NYU SHANGHAI 上海纽约大学

Data Science

Capstone Report - Fall 2023

# SecureStep: Data-Driven Walking Routes for Safety-Conscious New Yorkers

*Akhmed Adzhavi*

Supervised by
Professor Prométhée Spathis

## Declaration

I declare that this senior capstone was composed entirely by myself with the guidance of my advisor, and that it has not been submitted, in whole or in part, to any other application for a degree. Except where it is acknowledged through reference or citation, the work presented in this capstone is entirely my own.

## Preface

### Background and Credentials:

As a student that has gone through the challenging Computer Science coursework offered at NYU Shanghai, I have gained the crucial skills needed to pursue more complex topics. Although I don't have direct experience with routing problems, the course Basic Algorithms has given me the necessary abilities to tackle this project.

### Project Inspiration:

The experience that inspired this project is me attempting to navigate the hectic and intimidating streets of New York City during my study away semesters. That is when I realized that navigation applications have a major flaw, they ignore an extremely pivotal variable: safety.

### Target Audience and Usefulness:

The target audience for this project ranges from specialists in Data Science to simply individuals wishing to traverse the streets of New York in a peaceful manner. The techniques used and analysis produced in this paper aim to illuminate the solutions to pedestrian safety in bustling cities.

## Acknowledgements

I would like to thank my supervisor, Professor Prométhée Spathis, for assisting me with every stage of this research project.

**Abstract**

*Existing navigation applications solely take speed into consideration when calculating the route from point A to point B. This project aims to address this insufficiency by taking into account safety in the route calculation process. It incorporates crime incidents and taxi drop-offs to calculate the safety scores along a path. Solving this problem involves using existing multi-objective optimization algorithms in order to maximize both speed and safety at the same time. This paper utilizes NSGA-II (Non-dominated Sorting Genetic Algorithm II), a well-known multi-objective optimization algorithm. By balancing these variables, this project aims to transform navigation techniques. The implementation of NSGA-II provides a solution that avoids dangerous areas, resulting in a longer time duration, but a safer route between point A and point B.*

**Keywords**

**Multi-Objective; Routing Problems; Navigation; Crime; Safety; Data-Driven Solutions; NSGA-II**

# Contents

# 1 Introduction

## 1.1 Overview

This project is focused on developing a data-driven solution that calculates and presents walking routes in New York City, taking into account both distance and safety. The biggest issue with navigation applications is that they don't adequately consider the safety of a path taken from A to B, which is a significant concern, especially in densely populated cities like New York.

## 1.2 Scenario

Consider this scenario: an innocent civilian traversing unfamiliar neighborhoods using navigation applications that don't take safety into account. The individual could be directed to an area with a high crime rate which would put the pedestrian in harms way. To fully realize this situation, refer to Figure 1 which displays the crime rate across neighborhoods in Manhattan. Focusing on Lower Manhattan, we can evidently observe that it consists of neighborhoods with the same shade of red. Although the map provides a decent overview of crime across Manhattan, it lacks finer distinctions between areas, so it would be difficult to avoid certain areas in Lower Manhattan when all the blocks are the same shade of red. This is why calculating a safety score is important.

## 1.3 Objectives and Methodology

This project aims to address and fill this gap by intertwining crime and taxi drop-offs data into the route calculation process. The main objective is answering this question: "*How can we combine Data Science and Computer Science tools and techniques to provide both the safest and quickest route in New York City?*" To find the answer, we must find a method to strike a balance between speed and safety in the optimization process.

The fundamental building blocks of this project are shortest path algorithms. The most common algorithms include:

- Depth-First Search (DFS)
- Breadth-First Search (BFS)
- Dijkstra's Algorithm
- Bellman-Ford Algorithm

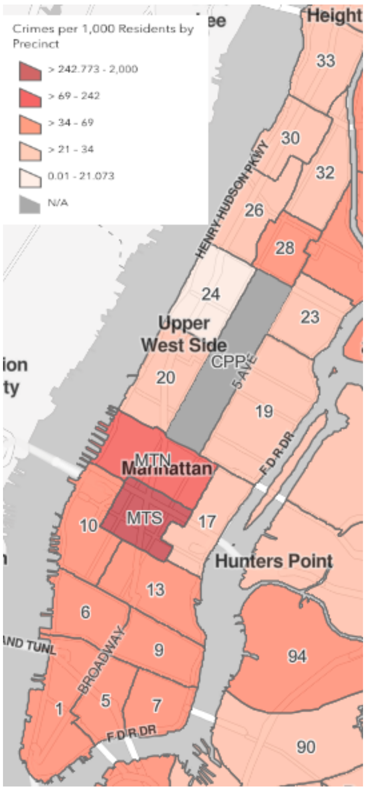


Figure 1: Manhattan Crime Map. Adapted from [1].

- Floyd-Warshall Algorithm

All these algorithms solve single-objective problems, the most common one being the vehicle routing problem. Now, let's consider multi-objective algorithms, that allow us to consider multiple objectives simultaneously. We will utilize the power of multi-objective optimization to present the safest and fastest route.

### 1.4 Structure

This paper presents its methodologies, analysis, and results in the following structured manner:

- **Related Works:** In this section, we go over existing literature relevant to multi-objective algorithms to provide background information and crucial findings.
- **Solution:** Here, we present our solution to finding the shortest and safest path from point A to point B. Details are provided about the techniques used to calculate safety scores, compute shortest paths, and optimize both objectives.
- **Results:** This section is for presenting the results from the solution applied on the data, which include tables, figures, and graphs.

- **Discussion:** Here, we discuss the results obtained from the previous section and address limitations about this project.

- **Conclusion:** This section summarizes the key findings of this paper and provides potential avenues for future research.

# 2 Related Work

## 2.1 Definition

Multi-objective optimization problems are present in our everyday lives. One simple example is a company attempting to maximize their efficiency and minimize their costs in a certain period. There are numerous other examples with different variables being optimized, thus it's crucial to formally define a multi-objective optimization problem. Every problem consists of a set of decision variables, objective functions, and constraints. The goal is to calculate an optimal solution that maximizes the objective vector with the appropriate constraints. Mathematically, this can represented as: $y = f(x) = (f_1(x), f_2(x), ..., f_k(x))$, where $x$ is the decision vector, representing the set of decision variables and $y$ is the objective vector.

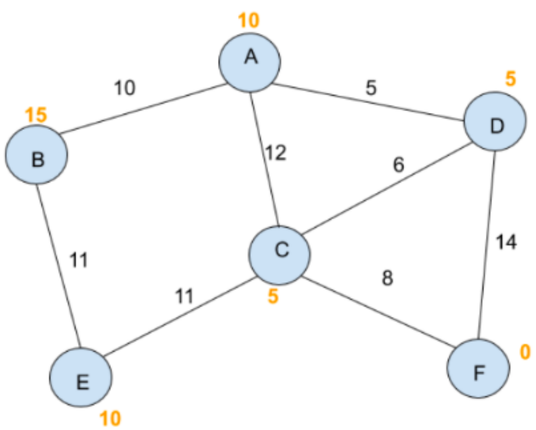


Figure 2: Multi-Objective Graph

## 2.2 SPEA

This section explores the SPEA in the context of multi-objective optimization. When considering the field of multi-objective optimization problems, we need to consider the work of Zitzler in "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications". This paper [2] that was published in 1999, tested numerous algorithms and specific theoretical problems and evaluated their performances.

Before we move into the contents of the paper, we need to define an evolutionary algorithm. These algorithms are a type of optimization algorithms that solve these complex problems based on the ideas of biological processes such as genetic variation and reproduction. They aim to find the optimal solution by evolving a set of potential solutions through the biologically inspired methods mentioned previously. These algorithms are appropriate in finding the optimal solution as they traverse the solution set efficiently.

The problems they utilized in this experiment are: multi-objective knapsack problem, multi-objective traveling salesman problem, and continuous test problems. They compared the strength of the algorithms based on the relative distance to the most optimal algorithm, the Pareto-optimal. Notably, the Strength Pareto Evolutionary Algorithm (SPEA) emerged as a top-performing algorithm in their analysis.

The 1980s experienced the rise of evolutionary multi-objective optimization (EMO) algorithms. In the beginning, it did not include elitism, but it was still crucial for multi-objective problems. With this limitation in mind, it led to a development of elitist multi-objective evolutionary algorithms. Eckart Zitzler introduced the SPEA2, which is an improved version of the regular SPEA [3]. It improves upon its limitations, and the experimental results show that it is more superior compared to the original and it's competitive with NSGA-II.

### 2.3 NSGA

This subsection discusses the concept of NSGA. The paper [4] published by Kalyanmoy Deb et al. in 2002 discusses the fact that multi-objective optimization problems produce numerous optimal solutions, not just one. The nondominated sorting genetic algorithm (NSGA) faced backlash based on complexity and lack of elitism. To get over these issues, the paper introduces a new and enhanced algorithm that builds upon the original, called NSGA-II. Extensive trials show NSGA-II's converging nature towards the true Pareto-optimal set. The proposed NSGA-II advances multiobjective evolutionary algorithms (MOEAs) by outperforming two other prominent MOEAs, PAES and SPEA.

In recent times, there has been significant progress made in the robotics industry. For instance, Tesla has been developing autonomous vehicles, which are complicated systems to create as they must be make intelligent decisions independently. The vehicle doesn't just attempt to avoid collision (safety measure), but also the optimal route from the start to the end. Most of the literature that exists simply identifies the path that has the shortest distance. But, in truth, there are far more variables that we need to consider, like path safety, surroundings and visibility. The paper [5] published by Faez Ahmed et al. integrates a Genetic Algorithm (GA) with a path representation method using B-splines. In addition, there are three different safety measures used with this algorithm. The authors have used MATLAB to implement the NSGA-II algorithm.

### 2.4 Applications of Evolutionary Algorithms in Optimization Problems

Building upon the usage of evolutionary algorithms on multi-objective optimization problems, Yetgin et al.'s paper [6] introduces the idea of optimizing routing in multi-hop wireless networks using NSGA-II and MODE algorithms. The paper involves comparative analysis between two multi-objective algorithms, NSGA-II and MODE, utilized with the goal of optimizing this set of variables: (delay, energy consumption). The outcome of the analysis determines that the performance of the MODE algorithm has a better performance.

The paper [7], published by Solwan M. Mostafa et al. in 2022, utilizes the travelling salesman problem (TSP) as a framework. The main idea of this paper is the usage of the multi-objective grasshopper optimization algorithm (MOGOA) to appropriately select the ant colony system (ACS) algorithm parameters, which finds an optimal path. To check the effectiveness of the proposed multi-objective method, it is compared to the Non-Dominated Sorting Genetic Algorithm (NSGA-II). This paper produced a table that evidently shows that the proposed method by the authors is superior compared to the NSGA-II.

### 2.5 Survey of Multi-Objective Routing Problems

The paper [8], published by Nicolas Jozefowiez et al., includes a bird's eye view of the available research on multi-objective routing problems. These sorts of problems typically consists of the graph, costs, and objectives. The graph is made up of nodes, which represent locations, and links, which represent routes (connections). Additionally, the costs are outlined along the links. The costs, e.g. time and safety, vary based on distance, location, and speed. Like in my project, the constraints usually interfere with each other, e.g. higher safety might mean a slower route. The paper provides a very useful timeline, but the limitation is that the timeline stops in 2006. The authors recognize the main methods of solving multi-objective problems: weighted and multi-objective evolutionary algorithms. The weighted strategy involves assigning a weight which could lead to limited approximations.

Rosario G. Garroppo et al.'s paper [9] delves into the complicated task of locating a path between a source and destination node, at the same time meeting the constraints. The authors introduces the multi-constrained optimal path (MCOP) problem, and it consequently provides the exact and approximate solutions. The paper includes the foundations, solutions, exact and approximate algorithms, and the time complexity of the algorithms. Figure 3 is the table displayed in the paper that provides even more literature which includes algorithms that are applicable to multi-objective optimization problems.

| | Solution type | Number of objectives | Technique | Complexity | Number of paths |
|---|---|---|---|---|---|
| Bellman–Ford (BF) | Exact | 1 | LC | $O(mn)$ | 1 |
| Dijkstra (DSP) | Exact | 1 | LS | $O(m + n\log n)$ | 1 |
| Martins and Santos | Exact | $M$ | LS | $O(DM^2\omega^2\log(\omega))$ | $\omega$ |
| Extend of Skriver and Andersen | Exact | $M$ | LC | $O(DM[2^M\omega + \omega^2\log(\omega)])$ | $\omega$ |
| Paixão and Santos | Exact | $M$ | RK | $O(n\, M^2D\omega^2)$ | $\omega$ |
| Raith and Ehrgott | Exact | 2 | 2P | $O\left(\xi \mathrm{DSP} + \xi\log\xi + \sum_{i=1}^{\xi}\mathrm{LS}(\omega_i)\right)$ | $\omega$ |
| Song and Sahni | Approximate | $M$ | IP + LC | $O(\, mM(n/\varepsilon)^{3M})$ | $\lceil n/\varepsilon\rceil^{M-1}$ |
| Xue et al. | Approximate | $M$ | SR | $O((\log\log M)(m(2n)^{M-1}) + m(n/\varepsilon)^{M-1})$ | 1 |
| Chen et al. | Approximate | 2 | SR, IP | $O((H/\varepsilon)\log(H/\varepsilon)(m + n\log n))$ | $O((H/\varepsilon)\log(H/\varepsilon))$ |
| Tsaggouris and Zaroliagis | Approximate | $M$ | IP + LC | $O\left(nm\left(n\log\left(n\widehat{C}\right)/\varepsilon\right)^{M-1}\right)$ | $\prod_{j=1}^{M}\lfloor\log_{1+\varepsilon}(nC_j)\rfloor$ |

Figure 3: Summary of Surveyed Path Computation Algorithms. Adapted from [9].

## 2.6 Exploring Algorithms for Multi-Criteria Shortest Path Problems

Multi-criteria shortest path problems, despite their presence in our daily life, have not received extensive attention in specialized literature. Most of the literature available discusses the networks with the constraints involving time delay, memory, etc. In this paper [10], two algorithms are discussed and proven to address the multi-criteria shortest path problem. One of them is an extension of Hansen's multiple labeling scheme algorithm for the bicriteria scenario. Building upon this algorithm, it is demonstrated that any pair of nondominated paths can be linked by other nondominated paths. The paper provides illustrations for both algorithms, which helps understanding the proofs and logic of the algorithms.

## 2.7 An Extended Algorithm for Multi-Objective Shortest Path Problems

The Multi-Objective Shortest Path (MOSP) problem is a highly researched area in the field of Multi-Objective Optimization. It consists of two primary types of objectives: formulated by a linear function and by a max-min function. These objectives are denoted as $(\sigma - S|\mu - M)$, where $\sigma$ signifies first type objectives and $\mu$ second type objectives. For example, objectives like time or distance belongs in the S-type category as we aim to minimize it, however qualities like safety are categorized as M-type objectives, aiming for maximization. The paper [11] published by Xavier Gandibleux et al. in 2006 provides an extension of Martins' algorithm made for $(\sigma - S|\mu - M)$ objective problems. The paper provides mathematical foundations, algorithmic descriptions, experimental results.

### 2.8 Conclusion

In conclusion, the literature reviewed here collectively addresses the intricate challenges of multi-objective optimization problems and route planning. The findings contribute valuable insights into algorithm selection and considerations for safety. The integration of safety into route optimization remains a central theme, emphasizing the ongoing efforts to balance speed and safety in navigation applications.

## 3 Solution

In this section, I illustrate the methodologies and techniques that were used to tackle the problem in this project.

### 3.1 Path Calculation

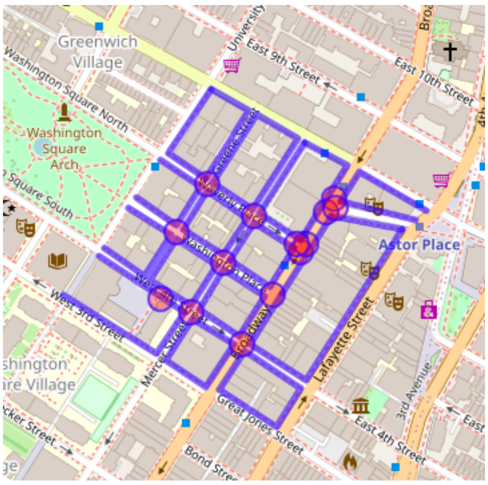


Figure 4: Conversion of Area around NYU Tisch School of Arts into a Graph

Using the OpenStreet API [12], we converted the map of Manhattan to a graph with nodes and edges so we can use it to extract data like shortest paths from point A to point B. To illustrate this visually, refer to Figure 4 which displays the graph created from the 150m radius around the

NYU Tisch School of Arts and we can see that the edges represent the pathways, while the nodes represent crossings between streets. Each node has an attribute that contains the latitude and longitude, which provided us with the ability to calculate the distance between two nodes using the Haversine formula:

$$
\begin{aligned}
a &= sin^2(\frac{\Delta\phi}{2}) + cos(\phi_1) \cdot cos(\phi_2) \cdot sin^2(\frac{\Delta\lambda}{2}) \\
c &= 2 \cdot atan2(\sqrt{a}, \sqrt{1-a}) \\
d &= R \cdot c
\end{aligned}
$$

Where: $\Delta\phi$ is the difference between the latitudes, $\Delta\lambda$ is the difference between the longitudes, R is the radius of the Earth (6,378 kilometers), a is the square of half the chord length between the points, c is the angular distance in radians, and d is the distance between the two points along the surface of the sphere.

Using this graph as a foundation, when we attempted to find a safe and fast path from A to B, we first utilized Yen's k-shortest paths algorithm [13] which involves the use of Dijkstra's Algorithm through the NetworkX library to find the few hundred shortest paths between two nodes through different neighborhoods.

### 3.2 Safety Score

After we have the paths and their time duration, we need to integrate both crime data and taxi data to calculate safety scores to analyze patterns across the neighborhoods in Manhattan. To calculate the safety score of a certain neighborhood, we will use this formula:

$$S = \frac{1}{N}\sum_{i=1}^{N}(W_i \cdot \frac{C_i}{Max(C_i)}) - \alpha \cdot \frac{D}{Max(D)}$$

In this formula, $N$ represents number of types of crimes. $W_i$ represents the weight reserved for the ith type of crime and $\alpha$ signifies the weight assigned to drop-offs. $C_i$ refers to the frequency/count of the ith crime, and $Max(C_i)$ represents the maximum frequency of the ith crime across all neighborhoods. Similarly, $D$ signifies the count of the drop-offs and $Max(D)$ is the maximum frequency of drop-offs across all neighborhoods. This formula takes into account all the relevant data together with the appropriate weights assigned to each data type and provides a value that represents the neighborhood safety.

The negative sign of the $\alpha$ represents a negative relationship: a higher number of taxi drop-offs in a neighborhood results in a lower safety score (safer). This relationship is assumed to be negative based on that a high number of taxi drop-offs indicates a higher level of public presence.

We included normalization in this formula by dividing each count of a specific data type by the maximum frequency extracted across all neighborhoods in Manhattan. This is done to make sure that the impact of each data type is proportional to its commonness and this will allow us to fairly compare the safety scores between different areas.

### 3.3 Multi-Objective Algorithm

After obtaining a few hundred paths from A to B with time values and safety scores, we will utilize the Non-dominated Sorting Genetic Algorithm II (NSGA-II) to find the non-dominated solutions. The NSGA-II algorithm works on the population of potential solutions (paths). This algorithm is created for the purpose of optimizing both objectives, safety and time in our case. The NSGA-II algorithm is based on using the non-dominated sorting method where it will categorize paths onto different fronts [4]. A path is non-dominated when there is no other path in the solution set that is better in both objectives (safer and faster). To understand this visually, Figure 5 illustrates how different possible solutions are grouped to different fronts based on their objective values. The NSGA-II returns the Pareto-optimal front where they are all non-dominated, but we can move along the front to trade-off the objectives. What makes this different to NSGA is that has a new feature: elitism. This new addition ensures that non-dominated solutions persist to the next generation. The NSGA-II was implemented through the popular Python library, pymoo.

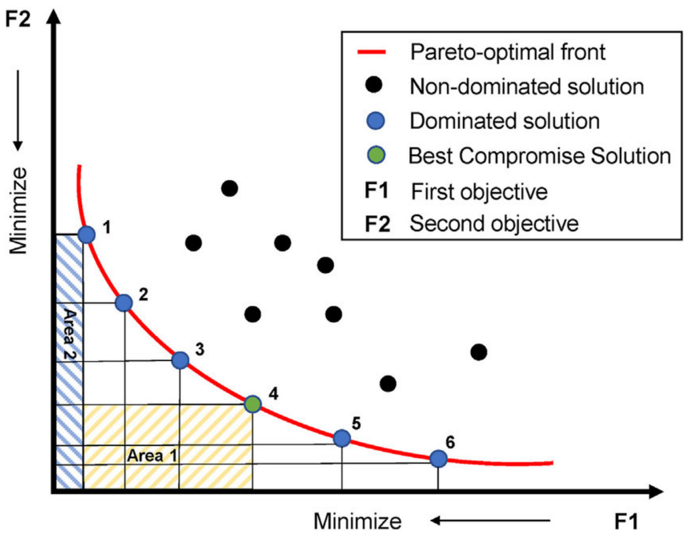


Figure 5: Pareto Front of Non-Dominated Solutions. Adapted from [14].

# 4 Results

## 4.1 Datasets

The relevant statistics that I collected for this project are: crime data from the NYPD (complaints [15] and shootings [16]) and green taxi data [17]. The crime complaints dataset contains rows of records with each having latitude, longitude, and crime type. Utilizing lexical analysis, I grouped the crime data into three parts: misdemeanors, violations, and felonies. This breakdown provides a more in depth insight into the distribution of crime across different neighborhoods in Manhattan. To comprehend the classification into these types of crime, here are a few examples of each:

- **Misdemeanors**: Petty theft, vandalism, minor drug offenses, and solicitation
- **Violations**: Jaywalking and traffic violations
- **Felonies**: Murder, assault, kidnapping, and robbery

Similarly, the shootings dataset also has latitude and longitude coordinates, thus we are able to perform spatial analysis over both datasets. This will enable us to identify which areas/neighborhoods have a high crime rate and are recommended to bypass.

The green taxi data offers a different perspective into the interactions between neighborhoods by examining the patterns of pickups and drop-offs across Manhattan. This dataset also contains records with latitude, longitude, and time values. By looking at the distribution of the data, we will be able to identify which is the popular neighborhood for drop offs.

## 4.2 Safety Score Analysis

| Minimum | Maximum | Median | 1st Quartile | 3rd Quartile |
|---|---|---|---|---|
| 2.531 | 525.0 | 43.49 | 16.55 | 124.9 |

Table 1: Safety Score Statistics

| $S < 16.55$ | $16.55 \leq S < 43.49$ | $43.49 \leq S < 124.9$ | $S \geq 124.9$ |
|---|---|---|---|
| Green | Yellow | Orange | Red |

Table 2: Heatmap Color Scale

Table 1 represents the statistics that describe the distribution of the safety scores across neighborhoods in Manhattan with the appropriate weights assigned shown in table 3. Based on these

| Violations | Felonies | Misdemeanors | Shootings | Dropoffs |
|---|---|---|---|---|
| 200 | 500 | 300 | 700 | 100 |

Table 3: Weights Assigned

values, we are able to create a color scale for a heatmap to analyze the distribution of safety scores across the neighborhoods in Manhattan.

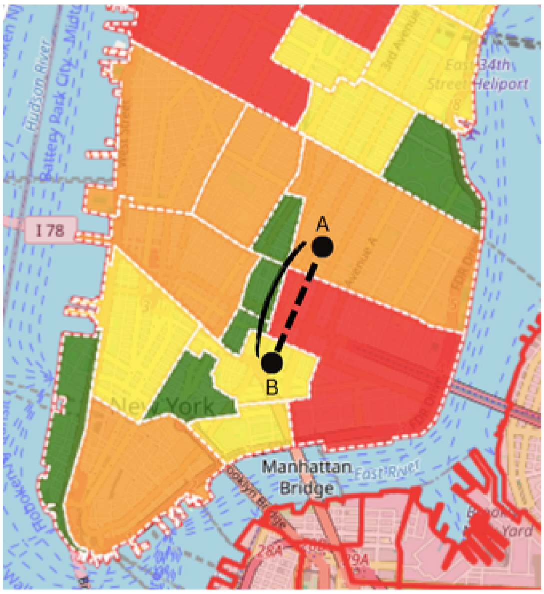


Figure 6: Lower Manhattan Heatmap

Figure 6 illustrates a heatmap of Lower Manhattan that we created based on the statistics that were calculated previously. This representation provides an overview over the patterns of safety scores over the areas. We can evidently see that this figure is more diverse than figure 1. Looking at this heatmap, there are two locations represented by point mark, call them A and B. We will take these points as starting and ending location for this project. There are two paths drawn from point A to point B. The dashed line represents the shortest path between as this path is perpendicular between the two points, but it passes through Lower East Side where the safety score illustrates that it is a dangerous area. While, the solid line represents an although longer pathway, it instead passes through Nolita where it's green color represents a quite safe region. This emphasizes the need for prioritizing safety in order to provide an alternative pathway that guides the user through a relatively safe region.

By examining the distributions of crime in the bar charts in figure 7 and 8 below for each neighborhood, we can clearly see that the actual counts of each crime is much lower in Nolita, especially with shootings being 0. We have each bar representing a different type of crime like misdemeanors, violations, felonies, and shootings. And this provides a clear visual representation of each crime group in these neighborhoods. With this in mind, we can definitely say that when we traverse the walking routes from point A to point B, we would rather go through the Nolita neighborhood and bypass Lower East Side.

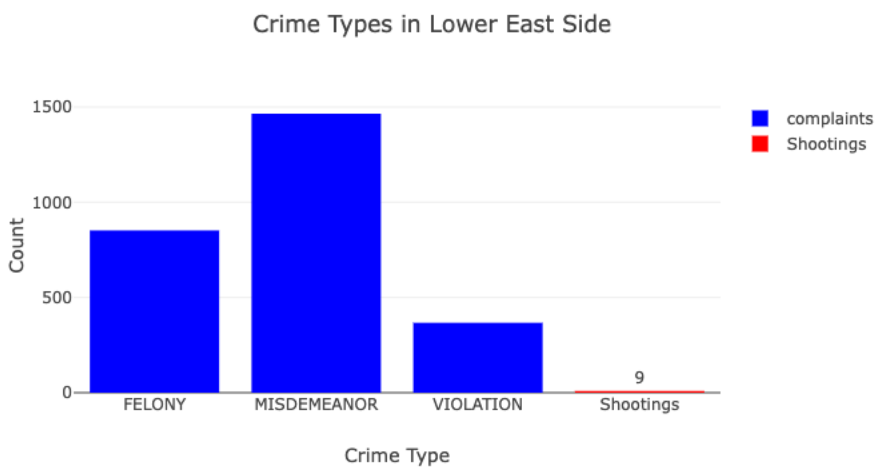


Figure 7: Crime Distribution in Lower East Side

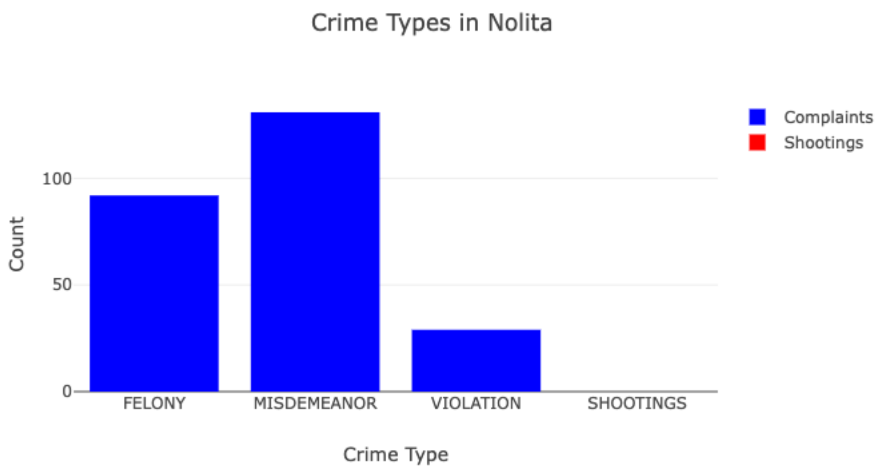


Figure 8: Crime Distribution in Nolita

### 4.3 NSGA-II Graph

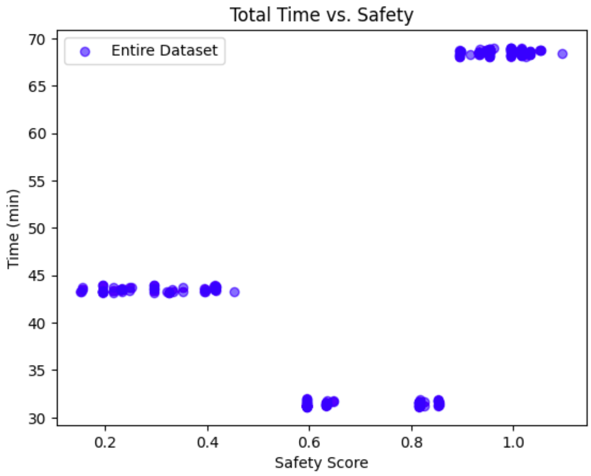


Figure 9: Scatter Plot of the Paths from A to B

The calculation of the safety scores along the paths produced by Yen's k-shortest path algorithm in Figure 9, where we plot the time vs safety score. By looking at the graph, its clear that we can group the data points into three distinct paths: the shortest path that goes through Lower East Side but high safety score (more dangerous), medium path that goes through Nolita but much lower safety score, and finally a long path that goes through multiple other neighborhoods with a high safety score. The reason why each group has multiple points with basically identical times is that the graph that was utilized contains nodes that are irrelevant. This will be discussed further in the discussion section.

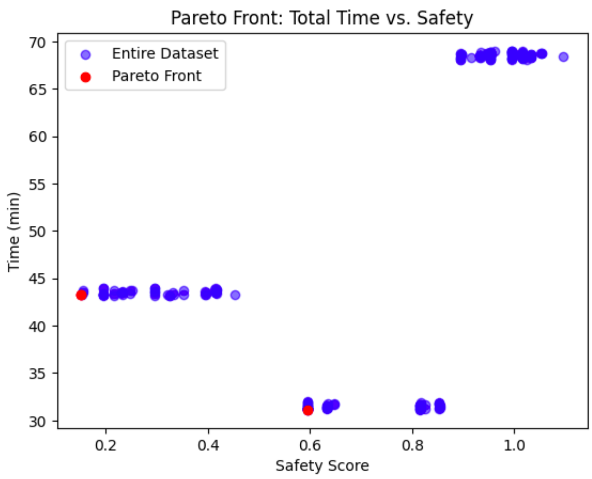


Figure 10: NSGA-II Graph

After running the NSGA-II on the data produced by Yen's k-shortest path algorithm, we can see that the red points in figure 10 are returned to be the Pareto front where we have two possible paths. With this in mind, this figure tells us that the long path with a high safety score should not be taken into account. The paths to take from A to B, we either bypass Lower East Side or go through it. But, as we previously saw from the crime distribution of these neighborhoods, we would like to avoid Lower East Side.


Figure 11: Path Recommended by Google Maps

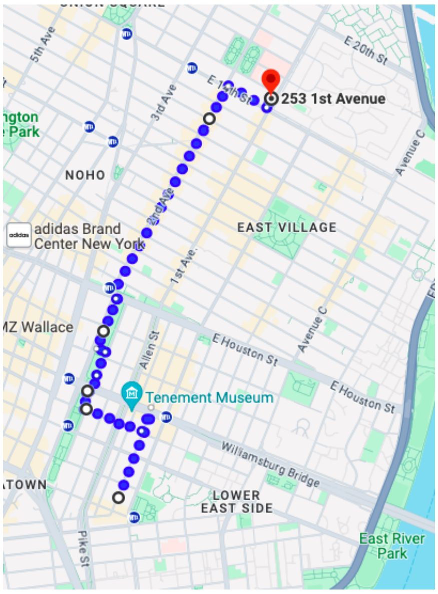


Figure 12: Path Recommended by NSGA-II

Figure 11 shows that Google Maps provides the path that is shortest but goes through the Lower East Side, which results in a higher safety score. But our calculation shows to take a longer path which can be seen in figure 12, where passes through Nolita which is a safer neighborhood and bypasses Lower East Side.

# 5 Discussion

## 5.1 Main Challenges

There were quite a few challenges encountered during the execution of this project. Firstly, the algorithm [13] used to calculate the few hundred shortest paths from point A to point B was a quite time-consuming process. This project could be improved by instead implementing a computationally efficient algorithm.

Secondly, the graph structure extracted from the OpenStreetMap API [12] had some flaws. Specifcally, the graph contained nodes that were literally a few steps away from each other, which made them irrelevant. This redundancy resulted in a more computationally expensive process with no benefit of processing these types of nodes. This could have been solved by extracting a graph structure from another source, or removing these nodes from the graph ourselves. This would've improved the performance level of the path calculation process and possibly given more diverse solutions.

Lastly, computing the safety scores along each path was a very time consuming process. This issue could've been solved by some parallelization methods to speed up the computations.

## 5.2 Advantages Over Previous Approaches

Compared to previous works that solely considered vehicle routing problems, this paper focuses only on walking routes because in cities like NYC, walking is a very common form of transportation. Furthermore, our project takes taxi drop-offs data into consideration and this provides a unique perspective on the dynamics of neighborhoods.

## 5.3 Other Approaches

It is worth considering utilizing other multi-objective optimization algorithms on the path data set to examine if the solutions produced are different compare to the NSGA-II algorithm.

# 6 Conclusion

Using the NSGA-II algorithm produced a path that avoided dangerous neighborhoods by using the data obtained from the calculation of the time duration and safety scores of each path.

The consideration of the taxi data offered a unique viewpoint on the safety aspect. This inclusion resulted in a more complete understanding of crime dynamics in different neighborhoods.

Although the NSGA-II produced promising results, it would be great for future work to implement various different multi-optimization algorithms. The combination of utilizing more computationally efficient methods to handle the computations and different multi-objective algorithms may result in more dynamic path recommendation systems.

# References


[1] Nyc crime map. [Online]. Available: https://www.arcgis.com/apps/instant/sidebar/index.htmlappid=8153f961507040de8dbf9a53145f18c4

[2] E. Zitzler, *Evolutionary algorithms for multiobjective optimization: Methods and applications*. Shaker Ithaca, 1999, vol. 63.

[3] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK report*, vol. 103, 2001.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] F. Ahmed and K. Deb, "Multi-objective path planning using spline representation," in *2011 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2011, pp. 1047–1052.

[6] H. Yetgin, K. T. K. Cheung, and L. Hanzo, "Multi-objective routing optimization using evolutionary algorithms," in *2012 IEEE wireless communications and networking conference (WCNC)*. IEEE, 2012, pp. 3030–3034.

[7] S. M. Mostafa, S. M. Habashy, and S. A. Salem, "A new framework for multi-objective route planning in smart cities," in *International Conference on Advanced Intelligent Systems and Informatics*. Springer, 2022, pp. 824–837.

[8] N. Jozefowiez, F. Semet, and E.-G. Talbi, "Multi-objective vehicle routing problems," *European journal of operational research*, vol. 189, no. 2, pp. 293–309, 2008.

[9] R. G. Garroppo, S. Giordano, and L. Tavanti, "A survey on multi-constrained optimal path computation: Exact and approximate algorithms," *Computer Networks*, vol. 54, no. 17, pp. 3081–3107, 2010.

[10] E. Q. V. Martins, "On a multicriteria shortest path problem," *European Journal of Operational Research*, vol. 16, no. 2, pp. 236–245, 1984.

[11] X. Gandibleux, F. Beugnies, and S. Randriamasy, "Martins' algorithm revisited for multi-objective shortest path problems with a maxmin cost function," *4OR*, vol. 4, no. 1, pp. 47–59, 2006.

[12] Overpass api documentation. [Online]. Available: https://wiki.openstreetmap.org/wiki/Overpass_API

[13] J. Y. Yen, "Finding the k shortest loopless paths in a network," *management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[14] R. Kyomugisha, C. M. Muriithi, and M. Edimu, "Multiobjective optimal power flow for static voltage stability margin improvement," *Heliyon*, vol. 7, no. 12, 2021.

[15] Nypd complaint data current (year to date). [Online]. Available: https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Current-Year-To-Date-/5uac-w243

[16] Nypd shooting incident data (year to date). [Online]. Available: https://data.cityofnewyork.us/Public-Safety/NYPD-Shooting-Incident-Data-Year-To-Date-/5ucz-vwe8

[17] 2015 green taxi trip data. [Online]. Available: https://data.cityofnewyork.us/Transportation/2015-Green-Taxi-Trip-Data/gi8d-wdg5/data