueprint of the program on paper, emulating the workings of the algorithm visually before anything was put into code. This phase is done to prevent excessive work caused by ideas being scraped in the coding phase.

Proof of Concept in Java - Late October - Mid November

Create a working program that at least works in simple, randomly-generated cases. A grid system and straight-line roads are all that are necessary for this phase, as complex, real-life roads are out of the scope for this phase of the project, and potentially the entire project.

This is the phase to work out any flaws that would need to be addressed before the program is taken further towards its final iteration. I expect to be constantly tweaking the software and its parameters during this phase, and all of these adjustments should be recorded.

Adaptations of Concepts (MatLab, Net2Plan, Java?) - November - December

If the proof of concept in Java is proven to be robust and effective, then more realistic and complex road data can be imported and the algorithms in the program can be adapted to work on these out-of-grid examples. The program could potentially stay in JAVA, or it could be moved over to MatLab or Net2Plan, both of which can create more realistic models that visually represent road maps. Even further, a version of the app that utilizes GPS measurements as well as inertial measurements could be deployed to pinpoint a truck's specific location and guide it along optimal paths (Munguia, 2014).

Final program / application - Before Christmas

Whether it be the initial concept or the more robust software, a finalized version of the program will be made that can potentially run as a standalone application or web applet. The final version should be presentable and easy to use for someone who has never seen it. Most to all of the work should be done on the computer's end, rather than the human user.