ABSTRACT

An optimal traffic evacuation plan is really important to reduce the devastating impact of tsunamis. In this study, we came up with a traffic evacuation strategy designed specifically for tsunami situations, covering the entire network. We set up a step-by-step evacuation system to help guide vehicles on where to go. In the first stage of the evacuation, the goal is to make sure everyone gets from the affected areas to temporary shelters as quickly as possible to ensure their basic safety. Then, in the second stage, the focus shifts to reducing the total time it takes to move everyone from the temporary shelters to the safe zones. For this solution, we defined the objective functions for Minimizing Maximum Evacuation Time (MM) and Link-Based System Optimization (LSO).

ABSTRACT

ACKNOWLEDGEMENT

CONTENTS

1. Introduction

Tsunami is one of the natural disasters that cause significant material and human losses with its devastating effects. According to the distance from the source to the shore and the time required to traverse this distance, tsunamis can be classified into three categories: local, regional, and distant tsunamis. Although there are differences among these categories, they all occur as a result of large earthquakes or underwater landslides and can inflict great damage on coastal settlements. In such disaster situations, evacuation—which is the most effective way to survive—constitutes the most important part of emergency planning.

Evacuation processes are influenced by many factors such as the demographic structure of the region, geographic features, the layout of the traffic network, and existing infrastructure. For example, in an area without high ground, vertical evacuation strategies become prominent, while in scenarios where time is limited, the effectiveness of horizontal evacuations increases. Thanks to advanced tsunami warning systems, there is sufficient time for people in the affected areas to reach safe zones. In this case, evacuations using vehicles along traffic networks play an important role in enabling people to reach safe areas.

In the event of a tsunami threat, it is critically important that evacuation information reaches the public quickly and accurately. Nowadays, due to the widespread use of communication tools such as mobile devices, social media, email, and instant messaging systems, local authorities can rapidly convey warnings and evacuation orders. This makes it possible to direct everyone affected to safe areas.

Human behavior changes during emergencies; factors such as panic, uncertainty, and crowd psychology can complicate evacuation processes. Therefore, preparing an effective evacuation plan not only involves determining routes but also developing methods that will facilitate people's adherence to these plans. In the article and in my work, studies were conducted on determining the most suitable routes on a theoretical basis by assuming that all affected individuals will comply with evacuation orders.

The study focused on optimizing evacuation processes using vehicles during tsunamis. In this context, a traffic network was examined by dividing it into three main regions: the affected area, the temporary shelter area, and the safe area. The journey starting from the affected area involves first reaching a temporary shelter area and then proceeding from there to a safe area. In both stages, the aim is to ensure that people can reach safe areas as quickly as possible and that the most efficient overall evacuation flow is achieved.

In the first stage, the focus was on minimizing individual evacuation times to ensure that everyone reaches the shelter as quickly as possible. This requires an approach aimed at reducing the longest time that any individual may encounter. In the second stage, I developed strategies to minimize the total evacuation time of the entire system in the process of moving from temporary shelters to safe areas. At this stage, the aim was to determine the most effective routing methods across the system in a calmer environment.

The study presents new approaches for optimizing vehicle routing strategies in emergencies such as tsunamis. In this way, both individual safety is maximized and overall system efficiency is increased. With the developed model and solution methods, the aim was to contribute to the creation of the most appropriate evacuation plans in the event of a disaster.

2. Methodology

In this section, three fundamental optimization methods for emergency evacuation planning will be addressed: Minimizing Maximum Evacuation Time (MM), Lexicographic Minimizing Maximum Evacuation Time (LMM), and Link-Based System Optimization (LSO).

2.1 Assumptions and research design

At the beginning of the study, several fundamental assumptions were made for modeling the evacuation process:

1. Vehicle Type and Passenger Groups: All evacuees use vehicles with standard characteristics for passenger transportation. Individuals sharing a vehicle are considered as a single group.

2. Access to Information: All participants in the evacuation have full access to dynamic route guidance and alerts provided by local authorities.

3. Compliance with Instructions: Everyone involved in the evacuation fully adheres to the provided guidance.

4. Traffic Flow: The evacuation traffic flow is modeled as an independent flow unaffected by external factors.

2.2 Objective Functions

2.2.1 Minimizing Maximum Evacuation Time (MM) Problem

The MM problem aims to minimize the longest evacuation time among all individuals in the affected area. This ensures that the evacuation for any individual in the system is completed in the shortest possible time. Mathematically:

$$Minimize\ Z = max\left(t_k\left(f_k^{sd} \right) \right) \quad \forall k, s, d$$

Constraints Considered in the Solution:

I. **Flow Balance at the Source Node:** The total traffic flow on all outgoing paths from a source node must equal the total number of individuals present at that node.

II. **Flow Balance on Source-Destination Paths:** The total traffic flow on all paths originating from a specific source node must match the total number of people to be evacuated from that node.

III. **Flow Balance for Individuals Waiting in Temporary Shelters:** The total traffic flow on all paths connected to the holding nodes in temporary shelters must equal the total number of individuals waiting in the network.

IV. **Overall Network Flow Balance:** The total traffic flow across all paths in the network must match the total number of individuals in the affected area.

V. **Capacity Constraint:** The traffic flow on any path (link) must not exceed the carrying capacity of that path. This prevents network congestion and ensures safety.

VI. **Non-Negative Flow:** The amount of traffic flowing on any path or route cannot be negative; flows must be zero or positive.

VII. **Path-Link Relationships, Length, and Travel Time Constraints:** The traffic flow on a path must be consistent with the flows on all links in that path. Additionally, the total length and travel time of a path must equal the sum of the lengths and travel times of all links within it.

2.2.2 Link-Based System Optimization (LSO) Problem

The primary goal of the LSO problem is to minimize the total travel time and flow across all links in the network. Mathematically:

$$Minimize\ ;\ Z\ =\sum_{\{a\ \in A\}} x_a\, t_a\,(x_a)$$

Constraints Considered in the Solution:

I. **Flow Balance for Temporary Shelters**: The total traffic flow on all paths connected to holding points in temporary shelters must equal the total number of individuals waiting there, ensuring the required traffic flow for each individual to reach the final safe area.

II. **Flow Balance for Exit Paths from Temporary Shelters**: The total traffic flow on all paths originating from any temporary shelter must match the number of people present at that shelter.

III. **Flow Balance for Access to Safe Areas:** The total traffic flow on all paths connected to safe areas must match the total number of people to be evacuated.

IV. **Capacity Constraint:** The traffic flow on each link must not exceed its capacity, minimizing congestion and ensuring smooth flow.

V. **Non-Negative Flow:** The traffic flow on any path or route cannot be negative.

VI. **Path-Link Relationships and Additional Constraints:**

- The traffic flow on a path must be consistent with the flows on all links in that path.
- Path lengths and travel times must equal the sum of the values of all links within them.
- Flow on certain paths must adhere to the links these paths traverse.

2.2.3 Lexicographic Minimizing Maximum Evacuation Time (LMM) Problem

In the MM problem, there may be multiple optimal solutions, making it difficult to determine which solution is the most equitable or appropriate. To eliminate this ambiguity and ensure a fair distribution, the lexicographic approach is applied.

Lexicographic minimization of maximum evacuation time not only reduces the longest evacuation time but also considers other evacuation times in a sequential manner. In the first step, the longest evacuation time for all individuals is minimized as much as possible. However, if multiple solutions meet this criterion, the lexicographic approach distinguishes these solutions by minimizing the second-longest evacuation time. This process continues until fair improvement is achieved across all evacuation times.

This method draws inspiration from John Rawls' principle of justice, which advocates improving the condition of the least advantaged individuals as a basis for a just order. In practice, the lexicographic approach focuses primarily on improving the situation of those in the worst condition, i.e., those with the longest evacuation times. If minimizing this longest time yields various solutions, subsequent steps aim to minimize the second and third longest times, ensuring overall fairness.

To make the lexicographic solution strategy more effective, the Tabu Search method is employed. In each stage of the lexicographic approach, the potential solution set is systematically evaluated, and unsuitable solutions are blocked using a "tabu list."

This process ensures that not just one individual, but all individuals' evacuation times are reduced equitably.

3. Experimental tests in Article

3.1 Study area

The study area is located at Waikiki Beach in Honolulu, Hawaii, one of the most visited and crowded beaches in the world.

In this study, two different scenarios are considered:

- **Scenario 1:** The study area is affected by a local tsunami caused by an earthquake near the Hawaii County.
- **Scenario 2:** The study area is affected by a tsunami caused by an earthquake occurring far from Hawaii.

It is assumed that the tsunami event occurs during the summer at noon, affecting a total of 33,000 people.

In Scenario 1, due to the very short tsunami warning time, approximately 16,000 people will evacuate by vehicles, while the remaining 17,000 people will take refuge in buildings. In Scenario 2, because the evacuation time is expected to be very long, a complete evacuation of all 33,000 people will be carried out.

3.2 Result discussions

For the first phase of evacuation, the Tabu Search algorithm is a more reasonable choice because it keeps the maximum travel time lower. However, in the second phase, thanks to the cooperation among evacuees, the LSO method—which results in a smaller total travel time—was utilized.

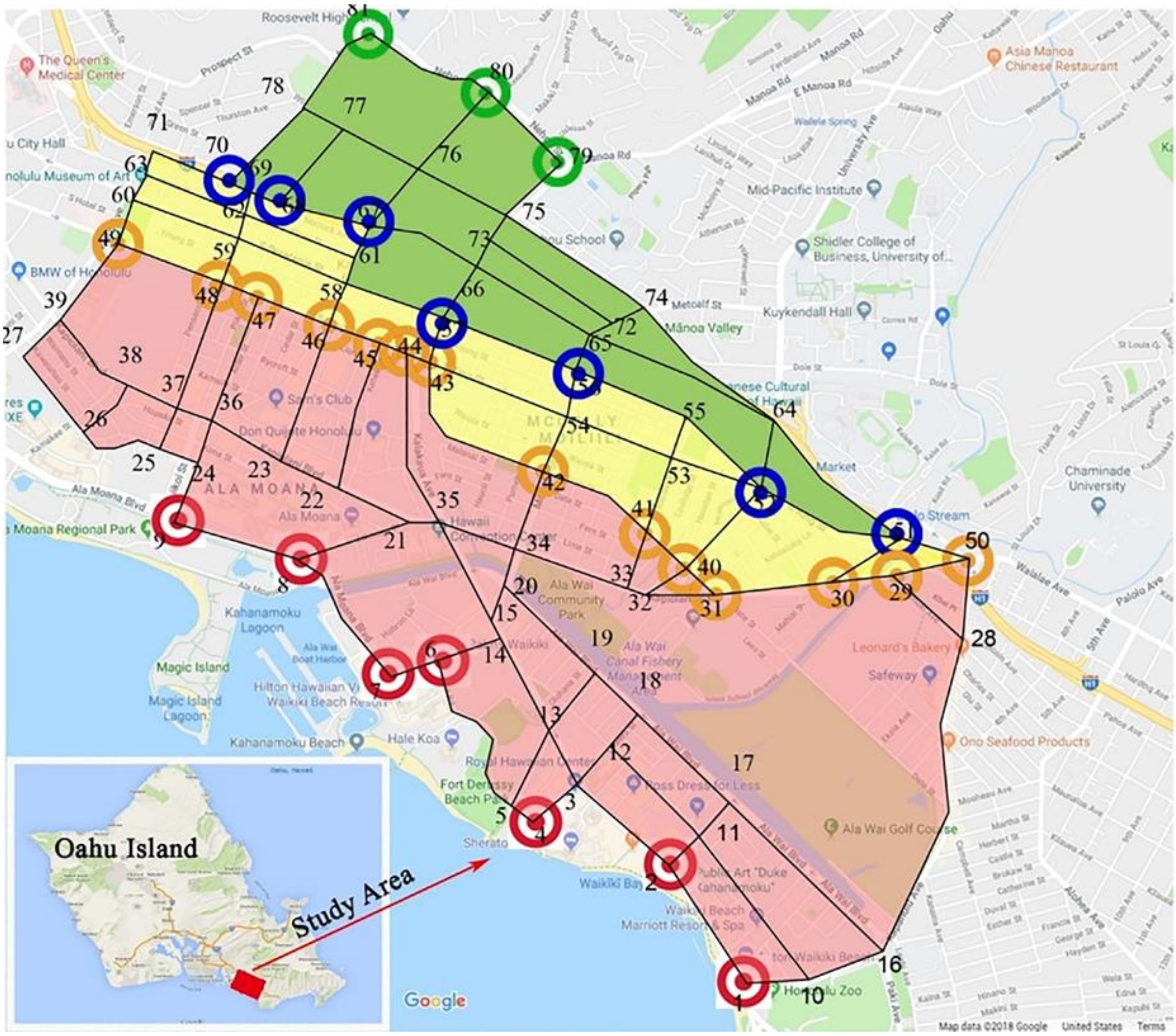


Fig. 1. Illustrative roadway network in the study area.

In Figure 1, the red dots represent the starting points, while the green dots represent the destination points. The red area indicates the zone that will be most affected by the tsunami, whereas the orange area represents a temporary zone. The entrances to this temporary zone are marked by orange dots, and the exits are marked by blue dots. When transitioning to Evacuation Phase 2, the exits from blue dots.

Comparison of proposed models with other traffic assignment methods.

Optimization goal	First phase of evacuation			Second phase of evacuation		
	LMM[a]	MM[b]	LSO[c]	LMM[a]	MM[b]	LSO[c]
Scenario 1						
Total Travel Time (s)	5,486,805	5,962,720	5,173,772	5,120,320	5,190,560	5,030,363
Maximum Travel Time (s)	458.81	459.32	896.79	342.86	343.27	378.06
Average Travel Time (s)	342.93	372.67	323.36	320.02	324.41	314.40
Scenario 2						
Total Travel Time (s)	20,020,397	21,614,080	18,452,672	18,228,480	18,486,080	17,340,197
Maximum Travel Time (s)	862.37	862.52	1342.68	780.11	780.33	978.17
Average Travel Time (s)	625.64	675.44	576.65	569.64	577.69	541.88

Figure 2

In Figure 2, the results of the first and second phases are presented. As stated in the Result Discussion section, the Tabu Search algorithm was preferred in the first phase due to its lower maximum travel time. In the second phase, the LSO algorithm was used because it provides a lower total travel time.

Evacuees' link-based routing details in Scenario 1.

Link No.	Flow[a]	Slack[b]	BPR[c]	Link No.	Flow[a]	Slack[b]	BPR[c]
1–10	2000	2400	43.17	28–50	1527	2873	53.26
2–11	1000	1200	23.98	32–31	0	2200	19.80
3–12	969	1231	25.94	32–40	0	2200	27.00
3–13	4031	2569	35.35	33–32	0	4400	10.80
5–13	0	4400	37.80	33–41	1360	840	53.58
6–14	3000	3600	35.97	34–33	1360	5240	66.67
8–21	2000	2400	83.94	34–42	4400	0	129.74
9–24	3000	3600	45.56	35–34	332	6268	48.73
10–16	2168	2232	60.34	35–44	4195	2405	170.84
11–10	168	4232	77.51	36–37	468	6132	21.72
11–12	0	4400	68.40	36–47	2200	0	202.39
11–17	832	1368	21.99	37–38	0	2200	28.80
12–18	969	1231	35.37	37–48	800	1400	84.48
13–14	600	6000	54.52	38–39	0	2200	50.40
13–19	3431	969	71.46	39–49	0	4400	52.20
14–15	3600	5200	13.65	**51–64**[d]	0	2200	90.00
15–20	1028	3372	23.26	**52–64**	0	4400	36.00
15–35	2527	4073	68.51	**56–65**	0	6600	27.00
16–28	3000	3600	235.02	**57–66**	600	3800	35.00
17–16	832	3568	118.87	**64–65**	0	4400	86.40
17–18	0	6600	68.40	**64–74**	0	4400	90.00
18–19	969	5631	27.75	**65–66**	0	13,200	79.20
19–20	4400	2200	118.00	**65–72**	0	6600	14.40
20–34	5428	3372	77.24	**66–73**	600	8200	23.52
21–22	0	6600	41.40	**67–76**	6600	0	145.31
21–35	2000	2400	23.98	**68–77**	2200	0	197.20
22–23	0	6600	32.40	**70–78**	6600	0	207.58
22–45	0	2200	81.00	**72–73**	0	2200	79.20
23–46	0	4400	77.40	**72–74**	0	4400	14.40
24–25	332	1868	22.23	**73–75**	600	10,400	23.47
24–36	2668	3932	31.67	**74–75**	0	6600	79.20
25–26	0	4400	52.20	**75–76**	0	6600	52.20
25–37	332	6268	25.27	**75–79**	600	10,400	39.72
26–27	0	4400	48.60	**76–77**	0	2200	73.80
26–38	0	4400	25.20	**76–80**	6600	4400	89.95
27–39	0	4400	28.80	**77–78**	2200	2200	45.68
28–29	1473	2927	58.95	**78–81**	8800	2200	170.48

Figure 3

In Figure 3, the results of LSO for Scenario 1 are presented. The flow on each path and the corresponding BPR value, which changes accordingly, are shown. (Based on my calculations, the α value is approximately 1.885, and the β value is approximately 2.2.)

4. Optimization Model

In this study, both mixed-integer nonlinear programming (MINLP) and integer linear programming (ILP) optimization solutions were implemented for evacuation processes, and both methods were successfully executed. The study presents a two-stage modeling approach for evacuation optimization in emergency scenarios. The objective function is defined as minimizing evacuation time. The model was formulated within the framework of nonlinear programming (NLP) using the Pyomo library. Additionally, network structures were created using the NetworkX library, and the routes to be taken were determined through initially defined pairs of possible paths and subsequent constraints added via the optimization model. The origin-destination paths, including intermediate points, were identified. The model minimized travel time while regulating passenger flow in accordance with capacity constraints.

In the MINLP model, the Bureau of Public Roads (BPR) formula was used, with alpha and beta values set to 2. For the ILP model, no additional formulation was applied for travel time.

The following constraints were incorporated into both the ILP and MINLP optimization models using the model.constraint structure. Some constraints were also handled during parameter initialization to ensure they were logically consistent:

1. **Flow Balance at Source Nodes:** The total traffic flow across all connections linked to any source node must equal the total number of individuals at that source node.

2. **Flow Balance Across All Nodes:** Traffic flows associated with all transit points (holdover nodes) and temporary shelters within the network must be balanced. In other words, the total traffic entering a node must equal the total traffic leaving that node.

3. **Connection Capacity Constraints:** The maximum capacity of each connection must not be exceeded. This constraint accounts for the physical capacity limitations of roads, preventing congestion and overloads during the evacuation process.

4. **Non-Negative Flow Constraint:** Traffic flow across all connections and paths must be non-negative.

5. **Relationship Between Connections and Paths:** Traffic flow on a connection is determined by whether that connection is part of a specific path.

6. **Relationship Between Travel Time and Traffic Flow:** The total travel time on any path is determined by the travel times of the connections on that path and the traffic flow on those connections.

These constraints were applied to solve the optimization model.

4.1. MINLP Solution

The model was mathematically defined using the Pyomo library and subsequently solved with the IPOPT solver. Pyomo is a Python library used to define and solve linear, nonlinear, and mixed-integer programming problems. The process of creating a model with Pyomo begins by defining sets that represent elements in the model, such as nodes or edges. Next, parameters that will be used as constants in the model, like capacities or travel times, are specified. Based on these parameters, variables to be optimized—for instance, the flow amounts on edges—are defined. The goal of the model is then expressed through an objective function, such as minimizing total travel time. Following this, mathematical constraints, such as capacity limitations, are added to the model. Once all these steps are completed, the model is sent to a solver to find the optimal solution.

4.2 ILP Solution

For the ILP model, the DOcplex library, which is the Python interface of IBM CPLEX Optimization Studio, was utilized. Although gap values were not printed, optimal solutions were generated. The objective function aimed to minimize the maximum evacuation time.

5. Showing My results

EDGE FLOWS AND TRAVEL TIMES:

Edge	Flow	Travel Time	Capacity Usage
Başlangıç51 -> Ara64	2200	270.00	100.0
Başlangıç52 -> Ara64	2295	55.59	52.2
Başlangıç56 -> Ara65	2437	34.36	36.9
Başlangıç57 -> Ara66	1862	46.45	42.3
Başlangıç67 -> Ara76	2963	70.72	44.9
Başlangıç68 -> Ara77	1590	139.86	72.3
Başlangıç70 -> Ara78	2649	95.20	40.1
Ara64 -> Ara65	4374	257.16	99.4
Ara64 -> Ara74	121	90.14	2.8
Ara65 -> Ara66	6793	121.15	51.5
Ara65 -> Ara72	18	14.40	0.3
Ara66 -> Ara73	8655	68.67	98.4
Ara72 -> Ara73	9	79.20	0.4

Ara72 -> Ara74	8	14.40	0.2
Ara73 -> Ara75	8664	52.43	78.8

The results printed in the table above show the flow between links, travel time (in terms of BPR), and capacity usage. The results have been presented and the values displayed as in the paper.

```
PATH FLOWS:
--------------------------------------------------------------------------------
Source      Target     Path #    Flow    Time
--------------------------------------------------------------------------------
Başlangıç51    Bit79          1      27    482.27
  Path: Başlangıç51 -> Ara64 -> Ara74 -> Ara75 -> Bit79
Başlangıç51    Bit79          2      10    267.86
  Path: Başlangıç51 -> Başlangıç52 -> Ara64 -> Ara74 -> Ara75 -> Bit79
Başlangıç51    Bit79          3      514    812.29
  Path: Başlangıç51 -> Ara64 -> Ara65 -> Ara66 -> Ara73 -> Ara75 -> Bit79
Başlangıç51    Bit79          4       1    716.07
  Path: Başlangıç51 -> Ara64 -> Ara65 -> Ara72 -> Ara73 -> Ara75 -> Bit79
Başlangıç51    Bit79          5       1    678.10
  Path: Başlangıç51 -> Ara64 -> Ara65 -> Ara72 -> Ara74 -> Ara75 -> Bit79
Başlangıç51    Bit79          7      21    597.88
  Path: Başlangıç51 -> Başlangıç52 -> Ara64 -> Ara65 -> Ara66 -> Ara73 -> Ara75 -> Bit79
Başlangıç51    Bit79          8      12    319.49
  Path: Başlangıç51 -> Başlangıç52 -> Başlangıç56 -> Ara65 -> Ara66 -> Ara73 -> Ara75 -> Bit79
Başlangıç51    Bit80          1      22    734.09
  Path: Başlangıç51 -> Ara64 -> Ara74 -> Ara75 -> Ara76 -> Bit80
```

I printed each connected link with its flow in this way. These links indicate which path passengers starting from the same origin choose and which destination they reach. It prints the flow of each passenger group, meaning the number of passengers, and their travel time. In this way, we can see how many passengers took which route and how long it took them. This enhances our analysis capacity and enables us to further refine the model.

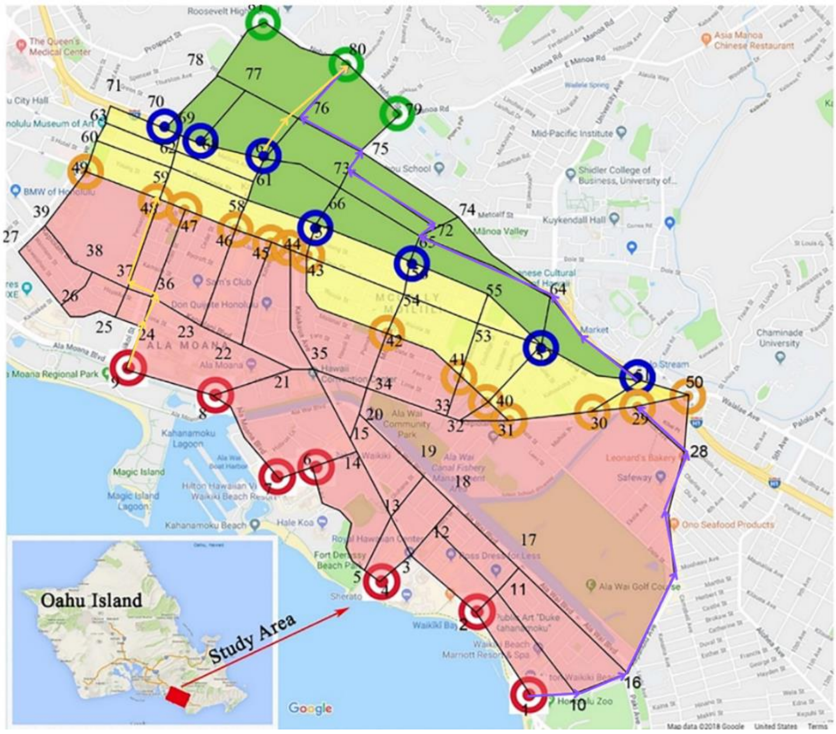


Fig 4

After my LSO model, on the Hawaii Honolulu map, the longest route taken (purple line) and the shortest route taken (yellow line) are shown.

6. COMPARİSON OF RESULTS

In this section, the results I obtained are compared with the results presented in the article.

6.1 MINLP

When the Ipopt model is run, it provides information about the structure of the optimization problem and the algorithm's progress. The results indicate the number of iterations and whether the solution is optimal, feasible, or infeasible.

6.1.1 SCENERIO 1

It has been tested on the map of Honolulu, Hawaii, with 16,000 people.

Method	**Total Travel Time (s)**	**Maximum Travel Time (s)**	**Average Travel Time (s)**
LMM (Article)	5,486,805	458.81	342.93
MM (Article)	5,962,720	459.32	372.67
LSO (Article)	5,173,772	896.79	323.36
MM	6,440,908	679.27	402.57
LSO	4,806,077	411.4	300.4

Table 1 Scenerio 1 Phase 1

According to the values in Table 1, considering the phase as well, it is observed that my LSO model performs better.

Method	**Total Travel Time (s)**	**Maximum Travel Time (s)**	**Average Travel Time (s)**
LMM (Article)	5,120,320	342.86	320.02

MM (Article)	5,190,560	343.27	324.41
LSO (Article)	5,030,363	378.06	314.40
MM	7,537,274	679.27	402.57
LSO	3,386,423	247.96	211.65

Table 2 Scenerio 1 Phase 2

In Table 2, it is observed that the MM function I wrote performs worse compared to the others, while the LSO function performs very well. Here, the second result has been found optimal based on the solver I used. The MM function performed 10,000 iterations here.

6.1.2 Scenerio 2

It has been tested on the map of Honolulu, Hawaii, with 33,000 people.

Method	Total Travel Time (s)	Maximum Travel Time (s)	Average Travel Time (s)
LMM (Article)	20,020,397	862.37	625.64
MM (Article)	21,614,080	862.52	675.44
LSO (Article)	18,452,672	1342.68	576.65
MM	19,108,159	858.73	579.24
LSO	18,887,059	859.02	572.33

Table 3 Scenerio 2 Phase 1

Here, my MM values result in a smaller total travel time. Therefore, in the first phase, we prioritize minimizing the total travel time, and for this purpose, the MM value is preferred.

Method	Total Travel Time (s)	Maximum Travel Time (s)	Average Travel Time (s)
LMM (Article)	18,228,480	780.11	569.64
MM (Article)	18,486,080	780.33	577.69
LSO (Article)	17,340,197	978.17	541.88
MM	17,632,810	1437.84	539.10
LSO	18,056,263	1489.85	547.19

Table 4 Scenerio 2 Phase 2

Since this is the second phase, using the lso values provided in the article would have been better for the total travel time. In the second phase, we are focusing on total travel time.

6.2 ILP

When the ILP optimization model is run with the same parameters, it produces lower results because the BPR formulation has not been applied. If the BPR formulation were integrated, higher values would be expected.

The ILP outputs were not presented since their format is very similar to the MINLP outputs. When executed, the model indicates that it finds the optimal solution.

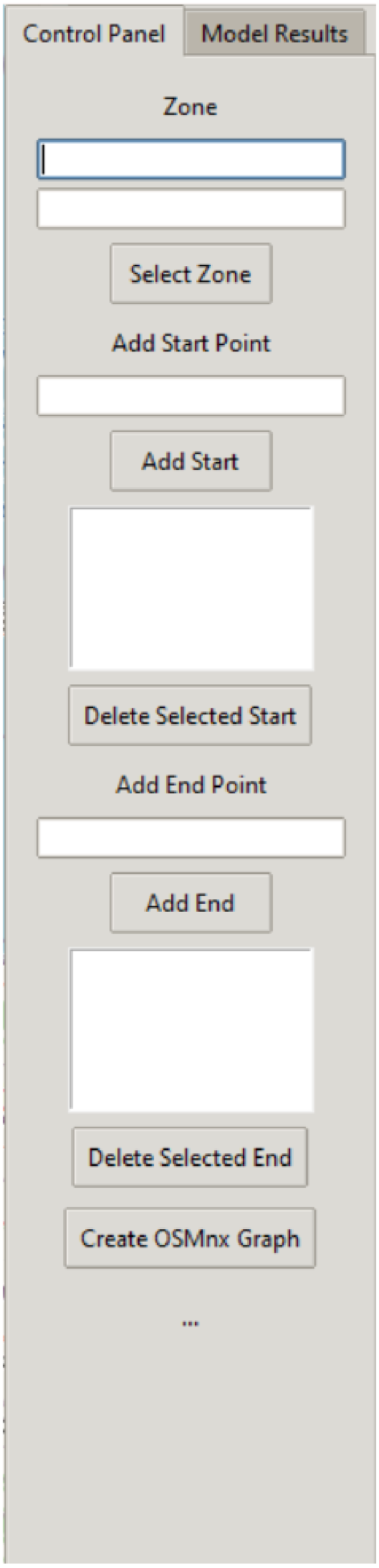
Additionally, while the final paths in the MINLP model are divided into many groups based on flows, in the ILP model these paths are grouped into fewer clusters. This difference highlights a notable variation in how each model organizes routes and flow groupings.

7. GUI

In this project, I develop an evacuation model by extracting the road network of a region using the OSMnx library. The primary goal is to complete the passenger transfer from specific points (Start) to other points (End) in the shortest possible time (or with the least total time) during a disaster or emergency situation. To achieve this, I am using a single-phase LSO (Link-Storage-Occupation) model. In this process, I consider road capacities, travel times, and passenger quantities. With a Tkinter interface, you can define the region on the map, select the start and end points, solve the model, and visualize the results on the map.

Fig 5

In this project, the capacities and travel times for the roads are calculated from the edge data of the graph returned by OSMnx. The distance of each edge (length in meters) and the average speed (speed_kph) information are taken; if the speed information is missing or incomplete, a default value (e.g., 30 km/h) is used. Then, using a simple formula (distance / speed), the time is calculated in hours and converted to minutes, thereby determining the base travel time for each edge. Capacities are estimated using a rough function based on the length of the road (for example, subtracting a portion of the distance from 5000) and applying a certain lower limit (300) to prevent very small capacity values. This way, the basic information from the map data is processed to make both road travel times and capacity values usable in the Pyomo model.


Control Panel
Model Results
Zone
Select Zone
Add Start Point
Add Start
Delete Selected Start
Add End Point
Add End
Delete Selected End
Create OSMnx Graph
...


Control Panel
Model Results
Model results will appear here.
Draw All Roads
Hide All Roads
Show Start Edge Capacities
Create/Solve Pyomo Model
Draw Solution on Map

Fig 6

7.1 How to Run

When you run the application, you will see a map on the left and two tabs on the right.

1. Control Panel Tab: Enter the corners of the region in the Zone fields in the format (lat, lon) and press the Select Zone button. A red rectangle will appear on the map. (If you right-click with the mouse, an option to copy the coordinates of the clicked location will appear.)

2. Start and End Points: Add as many coordinates as needed to the Add Start Point and Add End Point sections (e.g., 40.8470 29.2964), then press the Add Start / Add End buttons. These points will be marked on the map. (If you right-click with the mouse, an option to copy the coordinates of the clicked location will appear.)

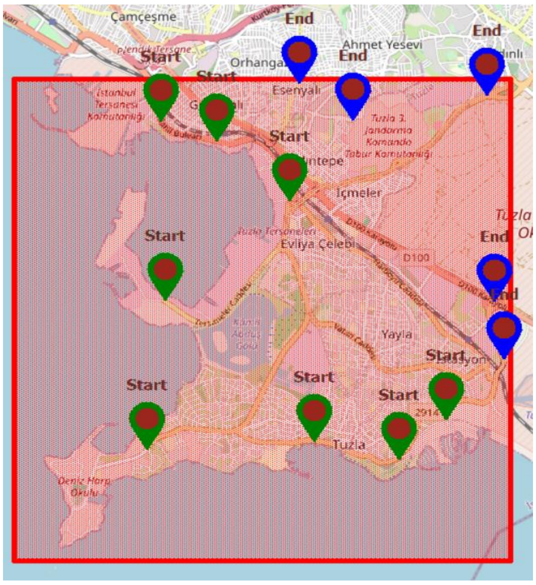


Fig 7

3. OSMnx Graph Creation: Click the Create OSMnx Graph button. This process downloads the road network of the region through OSMnx and populates the internal data structures.

4. Optional Road Visualization: If desired, you can view all roads as gray lines by clicking the Draw All Roads button.

Fig 8

5. Pyomo Model Creation and Solving: Click the Create/Solve Pyomo Model button. The system will create the Pyomo model, solve it, and display the results in the text area of the second tab.

6. Solution Visualization: Finally, click the Draw Solution on Map button to visualize the optimal flows found, highlighted on the map with colors.

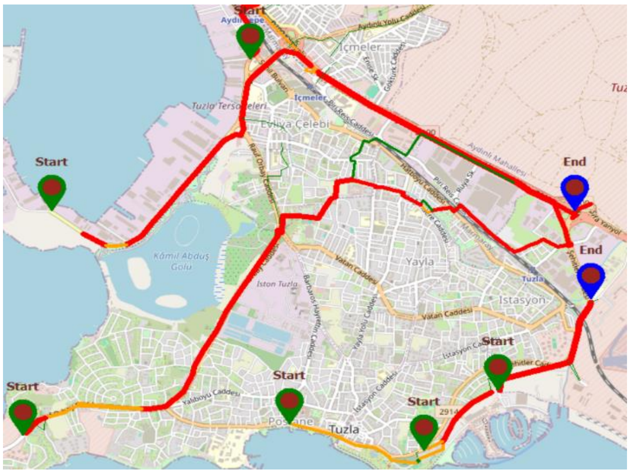


Fig 9

The GUI has been tested in three different locations: Tuzla, Kınalıada, and Honolulu.

7. Project Limitations

The main limitation of this study is the assumption that everyone can receive and follow evacuation instructions in a timely manner, which is not always feasible. In the chaotic aftermath of a tsunami, uncertainties may arise, and in economically disadvantaged areas, adequate warning systems, clean water, or medical supplies may be lacking. Additionally, language barriers, lack of access to technology, and unfamiliarity with evacuation routes can further complicate the process. In real disasters, those with insufficient resources are often unable to evacuate, leading to deviations between the models and real-world scenarios.

8. Challenges Encountered

While writing the optimization model, we faced numerous challenges. Although our problem was a non-linear integer problem, we initially approached it as a linear integer problem. Later, we transitioned from linear integer programming to non-linear integer programming. During this process, the library we used caused significant difficulties, and we were unable to implement the model exactly as we intended.

The formula that provided the non-linear programming feature was the BPR function. After transforming this function into a linear formula, we wrote it as an MILP using the docplex library. However, with this approach, we could not achieve the desired gap values, and the solution was returned in a very short time, appearing optimal. As a result, we attempted to rewrite the model in a non-linear form and adjusted it to find the optimal solution through iterative computations. In the map from the paper, sometimes the optimal solution is found so quickly that it finishes within 50 iterations. Could the reason for this be that I defined the variables differently than how they were defined in the paper?

Additionally, there were 1-2 rules not mentioned in the article but included in the algorithm's design. We resolved these rules by considering how to structure the algorithm and examining how they were applied in the article. While designing the GUI, the links that can pass through a specific zone were identified, and only the length data of these links were available. Based on this length data, capacity and travel time were calculated.

9. Conclusion

In this study, a two-stage tsunami evacuation framework with hierarchical optimization objectives was designed. In the first stage of evacuation, the objective was to minimize the maximum individual evacuation time from affected areas to temporary shelter zones. In the second stage, the goal was to minimize the total evacuation time of the entire system from temporary shelter zones to safe areas. The study was conducted in Honolulu, Hawaii, with two different scenarios. In the source article we referenced, three different models were developed, and it was recommended to use different models for the first and second phases of evacuation. However, in my study I developed only two models and ran them using different but closely related alpha and beta values. These values are close to those used in the original article, and the BPR functions also produce similar results. While my data produced different outcomes from the article in some tests, in other tests the models from the article generated very similar results.

In the tests I conducted, I generally observed that my LSO model performed better in both phases, while the MM model produced more stable values. Additionally, a GUI has been designed for the tests, which indicates the frequency of car usage on specific roads based on colors according to the LSO model. The GUI also returns the total travel time of the model.

10. Recommendations

I believe that this study can be applied to different cities. Particularly in smaller cities, it is likely to yield better results due to the constraints defined in the article. Additionally, I think the study can be adapted for natural disaster escape scenarios, such as avalanches. In a rapid evacuation scenario, with proper coordination by coordinators and effective guidance for people, I believe successful outcomes can be achieved. While defining the GUI zone, rectangles can be used instead of polygons.

TEST CASES

Test Case Gui İstanbul, Tuzla 1

According to the probabilistic tsunami hazard analysis, the probability of exceeding a tsunami wave height of 0.3 meters in the Tuzla region is over 90% within the next 50 years, and it rises to 95% within the next 100 years (2).

Fig 10

In figüre 10, the marked zone and the starting and ending points are shown.

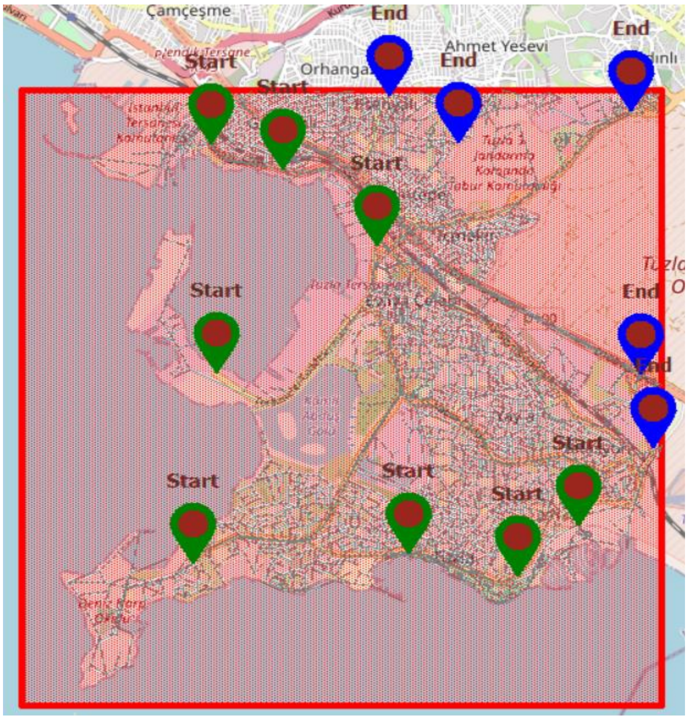


Fig 11

The gray areas represent the roads, with 5229 paths and 13,727 links created, as shown in Figure 11.

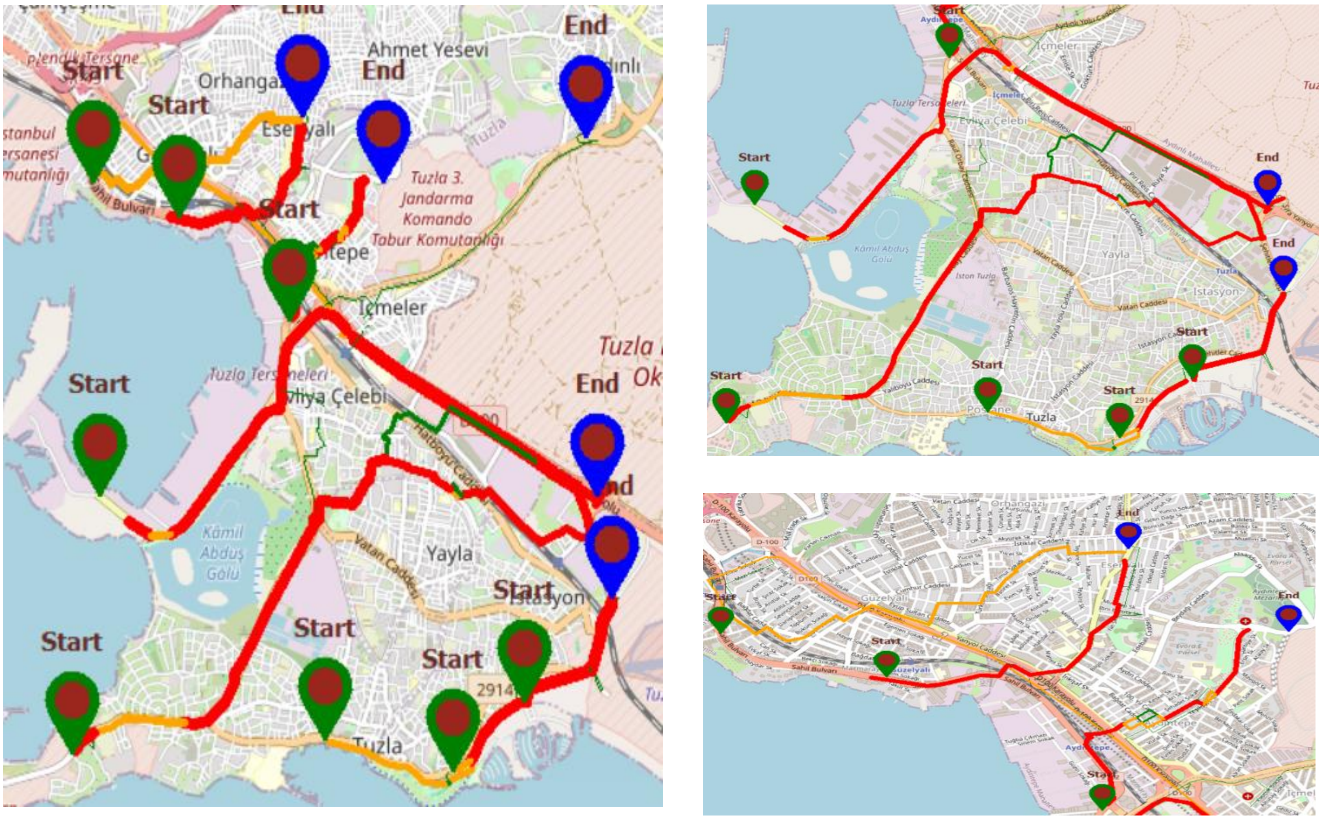


Fig 12

Paths are shown in Figure 12. The objective value was calculated as 258622. While obtaining these results, the travel times were automatically derived based on the lengths of the roads, and the travel times were calculated accordingly.

Test Case Gui İstanbul, Tuzla 2

Compared to the initial Tuzla test case, this study was carried out in a smaller area

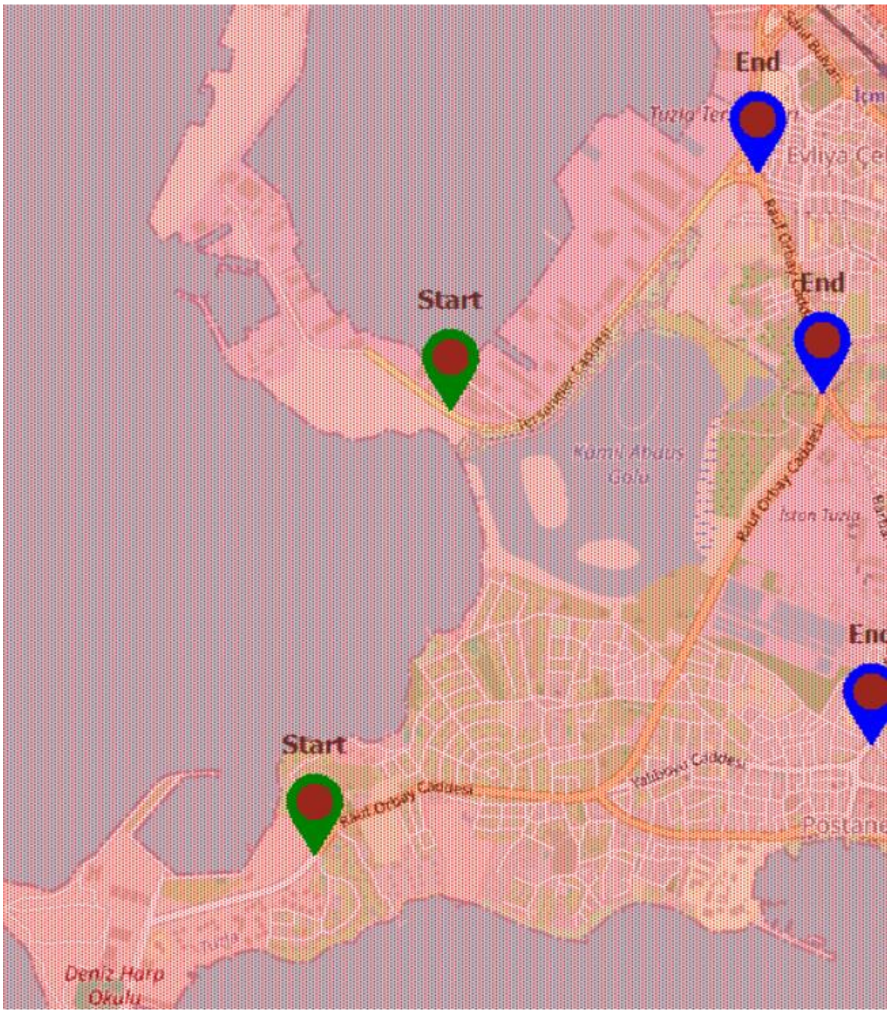


Fig 13

In fig 13, the marked zone and the starting and ending points are shown.

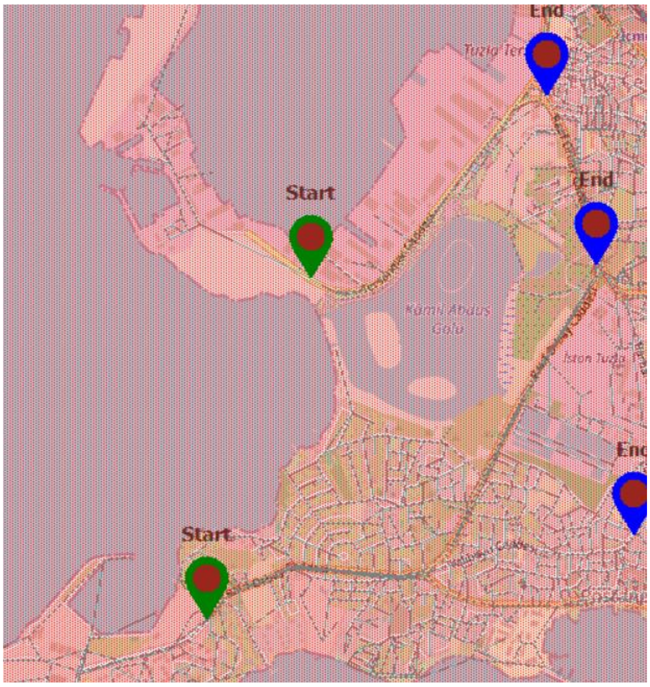


Fig 14

The gray areas represent the roads, with 1677 paths and 4185 links created, as shown in Figure 14.

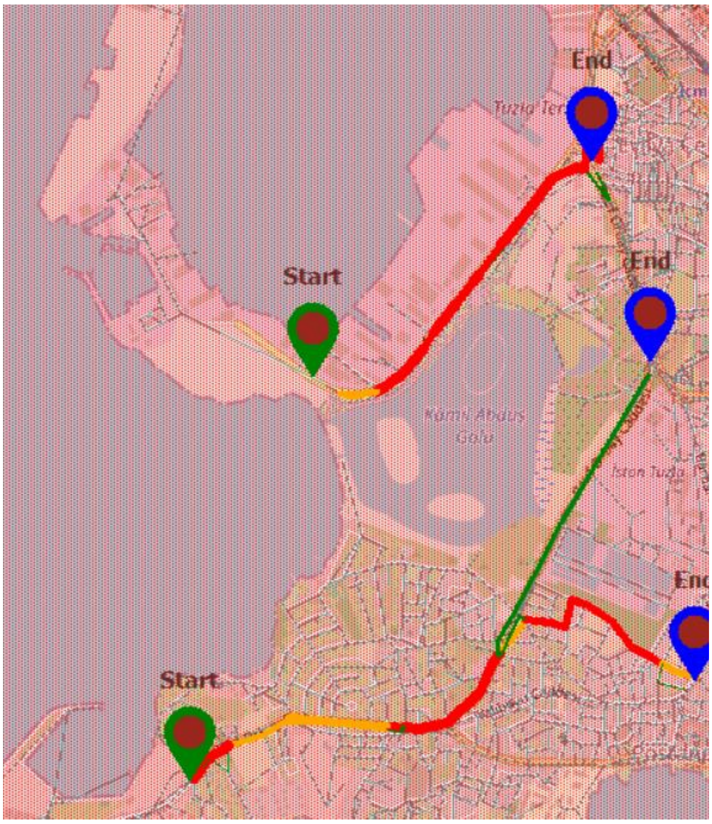


Fig 15

Paths are shown in Figure 15. The objective value was calculated as 258622.

Test Case Gui İstanbul, Kınalı Ada

This test was conducted based on Kınalı Ada, chosen as it is considered an ideal location for the test. In Figure 16, the start and end points are shown.

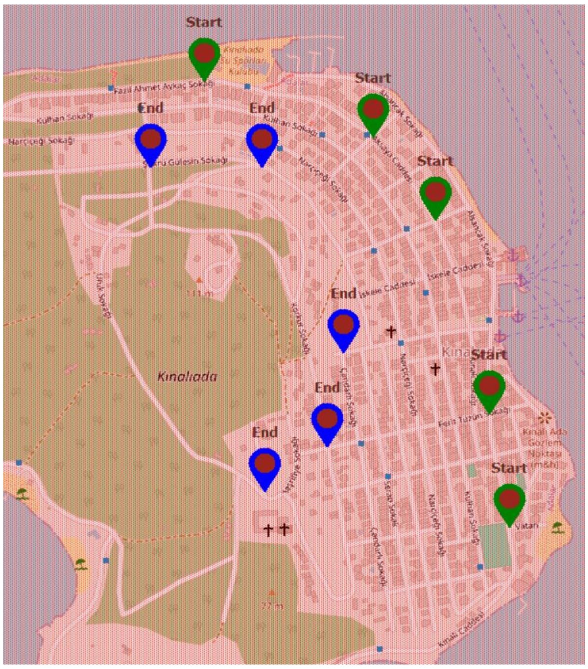


Fig 16

Fig 17

The gray areas represent the roads, with 135 paths and 418 links created, as shown in Figure 17.

Fig 18

Paths are shown in Figure 1. The objective value was calculated as 18604

Test Case Gui Hawaii, Honolulu 1

In this test, the starting and ending points specified as the study area in the article, Hawaii Honolulu, were used. Phase 1 and Phase 2 mentioned in the article were combined and considered as a single phase. The map shown in fig 19.

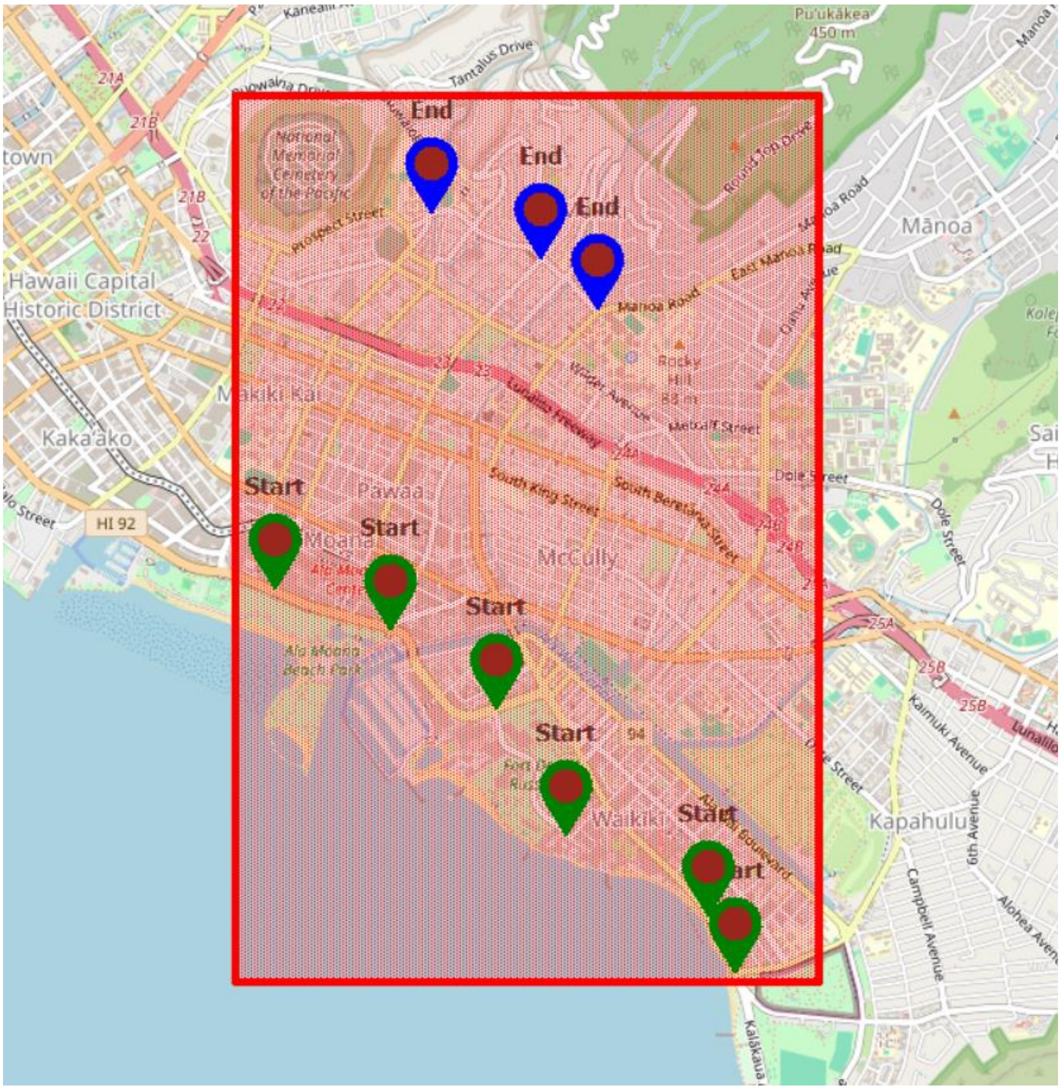


Fig 19

The gray areas represent the roads, with 4871 paths and 11052 links created, as shown in Figure 20.

Fig 20

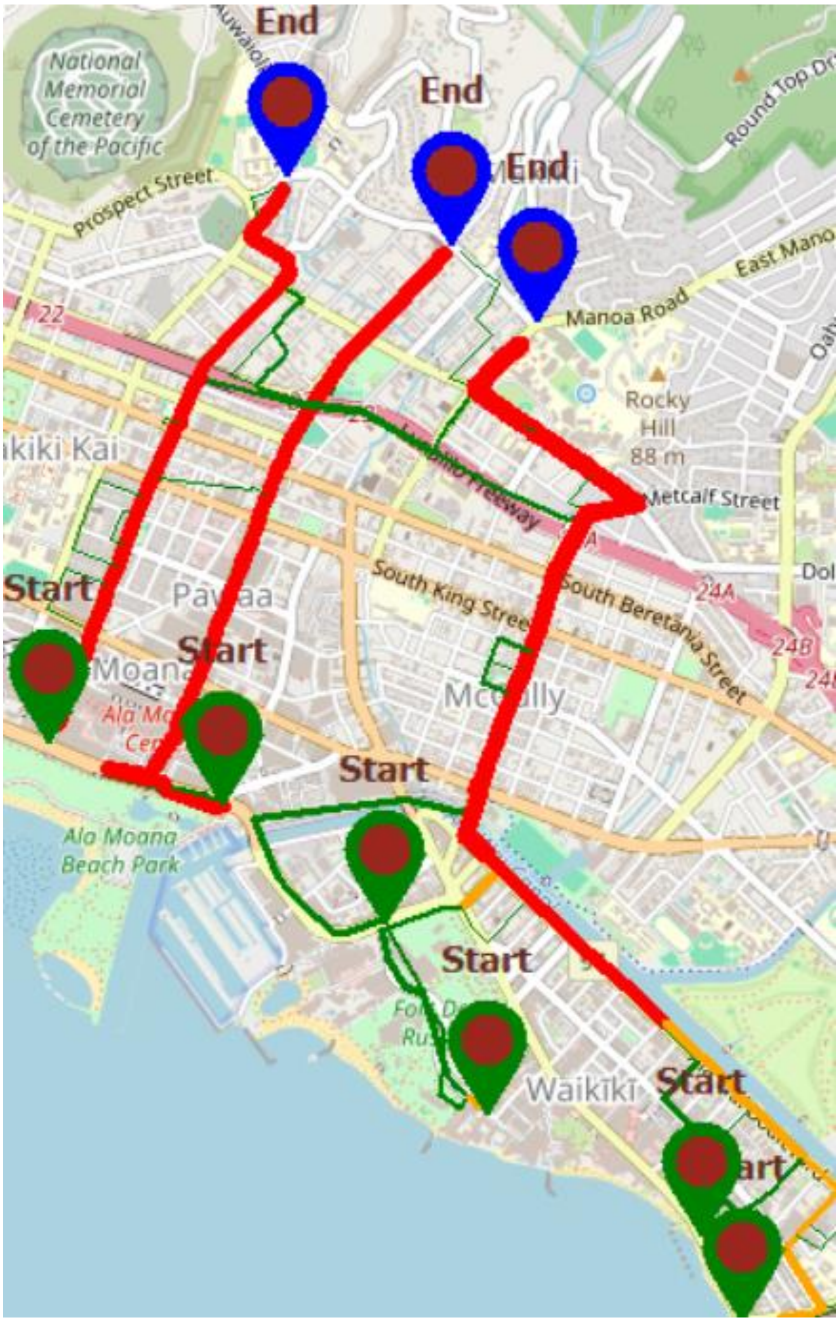


Fig 21

The objective value was calculated as 137970. Paths are shown in Figure 21.

Test Case Gui Hawaii, Honolulu 2

In this test, the Phase 1 temporary shelter entry points within the study area mentioned in the article were considered as the endpoints.

Fig 22

The gray areas represent the roads, with 4871 paths and 11052 links created, as shown in Figure 23.

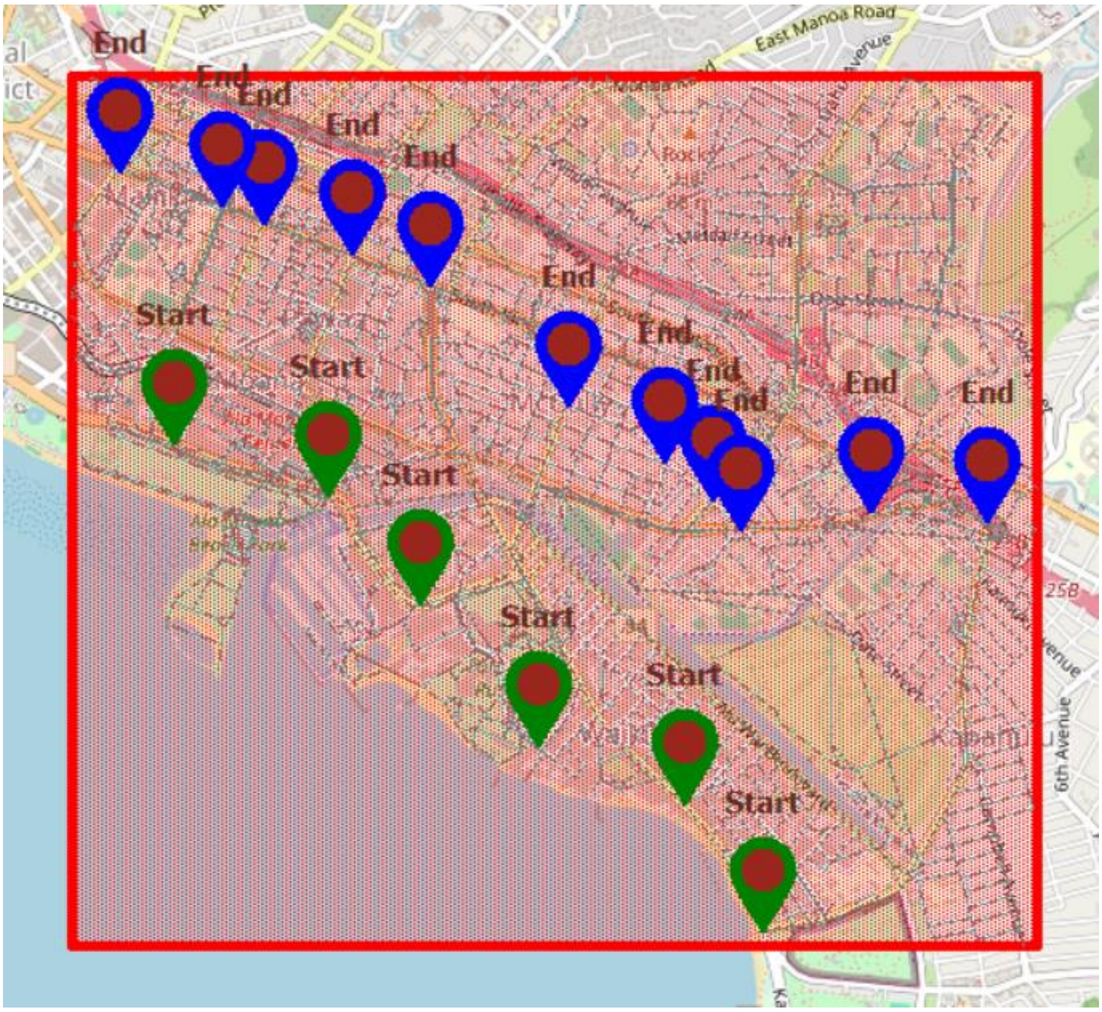


Fig 23

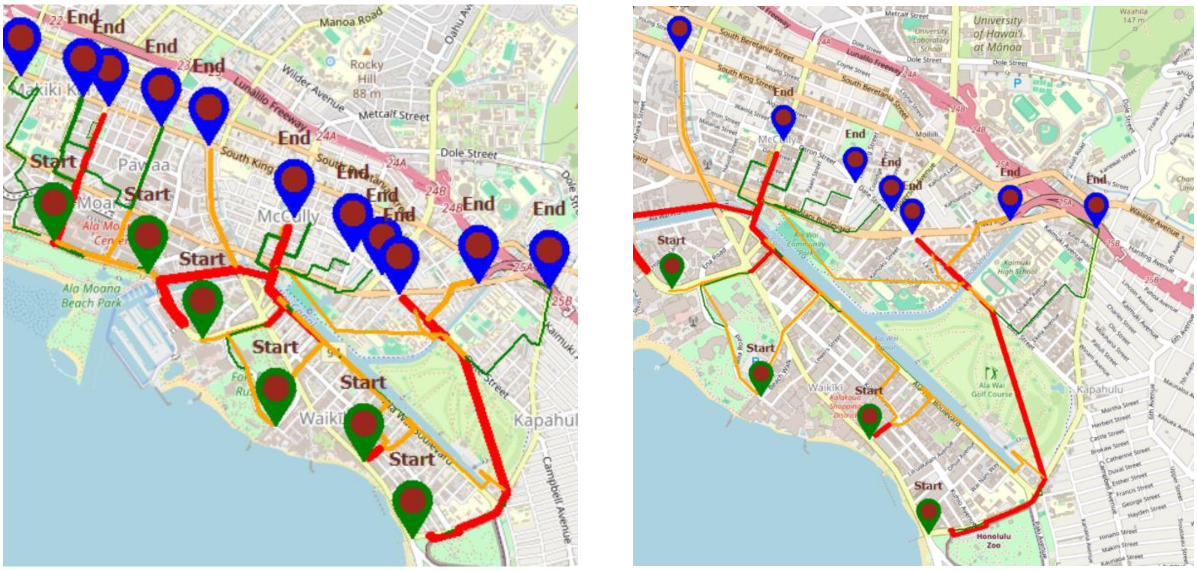


Fig 24

The objective value was calculated as 137970. Paths are shown in Figure 24.

Test Case 1

The first scenario in the article:

The initial passenger count is 16,000.

Stage 1 consists of 48 different paths, while Stage 2 has 20 different paths.

PHASE 1

Model	**Objective Value**	**Total travel time**	**Flow-Weighted Average Path Time**	**Maximum Path Time:**	**iteration**	**Optimization situation**
MM	225.39	6440908	402.57	679.27	70	Optimal
LSO	4806077.68	4806077.68	300.40	411.40	46	Optimal

PHASE 2

Model	**Objective Value**	**Total travel time**	**Flow-Weighted Average Path Time**	**Maximum Path Time:**	**iteration**	**Optimization situation**
MM	433563.40	7537274.17	471.17	1064.11	10000	Feasible
LSO	3386423.06	3386423.06	211.65	247.96	54	Optimal

Test Case 2

The second scenario in the article:

The initial passenger count is 33,000.

Stage 1 consists of 48 different paths, while Stage 2 has 20 different paths.

Phase 1

Model	Objective Value	Total travel time	Flow-Weighted Average Path Time	Maximum Path Time:	iteration	Optimization situation
MM	5533526.99	19108159.32	579.24	858.73	10000	Feasible
LSO	18887059.28	18887059.28	572.33	859.02	62	Optimal

Phase 2

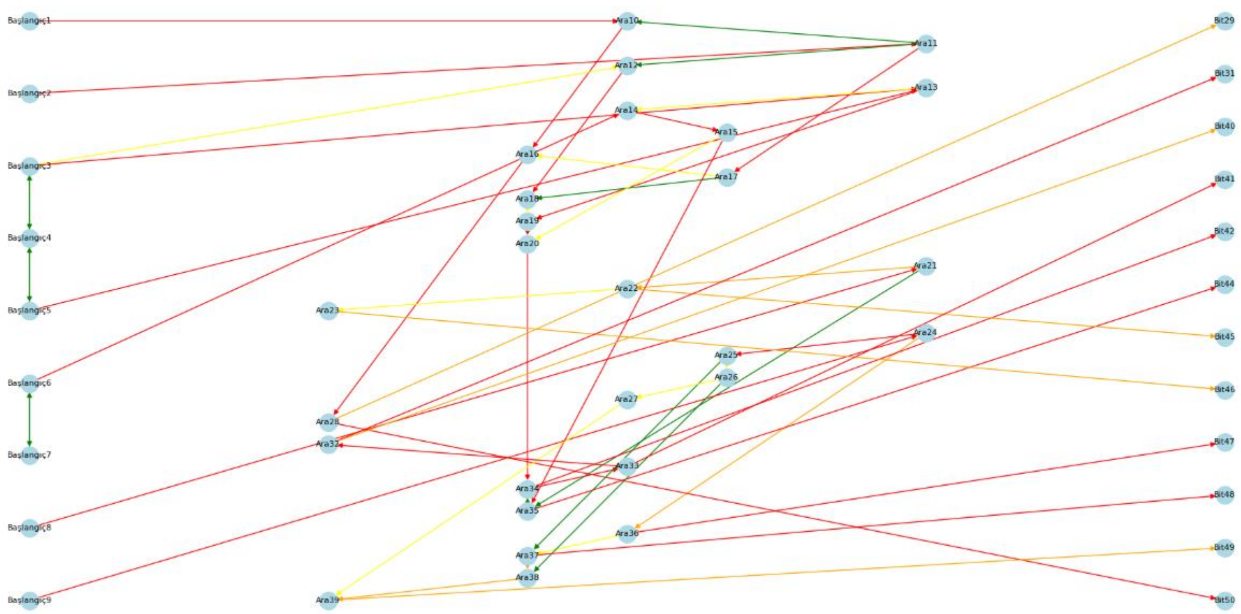
Model	Objective Value	Total travel time	Flow-Weighted Average Path Time	Maximum Path Time:	iteration	Optimization situation
MM	401528.27	17632810.73	539.10	1437.84	10000	Feasible
LSO	18056263.40	18056263.40	547.19	1489.85	73	Optimal

Test Case 3

For Stage 1, an easy path was chosen, whereas for Stage 2, a path that appears easy but has the capacity to handle the full flow—and is likely to encounter many constraints—was selected. While the optimal solution for Stage 1 was reached in 9 iterations, Stage 2 required 33 iterations to find the optimum.

Phase 1

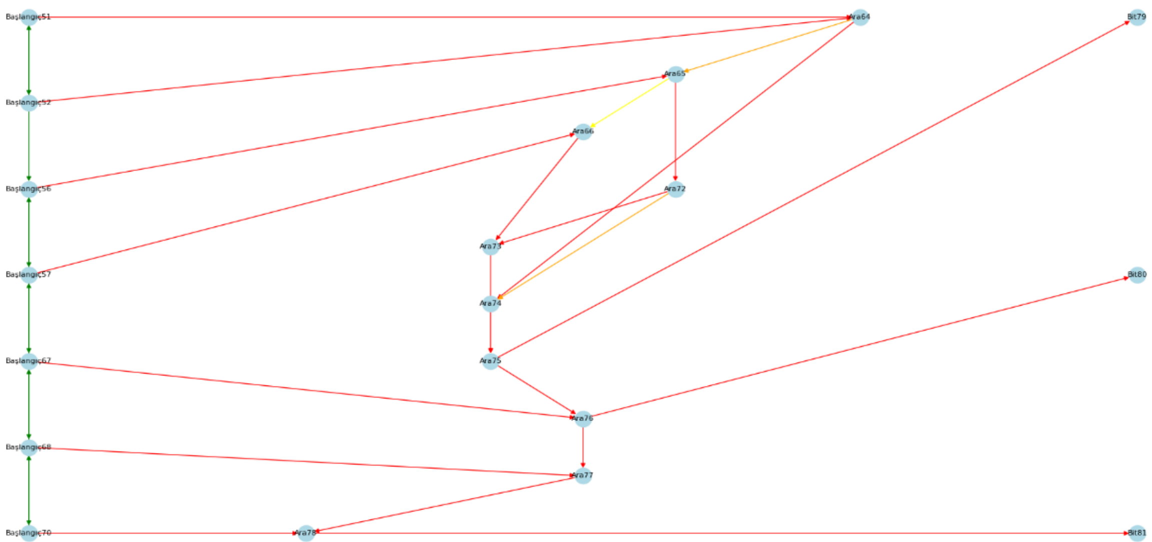
Model	Objective Value	Total travel time	Flow-Weighted Average Path Time	Maximum Path Time:	iteration	Optimization situation
MM	46.83	4563312.09	456.33	529.76	37	Optimal
LSO	18887059.28	18887059.28	572.33	859.02	62	Optimal



LSO result

Phase 2

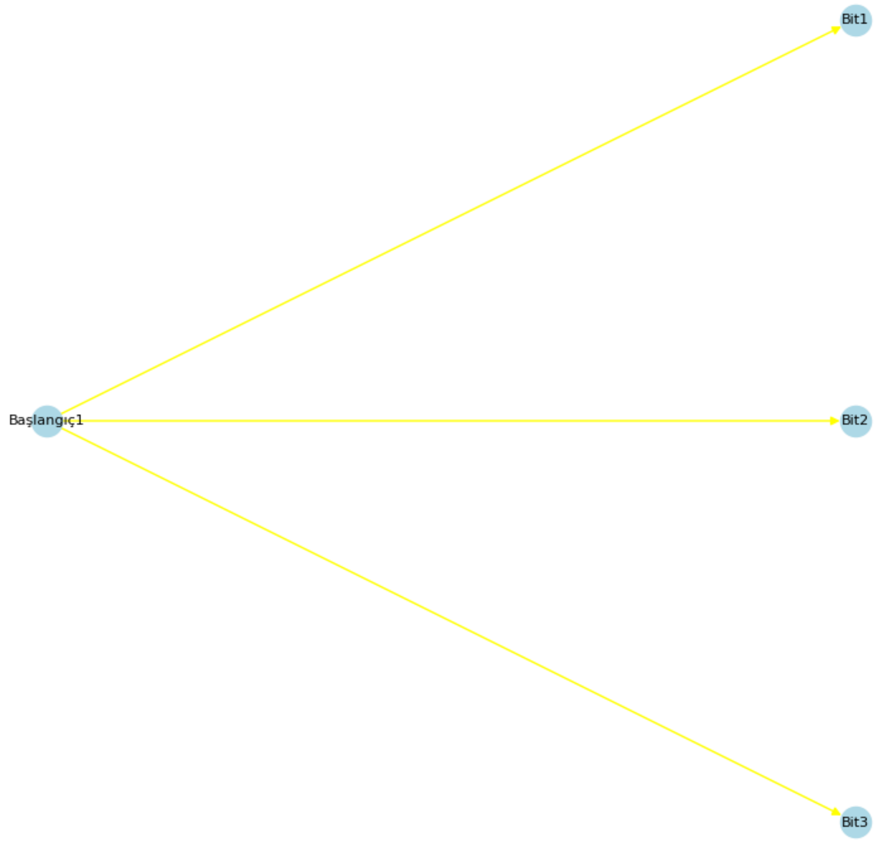
Model	Objective Value	Total travel time	Flow-Weighted Average Path Time	Maximum Path Time:	iteration	Optimization situation
MM	8910531886.26	4715233.33	471.52	607.50	10000	Feasible
LSO	18056263.40	18056263.40	547.05	1489.85	70	Optimal



LSO Result

TEST CASE 4

I defined a model with one starting point and three endpoints, where all travel times and capacities are equal, to demonstrate that the models I use work correctly.



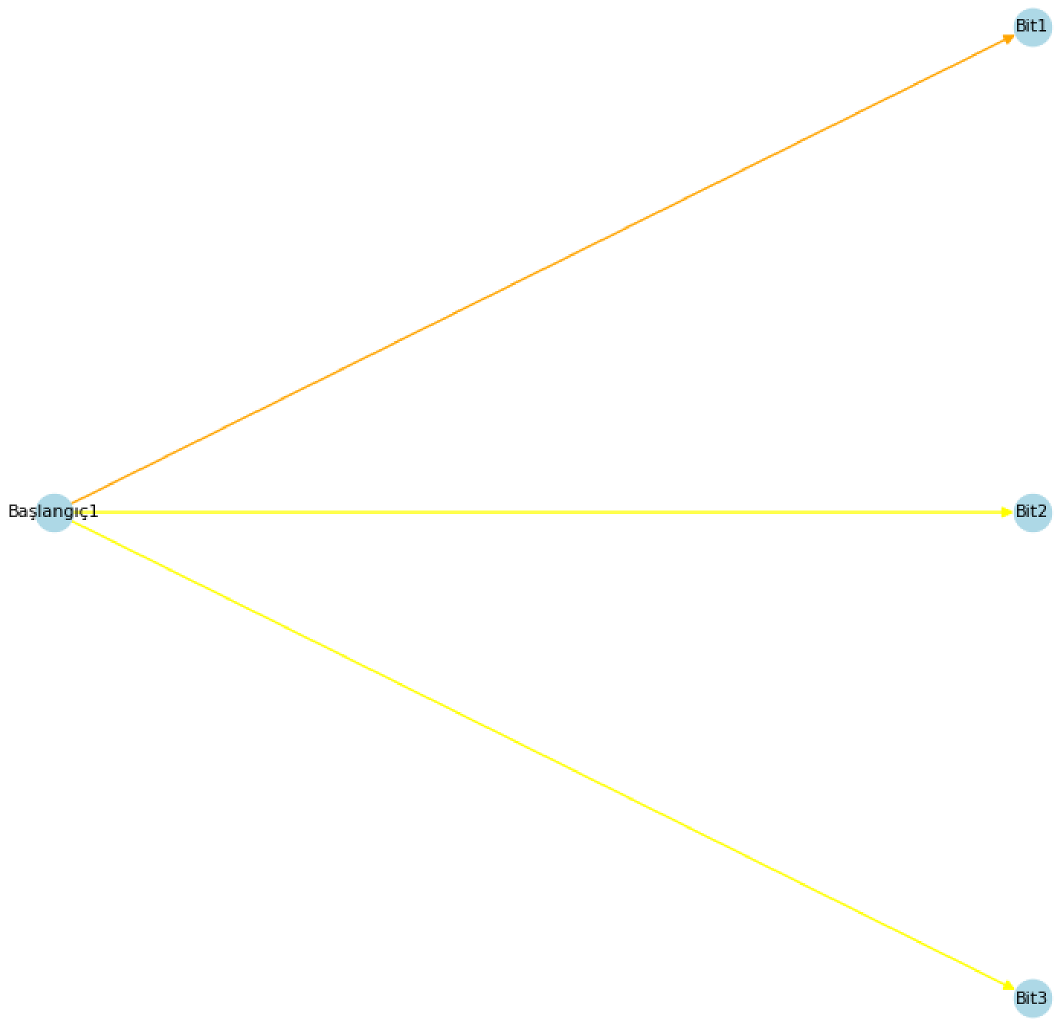
Both models returned identical results for all outcomes.

Edge	Flow	Travel Time	Capacity Usage
Başlangıç1 -> Bit1	500	75.00	50.0
Başlangıç1 -> Bit2	500	75.00	50.0
Başlangıç1 -> Bit3	500	75.00	50.0

Source	Target	Path #	Flow	Time
Başlangıç1	Bit1	1	500	75.00
Path: Başlangıç1 -> Bit1				
Başlangıç1	Bit2	1	500	75.00
Path: Başlangıç1 -> Bit2				
Başlangıç1	Bit3	1	500	75.00
Path: Başlangıç1 -> Bit3				

TEST CASE 5

Same simple model was used, and one edge's travel value was slightly increased.



While MM and LSO found similar values, they did not yield the exact same result.

TEST CASE 6

I defined the number of passengers at the starting point to exceed the total capacity.

Both models returned **infeasible** results, indicating that the constraints are working correctly.

Test Case 7

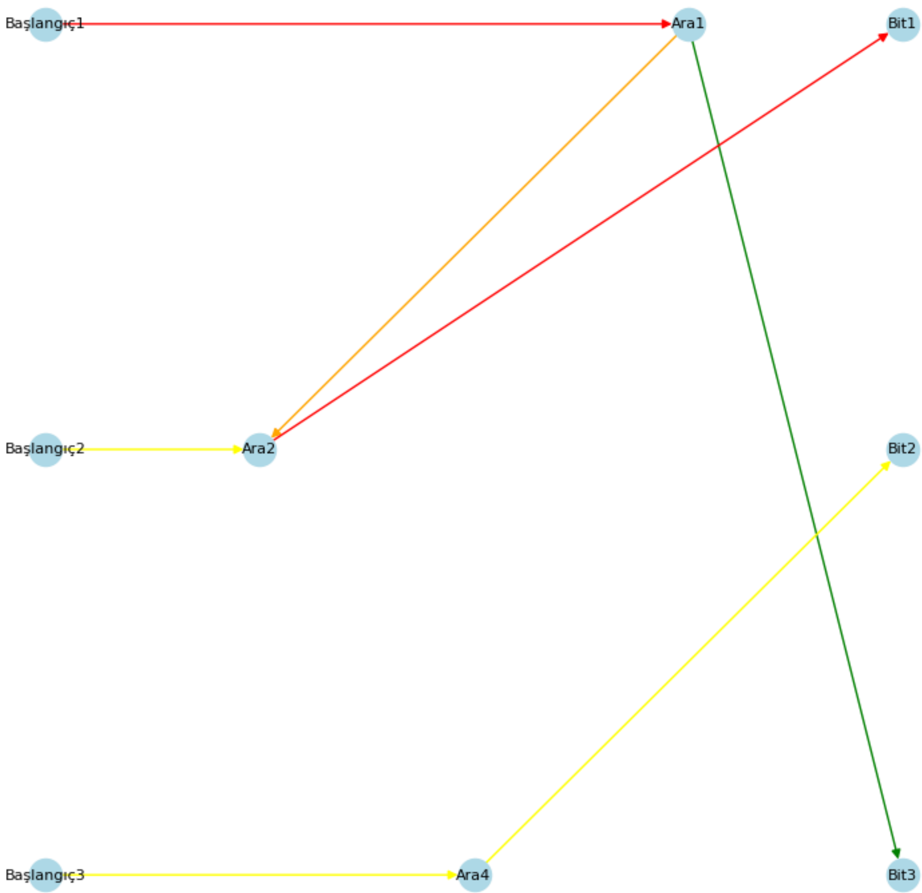
I defined the capacity of one road to be higher than the others while keeping the travel times constant.

```
PATH FLOWS:
--------------------------------------------------------------------------------
Source        Target        Path #      Flow      Time
--------------------------------------------------------------------------------
Başlangıç1    Bit1             1         83      50.69
   Path: Başlangıç1 -> Bit1
Başlangıç1    Bit2             1        833      50.69
   Path: Başlangıç1 -> Bit2
Başlangıç1    Bit3             1         83      50.69
   Path: Başlangıç1 -> Bit3
```

The flow was directed more toward the road with higher capacity, balancing the travel time on average.

Test Case 8

The roads were distributed with highly unbalanced travel times, and it was tested whether the model would still choose the most logical path, even in multi-step routes. While other roads were also utilized, the shorter path was preferred where appropriate.



Test Case 9

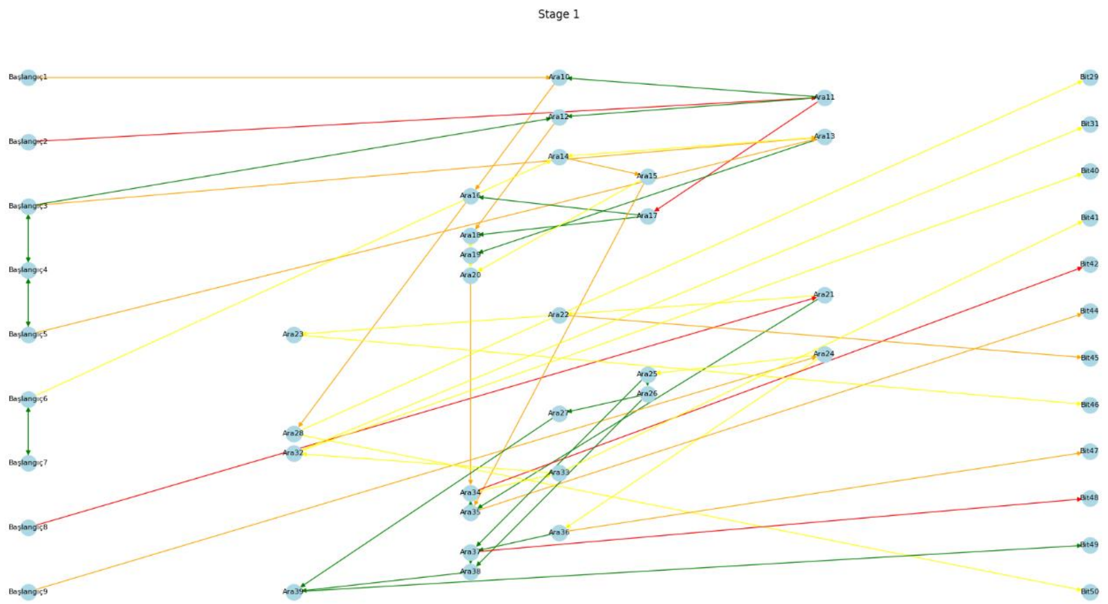
The first scenario in the article:

The initial passenger count is 22,000.

Stage 1 consists of 48 different paths, while Stage 2 has 20 different paths.

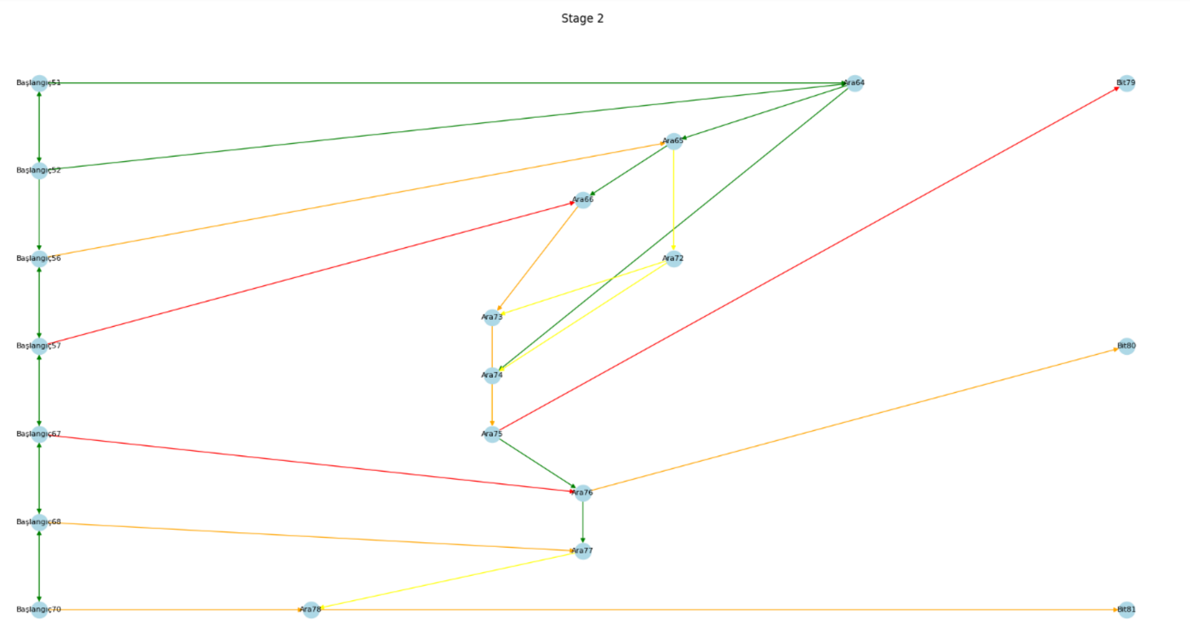
PHASE 1

Model	**Objective Value**	**Total travel time**	**Flow-Weighted Average Path Time**	**Maximum Path Time:**	**iteration**	**Optimization situation**
LSO	7184325.15	7184325.15	338.84	497.41	53	Optimal



PHASE 2

Model	**Objective Value**	**Total travel time**	**Flow-Weighted Average Path Time**	**Maximum Path Time:**	**iteration**	**Optimization situation**
LSO	5200185.25	5200185.25	236.37	339.45	57	Optimal



Github

https://github.com/Tk749/CSE496-Gradution-Project

References

[1] Li, Z., Yu, H., Chen, X., Zhang, G., & Ma, D. (2019). Tsunami-induced traffic evacuation strategy optimization. *Transportation Research Part D: Transport and Environment*, 77, 535-559.

[2] Bayraktar, H. B., & Ozer Sozdinler, C. (2020). Probabilistic tsunami hazard analysis for Tuzla test site using Monte Carlo simulations. *Natural Hazards and Earth System Sciences*, 20, 1741-1764.