GeoTrek

cd-099: Charles Henville, Michael Del Duca, Ritvik Jayanthi

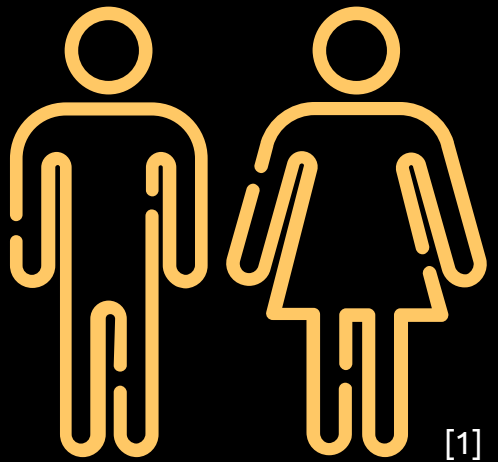
Table of Contents

01 Introduction

The Problem at Hand

The Urban Commuter:

- Budget-lover
- No personal vehicle
- Plans ahead



[1]

Customizability [1]

For ease of use

Route Visualization [1]

For one trip or many

Algorithmic Accuracy [1]

When finding routes

Responsiveness [1]

Fluid user Interactions (<200 ms)

[2, 3]

The Core Message

Geotrek, through innovative design, noteworthy results, and concrete future planning, excels in both catering to the Urban Commuter, and acting as an exemplary GIS for a wide array of users

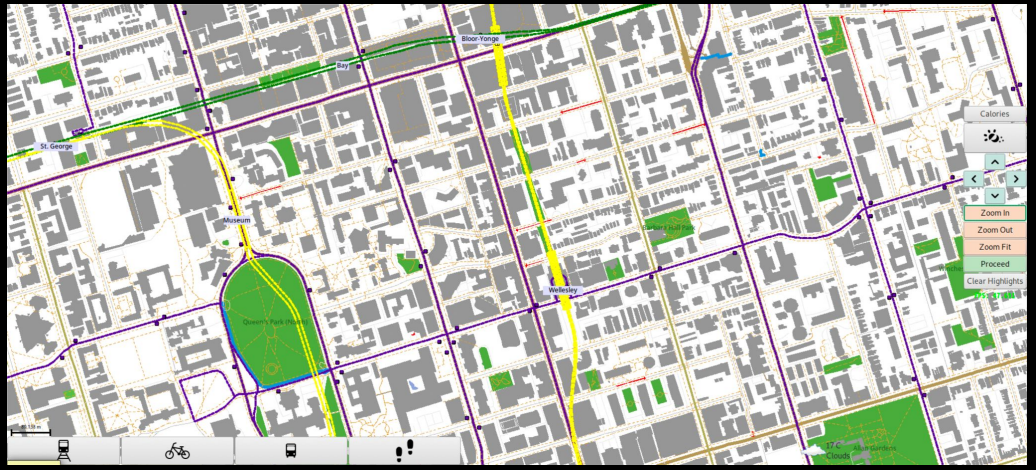
02 Status Report

Our Innovative Solutions

Customizability

Quality of Life (QoL) Improvements Towards a Seamless User Interface and Experience

Clutter Reduction for Usability

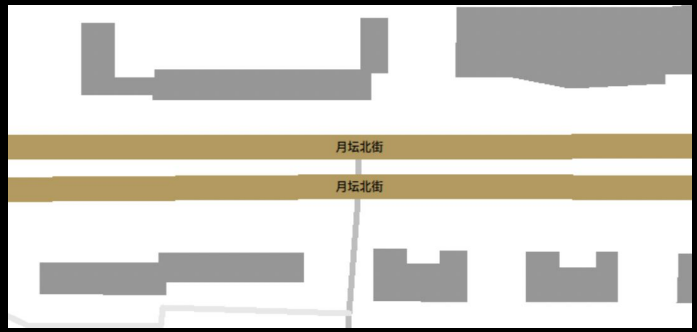


[4]

Familiar Interface Adjustments

Dynamic Colour Schemes

Non-Latin Character Support



A Screenshot from our map of Beijing, China.

Additional Functionality for Quality of Life

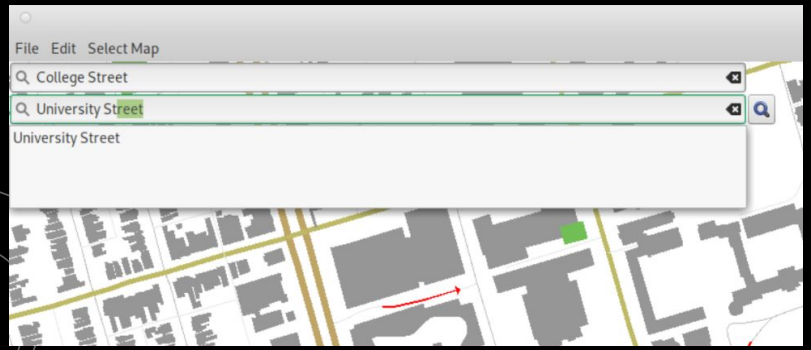
Rendering More Features & Labels

Weather API Integration

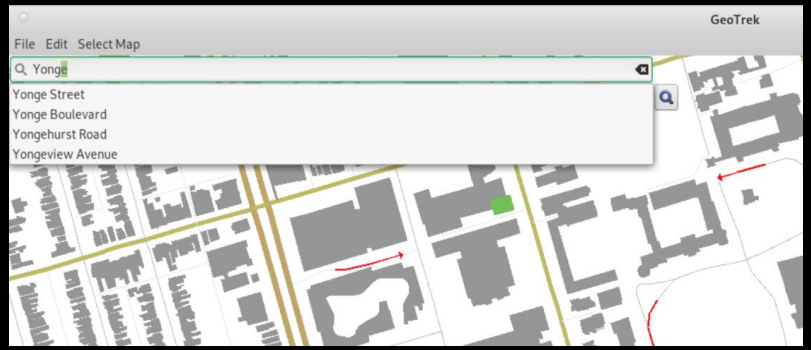
Route Visualization

Enhancing the functionalities of our GIS

Auto Filling Search Bar for Ease of Use

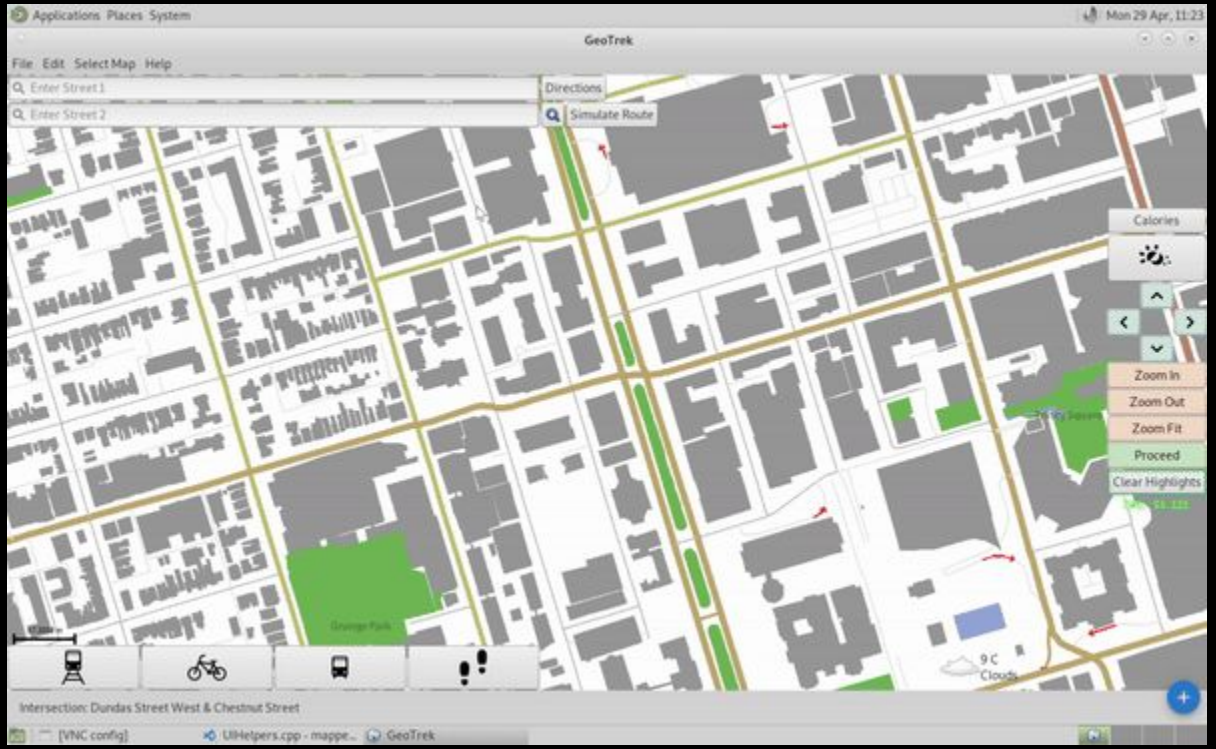


In-line AutoFill

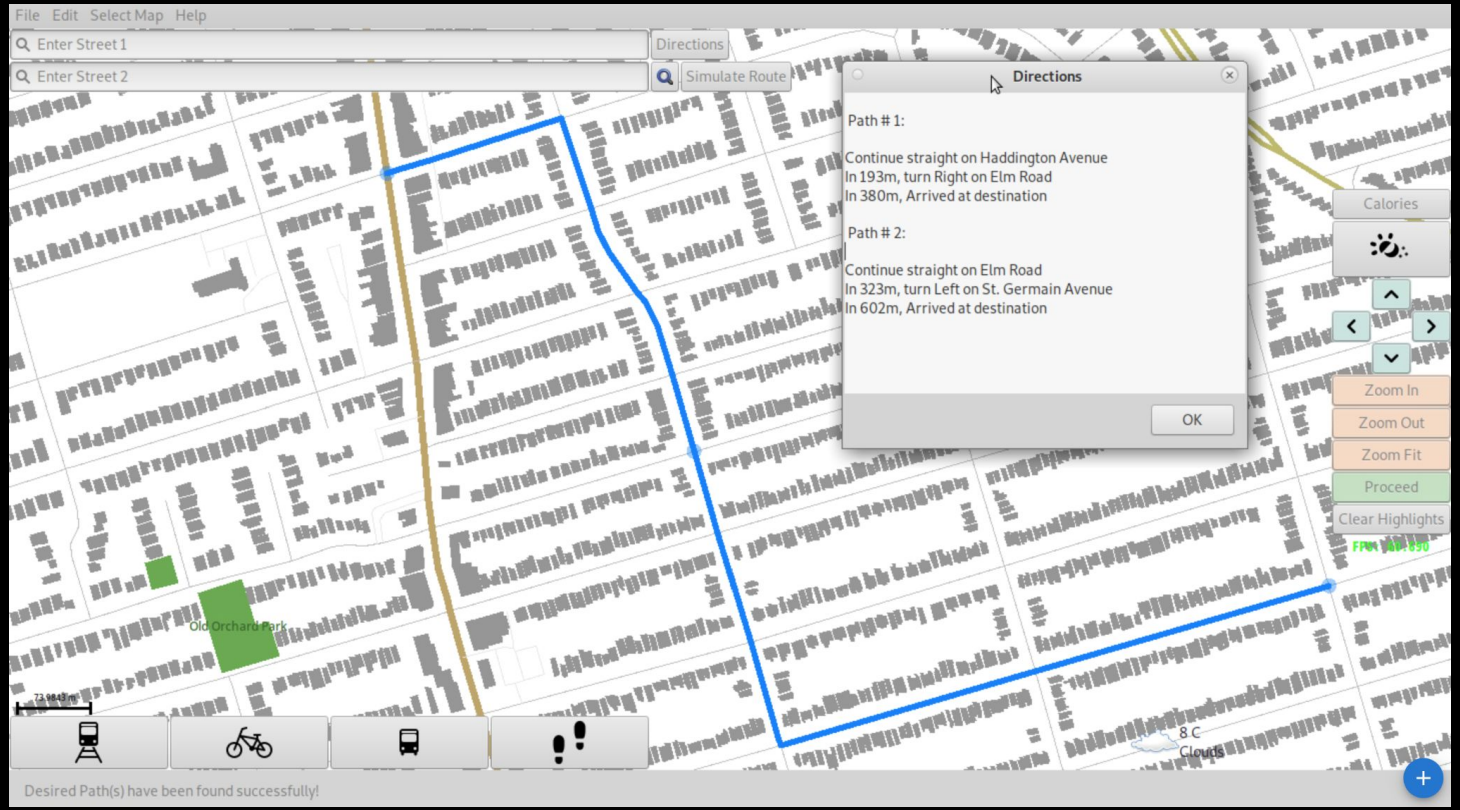


Drop Down Autofill

Improving Intuitiveness with Double Clicking



Sequential Pathfinding Improves Flexibility



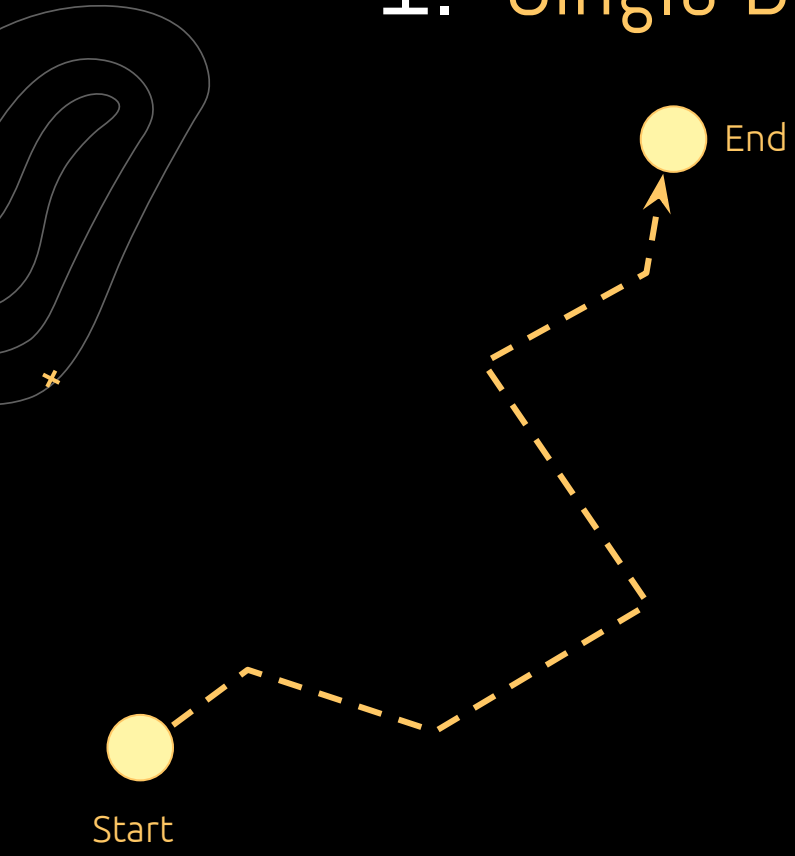
Route Simulation allows Easy Visualization

The user is capable of “simulating” the route that the software provides them with. Another optional feature.

Algorithmic Accuracy

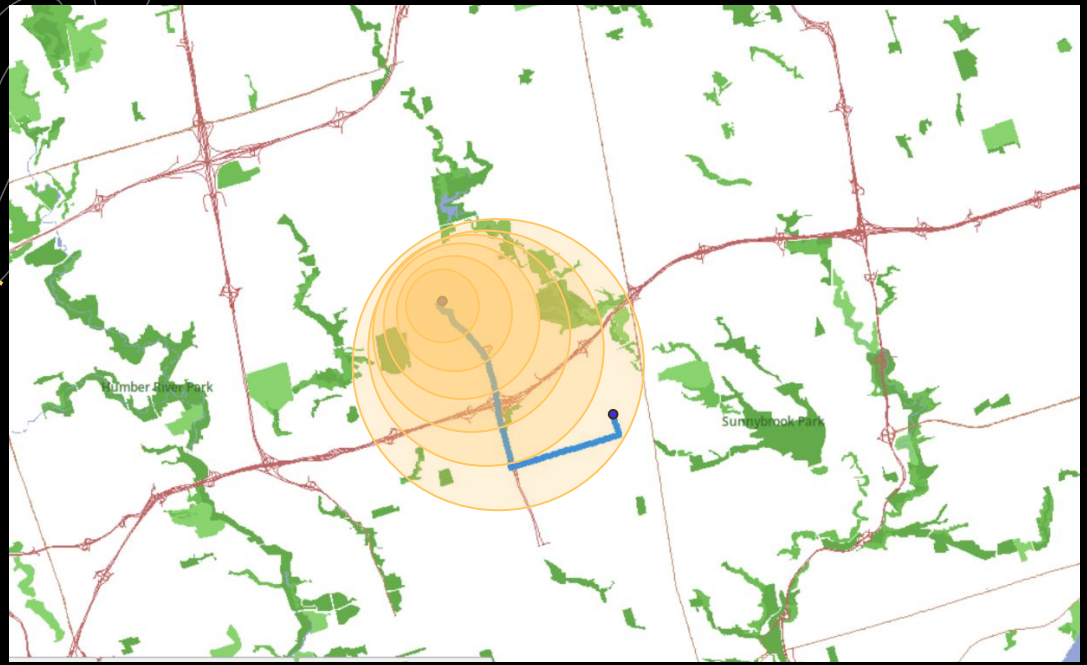
Finding the quickest route in the fastest time

1. Single Destination Algorithm



- Route between 2 points
- Quickest path
- Minimal loading

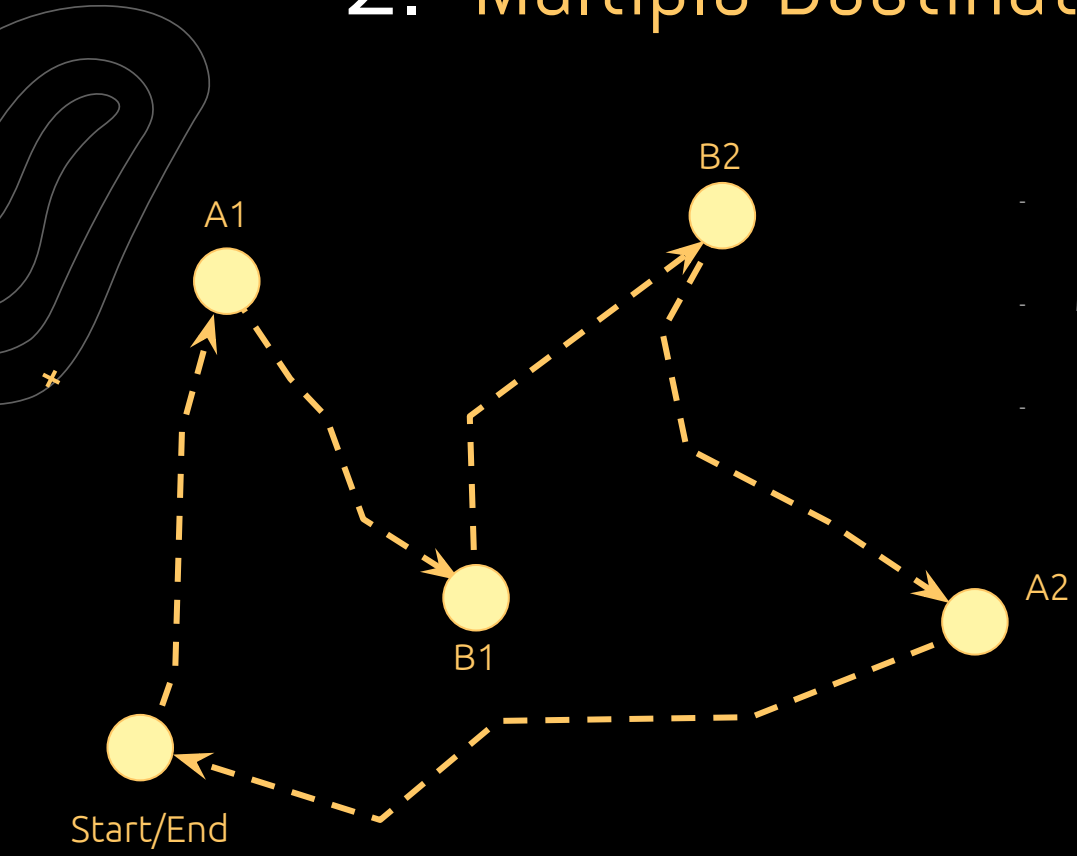
The State of the Art A* Algorithm



- 100% guarantee of the quickest path
- Avoids searching in incorrect directions

Appendix A

2. Multiple Destination Algorithm



- Round-trips from select starting points
- Multiple destinations
- Destination dependencies

GeoTrek's Optimized 4-Step Algorithm

1 Initial Loading

"Multi-Dijkstra" Algorithm

	A1	A2	...	Z1	Z2
A1		A1→A2	...	A1→Z1	A1→Z2
A2	A2→A1		...	A2→Z1	A2→Z2
...	...	...		...	...
Z1	Z1→A1	Z1→A2	...		Z1→Z2
Z2	Z2→A1	Z2→A2	...	Z2→Z1	

Adjacency Matrix

2 Potential Solutions

"Greedy Solution" Algorithm

Start → A1 → A2 → … → End

Start → A1 → B1 → … → End

Start → Z1 → Z2 → … → End

Start → B1 → Z1 → … → End

GeoTrek's Optimized 4-Step Algorithm

3 Improving the Solution

"Simulated Annealing" Algorithm

Start → A1 → B1 → Z1 → A2 → B2→ Z2 → End

Shift

Start → A1 → B1 → A2 → B2→ Z1 → Z2 → End

Swap

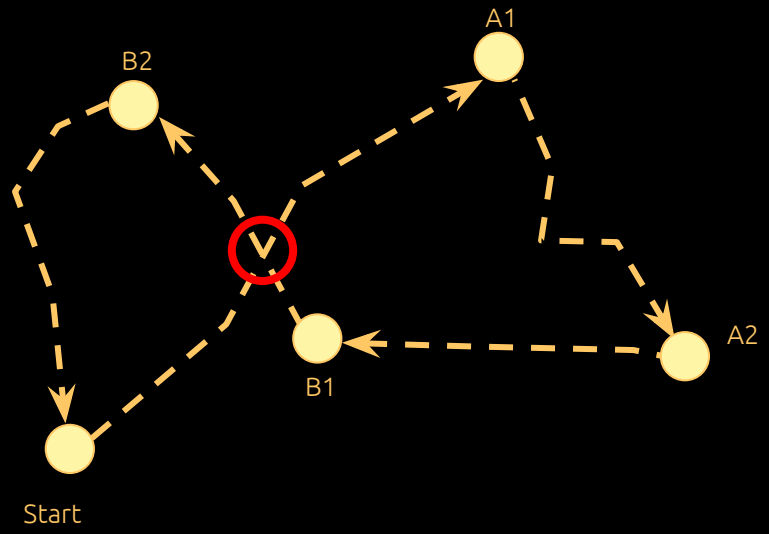
Start → A1 → A2 → Z1 → B1 → B2→ Z2 → End

Reverse

Start → Z1 → B1 → A1 → A2 → B2→ Z2 → End

4 Final Solution

"2-opt Inversion" Algorithm



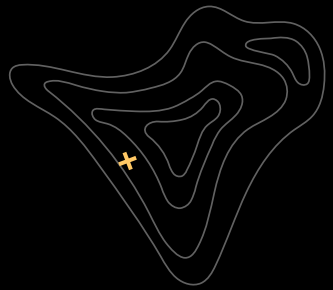
[5], Appendix B

Test Results Indicate the Success of our Algorithms

Single Destination Results

100%

Guarantee of quickest path



< 1 sec

For intra-city route finding

13.2 sec

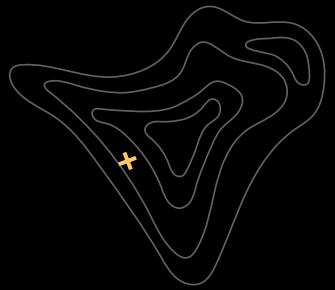
Average time for extremely distant routes

[3]

Multiple Destination Results

92nd Percentile

Compared to other algorithms



< 50 sec

For over 65 destinations

5% better

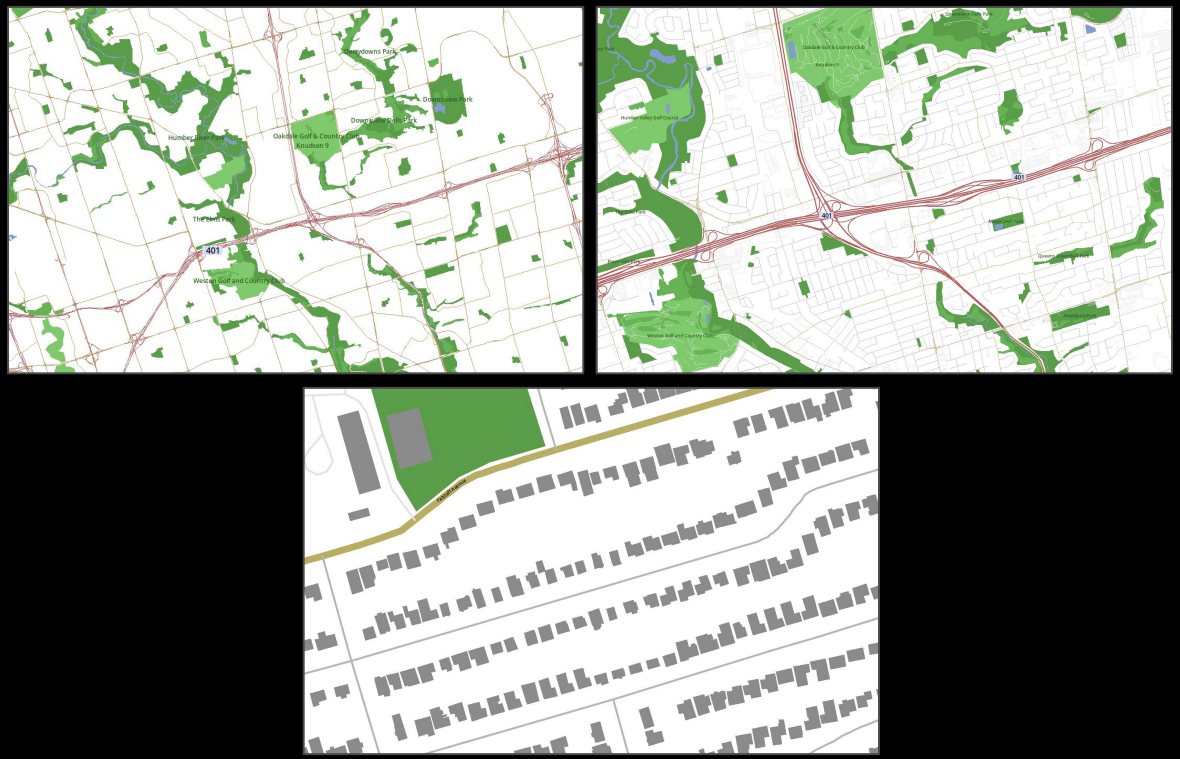
Than optimal benchmark

Responsiveness

For a seamless user experience

Cartographic Generalization

(More Zoom -> More Detail)



[4, 6]

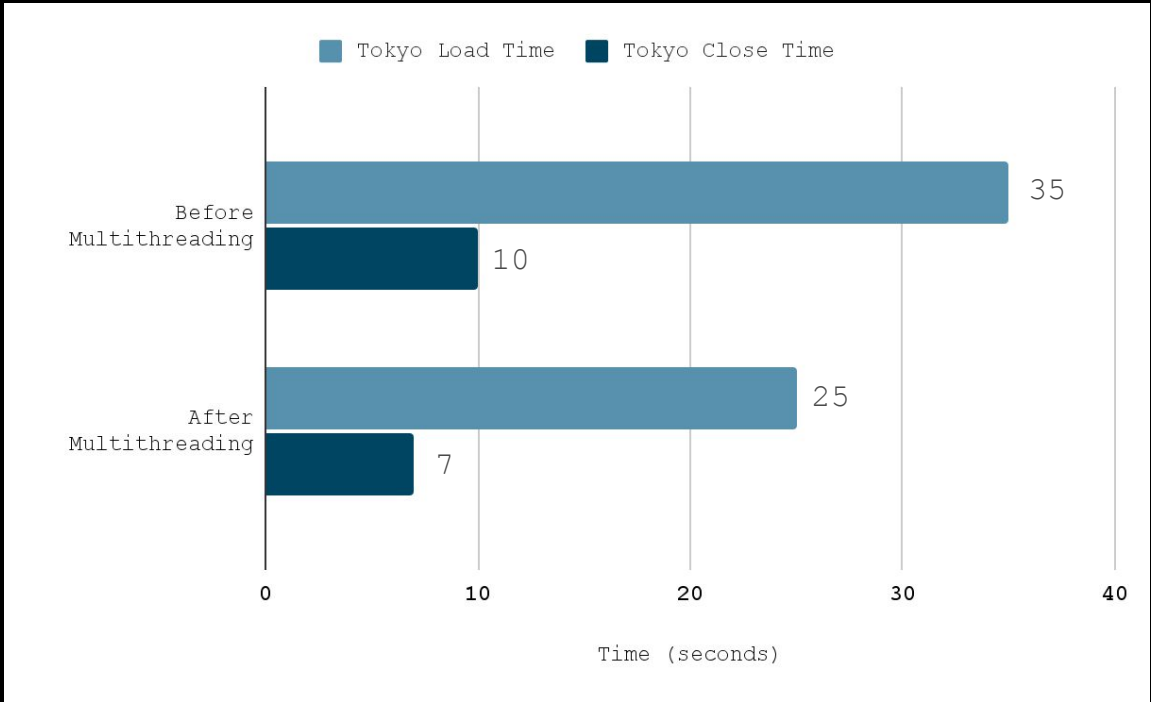
Multi Threading

1. Initial Loading
   - Extracting data for all map features
2. Dijkstra Algorithm
   - Populating the adjacency matrix rows
3. Greedy Algorithm
   - Finding numerous unique paths simultaneously

[2]

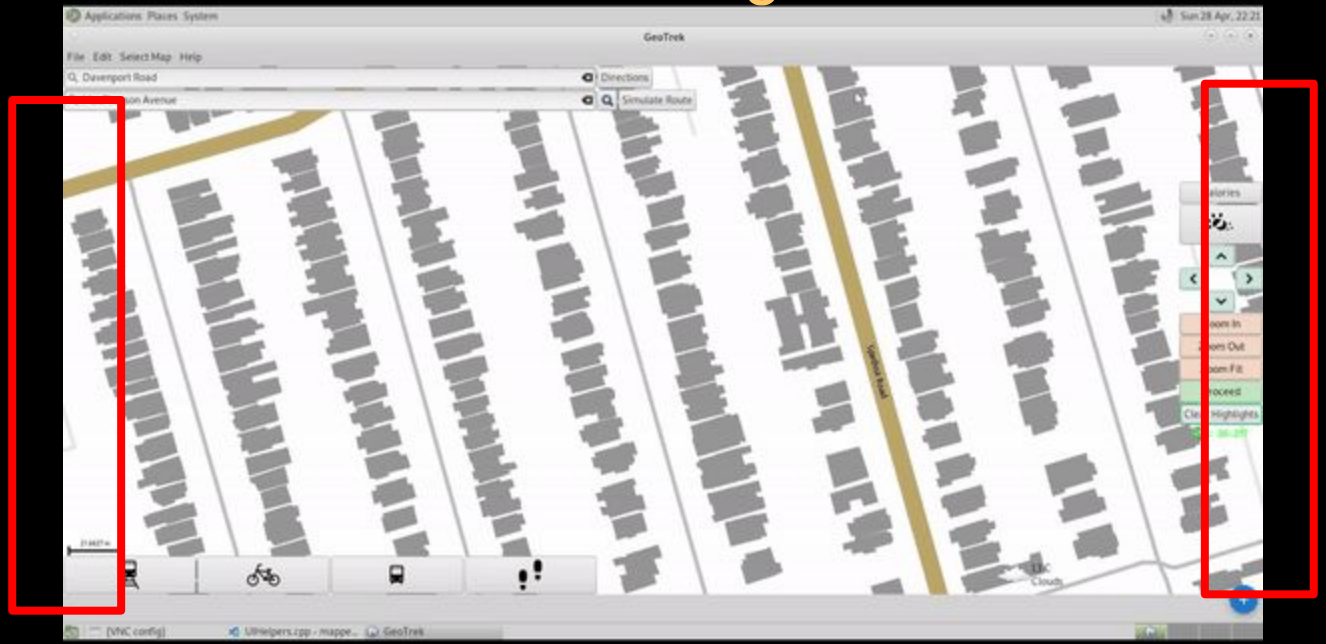
Multithreading Successfully Reduces Load Time

More Threads -> More Responsivity



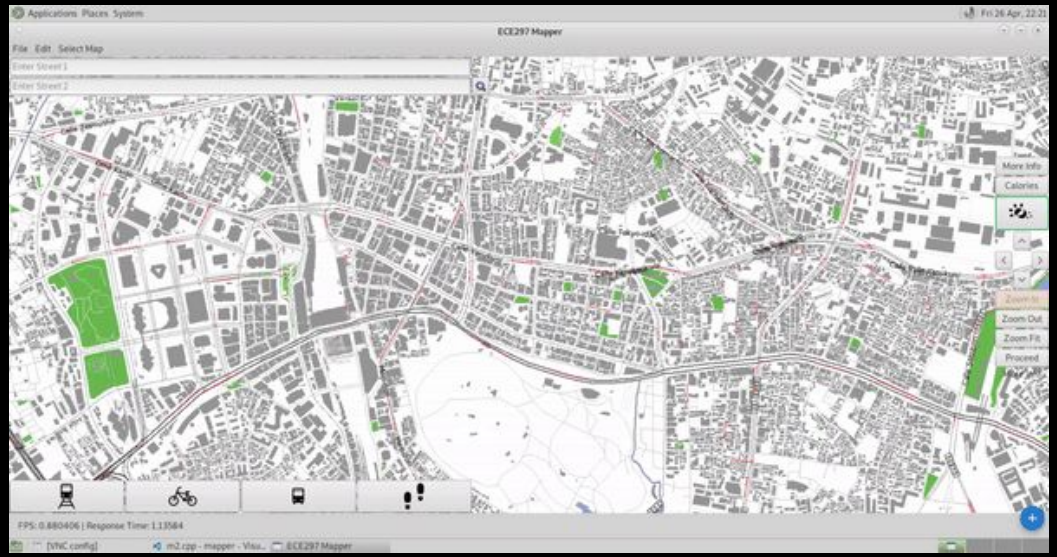
[2]

Culling



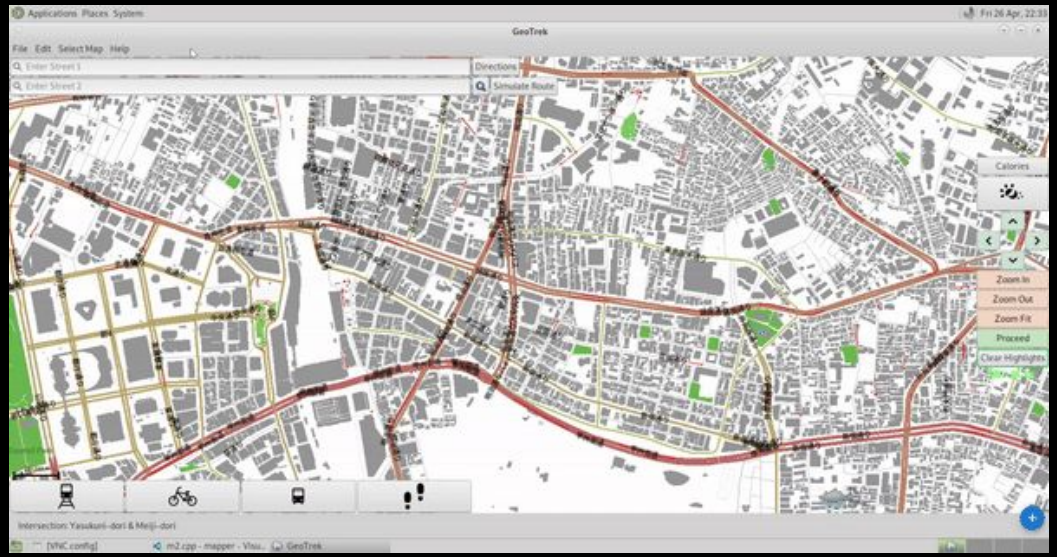
Objects disappear as they leave the screen

Before Culling: Unresponsive



Panning around Tokyo
(Pre-culling)

After Culling: Fluid Responses



Panning around Tokyo
(Post-culling)

Response Times Drastically Improved by Culling

City	Pre-Culling Response Time (ms)	Post-Culling Response Time (ms)	
Toronto	150	8	19x
New York	300	17	18x
Tokyo	1130	35	33x

03 Future Pitch

What we’re Planning

The Gap in the GIS Market

Lack of options that allow combination of transportation modes

[8]

Preset Options for travel

[7]

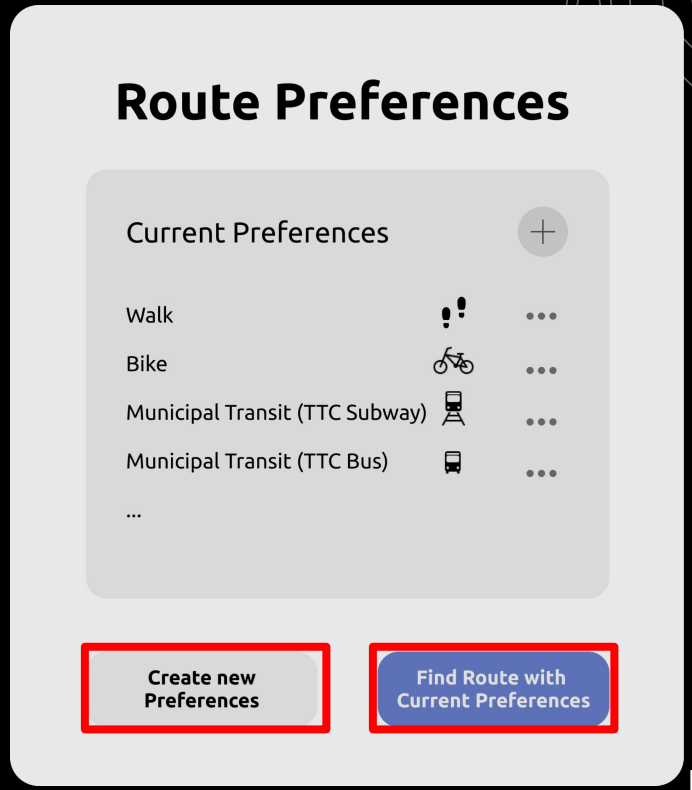
Our Goal to Bridge the Gap

Goal:

Allow the user to configure their transportation preferences via a new menu or survey.

What We Must Implement

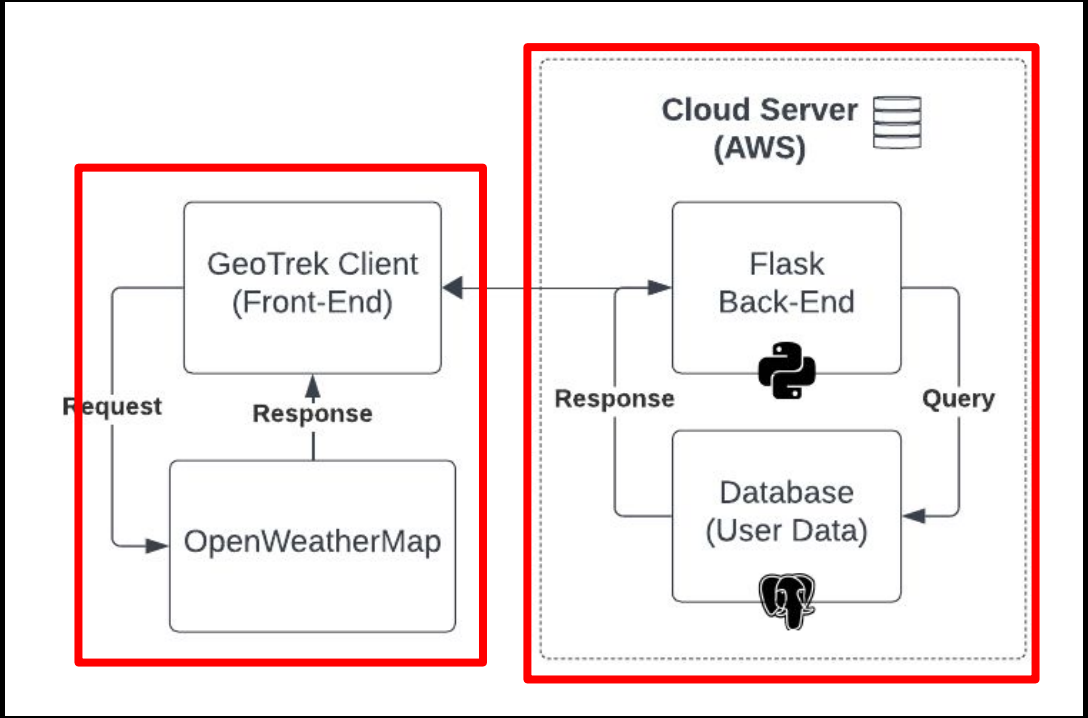
- The GUI shown to the right should appear when the user attempts to find a route.
- Modifications of existing pathfinding algorithm (to account for combinations of transportation methods).
- An account system to store user preferences (via a cloud-based solution).



[9]

Potential UI Design

A Potential System Design



[10]

Filling the Gap with our Solution

[8]

Preset Options for travel

Togglable options to allow for multiple options to be used in tandem

Caters the map to each user

04 Conclusion

The Takeaway

Takeaways

- GeoTrek is quick and responsive and in its current state
- It is usable and customizable, catering to urban commuters.
- The team has a plan going forward on how to set our application apart from other GIS, amending their existing issues.

References

[1] *Slidesgo*. Freepik. Accessed: Mar. 09, 2024. [Online] Available: https://slidesgo.com/

[2] T. Horbiński, P. Cybulski, and B. Medyńska-Gulij. 2021. "Web Map Effectiveness in the Responsive Context of the Graphical User Interface" *ISPRS International Journal of Geo-Information,* vol. 10, no. 3, pp. 134, 2021. [Online]. Available: https://www.mdpi.com/2220-9964/10/3/134 [Accessed Feb. 24, 2024].

[3] S. Yang, D. Yan, and A. Rountev, "Testing for poor responsiveness in android applications," in *2013 1st International Workshop on the Engineering of Mobile-Enabled Systems (MOBS)*, San Francisco, CA, USA, 2013, pp. 1-6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6614215

[4] M. Adnan and Z. Khan, "Usability Evaluation of Web-based GIS Applications: A Comparative Study of Google Maps and MapQuest," M. S. thesis, Comp. Sci., Blekinge Institute of Technology, Ronneby, Sweden, 2010. [Online]. Available: https://www.researchgate.net/profile/Muhammad-Adnan-163/publication/374420638_Usability_Evaluation_of_Web-based_GIS_Applications_A_Comparative_Study_of_Google_Maps_and_MapQuest/links/651d34aab0df2f20a20f831f/Usability-Evaluation-of-Web-based-GIS-Applications-A-Comparative-Study-of-Google-Maps-and-MapQuest.pdf

[5] M. G. Nejad et al., "A mathematical model and simulated annealing algorithm for solving the cyclic scheduling problem of a flexible robotic cell," *Advances in Mechanical Engineering*, vol. 10, no. 1, 2018. [Online]. Available: https://journals.sagepub.com/doi/10.1177/1687814017753912 [Accessed Apr. 09, 2024].

[6] K. S. Shea and R. B. McMaster, "Cartographic Generalization in a Digital Environment: When and How to Generalize." Cambridge University Press, [online document], 1989. Available:https://experts.umn.edu/en/publications/cartographic-generalization-in-a-digital-environment-when-and-how [Accessed: Feb. 25, 2024].

[7] A. Nivala, L. T. Sarjakoski, and T. Sarjakoski, “Usability methods’ familiarity among map application developers,” *International Journal of Human-Computer Studies,* vol. 65, no. 9, pp. 784-795, Sept. 2007. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1071581907000596?via%3Dihub [Accessed Feb. 24, 2024].

[8] *Google Maps*. Google. Accessed: Feb 27, 2024. [Online]. Available: https://maps.app.goo.gl/2aWbNN8muMcYRuwDA

[9] *Figma*. Figma Inc. Accessed: Apr. 09, 2024. [Online] Available: https://www.figma.com/

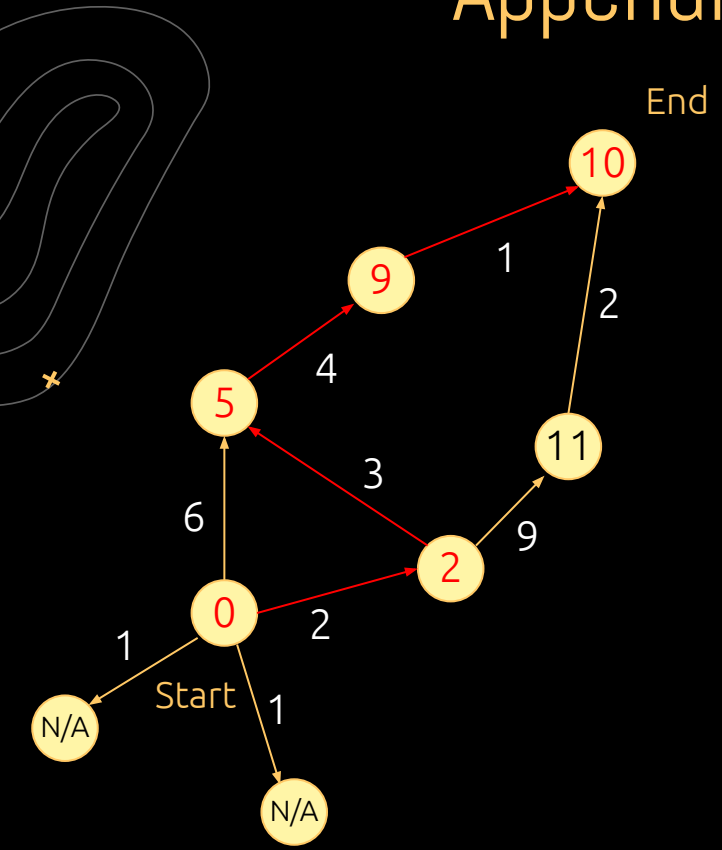
[10] *Lucidchart*. Lucid. Accessed: Apr. 09, 2024. [Online] Available: https://www.lucidchart.com/pages/

THANK YOU

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**

Please keep this slide for attribution

Appendix A: The A* Algorithm



- The A* algorithm only travels along the fastest path which tends generally towards the endpoint
- At each point it uses the absolute distance between the current point and destination, and the maximum speed limit of the map to find the quickest time you could reach the end in
- This heuristic limits (but still allows) searching in the “incorrect” direction to increase algorithmic speed

Appendix B: The Multi-Destination Algorithm

1. Multi-Dijkstra's
    - The Multi-Dijkstra's portion of the algorithm populates 2 matrices
        - Adjacency matrix (rows and columns correspond to each point in order)
        - Priority matrix (each row contains a priority queue with the closest point at the top)

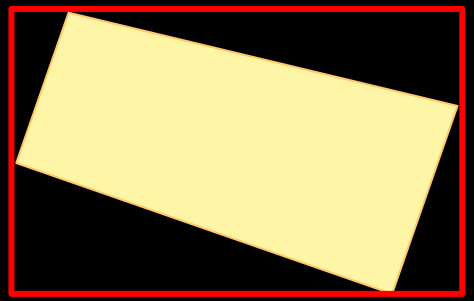
2. Greedy Solution
    - In parallel, 1000 possible solution paths are generated from each unique starting point.
    - The paths are a classic "nearest neighbour" greedy solution, but with a 5% chance to choose the second nearest neighbour each time

Appendix B: The Multi-Destination Algorithm

3. Simulated Annealing
    - The Simulated Annealing portion of the algorithm repeatedly performs a series of operations on randomized subsections of the route.
        - Shift Operator: Takes a single point and moves it to a randomized location in the order
        - Swap Operator: Randomly selects two points and swaps their locations in the order
        - Reverse Operator: Takes a random point in the route and reverses a subsection of random size beginning at that point
    - 2 operations are run in parallel on the 8 best paths from the greedy algorithm
    - After each operation, the variance from the original is compared to the constantly reducing "temperature," which determines whether to save that result or discard it

4. 2-Opt Inversion
    - The best result from step 3 is taken, and every possible subsection of points in the order is reversed and checked for a better solution.
    - This guarantees no overlapping pathways

Appendix C: The Culling Algorithm



- Each object on the screen has an associated “bounding box” which contains it
- If the entire bounding box is offscreen, the object is no longer drawn
- If any portion of the bounding box is within the bounds of the screen, the object is drawn


slidesgo