PUBLIC TRANSPORT ROUTE SELECTION METHODS WITH RESPECT TO SPECIFIC CRITERIA

ALİCAN BOZYİĞİT

MAY 2017

IZMIR UNIVERSITY OF ECONOMICS
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PUBLIC TRANSPORT ROUTE SELECTION METHODS WITH RESPECT TO SPECIFIC CRITERIA

ALİCAN BOZYİĞİT

MAY 2017

Approval of the Graduate School of School of Natural and Applied Sciences

Assoc. Prof. Dr. Devrim Ünay
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Computer Engineering.

Assoc. Prof. Dr. Cem Evrendilek
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Computer Engineering.

Prof. Dr. Efendi Nasiboğlu
Co-Supervisor

Asst. Prof. Dr. Gazihan Alankuş
Supervisor

Examining Committee Members

Asst. Prof. Dr. Gazihan Alankuş

Asst. Prof. Dr. Kamil Erkan Kabak

Prof. Dr. Efendi Nasiboğlu

Asst. Prof. Dr. Mehmet Türkan

Asst. Prof. Dr. Semih Utku

ABSTRACT

PUBLIC TRANSPORT ROUTE SELECTION METHODS WITH RESPECT TO SPECIFIC CRITERIA

Bozyiğit, Alican
M.Sc. in Computer Engineering
Graduate School of Natural and Applied Sciences
Supervisor: Asst. Prof. Dr. Gazihan Alankuş
Co-Supervisor: Prof. Dr. Efendi Nasiboğlu
May 2017

Public transport applications, which aim to propose the ideal route to end users, have commonly been used by passengers. However, the ideal route for public transport varies depending on the preferences of users. For instance, primary criterion of route selection for a user can be the shortest path; while it may be the least number of transfers for another user on the other hand. In this study, three different route selection criteria are determined. These criteria are named as; "the shortest path", "the least transfer", and "the least stop". Therefore, three different route selection methods, such that "The Modified Shortest Path Route Selection", "The Least Transfer Route Selection", and "The Modified Least Stop Route Selection" are evaluated with respect to our criteria, and the public transport network is modeled accordingly. Furthermore, in this model three cost functions are defined in order to calculate the distance of route, the number of transfers, and the number of stops on the route. The evaluated methods are experimented on a real world public transport network (İzmir, Turkey). The experiment results with regard to each method are examined and compared with each other by using our cost functions. Thus, it is aimed to emphasize each proposed public transport route selection methods' shortcomings and strengths with visualized results.

Keywords: Public transport, route selection methods, route selection criteria, the shortest path, the least transfer, the least stop, graph theory, Dijkstra's Algorithm, Breadth-first Search Algorithm

ÖZ

TOPLU TAŞIMADA SPESİFİK KRİTERLERE GÖRE ROTA SEÇİM METOTLARI

Bozyiğit, Alican
Bilgisayar Mühendisliği, Yüksek Lisans
Fen Bilimleri Enstitüsü
Tez Yöneticisi: Yrd. Doç. Dr. Gazihan Alankuş
İkinci Tez Yöneticisi: Prof. Dr. Efendi Nasiboğlu
Mayıs 2017

Kullanıcıları için ideal rota önermeyi amaçlayan toplu taşıma uygulamaları, yolcular tarafından sıklıkla kullanılmaktadır. Fakat toplu taşıma için ideal rota tanımı kullanıcıların tercihlerine göre değişebilir. Örneğin, bir kullanıcı için rota seçiminde birincil kriter en kısa yol olabilirken, başka bir kullanıcı için öncelik en az aktarma sayısı olabilmektedir. Bu doğrultuda, tez kapsamında üç farklı rota seçim kriteri belirlenmiştir; "en kısa mesafe", "en az aktarma sayısı" ve "en az durak sayısı". Çalışmamızda belirlenmiş olan kriterlere göre üç farklı rota seçim metodu değerlendirilmiştir. Bu metotlar; "En Kısa Yola göre Modifiye Edilmiş Rota Seçim", "En Az Aktarmaya göre Rota Seçim" ve "En Az Durağa göre Modifiye Edilmiş Rota Seçim" olarak isimlendirilmiştir. Bu metotların değerlendirilebilmesi için toplu taşıma ağı çizge olarak modelleniştir. Ayrıca bu modelde rota mesafesi, aktarma sayısı ve rota üzerinde durak sayısı hesaplanabilmesi için üç ayrı maliyet fonksiyonu tanımlanmıştır. Değerlendirilen yöntemler gerçek bir toplu taşıma ağı üzerinde (İzmir, Türkiye) test edilmiştir. Çalışmadaki her bir yönteme ait değerlendirme sonuçları, maliyet fonksiyonlarını kullanarak hesaplanmış ve birbirleriyle karşılaştırılmıştır. Böylelikle bu çalışmada sunulan toplu taşıma rota seçim metotlarının eksiklikleri ve güçlü yönleri açıklanmıştır.

Anahtar Kelimeler: Toplu taşıma, rota seçim metotları, rota seçim kriterleri, en kısa mesafe, en az aktarma sayısı, en az durak sayısı, çizge teorisi, Dijkstra Algoritması, sığ öncelikli arama

Acknowledgments

Firstly, I would like to thank my supervisor, Asst. Prof. Dr. Gazihan ALANKUŞ, for his continuous support, supervision and useful suggestions throughout this study. His guidance helped me in all the time of research and writing of this thesis.

Besides my supervisor, I would like to thank my co-supervisor, Prof. Dr. Efendi Nasiboğlu, for his motivation, insightful comments and encouragement.

Finally, I would like to offer my special thanks to my family for their support, patience and help. It would not have been able to complete this thesis without their support and help.

This study was partially supported under the project number 113E535 by The Scientific and Technological Research Council of Turkey (TUBITAK).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, firstly, there is a general introduction to the thesis in Section 1.1. Then, the motivation of the study is presented in Section 1.2. Lastly, the organization of the thesis is explained in Section 1.3.

1.1 General Introduction

Public transport is preferred by most of the people since it provides various advantages to users. As a result of this observation, numerous internet and mobile applications are developed for related users. These developed applications generally try to propose an ideal route for the user, who wants to go from one location to another in the urban areas. However, the ideal route for the public transportation varies depending on the preferences of the user. This is because the ideal route can be selected based on different criteria by each user. For example, a group of users may state that the primary criterion of the ideal route for them is the shortest path, while another group of users may express that primary criterion of the ideal route is the least number of transfers.

In the thesis, ***"the shortest path"***, ***"the least transfer"*** and ***"the least stop"*** are evaluated as the criteria for an ideal route; because it has been observed that "the shortest path" and "the least transfer" criteria are preferred by most users in a study (it is detailed in Chapter 2) [1]. Additionally, "the least stop" criterion is evaluated in the study with a novel approach. Therefore, in order to propose an ideal route regarding our criteria, three different methods are evaluated in the study. These methods are named as;

I. The Modified Shortest Path Route Selection
II. The Modified Least Stop Route Selection
III. The Least Transfer Route Selection

In the first method, The Modified Shortest Path Route Selection is evaluated by modifying commonly-known Dijkstra's Algorithm [2], in order to minimize number of transfers and repeated walking. In the second method, The Modified Least Stop Route Selection is evaluated with a novel approach. The last method, The Least Transfer Route Selection is evaluated by expanding Li's study [3] to the multi-mode public transport network.

Furthermore, the evaluated methods are tested on a real world transport network (İzmir, Turkey) with an experimental study. The experimental results of these methods are demonstrated in terms of the running times, the route distances, the number of transfers and the number of stops on the routes. By that, it is aimed to compare the evaluated methods, and additionally, to explain the evaluated methods' strengths and shortcomings.

1.2 Motivation

The primary motivation of the study is the fact that the public transport is highly preferred in urban transportation. Furthermore, it is actively used by different types of users in today's world. There are several reasons behind the more intensive use of public transport than other types of transportation (taxicab, hired buses etc.). First of all, most of the people do not have their own vehicles for transportation. These people prefer public transport because of its being more economical compared to other transport types.

Additionally, certain people who have their own vehicles may also prefer public transport. It is because; the public transport is more economical and creates less stress than driving in certain cases. Furthermore, the public transport can be faster than driving in the rush hours. At the same time, the public transport is very important for the protection of the natural life and the environment; since the carbon emission will decrease when larger numbers of people prefer the public transport (mass transit).

Besides these, the public transport is crucial to urban traffic planning. Undoubtedly the biggest and most important transportation problem of today is urban traffic congestions. Bridges, tunnels or alternative roads can be constructed in order to solve this problem; however, these solutions are costly and will take long time. Another less expensive and more practical solution is creating effective and useful public transport networks and encouraging the people to use these networks. By making use of this solution, the number of vehicles used in traffic will decrease and consequently the traffic congestion will diminish.

1.3 Organization of the Thesis

The organization of this thesis is as follows.

- In Chapter 1, the thesis is introduced.
- In Chapter 2, criteria for the route selection in the public transport are discussed. Furthermore, our route selection criteria and evaluated methods with respect to these criteria are explained.
- In Chapter 3, related works in the literature are discussed.
- In Chapter 4, basic definitions of the graph theory, graph types and data structures are introduced with examples in order to facilitate understanding of the public transport network model and methods in the following chapters.
- In Chapter 5, the public transport network is modeled for evaluating and comparing the route selection methods and describing the cost functions of the routes.
- In Chapter 6, the ideal route selection method with respect to the shortest path criterion is evaluated.
- In Chapter 7, the ideal route selection method with respect to the least stop criterion is evaluated.
- In Chapter 8, the ideal route selection method with respect to the least transfer criterion is evaluated.
- In Chapter 9, the experimental results of the route selection methods are detailed.
- In Chapter 10, the thesis is concluded.

Chapter 2

Background

In this chapter, it is aimed to explain the background of the study. Firstly, criteria for the ideal route selection in the public transport are discussed in Section 2.1. Additionally, a study that includes the users' preferences for the route selection criteria is evaluated. Then, “which criteria to propose for the route selection methods” is stated in Section 2.2. Lastly, evaluated methods for our route selection criteria are explained briefly in Section 2.3.

2.1 Criteria for the Route Selection

There are various criteria for the ideal route selection in the public transport network, because the primary criterion for each user may be different. As stated in the Chapter 1, some users may state that the primary criterion for their ideal route is the shortest path; on the other hand, another user may express that primary criterion for their ideal route is the least number of transfers.

The primary criteria that are recalled for the route selection are the shortest path, the least transfer, the minimum price and the minimum traveling time. There are also some uncommon criteria such as; the most comfortable journey, most touristic cruise, and etc. These are important at the primary level for some of the users; however, there are only a few applications that propose selections of routes regarding these criteria. Thus, the applications try to suggest the better route for the users according to generally accepted main criteria.

Nasibov et al. [1] have presented a study that consisted in 81 local residents from various locations in Izmir. In the study, it was aimed to learn participants’ experiences and ideas about route planning applications. Participants were asked to express their own priorities for the route selection in public transport. It was observed that, participants had many preferences for the public transport routes such as; less times of vehicle changes (the least transfer), less time consuming (the minimum travel time) -shortest path or the cheapest (the minimum price). The results of the mentioned study are stated in a table as follows;

Table 2.1: User Preferences with respect to the Route Selection Criteria [1]

User	**The Route Selection Criteria**		
Ages	**Least Transfer**	**Minimum Travel Time-Shortest Path**	**Minimum Price**
15-25	6	6	10
26-40	7	8	7
41-60	10	4	8
61-70	9	2	4
Total	32	20	29

As seen in Table 2.1, users were focused on four criteria in the survey; the least transfer, the minimum travel time - the shortest path and the minimum price. It has been observed that the shortest path and the minimum travel time were perceived as equivalent by the participants. It is obvious that the shortest path is regarded as an important reference for the minimum travel time in the public transport network. However, there are other factors that affect the duration of the travel such as; types of transit lines used on the route, the number of transfers, the traffic on the route, waiting times during the transfers, and etc. Therefore, it cannot be considered as a fact that the shortest path is exactly equivalent to the minimum travel time.

2.2 The Determined Criteria for Public Transport Routes

The primary criteria for the route selection are discussed in the previous section. Among these criteria, the shortest path and the least transfer are evaluated in the study. Furthermore, the least stop criterion for the route selection is proposed in this work, because this criterion offers certain advantages to the passengers. However, the minimum travel time is not evaluated in the thesis because of certain limitations. Additionally, the minimum price is not evaluated since it is highly related to “the least transfer” in most of the public transport networks.

Fifty-two people out of eighty-one preferred the shortest path and the least transfer as primary criteria for the route selection in the survey, as can be seen in Table 2.1. That is to say, about sixty-four percent of users’ primary criteria are the shortest path and the least transfer. Thus, these criteria are evaluated in the study.

Additionally, the least stop criterion is evaluated in our study. The least stop is not the first one to come to the mind as a criterion for the route selection in the public transport. However, this criterion may be useful for most of the users, because route with least number of stops much possibly use the rapid-transit lines such as; subway, metro, bus rapid-transit, ferry (it is explained in Section 7.1). Rapid-transit lines provide passengers many benefits, such as;

- The rapid-transit lines of the public transport network are separated from the traffic [8]. Therefore, the rapid transit lines travel faster than other modes of the public transport (bus transit lines etc.).
- By means of the pre-paid boarding system, the rapid-transit network systems speed up the passenger boarding time [9].
- High-frequency of the rapid-transit line service minimizes waiting times on the station and also minimizes the number of passengers per vehicle [10]. So, it allows passengers to travel in a more comfortable way.

The minimum travel time criterion is not evaluated in the thesis. Determining the optimal route with respect to the minimum travel time is exactly a challenging task because of the limitations. The main limitations are: getting the up to date data of the traffic conditions and locations of the public transport vehicles in uncertain environments. In order to get these data up to date, a comprehensive data network and technology are needed.

In addition to the minimum travel time, the minimum price criterion is not evaluated; since it is directly related to the least transfer (number of transfer) in most public transport networks. Additionally, there is no transfer fee within specified time intervals in some transport networks. For instance, there is no need to pay an extra fee for transfer in ninety minutes in the public transport network of Izmir.

To sum up, the least transfer, the shortest path, and the least stop are chosen as the criteria according to which we will create route selection methods.

2.3 The Evaluated Route Selection Methods

In order to propose ideal routes with respect to the shortest path, the least stop, and the least transfer (our route selection criteria in the study), following methods are evaluated respectively;

I. The Modified Shortest Path Route Selection
II. The Modified Least Stop Route Selection
III. The Least Transfer Route Selection

The Modified Shortest Path Route Selection Method is evaluated in Chapter 6, in order to propose the ideal route regarding "the shortest path". The Modified Least Stop Route Selection Method is evaluated in Chapter 7, in order to propose the ideal route with respect to "the least stop". The Least Transfer Route Selection Method is evaluated in Chapter 8, in order to propose the ideal route regarding "the least transfer". Since the question of ***"Why is an ideal route selection method that can provide all route selection criteria not evaluated?"*** might be addressed, it is because of the fact that, there is no route selection method that provides all of the criteria. This thesis is proved by following three cases. Therefore, three different methods are evaluated with respect to our criteria.

First Case: The shortest path may neither be the least transfer nor the least stop.

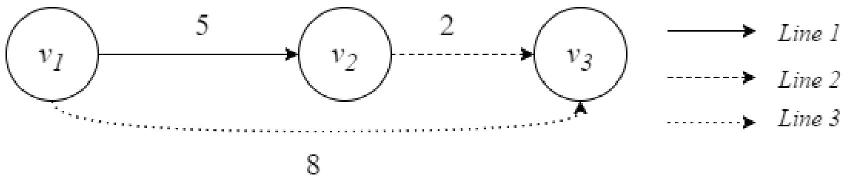

Figure 2.1: Public Transport Network Instance PTN1

In Figure 2.1, there is an instance of the public transport network PTN_1 that includes three transit lines and three stations v_1, v_2, v_3. The edges are labeled with their distances. Assume that, Line 3 is a rapid-transit line, i.e. a subway, and Line1, Line 2 are bus lines. There are two possible paths from the source station v_1 to the target station v_3. One of these paths is $P_1(v_1, v_3) = (v_1, v_2, v_3)$ that uses bus lines. The second one is $P_2(v_1, v_3) = (v_1, v_3)$ that uses the rapid-transit line.

$P_1(v_1, v_3)$ is the shortest path since its distance is less than the distance of $P_2(v_1, v_3)$. However, $P_1(v_1, v_3)$ uses more time for transfers than $P_2(v_1, v_3)$. Additionally, $P_1(v_1, v_3)$ is not the route that has least number of stop.

Second Case: The optimal route with respect to the least transfer may neither be the shortest path nor the least stop route.

In Figure 2.2, there is another instance of the public transport network PTN_2 that includes three transit lines and five stations: v_1, v_2, v_3, v_4, v_5. Assume that, Line 1 and Line 2 are rapid-transit lines. There are two possible paths from the source station v_1 to the target station v_3. The first one is $P_1(v_1, v_3) = (v_1, v_2, v_3)$ that uses rapid-transit lines. The second one is $P_2(v_1, v_3) = (v_1, v_4, v_5, v_3)$ that uses a bus transit line.

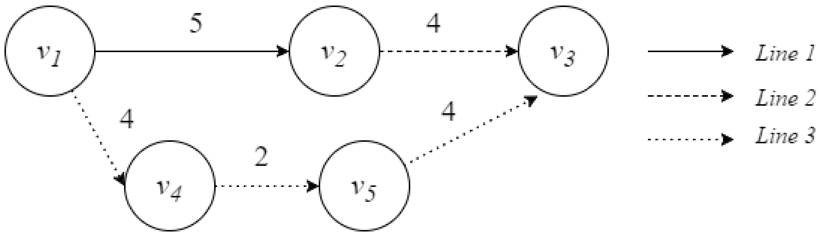

Figure 2.2: Public Transport Network Instance PTN2

Optimal route with respect to the least transfer criterion is $P_2(v_1, v_3)$. However, $P_2(v_1, v_3)$ is not the shortest path since its distance is more than the distance of $P_1(v_1, v_3)$. Additionally, $P_2(v_1, v_3)$ is not the route that has least number of stop.

Third Case: The optimal route with respect to the least stop may neither be the shortest path nor the least transfer route.

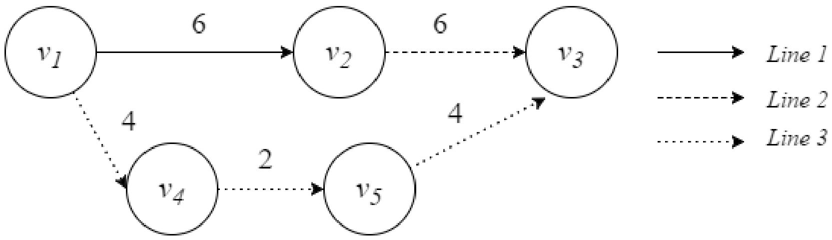

Figure 2.3: Public Transport Network Instance PTN3

In Figure 2.3, there is the last instance of the public transport network PTN_3 that includes three transit lines and five stations v_1, v_2, v_3, v_4, v_5. Assume that, Line 1 and Line

2 are rapid-transit lines. There are two possible paths from the source station v_1 to the target station v_3, such that; $P_1(v_1, v_3) = (v_1, v_2, v_3)$ and $P_2(v_1, v_3) = (v_1, v_4, v_5, v_3)$.

Optimal route with respect to the least stop criterion is $P_1(v_1, v_3)$ that uses rapid-transit lines (Line 1 and Line 2). However, $P_1(v_1, v_3)$ is not the shortest path since its distance is more than the distance of the path $P_2(v_1, v_3)$. Additionally, $P_1(v_1, v_3)$ uses more times of transfers than $P_2(v_1, v_3)$.

To sum up, three different route selection methods (The Modified Shortest Path Route Selection, The Modified Least Stop Route Selection, The Least Transfer Route Selection) are evaluated in order to propose ideal routes regarding to our criteria; because there is no ideal route selection method that can provide all route selection criteria.

Chapter 3

Related Work

In our research; the shortest path and the least transfer are determined as the foremost criteria that come to mind for route selection. Nasibov et al. [1] presented a study, in which the route selection applications were compared. In their study, participants expressed their own priorities for route selection in the public transport. These criteria are named as; the least transfer, the shortest path-the minimum travel time, and the minimum price. Additionally, travelers' needs for public transport were presented by Huang et al. [4]. The least transfer, least travel time-distance, and shortest walking path are stated as primary criteria for public transport route selection in their study. Furthermore, Meng et. al. [5] stated that the minimum number of transfers and the shortest path-the shortest travel time are very important for route guidance systems and so these systems should provide these options for related users. Moreover, Pun-Cheng and Chan [6] indicated that the least transfer is mostly preferred optimal route criterion by passengers. Nasiboğlu and Berberler [7] presented a study that the least transfer criterion is very important for public transportation planning and for passengers using public transportation systems. Thus, the least transfer and the shortest path are determined as primary route selection criteria to be evaluated in our study.

On the other hand, there are many studies that present the importance of rapid-transit lines for route selection in public transport. The benefits of the urban underground rapid-transit line were presented by Girnau and Blennemann [8]. Main advantages of light rapid-transit were presented by Luke and MacDonald [9], and the efficiency of metro systems were presented by Laporte and Mesa [10]. Thus, the least number of stops is determined as a route selection criterion, because a route with least number of stops much possibly use the rapid-transit lines (it is explained in Section 7.1) and this criterion has other advantages for users e.g. less boarding time.

Dijkstra's Algorithm [2] has mostly been used as the method to find the shortest path. However, a penalty system is implemented to Dijkstra's Algorithm in order to minimize number of transfers and walking distances in route selection, in our study. There are also relevant studies that modify the Dijkstra's Algorithm for public transport route selection. Wang et al. [11] stated that each transfer in a public route selection requires extra cost in their study. Therefore, they have designed public transport system as a new data model, and proposed a new shortest path algorithm in their study. Weaknesses of the shortest path algorithms (ignoring number of transfers) were analyzed and an improved Dijkstra's Algorithm was proposed by Xu et al. [12]. Xiaoyong and Xueqin [13] presented a heuristic algorithm that considers both transfer criterion and distance criterion for route selection. Jian-lin [14] explained that Dijkstra's Algorithm is not appropriate for route selection and proposed a new algorithm based on the least transfer. Wu and Hartley [15] presented a study that uses K-Shortest Paths Algorithm and they proposed the optimal path among the k-Shortest Paths by taking the user preferences into consideration. Ferreira et al. [16] proposed a new advisor system based on the integration of various data sources and they used a Dijkstra's Algorithm implementation in this system. Goel et al. [17] presented a variant of Dijkstra's Algorithm by precomputing transfer patterns between hub nodes in order to be used for a multi-mode (i.e., bus, train and walk) transport network of Mumbai city, India. Nguyen and MacDonald [18] presented a new data model "Exploded Graph" and they used Dijkstra's Algorithm in their new model to propose the path with the least number of transfers. Biswas [19] introduced a new fuzzy condition factor in their network graph and modifies Dijkstra's Algorithm in this way. Zhou et. al. [20] presented an iterative optimization method regarding individual thinking of bus travelers. In contrast, our approach is based on minimizing number of transfers and walking distance by slightly increasing distance of the path proposed by Dijkstra Algorithm. Thus, our modified algorithm strives to find an ideal path while keeping the path length short.

In our study, The Least Transfer Route Selection Method has been evaluated by extending a study that was presented by Li and Zhu [3]. Li and Zhu presented a study that bus transport network was modeled in space P, and then, the least transfer route was determined by using Breadth First Search algorithm [21]. In the topology of real-life infrastructure, two stations are adjacent if there is no station between them. In the space P, two stations are considered to be adjacent if there is at least one transit line that stops at these stations [22], [23]. Space P is mainly used in studies that analyses topologies of transport networks [24], [25]. In addition to Li and Zhu, Wang and Yang [26] also presented a study that uses space P for the public transport network route selection. However, Wang and Yang used matrix multiplication for each transfer that increases time complexity of the algorithm dramatically, in their study.

As distinct from studies using space P, Gao [27] presented a study that introduces novel network representation including layers for each route in order to propose the least transfer route selection by a matrix multiplication. Furthermore, Li et al. [28] classified public transport network into topologies based on path-stop network; and then proposed a new least transfer travel route model. However, using Breadth First Search Algorithm in space P, is the more efficient way for the least transfer route selection in terms of the running time.

Lastly, Breadth First Search algorithm is used for finding routes with least number of stops, in our study. Breadth First Search was developed by Lee [21], in order to find the connections on the paths. This algorithm was used in order to find the shortest path from the source to the target in public transport by Böhmová et al. [29].

Chapter 4

Graph Preliminaries

In this chapter, there is an introduction to the graph theory, which is used in the public transport network models, in this study. Firstly, the graph theory is explained briefly in Section 4.1. In the following sections, basic definitions of the graph theory, graph types and data structures are described with examples in order to facilitate understanding of the public transport network model and route selection methods in the following chapters.

4.1 The Origin of the Graph Theory

Graph Theory is simply modeling a problem with edges and vertices in order to represent this problem as a graph. Some features defined in the Graph Theory are used for solving this model in order to solve the real world problem. To define it simply; the real world problem is first modeled as a graph, and afterwards this model is solved, and the solution is applied to the real world problem [30].

"The Seven Bridges of Konigsberg" is a famous mathematical problem which constitutes the framework of Graph Theory [30]. It was inspired from the bridges in Konigsberg in the 18th Century. In the city of Konigsberg in Prussia, there are the Old Pregel and the New Pregel rivers. These rivers divide the city into four zones. Two of these zones are large islands. There are seven bridges connecting these zones. The problem was; *"Can anyone who wants walking around the town, turn to the point where he starts by crossing all the bridges once?"*. Figure 4.1 shows the visualized version of the problem.

Over time, the problem was brought to Leonard Euler, who was one of the famous mathematicians of the time. In order to simplify the problem a little bit more and to eliminate unnecessary components, Euler formed a new graph as shown in Figure 4.2. In the figure, land pieces are shown as vertices and bridges are shown as edges (connecting these vertices).

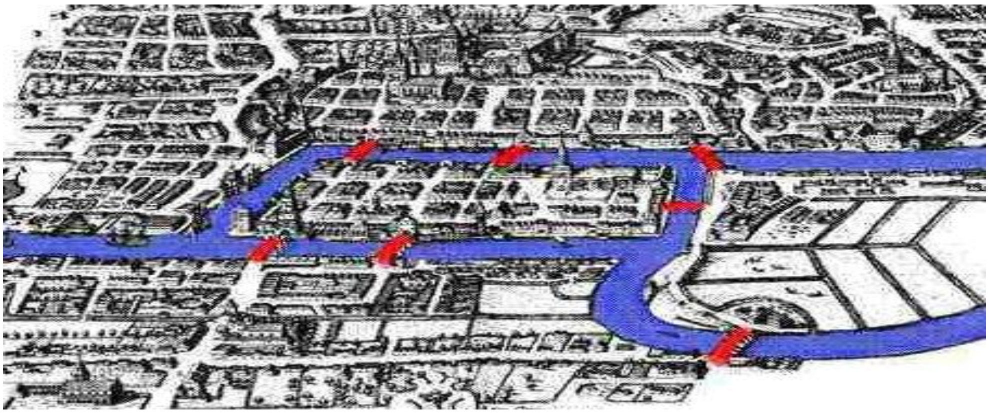

Figure 4.1: The Seven Bridges of Konigsberg [31]

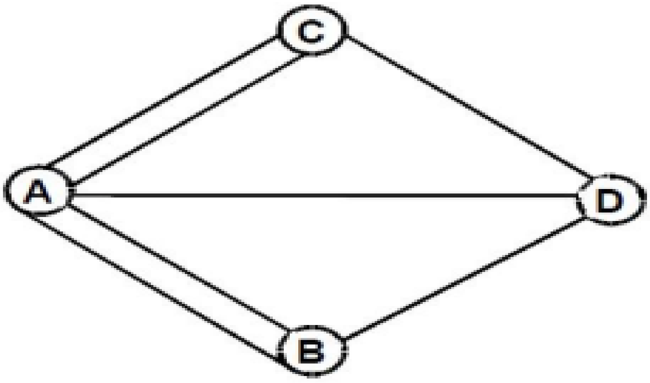

Figure 4.2: The Graph Representation of the Seven Bridges of Konigsberg

While Euler was struggling with the problem, he discovered the *Euler Path Theorem*. His claim was based on a simple discovery; “If you come to a vertex with an edge, you need a different edge to leave this vertex”. Therefore, if the degree of a vertex (number of edges incident to the vertex) is an odd number, it must be either a start vertex or an end vertex. Euler solved the problem and he proved that such a path cannot be found by using the theorem. Note that all degrees of vertices are odd numbers in Figure 4.2. The rule of traversing each vertex once leads to a contradiction. It is because of the fact that maximum two vertices may be the end points (start or end vertex).

Euler created a new field of mathematics while he was dealing with solving this problem. This new field, later known as Graph Theory, is now the study area of various disciplines. It is used in different fields ranging from sociology to computer science.

4.2 Basic Definitions

The graph is a kind of network structure that consists of vertices and edges which connect these vertices. It is stated by $G = (V, E)$. This definition states that a graph is a cluster of vertices and edges. The graph is considered as an ordered pair in most sources. Therefore, the vertex set is shown first and it is followed by the edge set.

The vertex set V is a set that contains all vertices of the graph. The **edge set** E is a set that contains all edges (connections between vertices) of the graph. Edges are represented as unordered pairs $\{u, v\}$ or ordered pairs (u, v) depending on the graph type where u and v are vertices in V.

The **order of the graph** is the number of elements in the vertex set. It is denoted by usually $|G|$ or $|V|$, sometimes m. The **size of the graph** is the number of elements in the edge set. It is usually denoted by $|E|$, sometimes n.

Vertices u and v are called **adjacent vertices** if these vertices are connected by some edge $e \in E$ where $u, v \in V$.

In a graph $G = (V, E)$, the **neighborhood of the vertex** v is a set of vertices that adjacent to v where $v \in V$. It is stated as follows;

$$N(v) = \{u | (u, v) \in E\}$$

In a graph $G = (V, E)$ the **degree of the vertex** v is the size of its neighborhood where $v \in V$. It is stated as follows;

$$d(v) = |N(v)|$$

Undirected Graph: It is a type of graph in which edges have no direction. There is an instance of undirected graph $G = (V, E)$ in Figure 4.3. The vertex set of the graph is $V = \{v_1, v_2, v_3, v_4, v_5\}$. The edge set of the graph is $E = \{e_1, e_2, e_3, e_4, e_5\}$ where edges are undirected and unordered pairs where $e_1 = \{v_1, v_2\}, e_2 = \{v_1, v_4\}, e_3 = \{v_4, v_5\}$, $e_4 = \{v_1, v_3\}, e_5 = \{v_3, v_5\}$.

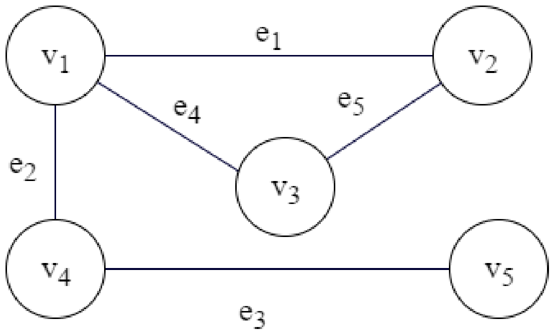

Figure 4.3: An Instance of Undirected Graph

In Figure 4.3, the order of the given graph is $|V| = 5$ and the size of the graph is $|E| = 5$. For instance, the neighborhood of the vertex v_3 is $N(v_3) = \{v_1, v_2\}$ and the order of this vertex is $d(v_3) = |N(v_3)| = 2$.

Directed Graph: It is another type of graph, in which the edges have directions. The directed graph's edges are ordered pairs. These directed edges are called as arcs sometimes. The directed graph is usually denoted by $G = (V, E)$ where V is the vertex set of the graph and E is the directed edge set of the graph.

If the vertex u and the vertex v are joined by a directed edge $e = (u, v)$, then it states that v is reachable from u in a graph $G = (V, E)$ where $u, v \in V$ and $e \in E$. Remark that the edge (u, v) has not same meaning as the edge (v, u). Because the directed edge (u, v) states that v is reachable from u but vice versa is not meant.

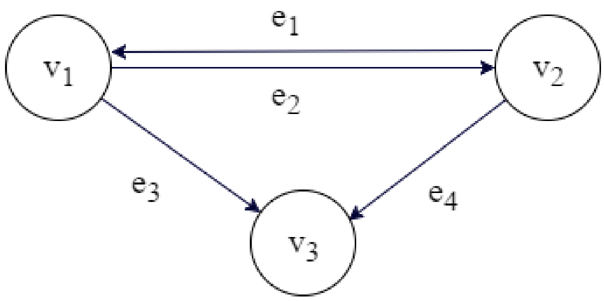

Figure 4.4: An Instance of Directed Graph

For example, there is a directed graph $G = (V, E)$ in Figure 4.4. All given definitions and notations are same for the directed graph except the notation of edges. The edge set of the graph is $E = \{e_1, e_2, e_3, e_4\}$ where edges are directed and ordered pairs as follows;

$$e_1 = (v_2, v_1),\ e_2 = (v_1, v_2),\ e_3 = (v_1, v_3),\ e_4 = (v_2, v_3)$$

Weighted Graph: A graph in which each edge has a numerical weight is called the weighted graph. Usually, the edge weights are non-negative numeric values. The weight of an edge (v_i, v_j) is denoted by $w_{i,j}$ where $v_i, v_j \in V$ and $(v_i, v_j) \in E$ in a weighted graph $G = (V, E)$.

4.3 Graph Representations

Graphs are represented in many forms in computer science. Most commonly used representations of the graphs are the adjacency matrix and the adjacency list. Both of these data structures are used in our methods and algorithms.

Adjacency Matrix: It is a two-dimensional array of size $|V| \times |V|$ where $|V|$ is the order of the graph G where $G = (V, E)$. The adjacency matrix is a representation of the graph as follows;

$$A_{nxn} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{bmatrix} \text{ where } a_{i,j} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \\ 0, & \text{else} \end{cases} \text{, where } v_i, v_j \in V$$

It is determined whether two vertices are adjacent in constant time $O(1)$ by using the adjacency matrix. Checking the value of the corresponding cell in the adjacency matrix is sufficient for this transaction. For example, it is determined whether vertices v_i and v_j are adjacent by checking the value $a_{i,j}$.

The first disadvantage of the adjacency matrix is taking $O(|V|^2)$ space even though a graph is sparse which contains a few number of edges. Secondly, we need to check all $|V|$ entries in i_{th} row of the adjacency matrix to determine which vertices are adjacent to the vertex v_i. Even if $d(v_i)$ is much less than $|V|$, the time complexity of this transaction still is $O(|V|)$.

The weighted adjacency matrix W is used for getting the weights of edges in the graph. The weights of the edges are usually positive numeric values. If two vertices are adjacent then $w_{i,j}$ is the weight of the related edge (v_i, v_j). If vertices v_i and v_j are not adjacent then their weight is represented by 0.

In Figure 4.5, there is an example of weighted directed graph $G = (V, E)$ as follows;

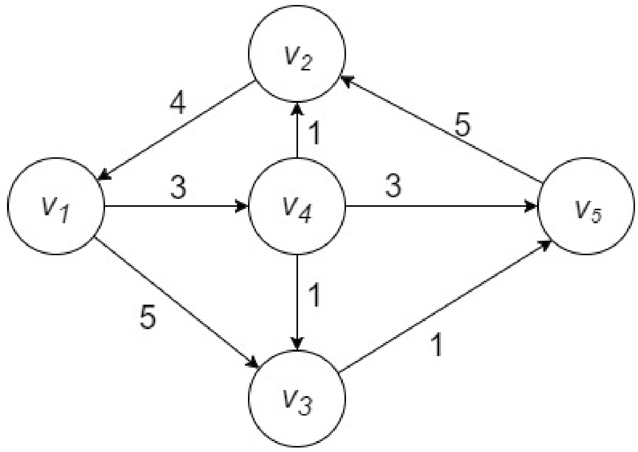

Figure 4.5: An Instance of Weighted Directed Graph

The adjacency matrix and the weighted adjacency matrix of the graph in Figure 4.5 as follows;

$$A_{nxn} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, W_{nxn} = \begin{bmatrix} 0 & 0 & 5 & 3 & 0 \\ 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Adjacency List: It is $|V|$ sized array of lists that represents $G = (V, E)$. The array's i_{th} index is the list of vertices that adjacent to vertex $v_i \in V$.

The adjacency list uses less space than the adjacency matrix. The space complexity of the adjacency list is $O(|V| + |E|)$. Additionally, determining which vertices are adjacent to a vertex v_i takes constant time $O(1)$. Getting the list of the array's i_{th} index is sufficient for this transaction.

However, the adjacency list is not efficient in terms of performance to determine whether vertex v_i and v_j are adjacent because we need to check all entries in the array's i_{th} index. The time complexity of this transaction is $O(|V|)$ where $d(v_i) = |V|$ in the worst case.

Weights can be stored in the adjacency list for weighted graphs. Two sized arrays are used in the list. The first value of the array gives the adjacent of the vertex and the second value gives the weight of the related edge.

For instance, the adjacency list of the graph G in Figure 4.5 as follows;

$$L = \begin{cases} \{v_{4,}\ v_3\} \\ \{v_1\} \\ \{v_5\} \\ \{v_2, v_3, v_5\} \\ \emptyset \end{cases}$$

The adjacency list with weights of the graph in Figure 4.5 as follows;

$$L = \begin{cases} \{\{v_4, 3\}, \{v_3, 5\}\} \\ \{\{v_1, 4\}\} \\ \{\{v_5, 1\}\} \\ \{\{v_2, 1\}, \{v_3, 1\}, \{v_5, 3\}\} \\ \emptyset \end{cases}$$

To sum up, the adjacency list and the adjacency matrix are data structures that are used to represent graphs. They are used by various graph algorithms in computer science. Both of them have advantages and disadvantages in some cases. Let graph be $G = (V, E)$, the comparison of these data structures with respect to criteria in the worst case as given in Table 4.1;

Table 4.1: Comparison between Adjacency Matrix and Adjacency List

	Adjacency Matrix	**Adjacency List**
Space complexity of the data structure	$O(\|V\|^2)$	$O(\|V\| + \|E\|)$
Query whether there is an edge from u to v and cost of the edge, where $u, v \in V$	$O(1)$	$O(\|V\|)$
Query the neighborhood of the vertex v, where $v \in V$	$O(\|V\|)$	$O(1)$

4.4 Path and Its Distance

Path: It is a sequence of vertices which are connected by the edges. The path is usually denoted by P.

In unweighted graphs, the **weight (length) of a path** is the number of edges that connects vertices in this path. In a weighted graph, the weight of a path is the sum of the weights of the edges in this path. It is denoted by $w(P)$, where P is a path.

$$w(P(v_1, v_2, \dots, v_n)) = \begin{cases} w_{1,2} + w_{2,3} + \cdots + w_{n-1,n} & \text{if G is a weighted graph} \\ n & \text{if G is an unweighted graph} \end{cases}$$

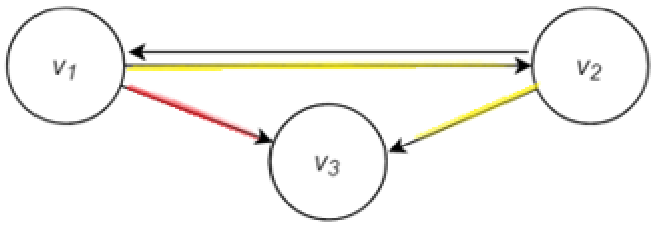

Figure 4.6: An Instance of Directed Graph

For example, there are two paths from vertex v_1 to v_3 in the graph shown in Figure 4.6 such that; $P_1(v_1, v_3) = (v_1, v_2, v_3)$ and $P_2(v_1, v_3) = (v_1, v_3)$. Since the graph is unweighted, the weights of paths are as follow;

$$w(P_1(v_1, v_3)) = |P_1(v_1, v_3)| = 2$$
$$w\big(P_2(v_1, v_3)\big) = |P_2(v_1, v_3)| = 1$$

Distance: From the source vertex u to the target vertex v, a path that its weight is less than other possible paths' weights is called the **shortest path** in the graph G where $G = (V, E)$ and u, $v \in V$. The distance between two vertices u and v is the number of edges in the shortest path for an unweighted graph $G = (V, E)$ where $u, v \in V$. Between two vertices u and v, the distance is the sum of the weight of edges in the shortest path from u to v for a weighted graph $G = (V, E)$ where $u, v \in V$. The distance is denoted by $dist_G(u, v)$.

For example, the graph in Figure 4.6 has two paths $P_1 = (v_1, v_2, v_3)$ and $P_2 = (v_1, v_3)$ from the vertex v_1 to the vertex v_3. The distance between these two vertices is $dist_G(v_1, v_2) = w(P_2) = 1$. Since P_2 is the shortest path from v_1 to v_3.

Chapter 5

Modeling the Public Transport Network

In this chapter, before the explanation of the proposed route selection methods, a basic graph representation of the public transport network is modeled. This model is then used for evaluating the proposed route selection methods. Furthermore, the cost functions are defined in this model which calculate the cost of the routes (the distance of the route, the number of transfers, and the number of stops on the route) proposed by the evaluated methods. These functions are used for comparing route selections of the evaluated methods in the experimental study.

5.1 The Public Transport Network Model

The public transport network is modeled as a directed graph in this thesis. Public transport network stations are represented as vertices and the connections between these stations are represented as edges in the model. Two data structures are used in order to represent this model. The first one is the transit line matrix that is used for finding transit lines, which traverse between the adjacent vertices. The second one is the adjacency matrix that is used for finding the distance between adjacent vertices. Lastly, public transport network routes are represented as paths in this model.

For instance, there is a small public transport network presented in Figure 5.1. Transit lines (bus and metro lines) are illustrated by straight solid lines. Walking paths are illustrated by straight dashed lines. Additionally, the distances between the stations are labeled on the lines. There are seven stations of A, B, C, D, E, F, G, in which A, B, C, D, F are bus stations and G, H are metro stations. There are five transit lines of 1,2,3,4,5, in which 1,2,3,4 are bus lines and 5 is a metro line.

The public transport network in Figure 5.1 is modeled as a directed graph $G = (V, E)$. The vertex set V represents all of the transport network's stations such as; bus stops, subway stations, ferry terminals, and etc. Simply, each vertex $v \in V$ refers to one of the public transport network's stations (see Figure 5.2.).

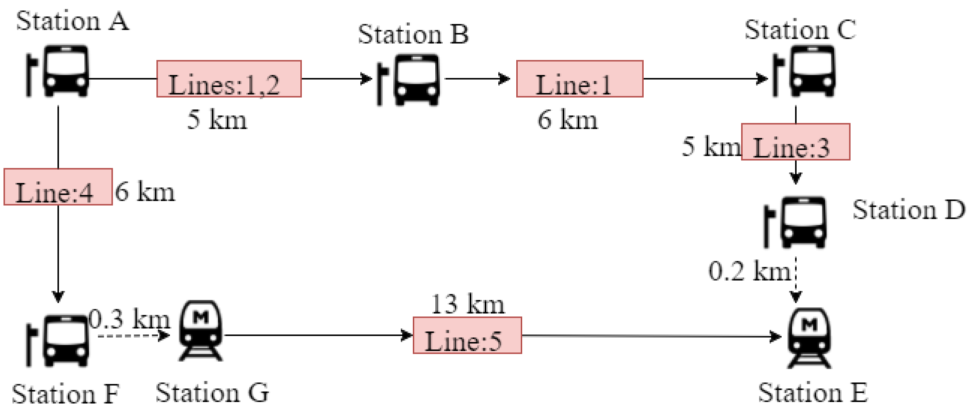

Figure 5.1: An Instance of Public Transport Network

The vertex v_t is adjacent to the vertex v_s, where $v_s, v_t \in V$, if v_t is reachable from v_s by at least one transit line or v_t is in walking distance of v_s. Therefore, each edge $e \in E$ refers to one of these connections (transit line or walking path) between adjacent vertices. The edge from the vertex v_s to the vertex v_t is denoted by (v_s, v_t). Given notation states that v_s is reachable from v_t by transit line or by walking path. To sum up, the vertices represent public transport stations and the edges represent connections between adjacent stations in our model.

For example, there is a graph $G = (V, E)$ in Figure 5.2 that represents the public transport network in Figure 5.1. Each element of $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ refers to stations A, B, C, D, E, F, G respectively. The vertices $v_1, v_2 \in V$ refer to adjacent stations A, B in the public transport network respectively. The edge $(v_1, v_2) \in E$ refer to transit line connection between these two adjacent stations.

To represent the model, two data structures are mainly used by our route selection methods and cost functions. These data structures are namely the weighted adjacency matrix W and the transit line matrix L.

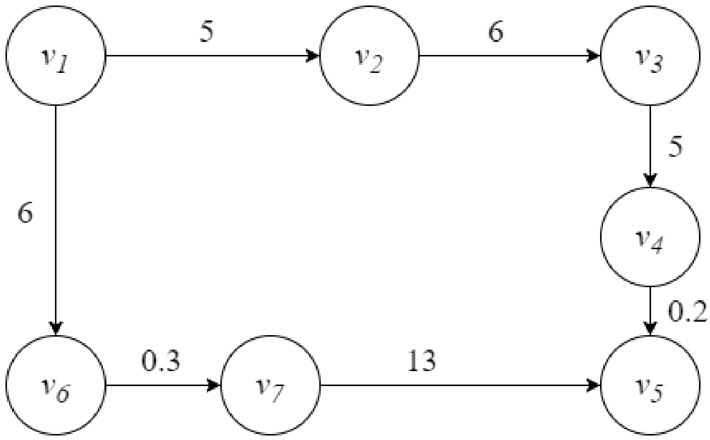

Figure 5.2: The Graph Representation of the Public Transport Network

Firstly, the **transit line matrix** is used for finding transit lines, which traverse between the adjacent vertices. The transit line matrix L is a matrix which is $|V| \times |V|$. Each cell of the matrix represents a set of transit lines between the related adjacent vertices. The transit line set of the edge (v_i, v_j) is stated by $l_{i,j}$, where $v_i, v_j \in V$, $(v_i, v_j) \in E$ and $l_{i,j} \in L$. The transit line matrix L is stated as follows;

$$L_{nxn} = \begin{bmatrix} \emptyset & \cdots & l_{1,n} \\ \vdots & \ddots & \vdots \\ l_{n,1} & \cdots & \emptyset \end{bmatrix} \text{ where } l_{i,j} \text{ is a set of transit lines of the edge } (v_{i,}, v_j) \in E$$

Secondly, the **weighted adjacency matrix** is used for finding the distances between adjacent vertices in the study. The weight of the edge is a positive numeric value that represents the distance from the source vertex to the target vertex. Calculating the real traveling distance between adjacent vertices is a challenging task, since the travels on transit lines are on complex paths (joined by too many edges) [3]. The approximate traveling distance is calculated by determining the straight-line distances between the adjacent vertices. Therefore, the function D in Equation 5.1 is used in order to calculate the straight line distances between adjacent vertices.

All vertices' coordinates are determined by using their related stations in the real world public transport network. The longitude and latitude of the vertex v_i is represented as lon_i and lat_i respectively. The straight-line distances between the adjacent vertices v_1 and v_2 are determined by using Haversine Formula [32] as follows;

$$D = 2r \sin^{-1}\left(\sqrt{\sin^2\left(\frac{lat_2 - lat_1}{2}\right) + \cos(lat_1) * \cos(lat_2) * \sin^2\left(\frac{lon_1 - lon_2}{2}\right)}\right) \quad (5.1)$$

For each edge $(v_i, v_j) \in E$, the weight of the edge is $w_{i,j} \in W$, where $v_i, v_j \in V$. The weighted adjacency matrix W is stated as follows;

$$W_{nxn} = \begin{bmatrix} 0 & \cdots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{n,1} & \cdots & 0 \end{bmatrix} \text{ where } w_{i,j} = \begin{cases} d(v_i, v_j), & \text{if } (v_i, v_j) \in E \\ 0 & \text{else} \end{cases}, v_i, v_j \in V$$

To explain the proposed data structures better, the transit line matrix and the weighted adjacency matrix are determined with respect to the graph $G = (V, E)$ in Figure 5.2. The distances between adjacent vertices are obtained from Figure 5.1, since the distances between referred adjacent stations are labeled in the figure. For instance, the distance between stations A and B (they are referred by v_1, v_2 respectively in the model) is stated as $w_{1,2} = d(v_1, v_2) = 5$ km. The weighted adjacency matrix W of the G in Figure 5.2 is as follows;

$$W_{7x7} = \begin{bmatrix} 0 & 5 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 & 13 & 0 & 0 \end{bmatrix}$$

Additionally, transit lines of this public transport network are given in Figure 5.1. For instance, the transit lines that traverse from station A to station B are stated as $l_{1,2} = \{1,2\}$ in this model. The transit line matrix L of the graph $G = (V, E)$ in Figure 5.2 is as follows;

$$L_{7x7} = \begin{bmatrix} \{\emptyset\} & \{1,2\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{4\} & \{\emptyset\} \\ \{\emptyset\} & \{\emptyset\} & \{1\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{3\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{5\} & \{\emptyset\} & \{\emptyset\} \end{bmatrix}$$

To sum up, the distances and transit lines between adjacent vertices are determined by the weighted adjacency matrix and the transit line matrix, respectively. For instance, in Figure 5.2, if it is aimed to go from the vertex v_1 to its adjacent vertex v_6, weight of the edge (v_1, v_6) is $w_{1,6} = 6$ km. Additionally, transit line set of the adjacent vertices v_1 and v_6 is $l_{1,6} = \{4\}$. If it is aimed to go from the vertex v_6 to its adjacent vertex v_7, weight of the edge (v_6, v_7) is $w_{6,7} = 0.3$ km. It is determined that; from the vertex v_6 to its adjacent vertex v_7 can only be reached by walking. Because $l_{6,7} = \{\emptyset\}$ states that there is no transit line between these two vertices.

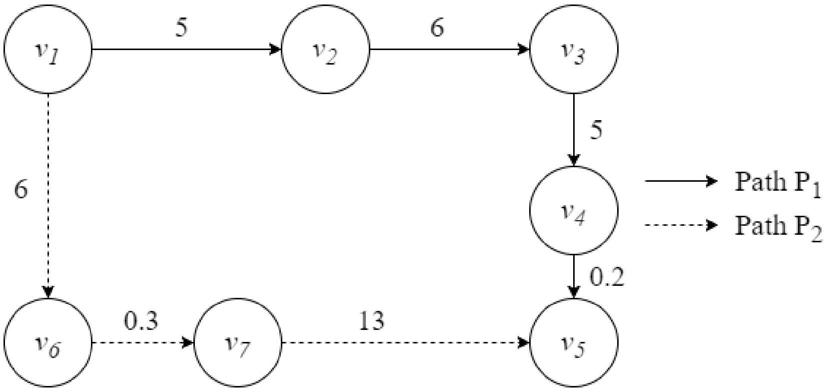

Figure 5.3: Possible Paths from Given Source to Target

Possible routes from the source station to the target station are represented as paths in the model. For instance, there are two possible routes from the source station A to the target station E in Figure 5.1. Remark that; vertices v_1, v_5 refer to the stations A, E respectively. Therefore, possible routes from the source station A to the target station E are represented as paths such that; $P_1(v_1, v_5) = (v_1, v_2, v_3, v_4, v_5)$ and $P_2(v_1, v_5) = (v_1, v_6, v_{7,} v_5)$. P_1 is illustrated by solid straight lines and P_2 is illustrated by dashed straight lines in Figure 5.3.

5.2 The Cost Functions of the Path

The experimental results of the evaluated methods are demonstrated in terms of the distances of the routes, the numbers of the transfers, and the numbers of stops on the routes. Firstly, the “The Distance of Route Function” is defined in order to calculate the distances of the routes. Secondly, the “The Number of Transfers Function” is defined in the model in order to calculate the numbers of transfers on the routes. Lastly, the “The Number of Stops Function” is defined in order to calculate the numbers of stops on the routes.

The primary cost of a path (route) depends on the route selection criteria in our model. For the route selection with respect to the shortest path, the primary cost of a path is the distance of the path. For the route selection with respect to the least transfer, the primary cost is the number of transfers. For the route selection with respect to the least stop, the primary cost is the number of stops.

To sum up, the costs of the routes in the public transport network are calculated by our model's cost functions are named as;

- The Distance of Route Function
- The Number of Transfers Function
- The Number of Stops Function

These functions are denoted by $\mathbf{dr}, \mathbf{nt}, \mathbf{ns}$, respectively.

The Distance of Route Function: This function calculates the weight (distance) of the path (explained in Section 4.5). It is denoted by $\mathrm{dr}(\mathrm{P}(\mathrm{v}_1, \mathrm{v}_n))$;

$$\mathrm{dr}\big(\mathrm{P}(\mathrm{v}_1, \mathrm{v}_n)\big) = \sum_{i=1}^{n-1} \mathrm{w}_{i,i+1}\text{, where } \mathrm{w}_{i,j} = \mathrm{d}(\mathrm{v}_{i,}\ \mathrm{v}_j) \text{ and } \mathrm{P}(\mathrm{v}_1, \mathrm{v}_n) = (\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_n) \quad (5.2)$$

The Number of Transfers Function: This function calculates minimum number of transit lines used on the path. It is denoted by $\mathrm{nt}(\mathrm{P})$ where is determined by using a function f as follows;

$$\mathrm{nt}(\mathrm{P}) = \mathrm{f}\big(\mathrm{l}_{1,2}, \mathrm{l}_{2,3}, \ldots, \mathrm{l}_{n-1,n}, \emptyset\big) = \begin{cases} \mathrm{f}\big(\mathrm{l}_{2,3},, \ldots, \mathrm{l}_{n-1,n}\big) & \text{if } \mathrm{l}_{1,2} = \{\emptyset\} \\ \mathrm{f}\big(\mathrm{l}_{2,3}, \mathrm{l}_{3,4},, \ldots, \mathrm{l}_{n-1,n}\big) + 1 & \text{if } \mathrm{l}_{1,2} \cap \mathrm{l}_{2,3} = \{\emptyset\} \\ \mathrm{f}\big(\mathrm{l}_{1,2} \cap \mathrm{l}_{2,3}, \mathrm{l}_{3,4},, \ldots, \mathrm{l}_{n-1,n}\big) & \text{if } \mathrm{l}_{1,2} \cap \mathrm{l}_{2,3} \neq \{\emptyset\} \end{cases} \quad (5.3)$$

where $\mathrm{l}_{i,j}$ is set of transit lines on the edge $\big(\mathrm{v}_{i,}, \mathrm{v}_j\big) \in \mathrm{E}$ and $\mathrm{P} = \mathrm{P}(\mathrm{v}_1, \mathrm{v}_n) = (\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_n)$.

The Number of Stops Function: This function calculates the order of the path (number of stops on the route). It is denoted by $\mathrm{ns}(\mathrm{P})$ as follows;

$$\mathrm{ns}\big(\mathrm{P}(\mathrm{v}_1, \mathrm{v}_n)\big) = |\mathrm{P}| = \mathrm{n} \text{ where } \mathrm{P}(\mathrm{v}_1, \mathrm{v}_n) = (\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_n) (5.4)$$

For instance, there are two possible paths from vertex v_1 to v_5 in Figure 5.3 such that $\mathrm{P}_1(\mathrm{v}_1, \mathrm{v}_5) = (\mathrm{v}_1, \mathrm{v}_2, \mathrm{v}_3, \mathrm{v}_4, \mathrm{v}_5)$ and $\mathrm{P}_2(\mathrm{v}_1, \mathrm{v}_5) = \big(\mathrm{v}_1, \mathrm{v}_6, \mathrm{v}_{7,}\ \mathrm{v}_5\big)$. Assume that a route selection method proposed P_1 as the ideal route and the another route selection method proposed P_2 as the ideal route. The route selections of these two methods are compared in terms of distance, the number of transfers and the number of stops in the experimental study. The cost functions are used for this comparison.

The distances of proposed routes in Figure 5.3 are calculated by using "The Distance of Route Function" as follows;

$$\mathrm{dr}(P_1(v_1, v_5)) = w_{1,2} + w_{2,3} + w_{3,4} + w_{4,5} = 5 + 6 + 5 + 0.2 = 17.2 \text{ km}$$
$$\mathrm{dr}(P_2(v_1, v_5)) = w_{1,6} + w_{6,7} + w_{7,5} = 6 + 0.3 + 13 = 19.3 \text{ km}$$

The proposed routes' numbers of transfers in Figure 5.3 are calculated by using "The Number of Transfers Function" as follows;

$$\begin{aligned}
&\mathrm{nt}(P_1(v_1, v_5)) = f(l_{1,2}, l_{2,3}, l_{3,4}, l_{4,5}, \emptyset) \\
&= f(\{1,2\}, \{1\}, \{3\}, \{\emptyset\}, \emptyset) \\
&= f(\{1\}, \{3\}, \{\emptyset\}, \emptyset) \\
&= 1 + f(\{3\}, \{\emptyset\}, \emptyset) \\
&= 2 + f(\{\emptyset\}, \emptyset) \\
&= 2 + f(\emptyset) = 2
\end{aligned}$$

$$\begin{aligned}
&\mathrm{nt}(P_2(v_1, v_5)) = f(l_{1,6}, l_{6,7}, l_{7,5}, \emptyset) \\
&= f(\{4\}, \{\emptyset\}, \{5\}, \emptyset) \\
&= 1 + f(\{\emptyset\}, \{5\}, \emptyset) \\
&= 1 + f(\{5\}, \emptyset) \\
&= 2 + f(\emptyset) \\
&= 2
\end{aligned}$$

The proposed routes' numbers of stops in Figure 5.3 are calculated by using "The Number of Stops Function" as follows;

$$\mathrm{ns}(P_1(v_1, v_5)) = |(v_1, v_2, v_3, v_4, v_5)| = 5$$
$$\mathrm{ns}(P_2(v_1, v_5)) = |(v_1, v_6, v_{7,} v_5)| = 4$$

Chapter 6

Method I: The Modified Shortest Path Route Selection

In this chapter, the route selection method with respect to the shortest path criterion is evaluated. In Section 6.1, Dijkstra's Algorithm [2] is evaluated in our public transport network model to propose the shortest path route selection. The main shortcomings of the Dijkstra's Algorithm for the route selection are demonstrated with examples in Section 6.2. In the last section, the modified Dijkstra's Algorithm is proposed as The Modified Shortest Path Route Selection Method in order to minimize the stated shortcomings. Remind that, there are also relevant studies that modified the Dijkstra's Algorithm for the route selection to minimize its shortcomings (explained in Chapter 3).

6.1 Dijkstra's Algorithm

Dijkstra's Algorithm is the most commonly used algorithm in the literature in order to find the shortest path between two vertices. Dijkstra's Algorithm finds the shortest paths from the source vertex to all other vertices (single-source shortest path problem) in a short time, assuming that the weights of all edges are non-negative. Therefore, the shortest path from the source vertex to the target vertex is determined by using Dijkstra's Algorithm in our public transport network model.

Let v_s be the source vertex, where $v_s \in V$. The cost of the vertex $v_i \in V$ is the distance between the source vertex v_s and v_i. Dijkstra's Algorithm assigns an initial value to the cost of each vertex. Afterwards, the algorithm develops the assigned costs of the vertices iteratively. The iterations are as follows;

I. Except for the source vertex, the cost of all vertices in the graph $v \in V$ are set to infinite. The cost of the source vertex is set to zero. Each vertex is marked as unvisited.

II. The unvisited vertex which has minimum cost is marked as the current vertex and this vertex is marked as visited.

III. The alternative cost is calculated for each adjacent vertex to the current vertex. The alternative cost of the adjacent vertex is the sum of the distance from the current vertex to the adjacent vertex and the cost of the current vertex. If the calculated alternative cost of the adjacent vertex is less than its exact cost, then the alternative cost is assigned to the cost of the adjacent.

IV. If there isn't any unvisited vertex left, the algorithm is terminated. Otherwise, the algorithm iterates from the second step.

These iterations are illustrated in Figure 6.1. Additionally, the pseudo code of Dijkstra's Algorithm is given in Algorithm 6.1.

```
Input: The weighted adjacency matrix of the graph W[n][n], source vertex s
Output: An array that stores cost of the vertices cost[n], N sized array that stores
the list of vertices which is path from the source to the related vertex path[n]
Function DijkstraAlgorithm(W, s)
Initialize N sized arrays cost[n], path[n]
Initialize vertex list unvisited
for each vertex v in the vertex set
        cost[v] ← ∞
        add v to unvisited
end for
cost[s] ← 0
while unvisited is not empty
        current ← v where v has min cost in unvisited list
        remove current from unvisited
        for each vertex adjacent to current
                alternativeCost ← cost[current] + W[current, adjacent]
                if alternativecost < cost[adjacent]
                        cost[adjacent] ← alternativeCost
                        path[adjacent] ← path[current] + current
                end if
        end for
end while
return path, cost
```

Algorithm 6.1: Dijkstra's Algorithm

Note that vertices are represented as their indices in the pseudo-code of the algorithm. For instance, in the pseudo code, s represents the vertex $v_s \in V$ in the graph.

Additionally, remind that, in a graph $G = (V, E)$ each edge $(v_i, v_j) \in E$ has weight $w_{i,j} \in W$ where $v_i, v_j \in V$. The weighted adjacency matrix in our model is as follows;

$$W_{nxn} = \begin{bmatrix} 0 & \cdots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{n,1} & \cdots & 0 \end{bmatrix} \text{ where } w_{i,j} = \begin{cases} d(v_i, v_j), & \text{if } (v_i, v_j) \in A \\ 0 & \text{else} \end{cases}, \text{where } v_i, v_j \in V$$

Dijkstra's Algorithm returns the path, which is an array list of shortest paths from the source to each vertex in the graph. Therefore, path[t] gives the shortest path from the source vertex to the target vertex v_t. Transit lines on the edges of the shortest path are determined by using the transit line matrix L;

$$L_{nxn} = \begin{bmatrix} \emptyset & \cdots & l_{1,n} \\ \vdots & \ddots & \vdots \\ l_{n,1} & \cdots & \emptyset \end{bmatrix}$$

where $l_{i,j}$ is a set of transit lines on the edge $\forall (v_i, v_j) \in P(v_s, v_t)$.

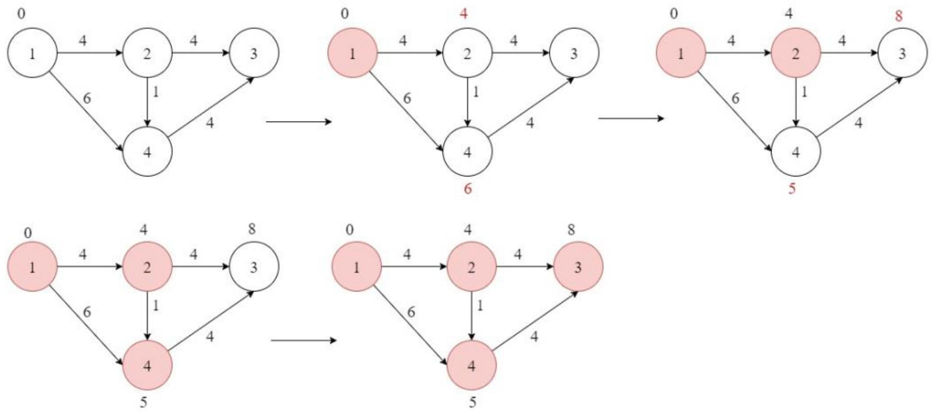

Figure 6.1: Dijkstra's Algorithm Iterations

6.2 The Shortcomings of Dijkstra's Algorithm for the Route Selection

Although Dijkstra's Algorithm is efficient in terms of running time to determine the shortest path optimally, this path is far from being the ideal algorithm for route selection; because Dijkstra's Algorithm does not take the number of transfers or the walking distances into account. These shortcomings are crucial disadvantages for the end-users in two ways. First; route selection with respect to the shortest path may include much more transfers than other possible paths whose costs are nearly equal to the distance of the shortest path.

For example, there is a directed weighted graph $G = (V, E)$ in Figure 6.2. The distances between adjacent vertices are labeled on edges. Assume that from v_1 to v_3 there is the transit line 1; from v_3 to v_4 there is the transit line 2; from v_4 to v_5 there is the transit line 3; and lastly the transit line 4 traverses v_1, v_2 and v_5.

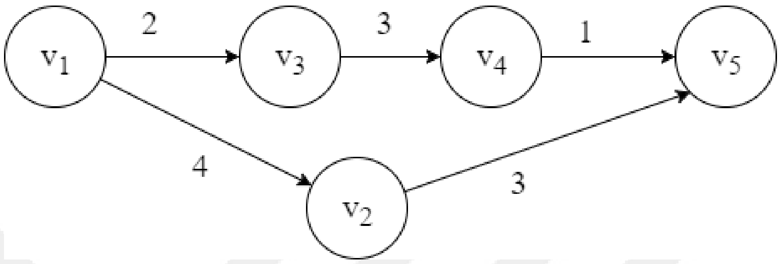

Figure 6.2: A Public Transport Network Instance

By the assumption, the transit line matrix of the graph in our model is as follows;

$$L_{5x5} = \begin{bmatrix} \{\emptyset\} & \{4\} & \{1\} & \{\emptyset\} & \{\emptyset\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{4\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{2\} & \{\emptyset\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{3\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} \end{bmatrix}$$

As clearly seen in Figure 6.2, the shortest path from the source vertex v_1 to the target vertex v_5 is $P_1(v_1, v_5) = (v_1, v_3, v_4, v_5)$. The cost of the path P_1 with respect to the shortest path is $dr(P(v_1, v_5))$ (The Distance of Route Function is explained in Section 5.2) as follows;

$$dr(P_1(v_1, v_5)) = 2 + 3 + 1 = 6$$

Cost of the path P_1 with respect to the least transfer is $t(P(v_1, v_n))$ (The Number of Transfer Function is explained in Section 5.2) as follows;

$$\begin{aligned} nt(P_1(v_1, v_5)) &= f(l_{1,3}, l_{3,4}, l_{4,5}, \emptyset) \\ &= f(\{1\}, \{2\}, \{3\}, \emptyset) \\ &= 3 \end{aligned}$$

By using Dijkstra's Algorithm, P_1 is proposed to the users as the ideal route from the v_1 to v_5; since P_1 is the shortest path. However, there is another path $P_2(v_1, v_5) = (v_1, v_2, v_5)$ from the source vertex v_1 to the target vertex v_5 where $dr(P_2(v_1, v_5)) = 7$ and

$nt(P_2(v_1, v_5)) = 1$. Although the distance cost of P_2 is only one unit higher than the distance cost of the proposed path P_1, there is no need for any transfers on the path P_2. In order to go one less unit of distance, Dijkstra's Algorithm proposes the shortest path P_1 that includes two transfers. However, most users prefer P_2 rather than P_1.

In addition, optimal route selection with respect to the shortest path may include long walking distances.

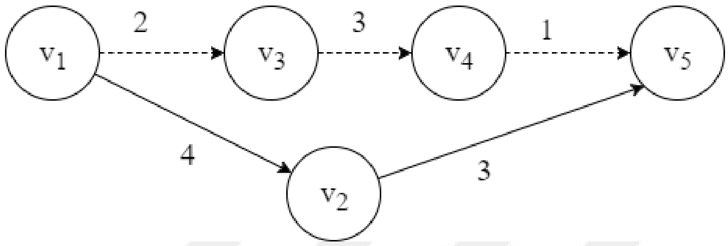

Figure 6.3: A Public Transport Network Instance

For instance, there is a directed weighted graph $G = (V, E)$ in Figure 6.3. Dashed straight lines illustrate walking paths between vertices. Solid straight lines illustrate transit lines between vertices in the figure. The shortest path from v_1 to v_5 is $P_1(v_1, v_5) = (v_1, v_3, v_4, v_5)$; which is a walking path. The distance of this path is $dr(P_1(v_1, v_5)) = 6$. However, there is another path that uses transit lines P_2=(v_1, v_2, v_5) and its distance is $dr(P_2(v_1, v_5)) = 7$. In order to go one less unit of distance, Dijkstra's Algorithm proposes the shortest path P_1 which is a long walking path. However, most users prefer P_2 rather than P_1.

6.3 The Modified Shortest Path Route Selection Method

The shortcomings of Dijkstra's Algorithm for the route selection are presented in the previous section. In order to minimize these shortcomings, Dijkstra's Algorithm is modified by implementing a penalty system. The "small penalty cost" is added to the alternative cost in cases of stated shortcomings. Thus, the number of repeated walking and the number of transfers are minimized by slightly increasing the distance of the proposed path in the new route selection method.

In order to avoid repeated walking and multiple transfers on the proposed path, a rule set is defined on Dijkstra's Algorithm. The rule set of the penalty system is as follows;

- Assume that the current vertex is reached by walking. If the adjacent vertex of the current vertex is only reachable by walking again, the repeated walking will occur. In this case, the **walking penalty** is added to the alternative cost of the adjacent vertex to prevent the repeated walking.
- Assume that the current vertex is reached by walking. If the adjacent vertex of the current vertex is reachable by a transit line, the number of transfers will be increased on the path. In this case, the **transfer penalty** is added to the alternative cost of the adjacent vertex to prevent numerous transfers on the proposed path.
- Assume that the current vertex is reached by a transit line. If its adjacent vertex is only reachable by a different transit line, the number of transfers will increase on the path. In this case, the **transfer penalty** is added to the alternative cost of the adjacent vertex.

By following the given rule set, the penalty cost from the current vertex to its adjacent vertex is determined by using the Penalty Function. Pseudo code of the function is given in Algorithm 6.2.

```
Input: List of lines reached to the current vertex currentLines, list of lines reaches
to the adjacent vertex from the current vertex adjacentLines
Output: Numeric value penalty, list of intersected lines intersectedLines
Function PenaltyFunction (currentLines, adjacentLines)
Initialize a new list intersectedLines
penalty ← 0
if currentLines is null
        if adjacentLines is null
                penalty ← walkingPenalty
        else if adjacentLines is not null
                penalty ← transferPenalty
                intersectedLines ← adjacentLines
        end if
else if currentLines is not null and adjacentLines is not null
        for each line l in currentLines
                if adjacentLines contains l
                add l to intersectedLines
                end if
        end for
        if intersectedLines is null
                penalty ← transferPenalty
                intersectedLines ← adjacentLines
        end if
end if
return penalty, intersectedLines
```

Algorithm 6.2: Penalty Function

Thus, the new penalty function is implemented to Dijkstra's Algorithm. This function is used for adding a penalty to alternative cost if it is needed. Additionally, the transit line matrix L is taken as an input parameter in this new method in order to obtain the lines between the adjacent vertices. Furthermore, an array list is used to store the data of the lines that reach to the vertices. The pseudo code of the new method is given in Algorithm 6.3.

Input: The weighted adjacency matrix of the graph W[n, n], the transit line matrix of the graph L[n, n] and the source vertex s
Output: An array that stores cost of the vertices cost[n], N sized array list that stores the list of vertices which is path from the source to the related vertex path[n]
Function ModifiedShortestPathRouteSelection(W, L, s)

```
Initialize N sized arrays cost [n], path [n],lines[n]
Initialize vertex list unvisited
for each vertex v in the vertex set
        cost[v] ← ∞
        add v to unvisited
end for
cost[s] ← 0
while unvisited is not empty
        current ← v where v has min cost in unvisited list
        remove current from unvisited
        for each vertex adjacent to current
                initialize new line list insersectedLines
                penalty, insersectedLines ←
                        PenaltyFunction(lines[current], L[current, adjacent])
                alternativeCost ← cost[current] + W[current, adjacent] +
                                                                penalty
                if alternativecost < cost[adjacent]
                        cost[adjacent] ← alternativeCost
                        path[adjacent] ← path[curent] + current
                        lines[adjacent] ← intersectedLines
                end if
        end for
end while
return path, cost
```

Algorithm 6.3: Modified Shortest Path Route Selection

In this new method, repeated walking and the numbers of transits used on the path are minimized. However, in proportion to this, the distance of the path is slightly increased. Thus, it can be stated that, an ideal route selection with respect to the shortest path is calculated by this proposed method.

Chapter 7

Method II: The Modified Least Stop Route Selection

In this chapter, the route selection method with respect to the least stop criterion is evaluated. In this method, a route with the least stops from the source to the target is proposed as the ideal route with a new approach. The new approach is explained in Section 7.1. In this approach, Breadth First Search (BFS) [21] algorithm is used in order to determine the route with the least stops in Section 7.2. Lastly, a few implementations are added to this method in order to avoid some of the shortcomings, in Section 7.3.

7.1 A New Approach for the Least Stop Route Selection

Our novel approach for the least stop route selection method is based on a claim, which is: "if a route from the source station to target station includes less number of stops than other possible routes, then rapid-transit lines are most probably used on this route". This claim is brought by the observation that the distance between the adjacent stations of the rapid-transit networks is much longer than the distance between the adjacent stops of the bus transport networks. The public transport network of İzmir is taken as an example in order to verify this observation, in the study.

The public transport network of İzmir includes rapid-transit lines such that; ferries, metro lines, and light railways. In addition to these, this network also includes numerous bus transit lines. The number of adjacent stations in the rapid-transit network is 74. The average distance between these adjacent stations is 2,986 meters. On the other hand, the number of adjacent stops in the bus transport network is 8,763 and the average distance between these adjacent stops is 509 meters. The comparison is illustrated in Table 7.1.

Table 7.1: Comparison between Rapid-transit and Bus Transit Network

	Number of Adjacent Stations	**Average Distance between Adjacent Stations (meter)**
Rapid-transit Network	74	2,986
Bus Transport Network	8,763	509

For instance, in Figure 7.1 metro line and bus line between Konak and Üçyol are illustrated. The bus route includes five stops, where the metro route includes two stations.

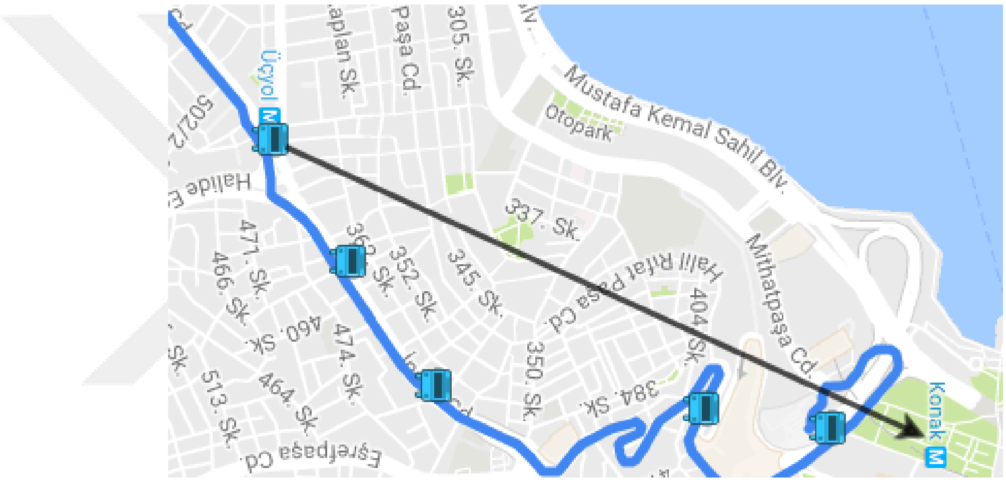

Figure 7.1: Bus Path and Metro Line from Üçyol to Konak

7.2 The Least Stop Route Selection Method

The ideal route from a source vertex to the target vertex with respect to the least stop criterion is the path with least order. Remark that, in Section 5.2, it is stated that primary cost of a path is the number of stops for the least stop route selection. It is stated as follows;

$$ns\big(P(v_1, v_n)\big) = |P| = n \text{ where } P(v_1, v_n) = (v_1, v_2, \dots, v_n)$$

This ideal route definition is also known as the shortest path from the source vertex to the target vertex in the unweighted graph. To find the shortest path from the source vertex to the target vertex in unweighted graphs, Breadth First Search Algorithm (BFS) is used in this method. BFS algorithm determines a tree data structure by traversing the given graph. The root of the determined tree is the source vertex. From the root vertex to its child (other vertices in the graph) node, there is only one path. Furthermore, this path is the shortest path from the root (the source vertex) vertex to the related vertex. Pseudo code of the BFS algorithm is given in Algorithm 7.1.

```
Input: N sized array of list that contains adjacent vertices adjacencyList[n] and
source vertex s
Output: N sized array that stores of parent of the related child parent[n]
Function BFSAlgorithm (adjacencyList, s)
Initialize N sized array parent[n] and a new queue Q
parent[source] ← null
Enqueue source to Q
while Q is not empty
        current ← Q.Dequeue()
        for each vertex adjacent in adjacencyList[current]
                if parent[adjacent] is null
                        Enqueue adjacent to Q
                        parent[adjacent] ← current
                end if
        end for
end while
return parent
```

Algorithm 7.1: BFS Algorithm

Note that the adjacency list is used in the algorithm; because BFS algorithm queries the neighborhood of a vertex for each iteration. The cost of this query is determined as $O(1)$ by using the adjacency lists (it is explained in detail in Section 4.3).

The path from the source vertex v_s to the target vertex v_t is determined by using ProposedPath function. The target vertex and the output of BFSAlgorithm (parent) is taken as an input parameter in this function. The function follows parents of the target vertex iteratively until the iteration arrives at the source vertex. Thus, this function returns the proposed path from the source to the target. Pseudo code of the algorithm is given in Algorithm 7.2.

```
Input: N sized array that stores the parent vertex of the related vertex parent[n],
source vertex s, target vertex t
Output: Stack P that gives the proposed path from the source vertex s to the target
vertex t
Function ProposedPath (parent, s, t)
Initialize new stack P
current ← t
do
        push current to P
        current ← parent [current]
while current ≠ s
return P
```

Algorithm 7.2: Proposed Path Algorithm

Lastly, the line information between the adjacent vertices of the proposed path is determined by using the transit lines matrix L.

For example, let an unweighted graph G represent the public transport network in Figure 7.1. In this graph, Üçyol is the source vertex v_s and Konak is the target vertex v_t. Bus stations are represented by vertices $v_2, v_3, v_4, v_5, v_6 \in V$. Konak and Üçyol metro stations are represented by $v_1, v_7 \in V$. The walking paths are illustrated as dashed lines. The modeled graph G is illustrated in Figure 7.2.

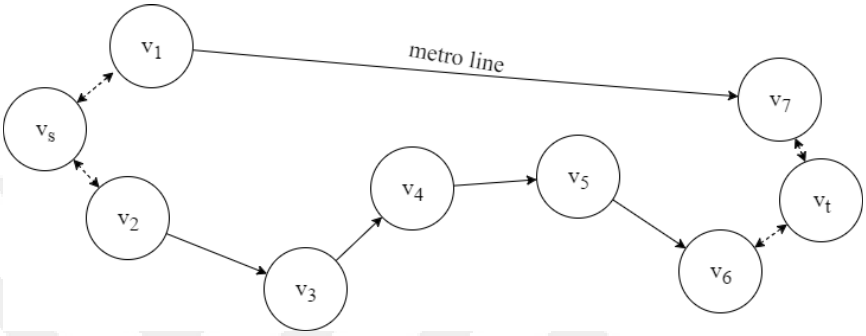

Figure 7.2: Graph Representation of the Problem

For graph G in Figure 7.2, BFS algorithm determines the tree structure as given in Figure 7.3.

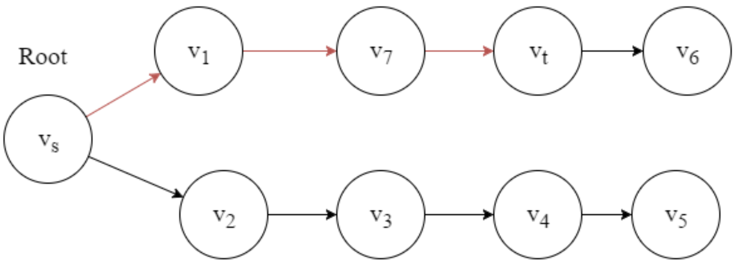

Figure 7.3: Proposed Tree Structure by BFS Algorithm

We can find the proposed path with respect to the least stop route selection from the source vertex to the target vertex by following the parents of the target vertex iteratively; until the iteration arrives at the source vertex. For instance, the ideal route is determined as $P_1 = (v_s, v_1, v_7, v_t)$ by following the parents of the target vertex v_t in Figure 7.3.

7.3 The Modified Least Stop Route Selection Method

Although BFS algorithm returns the shortest path in unweighted graphs in a short time, the proposed path may include a shortcoming. There is an example of this shortcoming in the following.

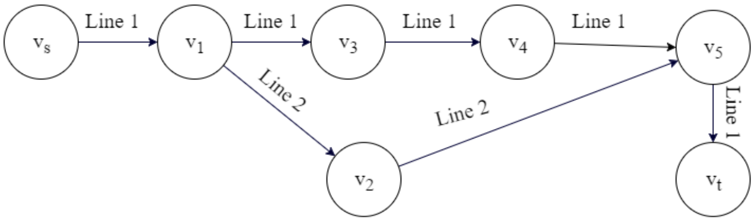

Figure 7.4: A Public Transport Network with a Special Case

In Figure 7.4, there is a graph G where transit lines are labeled on the edges of the graph. There are two possible paths from the source vertex v_s to the target vertex v_t as; $P_1(v_s, v_t) = (v_s, v_1, v_3, v_4, v_5, v_t)$ and $P_2(v_s, v_t) = (v_s, v_1, v_2, v_5, v_t)$. P_1 and P_2 are illustrated in Figure 7.5 and Figure 7.6, respectively.

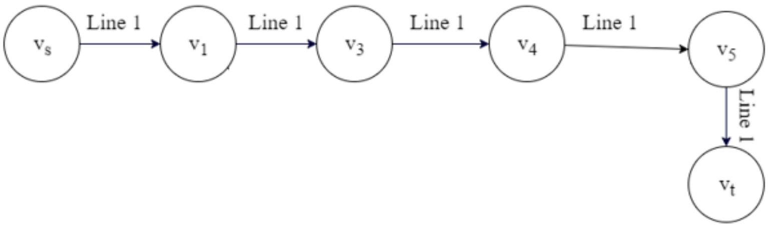

Figure 7.5: Path P_1

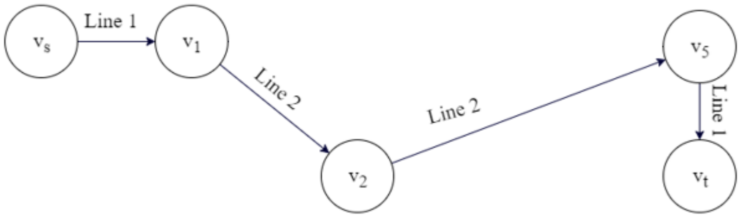

Figure 7.6: Path P_2

By the BFS algorithm, path P_2 is proposed as the ideal path; since its order is less than the order of the P_1. In order to traverse one less vertex, two more transfers are needed on the proposed path P_2. However, most users choose P_1; since they don't prefer two transfers in order to travel one less station.

To prevent the above-mentioned shortcoming, a method is developed for fixing the solution path of the BFS algorithm. The following method, named as fixPath, is based on a trade-off between the number of transfers and the order of the shortest path. For a given proposed path $P = (v_1, v_2 \dots v_n)$, structure of the method is as follows;

1. Find next transfer vertex v_{tr} of the given path, where there is no intersected line between (v_{tr-1}, v_{tr}) and (v_{tr}, v_{tr+1}).

2. Get the edges after the transfer vertex on the path; (v_{tr+1}, v_{tr+2}), (v_{tr+2}, v_{tr+3}).
3. Check whether $l_{tr+1,tr+2}, l_{tr+2,tr+3}$ (line lists of the edges (v_{tr+1}, v_{tr+2}), (v_{tr+2}, v_{tr+3}) respectively) contain any line of $l_{tr-1,tr}$. If so, state this vertex as v_{tr+i}, v_{tr+i+1} where $0 < i < 3$.
4. An alternative sub-path between v_{tr} and v_{tr+i+1} is determined by the intersected line.
5. If the order of the alternative sub-path is not greater than $i + 3$, then fix given proposed path with the alternative sub-path.

For example, $P_2 = (v_s, v_1, v_2, v_5, v_t)$ is proposed by BFS algorithm from v_s to v_t for the graph in Figure 7.4. By fixPath method,

1. The next transfer vertex is determined as v_1.
2. For v_1, next edges to be examined are (v_2, v_5) and (v_5, v_t).
3. $l_{5,t}$ contains a transit line, which is an element of $l_{s,1}$ (the edge before the transfer vertex v_1 on the path). Note that $i = 2$.
4. $P_1(v_1, v_5) = (v_1, v_3, v_4, v_5)$ is determined as alternative path from v_1 to v_5.
5. The order of the alternative sub-path is $|P_1(v_1, v_5)| = 4 \ngtr i + 3$. Therefore, the proposed path is fixed as $P_2' = (v_s, v_1, v_3, v_4, v_5, v_t)$.

The order of the path is increased by 1. However, the number of the transfers on the path are decreased by 2.

Chapter 8

Method III: The Least Transfer Route Selection

In this chapter, the route selection method with respect to the least transfer criterion is evaluated. In Section 8.1, our public transport network model is extended to an unweighted complex graph in space P [22], [23]. The route selection method on this model is explained in Section 8.2.

8.1 The Public Transport Network in Space P

To determine optimal route with respect to the least transfer route criterion, the public transport network model extended to an unweighted complex graph in space P. In this model, a vertex v_i is assumed adjacent to the vertex v_j if v_i is reachable from v_j by a transit line (these vertices do not need to be exactly adjacent).

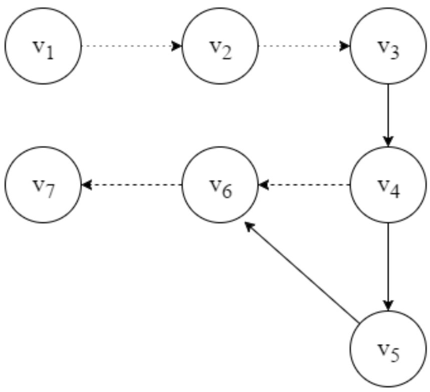

Figure 8.1: The Graph Representation of a Public Transport Network

For example, there is a graph $G = (V, E)$ that is the representation of some public transport network in Figure 8.1. The dotted straight lines refer to transit line 1, the solid straight lines refer to transit line 2 and dashed straight lines refer to transit line 3 in the modeled graph. The public transport network model in space P is a complex graph $G' = (V, E')$ in Figure 8.2.

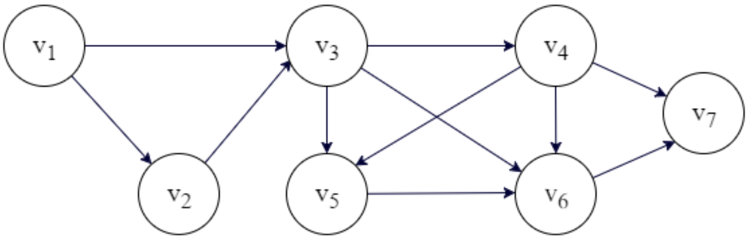

Figure 8.2: Space P Model of the Graph

Note that both of vertices v_2 and v_3 are reachable from v_1 by a transit line in Figure 8.1. The neighborhood of the vertex v_1 is $N(v_1) = \{v_2, v_3\}$ in the extended model. Hence the adjacency matrix of the model G' as follows;

$$A'_{7x7} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The transit line matrix of the new model G' as follows;

$$L'_{7x7} = \begin{bmatrix} \{\emptyset\} & \{1\} & \{1\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} \\ \{\emptyset\} & \{\emptyset\} & \{1\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{2\} & \{2\} & \{2\} & \{\emptyset\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{2\} & \{2,3\} & \{3\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{2\} & \{\emptyset\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{3\} \\ \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} & \{\emptyset\} \end{bmatrix}$$

8.2 The Least Transfer Route Selection Method

Breadth first search based algorithm is used in this method in order to find the least transfer path from the source vertex to the target vertex. Instead of the adjacent matrix, the adjacency list is used, because the adjacency list is compatible with the BFS algorithm (it is described in detail in previous chapters).

The pseudo code of the method is given in Algorithm 8.1.

```
Input: Adjacency lists adjacencyByTransit[n], adjacencyByWalking[n] and
source vertex s
Output: N sized array that stores the list of vertices which is path from the source to
the related vertex path[n]
Function LeastTransferMethod (adjacencyByTransit, adjacencyByWalking, s)
Initialize N sized array list path[n]
Initialize a new queue Q and nextLayer
Initialize a new list walkingCheckList
path[s] ← null
Enqueue s and each vertex in adjacencyByWalking[s] to Q
for each vertex v in adjacencyByWalking[s]
        path[v ] ← current
end for
while Q is not empty
        current ← Q.Dequeue()
        for each vertex adjacent in adjacencyByTransit [current]
                if parent[adjacent] is null
                        Enqueue adjacent to nextLayer
                        path[adjacent] ← path[current] + current
                        Add adjacent to walkingCheckList
                end if
        end for
        if (Q is empty)
                for each vertex v in walkingCheckList
                        for each vertex wadjacent in adjacencyWalking[v]
                                if parent[wadjacent] is null
                                        Enqueue wadjacent to nextLayer
                                        path[wadjacent] ← path[v] + v
                                end if
                        end for
                end if
                Q ← nextLayer.Copy
                Clear nextLayer
        end if
end while
return path
```

Algorithm 8.1: Least Transfer Route Selection Algorithm

The structure of the method is as follows;

- Firstly, the algorithm enqueues the source vertex v_s and its walking neighborhood. These vertices are marked as visited.
- After the first step, the algorithm traverses adjacent vertices to each vertex in the queue. The traversed vertices are added to next layer if they are not visited before. Additionally, the traversed vertices are marked as visited.
- Whenever there are no vertices left in the queue, the walking neighborhood of each element in the next layer is traversed. If traversed vertices are

unvisited, then they are added to next layer and they are marked as visited. Lastly, each element of next layer is added to the queue and next layer is cleared. And this iteration goes on until there is no unvisited vertex left in the graph.

For example, from v_1 to v_7 in Figure 8.2 the route selection method iterations are illustrated in Figure 8.3. The method returns the proposed path from v_1 to v_7 as $P(v_1, v_7) = (v_1, v_3, v_4, v_7)$. The line information between adjacent vertices on the path is determined by using the transit line matrix of this graph, such that from v_1 to v_3 the transit line set is $l_{1,3} = \{1\}$, from v_3 to v_4 the transit line set is $l_{3,4} = \{2\}$, lastly from v_4 to v_7 the transit line set is $l_{4,7} = \{3\}$.

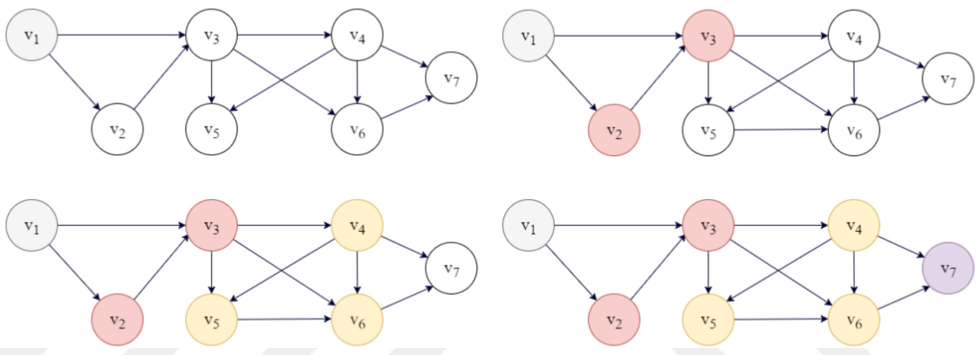

Figure 8.3: BFS Algorithm Iterations

Chapter 9

Experimental Study

In this chapter, experimental results of the route selection methods are demonstrated. Let's remember that, in order to propose ideal routes with respect to the shortest path, the least stop, and the least transfer (our route selection criteria in the study) routes, following methods are evaluated respectively;

I. The Modified Shortest Path Route Selection
II. The Modified Least Stop Route Selection
III. The Least Transfer Route Selection

These methods are tested on 2000 source-target pairs that are stations of the public transport network of İzmir. For each source-target pair, three routes (one for each method) are calculated. Costs (the distance of route, the number of transfers, and the number of stops) of these routes are calculated by using the cost functions in our model (as explained in Section 5.2). Therefore, the proposed methods are compared with each other in terms of these cost functions.

Firstly, the developed application for experimental study is explained in Section 9.1. Furthermore, details of the used dataset in the application are represented in this section. In section 9.2, results of the study are explained with figures. In section 9.3, the results of the study are discussed. The small samples of route selections for each method in this study are represented in the last section.

9.1 The Application and Dataset

Each algorithm of the proposed methods and the cost functions are developed by using C# on a personal computer with Intel i5-4440 3.1 GHz processor, 8 GB 1600 MHz DDR 3 RAM and Windows 10 operating system.

To observe the results of the proposed methods, a visual form application is developed. Additionally, the application includes the GMap.NET (map component) [33]. The application is illustrated in Figure 9.1.

In the application, the source and the target stations of the routes are entered by using text boxes. The text boxes of the application have the auto-complete property that displays possible stations as a list. After entering the source and target stations, route is determined with respect to the route selection methods by clicking the related buttons. The costs of the proposed routes; the distances of routes, the numbers of transfers, the numbers of stops on the routes are calculated by using “The Distance of Route Function” **dr**, “The Number of Transfers Function” **nt** and “The Number of Stops Function” **ns** (as detailed in Section 5.2) respectively. These costs are returned as results in the application. Additionally, the proposed route is drawn on the map by using the GMap.NET component of the application.

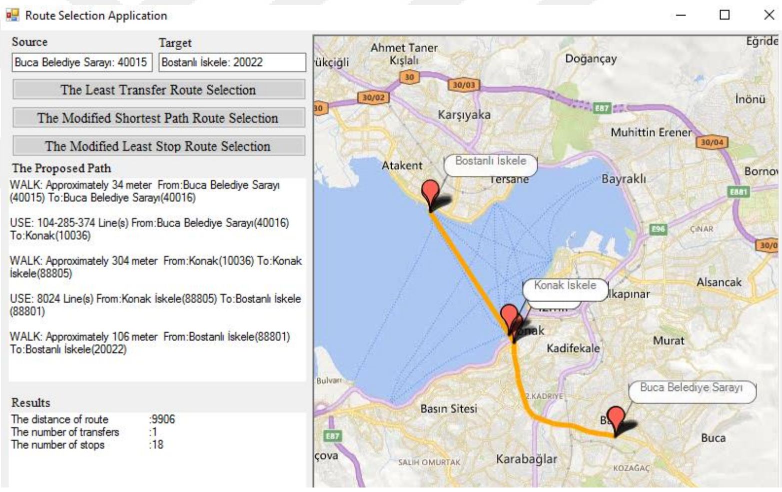

Figure 9.1: The Route Selection Application

The application uses the real-world transport network of İzmir (Turkey) as the dataset. The dataset is obtained from the web page of ESHOT [34]. Experiments are conducted with this dataset. This dataset is presented in the table below.

Table 9.1: Comparison between Standard Model and Space P Model

Network Model	Number of Stations	Number of Walking Connections	Number of Transit Connections
The Public Transport Network Model	7,704	34,630	8,837
The Public Transport Network Model in Space P	7,704	34,630	289,875

Public transport network of İzmir has four transit modes as; bus, metro, ferry and light railway. There are 7,704 stations (vertices) in the public transport network of İzmir that includes ferry stations, metro stations, light railway stops, and mostly the bus stops. There are 8,837 transit line connections (edges) between all of these stations. The number of these connections is 289,875 in the space P model (remark that, it is used in the least transfer method). Additionally, there are 34,630 walking connections between these stations. Note that walking distance is taken as 300 meters as a constant in this study.

9.2 Results of the Study

In this study, The Modified Shortest Path Route Selection (*SPRS*), The Least Transfer Route Selection (*LTRS*), and The Modified Least Stop Route Selection (*LSRS*) methods are tested on 2000 source-target pairs that are stations of the public transport network of İzmir. For each source-target pair, three routes (one for each method) are calculated. Costs (the distance of route, the number of transfers, and the number of stops) of these routes are calculated by using the cost functions in our model (as explained in Section 5.2). Firstly, the average results of each methods' route selections are stated in this section. Additionally, relationships between these average results for each method are illustrated. Secondly, relationships between costs of the routes are demonstrated with scatter plots for each method. Lastly, relationships between the distances of routes and the straight line distances of routes for each method are explained with scatter plots.

Table 9.2: The Average Results of the Route Selection Methods

Evaluated Route Selection Method	**Average Distance (km)**	**Average Number of Stops**	**Average Number of Transfers**	**Average Running Time(ms)**
The Modified Shortest Path Route Selection	33,598.75	56.00	2.48	309.37
The Least Transfer Route Selection	37,535.04	65.70	1.77	43.23
The Modified Least Stop Route Selection	34,976.82	45.02	3.32	1.02

Average results of each method's route selections are shown in Table 9.2. Note that, the best method for each cost is illustrated in the green cells in the table. Additionally, the worst method for each cost is illustrated in the red cells.

It is observed that the average running times of all evaluated route selection methods are remarkably fast enough to be used in the public transport applications. Although *SPRS* method has the worst running time, its average running time is only 0.3 second. Additionally, the average distances of routes proposed by evaluated methods are not considerably different from each other. However, the average results of the evaluated methods are not close to each other in terms of the number of transfers and the number of stops on the route.

For each method, relationships between ***average distance of routes*** and ***average number of stops on the route*** are illustrated in Figure 9.2.

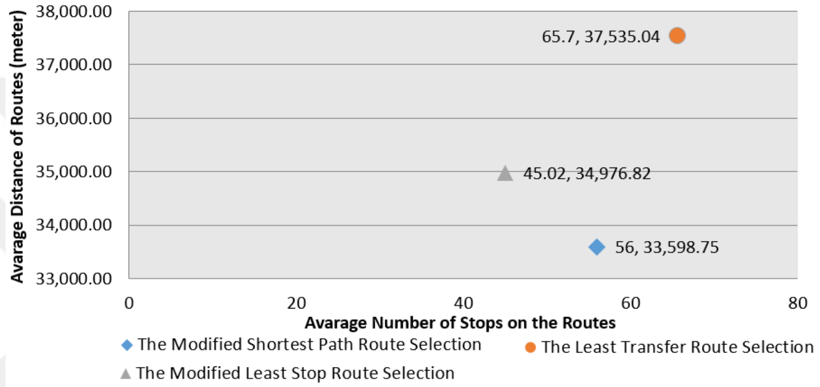

Figure 9.2: Relationships between Average Distance and Average Number of Stops on the Route

In Figure 9.2, it is observed that *LTRS* method has the worst average results in terms of distance of route and number of stops on the route. Additionally, it is observed that *LSRS* method is significantly better than others in terms of average number of stops on the route since *LSRS* method mainly uses rapid-transit lines if they are on the route. Remind that, in public transport network of İzmir the average distance between rapid-transit stations is approximately 2.5 km and the average distance between bus stops is approximately 0.5 km. Hence, the route with rapid-transit lines includes less number of stops than other routes for the same distance. Furthermore, *SPRS* is the best method in terms of average distance of route since this method is based on the shortest path algorithm. However, its average distance is slightly close to others methods' average distance, because distances of its proposed routes are slightly increased in order to minimize the number of transfers and walking distance.

For each method, relationships between ***average number of transfers*** and ***average distance of routes*** are illustrated in Figure 9.3. *LTRS* method is significantly better than others in terms of average number of transfers since *LTRS* method is optimized for least transfer route selection. On the other hand, it is observed that *LSRS* method is considerably worse than other methods in terms of the number of transfers, because *LSRS* method mainly tries to propose a route with the least number of stop.

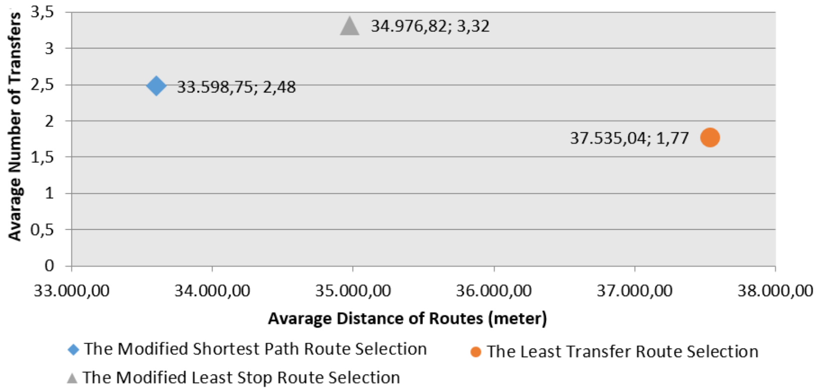

Figure 9.3: Relationships between Average Number of Transfers and Average Distance of Route

For each method, relationships between ***average number of transfers*** and ***average number of stops*** on the route are illustrated in Figure 9.4. It is observed that *LSRS* is the best in terms of average number of stops on the route however it is worst in terms of average number of transfers. On the other hand, it is observed that *LTRS* is the best in terms of average number of transfer; however, it is the worst in terms of average number of stops on the route. *STRS* method give more balanced results than others since *STRS* is the second method in both of the average results.

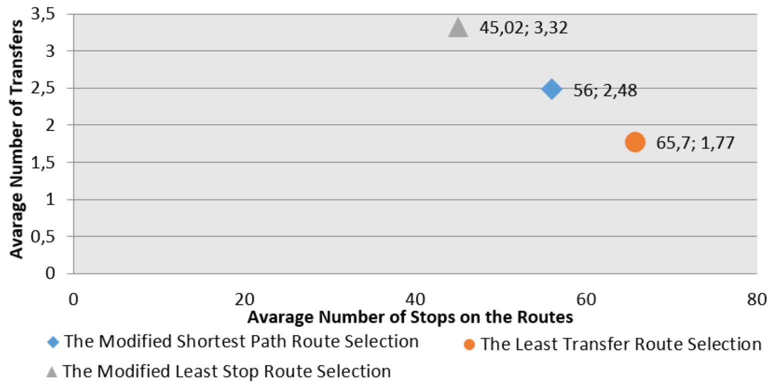

Figure 9.4: Relationships between Average Number of Transfers and Average Number of Stops

Remind that, the relationships between ***average distance of routes*** and ***average number of stops on the route*** are illustrated in Figure 9.2. For each method's route selections in the study, relationships between the ***distance of the route*** and the ***number of stops on the route*** are represented on scatter plots in Figure 9.5. Although there are some outliers on the plot, we can say that for each evaluated method's route selections there are moderate positive correlations between the number of stops and the number of transfers. Furthermore, it is observed that for the same number of stops, distances of routes that are proposed by *LSRS* method is slightly longer than others. Because *LSRS* method mainly uses rapid-transit lines if they are on the route.

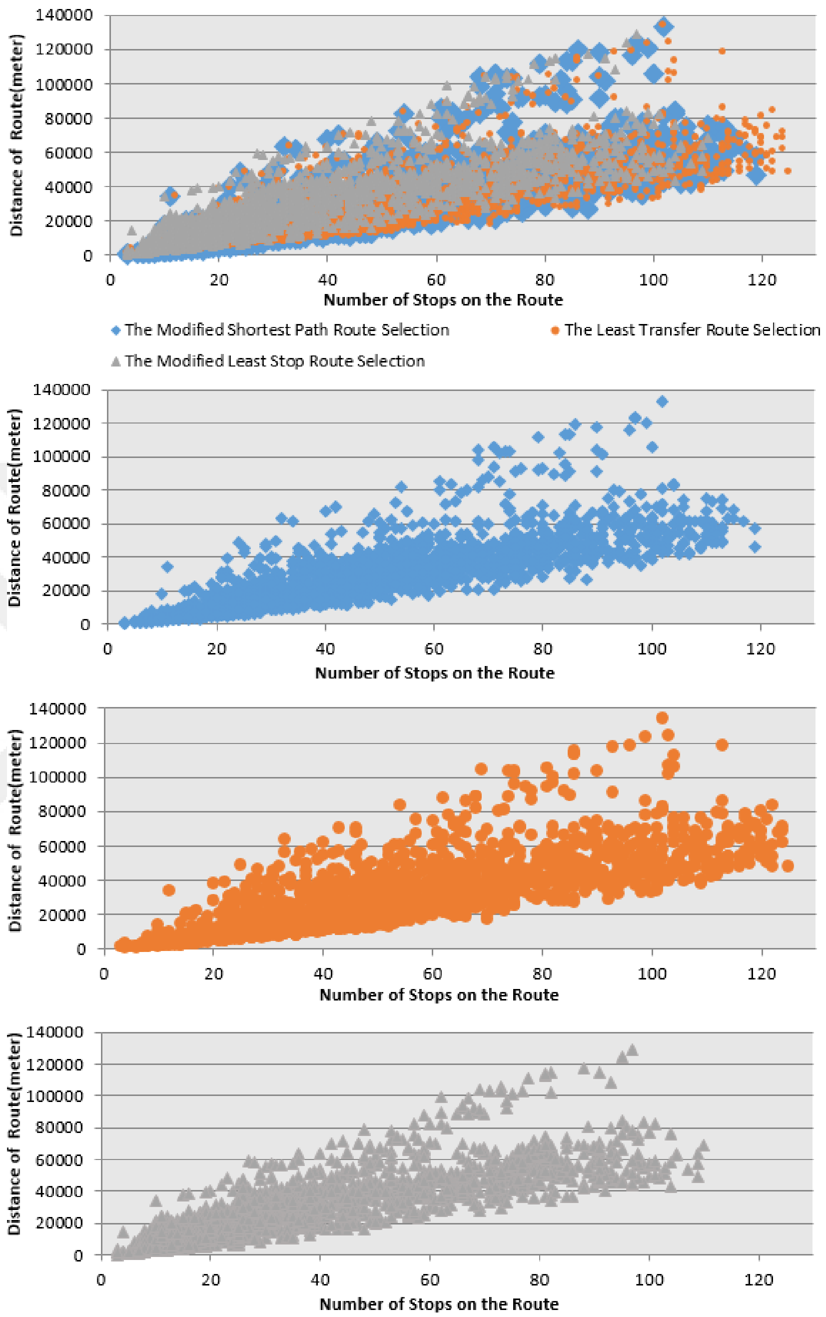

Figure 9.5: Relationships between Distance of Route and Number of Stops on the Route

Remind that, the relationships between ***average number of transfers*** and ***average distance of routes*** are illustrated in Figure 9.3. For each method's route selections, relationships between the ***number of transfers*** and the ***distance of route*** are represented on scatter plots in Figure 9.6. There seems to be no correlation between the number of transfers and the distance of route. However, it is clearly observed in the figures that *LTRS* method route selections are pretty good in terms of the number of transfers. Additionally, it is clearly observed that *LSRS* method's results are considerably bad in terms of the number of transfers. Remark that the number of transfers can be up to 6 for *LSRS* method, in order to propose a route with less number of stops.

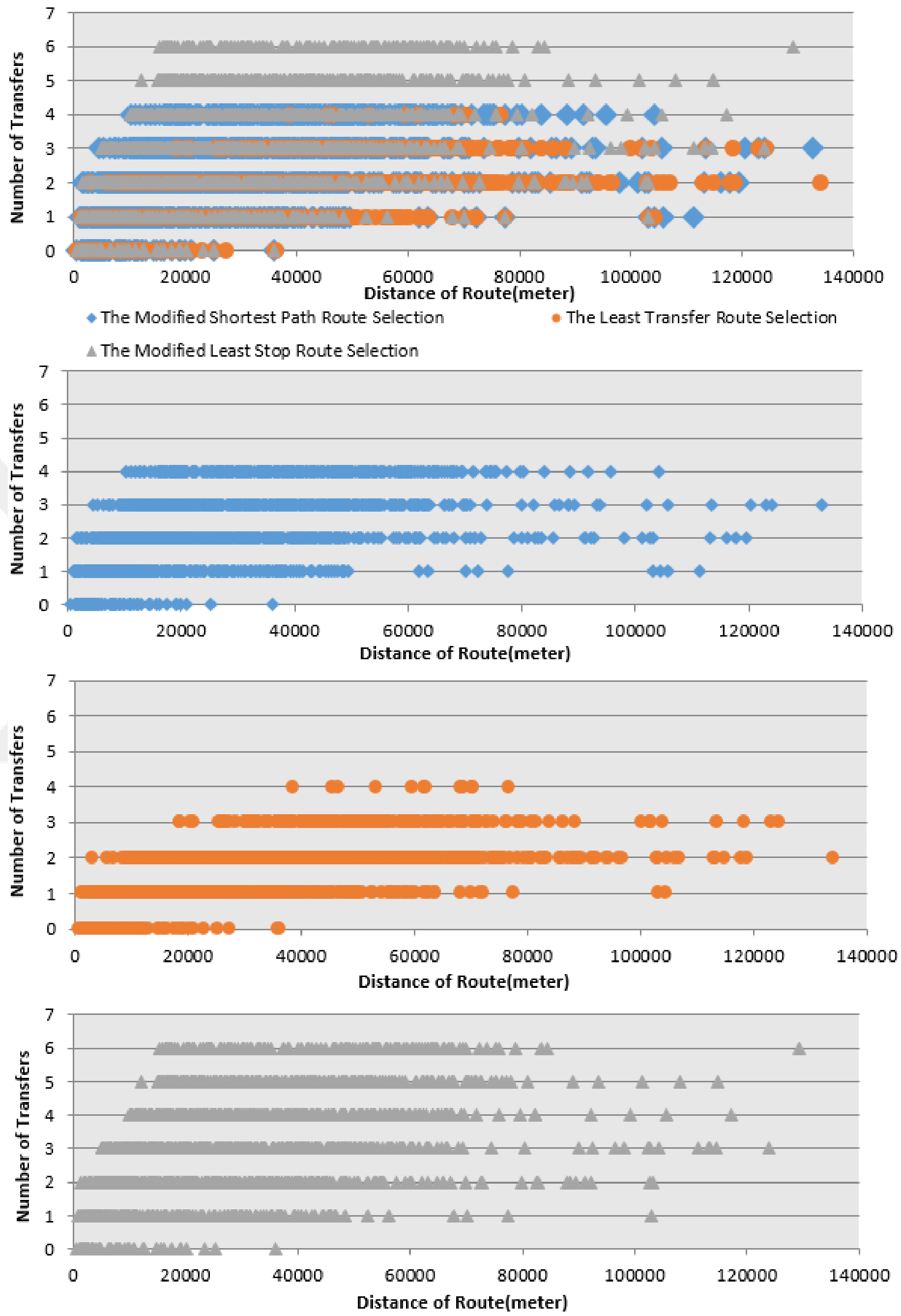

Figure 9.6: Relationship between Number of Transfers and Distance of Route

Remind that, the relationships between ***average number of transfers*** and ***average number of stops on the route*** are illustrated in Figure 9.4. For each method's route selections, relationships between the ***number of transfers*** and the ***number of stops*** on the route are represented on scatter plots in Figure 9.7. There seems to be no correlation between the number of transfers and the number of stops. However, it is observed once again that *LTRS* is the best in terms of the number of transfers where *LSRS* is the worst.

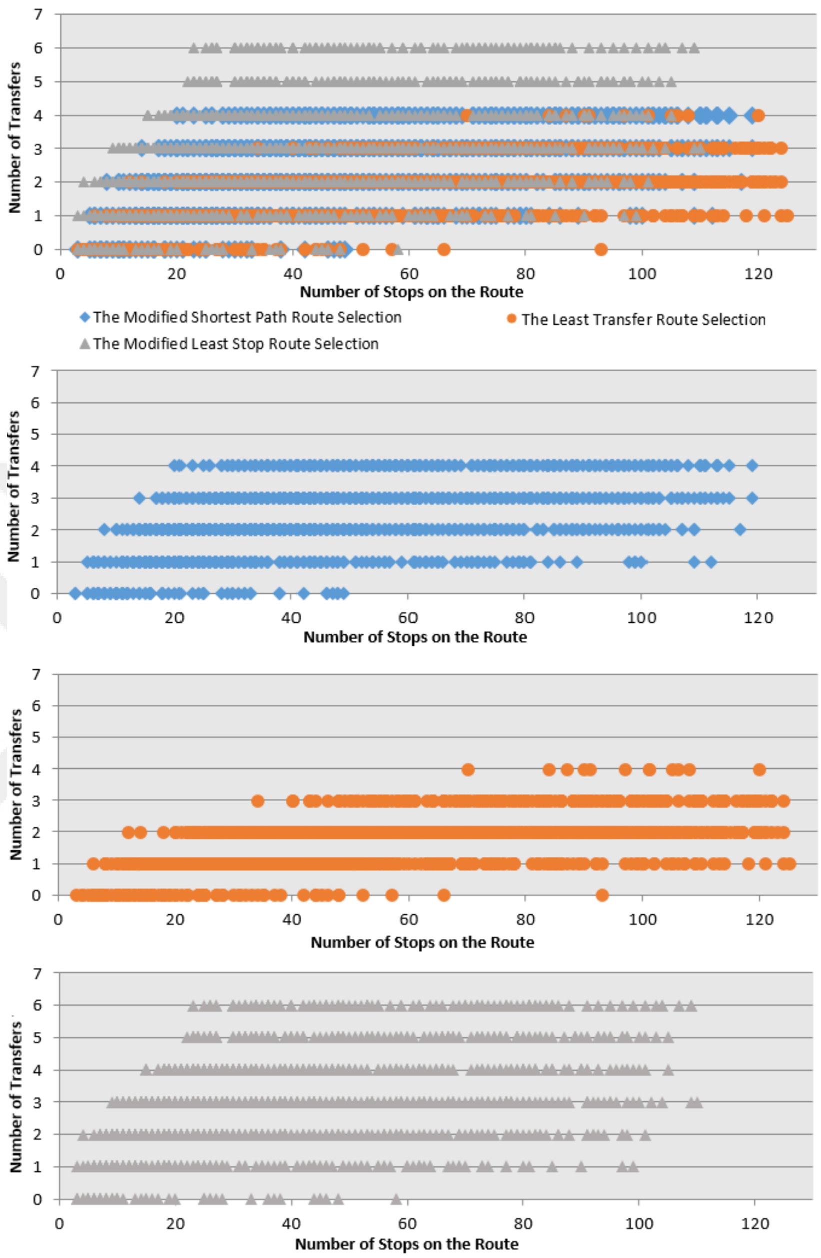

Figure 9.7: Relationship between Number of Transfers and Number of Stops on the Route

For each method's route selections, relationships between the ***distance of route*** and ***straight line distance*** from source to target on the route are represented on scatter plots in Figure 9.8. Although there are some outliers on the plot, we can say that there are strong positive linear correlations between the distance of route and straight line distance from source to target on the route, for each evaluated method. Additionally, it is observed that results of *SPRS* method is slightly better than others. Because *SPRS* method is based on the shortest path, although this method is a modified in order to minimize number of transfers and long distance walks.

Figure 9.8: Relationship between Distance of Route and Straight Line Distance

9.3 Discussion

Results of the experimental study are discussed for each method in this section.

The Least Transfer Route Selection Method is far better than other methods in terms of the number of transfers since it proposes optimal route with respect to the least transfer from the source to the target. Therefore, passengers that dislike transfers in the public transport certainly prefer route selections of this method. On the other hand, this method has the worst average in terms of the number of stops. Because, it is not preferred to transfer for using rapid-transit lines in this route selection method. Most users prefer to make one more transfer in order to use rapid-transit lines in some situations. Therefore, this method is not favorable for these types of users in these situations. Furthermore, this method is slightly worse than other methods in terms of the distance of route.

The Modified Least Stop Selection Method is significantly better than other methods in terms of the number of stops on the route. Because this method mainly uses rapid-transit lines if they are on the route. Although this method is favorable for using transit lines, it is considerably worse than the other methods in terms of the number of transfers. In some cases, using rapid-transit lines would be useful. However, there are some undesirable cases such as making two or more transfers in order to use a rapid-transit line. Therefore, using rapid-transit lines have both advantage and disadvantage for this method. Lastly, this method is quite close to the best average in terms of the distance of route.

The Modified Shortest Path Route Selection Method has the best average distance since this method is based on the shortest path algorithm. Furthermore, it is the second method in terms of the number of transfers owing to implemented penalty system (as explained in Section 6.3). Additionally, it is the second method in terms of the average number of stops on the route. Results of this method are in the middle or first place for all criteria of routes. It is not the worst method for any criteria of routes. Therefore, it would be better route selection method for users that have not certain primary criterion for route selection.

9.4 Some Route Selection Results in the Study

There are 2000 source-target pairs in the study. Since it is not possible to present all of source-target pairs, only some of these are presented in following tables for each method.

Table 9.3: The Modified Shortest Path Route Selection Method Results

Source Station	Target Station	The Distance (meter)	The Number of Stops	The Number of Transfers	Running Time (ms)
Konak Anadolu	Alsancak Gar	4,995	13	0	315
Yüksek Teknoloji	Vicdan	56,940	120	3	308
Karşıyaka İskele	Olimpiyat Köyü	14,412	35	3	303
Kehribar	Cengiz Parkı	42,176	97	4	296
Göztepe	Fahrettin Altay Aktarma Merkezi	1,849	6	0	306
Şelale Parkı	Hocazade Cami	4,593	12	0	325
Bornova Metro	Ballıkuyu Çeşme	7,621	12	1	306
Tınaz Tepe	Eşrefpaşa	7,504	21	1	302
Betontaş Lisesi	Karakuyu İlköğretim Okulu	48,342	106	4	311
Söğüt	Bahribaba	5,098	14	1	345
Belediye Sarayı	Evka 5 Sondurak 3	15,194	34	3	302
Bornova Metro	Buca Endüstri Meslek Lisesi	13,491	26	2	303
Çevik Bir	Osmangazi İlköğretim Okulu	12,901	24	2	304
Dikili Terminal	İzban-Salhane	103,077	74	1	302
Gaziemir Belediyesi	Metro-Sanayii	18,039	21	2	301
Halkapınar Metro	Adem Yavuz	2,257	10	2	302
Aliağa Organize Sanayi Yol Ayrım	Gediz Üniversitesi	45,277	35	2	306
Bornova Metro	Opel	47,161	62	2	320
Dikili Lise	Şehitler Parkı	113,304	85	2	303
Eser	Buca Doğum Evi	16,131	43	4	303
Gaziemir Aktarma Merkezi	Pazaryeri	34,670	73	3	307
İnciraltı	Özkent	67,365	111	3	304
İzban-Semt Garajı	Turgut Mezarlık	45,284	77	3	296
Narlıdere Piknik Yeri	Çam	28,050	44	2	306
Metro-Sanayii	Urla	42,886	77	3	330
Tekel Lojmanlar	Ulucak Cumhuriyet	33,971	53	4	302
Çiğli Aktarma Merkezi	Aydın	19,523	23	2	300

Table 9.4: The Least Transfer Route Selection Method Results

Source Station	Target Station	The Distance (meter)	The Number of Stops	The Number of Transfers	Running Time (ms)
Konak Anadolu	Alsancak Gar	4,995	13	0	49
Yüksek Teknoloji	Vicdan	57,693	126	2	48
Karşıyaka İskele	Olimpiyat Köyü	14,814	18	2	48
Kehribar	Cengiz Parkı	54,679	129	4	50
Göztepe	Fahrettin Altay Aktarma Merkezi	1,849	6	0	51
Şelale Parkı	Hocazade Cami	4,593	12	0	51
Bornova Metro	Ballıkuyu Çeşme	7,910	15	0	50
Tınaz Tepe	Eşrefpaşa	8,395	25	0	47
Betontaş Lisesi	Karakuyu İlköğretim Okulu	50,031	98	3	42
Söğüt	Bahribaba	5,357	13	0	52
Belediye Sarayı	Evka 5 Sondurak 3	23,565	54	1	48
Bornova Metro	Buca Endüstri Meslek Lisesi	13,773	29	1	51
Çevik Bir	Osmangazi İlköğretim Okulu	14,896	38	1	42
Dikili Terminal	İzban-Salhane	103,077	75	1	32
Gaziemir Belediyesi	Metro-Sanayii	18,340	40	1	45
Halkapınar Metro	Adem Yavuz	2,267	11	1	38
Aliağa Organize Sanayi Yol Ayrım	Gediz Üniversitesi	45,277	37	2	31
Bornova Metro	Opel	48,175	67	2	48
Dikili Lise	Şehitler Parkı	113,246	86	2	32
Eser	Buca Doğum Evi	21,472	51	2	50
Gaziemir Aktarma Merkezi	Pazaryeri	38,707	83	2	39
İnciraltı	Özkent	87,841	170	2	44
İzban-Semt Garajı	Turgut Mezarlık	48,326	116	2	36
Narlıdere Piknik Yeri	Çam	27,030	59	2	47
Metro-Sanayii	Urla	46,251	94	2	41
Tekel Lojmanlar	Ulucak Cumhuriyet	38,529	70	4	42
Çiğli Aktarma Merkezi	Aydın	23,767	39	1	34

Table 9.5: The Modified Least Stop Route Selection Method Results

Source Station	**Target Station**	**The Distance (meter)**	**The Number of Stops**	**The Number of Transfers**	**Running Time (ms)**
Konak Anadolu	Alsancak Gar	7,182	9	1	1
Yüksek Teknoloji	Vicdan	61,975	111	4	1
Karşıyaka İskele	Olimpiyat Köyü	13,852	8	2	1
Kehribar	Cengiz Parkı	42,960	92	4	1
Göztepe	Fahrettin Altay Aktarma Merkezi	1,979	6	0	1
Şelale Parkı	Hocazade Cami	4,593	12	0	1
Bornova Metro	Ballıkuyu Çeşme	13,544	11	2	1
Tınaz Tepe	Eşrefpaşa	7,548	17	3	1
Betontaş Lisesi	Karakuyu İlköğretim Okulu	49,669	90	4	1
Söğüt	Bahribaba	4,990	11	1	1
Belediye Sarayı	Evka 5 Sondurak 3	16,740	25	3	1
Bornova Metro	Buca Endüstri Meslek Lisesi	12,970	17	3	1
Çevik Bir	Osmangazi İlköğretim Okulu	14,419	18	3	1
Dikili Terminal	İzban-Salhane	103,077	73	1	1
Gaziemir Belediyesi	Metro-Sanayii	20,215	10	2	1
Halkapınar Metro	Adem Yavuz	2,256	8	2	1
Aliağa Organize Sanayi Yol Ayrım	Gediz Üniversitesi	45,277	35	2	1
Bornova Metro	Opel	47,837	56	4	1
Dikili Lise	Şehitler Parkı	113,196	81	3	1
Eser	Buca Doğum Evi	15,903	25	4	1
Gaziemir Aktarma Merkezi	Pazaryeri	38,392	38	4	1
İnciraltı	Özkent	62,810	84	5	1
İzban-Semt Garajı	Turgut Mezarlık	58,169	98	3	1
Narlıdere Piknik Yeri	Çam	29,572	35	3	1
Metro-Sanayii	Urla	50,273	84	6	1
Tekel Lojmanlar	Ulucak Cumhuriyet	37,513	44	6	1
Çiğli Aktarma Merkezi	Aydın	20,667	19	1	1

Chapter 10

Conclusion

In this chapter, the thesis is summarized in Section 10.1. Additionally, the direction for the future researches is explained in the last section.

10.1 Summary

To briefly summarize the thesis; firstly, the public transportation network and the route selections in this network are introduced. Afterwards, route selection criteria for the public transport are discussed, and the route selection criteria to be evaluated in this study are determined. These evaluated criteria are named as; "the shortest path", "the least transfer" and "the least stop" routes. Basic definitions of the graph theory, graph types and graph data structures are explained for our public transport network model and our route selection methods. After that, the public transport network is modeled in order to evaluate the route selection methods with respect to our criteria, and to describe the cost functions of the routes. Therefore, route selection methods named as; The Modified Shortest Path Route Selection Method, The Modified Least Stop Route Selection Method and The Least Transfer Route Selection Method are evaluated with respect to our criteria. These route selection methods are experimented on a real world public transport network. Their route selections' average results are compared in terms of the running time, the distance of the routes, the number of the transfers and the number of the stops on the routes. Thus, strengths and shortcomings of evaluated methods are observed.

10.2 Future Work

For our future researches, it is aimed to develop a public transport route selection application that creates a specific characterization for each user. Thus, it is aimed to propose an ideal route selection specific to the user with respect to his/her characterization.

Chapter 11

Bibliography

[1] Nasiboglu, E., Bozyigit, A., and Diker, Y. 2015. *Analysis and Evaluation Methodology for Route Planning Applications in Public Transportation*, in Application of Information and Communication Technologies (AICT): 2015 9th International Conference on IEEE, 477-481.

[2] Dijkstra, E. W. 1959. *A Note on Two Problems in Connexion with Graphs*. Numerische mathematik, Vol. 1(1): 269-271.

[3] Li, Z. W. and Zhu, M. 2010. *Breadth-First Search Based Bus Transport Transfer Algorithm*, in 9th WSEAS Int. Conference Applied Computer and Applied Computational Science, 106-110.

[4] Huang, Z., Liu, X., and Li, J. 2009. *Information Needs of Urban Transit Travelers—The Case of Wuhan, China*, in ICCTP 2009: Critical Issues in Transportation Systems Planning, Development, and Management, Harbin, China, 1-7.

[5] Meng, F. H., Lao, Y., Wai, L. H., and Chuin, L. H. 1999. *A multi-criteria, multi-modal passenger route advisory system*.

[6] Pun-Cheng, L. S., and Chan, A. W. 2016. *Optimal route computation for circular public transport routes with differential fare structure*. Travel Behaviour and Society, Vol. 3: 71-77.

[7] Nasiboğlu, E., and Berberler, M. E. 2016. *Toplu Taşımada En Az Aktarma Kriterine Göre Seyahat Planlama İçin Matematiksel Model*. Türkiye Bilişim Vakfi Bilgisayar Bilimleri Ve Mühendisliği Dergisi, Vol. 8(2).

[8] Girnau, G., and Blennemann, F. 1989. *Cost-Benefits in Underground Urban Public Transportation*. Tunneling and Underground Space Technology, Vol. 4(1): 23-30.

[9] Luke, S., and MacDonald, M. 2006. *Public Transport Mode Selection: A Review of International Practice*, in European Transport Conference, Strasburg, France.

[10] Laporte, G., and Mesa, J. A. 2015. *The Design of Rapid Transit Networks, in G.* Laporte, S. Nickel, and F. Saldanha da Gama, ed., Location science. New York: Springer International, 581-594.

[11] Wang, H., Hu, M., and Xiao, W 2010. *A New Public Transportation Data Model and Shortest-Path Algorithms*, in Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on IEEE, Vol. 1: 456-459.

[12] Xu, X., Yin, Y., and Liu, L. 2016. *Improved Dijkstra's Algorithm and its Application in Intelligent Transportation System*. Journal of Residuals Science and Technology, Vol. 13(7).

[13] Yan, X., and Niu, X. 2005. *Study on Heuristic Algorithm for Public Transport Network Multi-Path Selection.* Urban Transport of China, Vol. 3.

[14] Jian-lin, W. A. N. G. 2005. *The Public Transportation Optimum Route Algorithm Based on The Least Transfer [J]*. Economic Geography, Vol. 5: 673-676.

[15] Wu, Q., and Hartley, J. 2004. *Using K-Shortest Paths Algorithms to Accommodate User Preferences in The Optimization of Public Transport Travel*, in Applications of Advanced Technologies in Transportation Engineering, 181-186.

[16] Ferreira, J. C., Filipe, P., and Silva, A. 2011. *Multi-modal transportation advisor system*, in Integrated and Sustainable Transportation System (FISTS): 2011 IEEE Forum on, Vienna, 388-393.

[17] Goel, N. et al. 2017. *Optimal routing algorithm for large scale multi-modal transit networks*.

[18] Nguyen, D., and MacDonald, T. 2010. *TDplanner: Public transport planning system for mobile devices*. Department of Mathematics and Computer Science, Suffolk University, Boston, Tech. Rep.

[19] Biswas, S. S. 2017. *Fuzzy Real Time Dijkstra's Algorithm*. International Journal of Computational Intelligence Research, Vol. 13(4): 631-640.

[20] Zhou, Y., Thill, J. C., and Huang, Z. 2011. *Design of a user-centric decision support tool for fixed-route bus travel planning*. Applied Geography, Vol. 31(3), 1173-1184.

[21] Lee, C. Y. 1961. *An Algorithm for Path Connections and Its Applications*. IRE transactions on electronic computers, Vol. 3: 346-365.

[22] Kurant, M. and Thiran, P. 2006. *Extraction and analysis of traffic and topologies of transportation networks[J]*. PHYSICAL REVIEW. E., Vol. 74(3):036114.

[23] Sienkiewicz, J. and Holyst, J.A. 2005. *Statistical analysis of 22 public transport networks in Poland[J]*. PHYSICAL REVIEW. E., Vol. 72(4): 046127.

[24] Chen, Y. Z., Li, N., and He, D. R. 2007. *A study on some urban bus transport networks*. Physica A: Statistical Mechanics and its Applications, Vol. 376: 747-754.

[25] Zhang, X. et al. 2016. *Modeling and analysis of bus weighted complex network in Qingdao city based on dynamic travel time*. Multimedia Tools and Applications, Vol. 75(24): 17553-17572.

[26] Wang, B. and Yang, X. H. 2010. *A Transfer Method of Public Transport Networks by Adjacency Matrix,* in Proceedings of the 2010 international conference on Applied computing conference, 123-127.

[27] Gao, Y. 2008. “Finding Least-Transfer Path in Multi-Route/Modal Public Transit System”. PhD Thesis. Nanjing University of China.

[28] Li, Y., Xing, J., and Huang, G. 2010. *Least Transfer Cost Model for Optimizing Public Transport Travel Routes,* in Signal Processing Systems (ICSPS): 2010 2nd International Conference on IEEE, Vol. 2: 828-831.

[29] Böhmová, K. et al. 2016. *Computing and Listing St-Paths in Public Transportation Networks,* in International Computer Science Symposium, Russia, 102-116.

[30] Biggs, N., Lloyd, E. and Wilson, R. 1986. *Graph Theory*. Oxford University Press.

[31] Bridges of Konigsberg. 2017. URL: <http://www-history.mcs.st-andrews.ac.uk/ Diagrams/Konigsberg_colour.jpeg>. [27 October 2016]

[32] Brummelen, G. R. 2013. *Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry.* Princeton University Press.

[33] CodePlex, 2016. GMap.NET - Great Maps for Windows Forms & Presentation. URL: <https://greatmaps.codeplex.com>. [Jul 10, 2016].

[34] ESHOT, 2016. Durak Bilgileri. URL: <http://www.eshot.gov.tr>. [March 10, 2016].