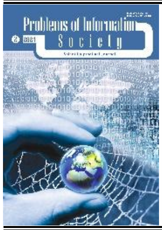

www.jpis.az

15 (2)
2024

Comparative analysis of the optimal route recommendation models based on city public transportation network data

Resmiye Nasiboglu[1], Mohd S. Rusiman[2], M. Halil Ibrahim Koroglu[3], Efendi Nasibov[4]

[1,4]Dokuz Eylul University, Kültür, Cumhuriyet Blv No:35210 no 144, 35220, Izmir, Türkiye
[2]University Tun Hussien Onn Malaysia, Persiaran Tun Dr. Ismail, 86400 Parit Raja, Johor, Malaysia
[3]Turkan Yamanturk School, Ayrancı, Reşat Nuri Cd No:39, 06540 Çankaya/Ankara, Türkiye

[1]resmiye.nasiboglu@deu.edu.tr, [2]saifulah@uthm.edu.my, [3]mhik_06@hotmail.com, [4]efendi.nasibov@deu.edu.tr

[1]*orcid.org/0000-0003-1739-1469*, [2]*orcid.org/0000-0001-8255-9884*, [3]*orcid.org/0009-0009-0773-8706*, [4]*orcid.org/0000-0002-1889-6410*

ARTICLE INFO

http://doi.org/10.25045/jpis.v15.i2.02

Article history:
Received 09 January 2024
Received in revised form
10 March 2024
Accepted 15 May 2024

Keywords:
Public transportation
Optimal path
Dijkstra algorithm
Smart card data
Fuzzy model

ABSTRACT

In recent years, studies on smart cities have attracted much attention in the literature. In the smart city concept, urban public transportation network studies are one of the most important issues. After the 2000s, smart cards have been widely used in city public transportation systems. Using the data collected through the smart card, many models have been created in line with the smart city concept. In the literature, there are various optimal route recommendation models using urban public transportation network and city smart card data. In this study, some current models built on the basis of urban public transportation network and smart card data are discussed. The specifications of the models, their important differences, the topologies they use, optimization criteria, and computational complexities are analyzed. Dijkstra's algorithm, which is widely used for the solution of optimal route recommendation models, and its various modifications are analysed. Additionally, various models developed by applying fuzzy logic are examined. Comparative analysis of public transportation network models is given.

1. Introduction

Recently, with the spread of the smart city concept, the importance of various models created on the city public transportation network has also increased. One of the most notable of these models is the optimal route recommendation models, which calculate the best way to get from one location of the city to another by public transportation.

Optimization models based on urban public transport networks have long been included in the scientific literature (Ceder, 2007; Chen et al., 2007; Ferber et al., 2007, 2009; Marc, 2011; Sienkiewicz and Hołyst, 2005; Zhang et al., 2013a,b). Various topologies of the public transport network are used in these models. These topologies are mainly P-space, C-space, B-space and L-space topologies and their multigraph variants P'-space, C'-space, B'-space and L'-space topologies, respectively (Ferber et al., 2009; Hui et al, 2007; Marc, 2011; Panagiotis and Fisk, 2006; Sienkiewicz and Hołyst, 2005; Sui et al., 2019; Vito and Marchiori, 2001; Xu et al., 2013; Zhang et al., 2013; Zhu et al., 2008). Classic models built on the urban public transportation network (PTN) are models based on the static structure of PTN. However, recently smart electronic cards have been used in big cities for automatic fare allocation. In addition to collecting fares, many other instant data such as boarding time, boarding coordinates, line and stop information can also be collected through these cards. A variety of studies based on data mining have been conducted on these data (Bagchi and White, 2005; Barry et al., 2002; Munizaga and Palma, 2012; Nasiboglu et al., 2012; Nasibov et al., 2016; Pelletier et al., 2011; Trepanier, 2007).

In this study, various optimal route recommendation models on the urban public transportation network are discussed and their differences, the topologies they use, the fuzzy or heuristic approaches they use, the use of smart card data, and their computational complexity are comparatively analyzed. In the rest of the study, in section 2, other studies that form the basis of this study are mentioned. In section 3, the network topologies and various modeling approaches used by urban PTN models are given. In section 4, solution algorithms are discussed. Comparative analysis and discussion of various models is given in section 5. To improve readability, the results are presented in tab form. Finally, in the Conclusion section, general evaluations are made and hints are given about future studies.

2. Related work

One of the most notable urban PTN models is study based on optimal route calculation between various locations of the city (Schrijver, 2012). A popular approach used in optimal route planning studies is mathematical-heuristic approaches (García-Heredia et al., 2021; Long and Tan, 2020; Lopez and Lozano, 2014). For example, in the study (Long and Tan, 2020), an improved genetic algorithm is used to find the optimal route. Model constraints such as starting and ending point walking distance, transfer distance, driving distance and cost are taken into account. Based on the optimal recursive selection properties of the developed genetic algorithm, the optimal path selection model of the public transportation network is created. In the study (García-Heredia et al., 2021), the problem of finding the shortest path between two specific nodes for each network within a collection is addressed. This problem is defined as an integer programming problem. A parallelizable heuristic approach is proposed for the solution, which consists of three stages: generation of feasible solutions, combination of solutions, and improvement of the solution. In the study (Lopez and Lozano, 2014), various techniques used to find the shortest paths in multimodal transportation networks are examined.

Fuzzy models of the public transport network have also been discussed by various authors in the literature (Deng et al., 2012; Ji et al., 2007; Mahdavi et al, 2009; Nasibov et al, 2016; Nasiboglu, 2021, 2022). In the study (Ji et al., 2007), the fuzziness of the model arises from the fact that the distance between stops is treated as fuzzy numbers. In the study, three types of models are formulated and a hybrid heuristic approach integrating simulation and genetic algorithm is proposed to solve these models. In (Deng et al., 2012), a generalized Dijkstra algorithm that can work on a graph with fuzzy edge lengths is proposed. To solve problems such as collecting fuzzy edge lengths and comparing fuzzy lengths, the concept of the graded mean integration representation of fuzzy numbers and the fuzzy algorithm, which is an expansion of the classical Dijkstra algorithm, are proposed.

In the study (Nasibov et al, 2016), fuzziness is discussed from a different perspective. The fuzzy stop accessibility degree proposed in this study is calculated depending on factors such as walking distance to the stops, mobility density of the stop, and the number of lines passing through the stop. In addition to these features, the stop-line connection degree is also recommended. As a result, the route with the highest degree of connection between the source and destination stops is calculated. In the studies (Nasiboğlu, 2021, 2022), the concept of line-based fuzzy connection degree is developed. In these studies, the connection degree of the line is based on the bus crowding index on the line using smart card data.

Among the optimal route calculation algorithms on the public transport network, algorithms based on the Dijkstra algorithm have additional importance (Bozyiğit et al. 2017,2018; Dijkstra, 1959; Khaing et al. 2018; Nasiboglu, 2022; Ray et al., 2022; Tirastittam and Waiyawuththanapoom 2014). In Dijkstra-based algorithms, the optimal route calculation is generally made between any two specific stops in the city using P-space topology. The stops of the public transportation network are considered as the vertices of the graph, and if there is a line between any two stops, there is an edge between these vertices. The weights of the edges are considered as the distance between the stops.

In the study (Khaing et al. 2018), Dijkstra Algorithm is applied to calculate the shortest path on Yangon city bus network. GIS data is used to realize the algorithm. In the study (Bozyiğit et al., 2017), the length of the line between the stops is considered as the number of stops on the route between these stops. Additionally, penalty points are applied when transferring from one line to another on the route. Thus, when calculating the optimal route between source and target stops, not only the shortest route but also the route with the least number of line transfers comes to the fore. In the study (Bozyiğit et al., 2018), in addition to the previous study, a modified Dijkstra algorithm is proposed, which minimizes the number of walks between stops. In the algorithm, penalty points are applied for each line change on the route and for each consecutive walk to more than one stop during these line changes. In the study (Nasiboglu, 2022), a model based on the Dijkstra algorithm is proposed to produce routes with the shortest path, the least transfer criteria and the highest degree of accessibility. Concepts such as fuzzy stop ratings, fuzzy inner-bus density, fuzzy line accessibility are used to calculate the accessibility degrees of the routes. In the proposed algorithm, fuzzy penalty function is used to produce routes with higher connectivity degrees.

3. Network topologies and modeling approaches

In this section, the topologies used in urban public transportation networks and optimal route recommendation models will be discussed in details.

3.1. Network topologies

Naturally, urban public transport networks are modeled as a graph. In these graph models, L-space, P-space, C-space, B-space topologies and their multigraph approaches, L'-space, P'-space, C'-space topologies, are used. Among these topologies, P-space, P'-space, L-space and L'-space topologies are frequently used topologies. In P-space topology, stops are considered as vertices of the graph, and if there is any direct line between two stops, there is an edge between these stops. In P'-space topology, the multigraph structure is used by considering the different lines between the stops as separate edges. In L-space topology, the lines are given as a list of the stops on them or as a line-stop relationship matrix. In B-space topology, stops and lines are two disjoint vertex sets of a bipartate graph. If there is a relationship between the line and the stop, edges are drawn between suitable nodes. In C-space topology, the vertices of the graph show the lines. If there is a common stop between two lines, edges are drawn between these nodes. It should be noted that, while P-space topology on a directed graph is generally used in road recommendation problems for private vehicles, P'-space topology on a directed graph is used in PTN models. Because, in PTN problems, there are different lines between the stops and this line difference affects the solution.

3.2. P'-space topology-based model

In some studies, the relationship matrix between lines and stops is in the form of a square matrix containing stop to stop rows and columns. This model fits the P'-space model. In this matrix, the list of lines running between stop i and stop j at the intersection of any (i, j) row and column is given. For example, (Bozyigit et al., 2017, 2018) use a matrix as follows in their studies:

$$L = \begin{bmatrix} \emptyset & l_{1,2} & l_{1,3} & \dots & l_{1,n} \\ l_{2,1} & \emptyset & l_{2,3} & \dots & l_{2,n} \\ & & \dots \dots \dots & & \\ l_{n,1} & l_{n,2} & \dots & l_{n,n-1} & \emptyset \end{bmatrix}, \quad (1)$$

Here, $l_{i,j}$ is the list of lines passing between the stop i and the stop j. If there is no line between two stops, this list will be empty. The additional matrix $W: n \times n$ from matrix L is also used and defined as follows:

$$W = \begin{bmatrix} 0 & w_{1,2} & w_{1,3} & \dots & w_{1,n} \\ w_{2,1} & 0 & w_{2,3} & \dots & w_{2,n} \\ & & \dots \dots \dots & & \\ w_{n,1} & w_{n,2} & \dots & w_{n,n-1} & 0 \end{bmatrix} \quad (2)$$

Here, $w_{i,j}$ value is the number of stops between stops i and j. If there is no line between any pair of stops, this value is marked as INFINITY. In more detailed studies, this matrix can also be defined on a line basis. That is, instead of the W matrix, a separate W^l matrix can be defined for each $l \in L$ line. The cost value of the route for each stop pair (i, j) is calculated through the matrix $W: n \times n$ in the classical Dijkstra algorithm.

Let's mark the route from a specific origin stop s_O to the destination stop s_D with π_{s_O,s_D}. Let the lines used in this route be $l_i, i = 1, \dots, m$. Thus, the total length of the route from the origin point to the destination will be as follows:

$$w(\pi_{s_O,s_D}) = \sum_{t=1}^{m} w(l_i), \quad (3)$$

where $w(l_i)$ denotes the length of the route segment traveled by the corresponding line l_i. The W matrix is used to calculate these lengths. In most optimal route recommendation models, the model is based on this length and this path length is considered the most important criterion. In the classical Dijkstra algorithm, the shortest route between the start and destination stops is calculated taking into account the following criteria:

$$w(\pi_{s_O,s_D}) \longrightarrow min \quad (4)$$

In the study (Bozyiğit et al., 2017), it is suggested to replace the Dijkstra algorithm by using penalties. As the input of the algorithm, the weight matrix $W: n \times n$, which holds the length of the route between the stops for each (i, j) stop pair, and the matrix $L: n \times n$, consisting of the lines providing this route, are given. The optimization criterion used in the modified Dijkstra algorithm can be expressed as follows:

$$c(\pi_{s_O,s_D}) = w(\pi_{s_O,s_D}) + w\_Penalty + t\_Penalty \longrightarrow min, \quad (5)$$

where $c(\pi_{s_O,s_D})$ is the cost value of the route, and $w\_Penalty$ is the penalty per transfer used to minimize the number of transfers between lines. $t\_Penalty$ is the penalty per walk used to minimize uninterrupted repetitive pedestrian walks between the stops.

3.3. Discrete programming-based model

The data structures used in PTN optimal route recommendation models vary. In some studies, the relationship matrix between lines and stops is used. For example, in the study of (Nasibov et al., 2016), the relationship matrix between the line and the stops is taken as a basis, in accordance with the L'-space model. In this matrix, there is a line in each row and a separate column for each stop. In the cell at the row-column intersection, the number of the stop on this line is given. If any line does not pass through any stop, the value 0 is included in the appropriate intersection cell.

To determine the degree of preference of a stop, taking into account the walking distance between stops, definitions as follows are used in the study (Nasibov et al., 2016).

The fuzzy neighborhood relationship between any two stops $s_i,\ s_j \in S$ is defined as follows:

$$\mu_{s_i}(s_j) = max\left\{0,\ 1 - \frac{d(s_i,s_j)}{d_{walk}}\right\}, \quad (6)$$

where d_{walk} parameter specifies the maximum reasonable walking distance, $d(s_i, s_j)$ is the distance between the stops s_i and s_j.

The fuzzy set of neighboring stops of any stop is defined as follows:

$$N_s = \{(\mu_s(s')/s')\ |s' \in S\}. \quad (7)$$

In this case, the membership degree of any stop s' to the fuzzy set N_s is shown as follows:

$$N_s(s') = \mu_s(s'). \quad (8)$$

For a given level $\gamma \in [0,1]$, the γ-level set of the fuzzy set N_s is defined as follows:

$$N_s^\gamma = \{s' \in S : N_s(s') \geq \gamma\}. \quad (9)$$

The set N_s^γ is also called the γ-neighbour set of stop s. Naturally, with the degree of γ, there is an ε distance between stops corresponding to this degree:

$$\{s' \in S : N_s(s') \geq \gamma\} \equiv \{s' \in S : d(s, s') \leq \varepsilon\}. \quad (10)$$

In some studies, whether transfer between lines is possible is determined by the existence of exactly common stops of the lines. However, in the study of (Nasibov et al., 2016), the concept of fuzzy neighboring stops was proposed to ensure the transition between lines that do not have a common stop between each other. This concept was used in the formulation of the fuzzy problem and the creation of the heuristic search algorithm.

A fuzzy neighbor line to any line l is any other line, passing through any fuzzy neighbor stop belonging to that line l. Lines that are not directly connected to each other are connected using the concept of γ-neighborhood of a line and in the relevant case, the passenger can reach the destination with some transfers. Thus, for any given origin-destination stops s_O and s_D, the fuzzy multi-criteria route planning problem can be stated as a decomposition into γ-levels as follows:

$$\max : \gamma, \quad (11)$$

$$\min : |\pi_{s_O s_D}| = \sum_{t=1}^{m} |(s_{i_t}, r_{j_t}, s_{k_t})|, \quad (12)$$

s.t.:

$$RS(r_{j_t}, s_{i_t}) \geq 1,\ t = 1, \dots, m, \quad (13)$$

$$RS(r_{j_t}, s_{k_t}) > RS(r_{j_t}, s_{i_t}),\ t = 1, \dots, m. \quad (14)$$

$$\mu_{s_{k_{t-1}}}(s_{i_t}) \geq \gamma,\ t = 2, \dots, m; \quad (15)$$

$$s_{i_1} = s_O, \quad (16)$$

$$s_{k_m} = s_D, \quad (17)$$

$$\gamma \in (0,1], \quad (18)$$

Here, m is the number of lines used on the proposed route between s_O and s_D stops. $|(s_{i_t}, r_{j_t}, s_{k_t})|$ is the number of stops between s_{i_t} and s_{k_t} stops when the r_{j_t} line is used. While criterion (11) refers to choosing among possible routes with as high a degree of preference as possible, (12) indicates choosing shorter routes possible in terms of the number of stops. Constraints (13)-(14) are the condition that every link on the route is a possible link. Constraints (15)-(18) enable a route to form a fuzzy γ-connection.

The solution of the problem (11)-(18) ensures that the shortest possible routes connecting the selected origin and destination stops s_O and s_D are selected among highly preferable fuzzy routes. To increase practicality, the proposed solution results can be sorted in descending order so that only the first q set of alternative routes Π_{XY}^q can be considered:

$$\Pi_{XY}^q = \left\{ \begin{matrix} \pi_{s_X s_Y}^{(1)}, \dots, \pi_{s_X s_Y}^{(q)} \mid \pi_{s_X s_Y}^{(i)} \succ \pi_{s_X s_Y}^{(i+1)}, \\ i = 1..q-1 \end{matrix} \right\}, \quad (19)$$

where $\pi_{s_X s_Y}^{(i)} \succ \pi_{s_X s_Y}^{(i+1)}$, indicates that the path $\pi_{s_X s_Y}^{(i)}$ is preferred over the path $\pi_{s_X s_Y}^{(i+1)}$. The non-dominant set of solutions of the multi-criteria problem (11)-(18) is in the pareto-optimal solution space of the problem. This problem can be solved using various approaches such as the lexicographic ordering of criteria, the weighting of criteria, etc.

3.4. Inner-bus density-based model

In the studies by Nasiboglu (2021, 2022), fuzzy models that take into account passenger density inside the bus using smart card data are proposed.

We can define these models as follows. Let the set of all stops be labeled as S and the set of all lines be labeled as L. In this case, each line $l \in L$ can be given as an ordered sequence of the stops $s \in S$ through which it passes. We can define this as a line-stop relationship as follows:

$$LS(l,s) = \begin{cases} i, & \text{the } stop\ s\ is\ the\ i.th\ stop\ on\ the\ line\ l, \\ 0, & the\ stop\ s\ is\ not\ on\ the\ line\ l. \end{cases} \quad (20)$$

The set of stops that a certain line l passes through, is denoted as follows:

$$S(l) = \{s: LS(l,s) \geq 1\} \quad (21)$$

A triple (s_b, l, s_a) that satisfies the following conditions is called a possible connection between the boarding stop s_b and the alighting stop s_a using the line l:

$$s_b \in S(l) \quad (22)$$

$$s_a \in S(l) \quad (23)$$

$$LS(l, s_b) < LS(l, s_a) \quad (24)$$

In other words, a triple (s_b, r, s_a) is a possible connection if it satisfies the following conditions:

$$LS(l, s_b) \geq 1 \quad (25)$$

$$LS(l, s_b) < LS(l, s_a) \quad (26)$$

Provided that the triple (s_b, l, s_a) is a possible connection, the set of stops between s_b and s_a on the line l will be marked $S(s_b, l, s_a)$.

Let us denote $\mu(l; s)$, the occupancy rate of a bus departing from stop s on line l. This also can be called as the inner-bus fuzzy density degree. The degree is calculated as the ratio of the number of passengers in the bus to the maximum number of passengers allowed for this bus type, i.e.:

$$\mu(l;s) = \begin{cases} \frac{p_i}{p_{max}}, & if\ p_i \leq p_{max}, \\ 1, & d.d. \end{cases} \quad (27)$$

Here, p_i is the number of passengers inside the bus when starting from the bus stop i, and p_{max} is the maximum number of passengers allowed for this bus model. p_i values are obtained from statistics based on city smart card data.

The degree of congestion inside the bus is taken into account for all possible connections. The concept of fuzzy possible connection accepts that the preference of a line is realized not with the classical 0/1 logic, but with the degree of fuzzy line preference. Clearly, the line preference degree is inversely proportional to the inner-bus density degree. So, the fuzzy line preference degree of some possible connection (s_b, l, s_a) is determined as follows:

$$\mu^{Line}(s_b, l, s_a) = 1 - \mu^{Density}(s_b, l, s_a) \quad (28)$$

As can be seen from the formula (31), in the case where the bus is fully occupied, the fuzzy line preference degree of the possible connection (s_b, l, s_a) will be 0, therefore this situation will definitely not take place in the solution space. However, this situation may be preferred by some passengers in real life. In this regard, a parametric formula as follows can be used instead of formula (28) where the degree of preference will take value greater than 0 for any $w > 1$ parameter (Nasiboglu, 2022):

$$\mu^{Line}(s_b, l, s_a) = \frac{1}{(1+\mu^{Density}(s_b,l,s_a))^w} \quad (29)$$

Let $s_1, s_2 \in S$ be any two stops. The fuzzy neighborhood relationship of stop s_2 with respect to stop s_1 can be defined through the membership function $\mu_{s_1}(s_2) \in [0,1]$. In some cases, this membership function can be used as an exact relation defined with respect to a certain set of γ-levels, as stated in formula (30):

$$\mu^{\gamma}_{s_1}(s_2) = \begin{cases} 1, & if\ \mu_{s_1}(s_2) \geq \gamma, \\ 0, & \text{otherwise.} \end{cases} \quad (30)$$

In cases where the preference degrees of all stops are considered the same within a certain $d_{\max}$ walking distance radius, the neighborhood relationship between stops can be defined as follows:

$$\mu_{s_1}(s_2) = \begin{cases} 1, & if\ d(s_1, s_2) \leq d_{\max}, \\ 0, & \text{otherwise.} \end{cases} \quad (31)$$

A possible route with m possible connections from a given origin stop s_O to the destination stop s_D will be denoted as follows:

$$\pi_{s_O,s_D} =< (s_{b_1}, l_1, s_{a_1}), (s_{b_2}, l_2, s_{a_2}), \ldots, (s_{b_m}, l_m, s_{a_m}) > \quad (32)$$

Obviously, for this route to be preferable, the following conditions will need to be met:

$$s_{b_1} = s_O \quad (33)$$

$$s_{a_m} = s_D \quad (34)$$

$$\mu^{line}(s_{b_t}, l_t, s_{a_t}) > 0 \quad (35)$$

$$\mu_{s_{a_{t-1}}}(s_{b_t}) > 0, \quad (36)$$

The representation $\mu_{s_{a_{t-1}}}(s_{b_t}), t = 2, \ldots, m,$, in the formula (36) above indicates that the last stop of the possible connection number $(t-1)$ on the route is $s_{a_{t-1}}$ and s_{b_t}, which is the first stop of the next possible connection, show that there is a walking distance between them. Equations (33) and (34) guarantee the origin and destination stops be equal to the first and last stops of the route, respectively. It is clear that the route π_{s_O,s_D} consisting of m possible connections contains $(m-1)$ line transfers.

The fuzzy preference degree of the possible route can be defined as the minimum of the preference degrees of the possible connections that form it, as stated in the formula (37):

$$\mu(\pi_{s_O,s_D}) = \left[\wedge_{t=1}^{m} \mu^{Line}(s_{b_t}, l_t, s_{a_t})\right] \quad (37)$$

Thus, the multi-criteria optimal route recommendation problem, created to find the optimal route with the least passenger density from a certain stop s_O to the s_D stop, is defined as follows:

$$\mu(\pi_{s_O,s_D}) \to max \quad (38)$$
$$|\pi_{s_O,s_D}| = \sum_{t=1}^{m} |(s_{b_t}, l_t, s_{a_t})| \to min \quad (39)$$
$$m \to min \quad (40)$$

s.t.:

$$s_{b_1} = s_O \quad (41)$$
$$s_{a_m} = s_D \quad (42)$$
$$LS(l_t, s_{b_t}) \geq 1, t = 1, \dots, m, \quad (43)$$
$$LS(r_t, s_{a_t}) > LS(r_t, s_{b_t}), t = 1, \dots, m, \quad (44)$$
$$s_{b_t}, s_{a_t} \in S,\ t = 1, \dots, m, \quad (45)$$
$$l_t \in L,\ t = 1, \dots, m \quad (46)$$

Criterion (38) above will ensure that the route has as high a degree of preference as possible, criterion (39) will ensure that the shortest route as possible among possible routes is selected, and criterion (40) will ensure that the number of transfers along the route is minimized. Thus, the problem defined by the objective criteria (38)-(40) is a multi-criteria decision-making problem. In this case, the solution space of the problem will consist of Pareto optimal solutions. The decision variables of the problem (38)-(46) are the stops specified in formula (45) and the lines specified in formula (46). Thus, the solution to the problem will be sought in the discrete cartesian product space $S \times L \times S$ for each possible connection. The general solution will be in $m \times S \times L \times S$ space, with the total number of m possible connections on the route.

4. Algorithms

Optimal route recommendation algorithms on the urban public transport network are generally based on the Dijkstra's algorithm logic. Dijkstra's algorithm was first proposed by Dijkstra (1959) and then developed by many researchers (Bozyiğit et al., 2017,2018; Khaing et al., 2018; Nasiboglu, 2022; Ray et al., 2022; Tirastittam and Waiyawuththanapoom, 2014). The classical Dijkstra algorithm is an algorithm that finds the shortest path between vertices in an undirected graph, taking into account the edge weights. Modifications of Dijkstra's algorithm have been discussed in various studies. In the studies (Bozyiğit et al., 2017, 2018), a Modified Dijkstra algorithm is proposed that, in addition to the shortest path, also minimizes the number of transitions between lines. The pseudocode of this algorithm is given in Figure 1. In the algorithm, the W and L matrices specified previously in the section 3.2 are used.

```
function ModifiedDijkstra(s,L,W):
Input: s - a given stop; L - matrix as
in eq. (1); W - matrix as in eq. (2);
Output: c[]- cost vector of stops;
p[]- path vector of stops.
begin
    create an empty list unvisited
    create empty arrays l[],c[],p[]
    for each vertex v in Graph:
        c[v] = infinity
        p[v] = undefined
        add v to the list unvisited
    c[s] = 0
    while unvisited is not empty:
        u = vertex with min cost in
the unvisited list
        remove u from the list
unvisited
        for each neighbor v of u:
           create lines' list comLines
           penalty, comLines =
Penalty(l[u],L[u,v])
            alt = c[u] + W[u,v] +
penalty
            if alt < c[v]:
                c[v] = alt
                p[v] = p[v] + u
return c[], p[]
```

Fig. 1. The pseudocode of the Modified Dijkstra algorithm.

```
function Penalty(curLines,
adjLines,w_Penalty,t_Penalty):
Input: curLines  - list of current
lines; adjLines - list of adjacent
lines; w_Penalty- penalty per
repetitive walk; t_Penalty - penalty
for a line transfer.
Output: penalty - penalty for an edge;
comLines - common lines of a path.
begin
    Create list comLines; penalty = 0
    if curLines is null:
        if adjLines is null:
            penalty = w_Penalty
        else:
            penalty = t_Penalty
            comLines = adjLines
    else:
        if adjLines is null:
            penalty = w_Penalty
        else:
            for each line in curLines:
                if line ∈ adjLines:
              add line to comLines
            if comLines is null:
                penalty = t_Penalty
                comLines = adjLines
return penalty, comLines
```

Fig. 2. The penalty function of the Modified Dijkstra algorithm

```
function FuzzyDijkstra(s,L,W,M,D):
Input: s – a given stop; L – matrix of
paths' lines; W – matrix of paths'
lengths; M – matrix of paths' degrees;
D – distance matrix between the stops.
Output: fuzzyDegree[] –fuzzy degree
vector of the stops; cost[] – cost
vector of the stops; path[] – path
vector of the stops.

    create list unvisited
    create arrays
lines[],cost[],path[], fuzzyDegree[]
    for each vertex v in Graph:
        cost[v] ← INFINITY
        path[v] ← UNDEFINED
        fuzzyDegree[v] ← MaxInt
        add v to unvisited
        cost[s] ← 0
    while unvisited is not empty:
        u ← vertex in unvisited with
min cost
        remove u from unvisited
        for each neighbor v of u:

comDegree,fuzzyPenalty,comLines ←
FuzzyPenalty(cur,adj,lines[u],L,W,M,D)
            alt ← cost[u] + W[u,v] +
fuzzyPenalty
            if alt < cost[v]:
                cost[v] ← alt
                fuzzyDegree[adj] ←
min(fuzzyDegree[cur], comDegree)
                path[v] ← path[v] + u
    return fuzzyDegree[],cost[],path[]
```

Fig. 3. The pseudocode of the FuzzyDijkstra algorithm.

The main difference between the Modified Dijkstra algorithm and the classical Dijkstra algorithm is that the Modified Dijkstra algorithm also includes a penalty calculation procedure. This procedure is used to reduce the number of transfers between lines and repetitive walking situations. The parameters of the function, $curLines$ and $adjLines$, are the list of lines from the starting point to the current stop and the list of lines between the current stop and the next trial stop, respectively. Other parameters, $w\_Penalty$, is the penalty applied to minimize uninterrupted repetitive pedestrian walks between stops, and $t\_Penalty$ is the penalty used to minimize the number of transfers between lines. The pseudocode of this procedure is given in Figure 2.

In the study (Nasiboglu, 2022), the FuzzyDijkstra algorithm, which is a fuzzy development of the Modified Dijkstra algorithm, is proposed. In the FuzzyDijkstra algorithm, a $fuzzyPenalty$ value consisting of total fuzzy walking and fuzzy accessibility degrees is used.

Let us mark the path from the starting s_O stop to the target s_D stop with π_{s_O,s_D}:

$$fuzzyPenalty = 1 - \mu(\pi_{s_O,s_D}) \quad (47)$$

```
function FuzzyPenalty(cur, adj,
curLines,L,W,M,D):
Input: cur – a current edge; curLines
– list of current lines; W – matrix of
paths' lengths; L – matrix of paths'
lines; M – matrix of paths' degrees; D
– distance matrix between the stops.
Output: comDegree –fuzzy degree of a
common line; fuzzyPenalty – fuzzy
penalty for an edge; comLines – common
lines of a path.

    walkPenalty ← CONST1
    transPenalty ← CONST2
    adjLines ← L[cur,adj]
    adjDegree ← max{M[cur,adj][l]: l
in adjLines}
    walkDegree ← max{0, 1-
D[cur,adj]/maxWalk}
    create list commonLines
    penalty ← 0
    comDegree ← 1
    if curLines is null:
        if adjLines is null:
            penalty ← walkPenalty
            comDegree ← walkDegree
        else:
            comLines ← adjLines
            penalty ← transPenalty
            comDegree ← adjDegree
    else:
        if adjLines is null:
            penalty ← walkPenalty
            comDegree ← walkDegree
        else:
            for each line in curLines:
                if adjLines contains
line:
                    add line to comLines
            if comLines is null:
               comLines ← adjLines
               penalty ← transPenalty
               comDegree ←
max{M[cur,adj][l]: l in comLines}
            if comLines is empty:
               fuzzyPenalty = penalty +
(1 – walkDegree)
           else:
               fuzzyPenalty = penalty +
(1 – comDegree)
    return comDegree, fuzzyPenalty,
comLines
```

Fig. 4. The pseudocode of the FuzzyPenalty function of FuzzyDijkstra algorithm.

Here, $\mu(\pi_{s_O,s_D})$, is calculated as in equation (41) in section 3.4. Thus, the total cost of the journey from the starting point to the destination is calculated as follows:

$$cost(\pi_{s_O,s_D}) = w(\pi_{s_O,s_D}) + \left(1 - \mu(\pi_{s_O,s_D})\right) \quad (48)$$

The pseudocode of the FuzzyDijkstra algorithm is given in Figure 3.

The fuzzy penalty value used in the FuzzyDijkstra algorithm is calculated using a separate FuzzyPenalty function. The pseudocode of the FuzzyPenalty function is given in Figure 4.

5. Discussion

The solution algorithms of models built on the urban public transportation network use different topologies, and may show different time complexity depending on the data structures, space topology and algorithms they use.

Dijkstra's algorithm, a widely used algorithm, uses a data structure to store and query partial solutions ordered by distance from the origin. Dijkstra's original algorithm does not use minimum priority queues and runs in $O(|V|^2)$ time where $|V|$ is the number of vertices in the graph. An improved variant of this algorithm can use a minimum-priority ordered Fibonacci heap, which can reduce the runtime complexity to $O(|E| + |V|log(|V|))$. This algorithm is asymptotically the fastest known single-source shortest path algorithm for any directed graph with infinite non-negative weights.

PTN models proposed in the literature may also vary in terms of optimization criteria, use of fuzzy or heuristic approaches, and use of city smart card data. While models based on the classical Dijkstra algorithm came to the fore in earlier years, modified Dijkstra algorithms containing penalty functions and fuzzy models, especially based on city smart card data, have been discussed in recent years. Comparison results of some urban PTN models in the literature are briefly given in Table 1. In the table, the algorithms are categorized according to the type of network topology they use, the optimization criteria they use, whether they use fuzzy or heuristic approaches, and whether they use city smart card data. Table 1 also shows the basic differences of the algorithms and their computational complexity.

Table 1. Comparison results of some PTN models

Authors	Used network topology	Optimization criterion	Fuzzy/heuristic approach usage	Smart card data usage	Basic differences	Computational complexity
Dijkstra, 1959	Undirected graph, P topology.	The shortest path suggestion in graph.	Not used.	Not used.	Algorithmic approach that forms the basis of shortest route problems.	$O(\|V\|^2)$, where $\|V\|$ is the number of vertices of the graph.
Ji et al., 2007	Undirected graph, P topology.	The shortest path suggestion in graph.	Edge lengths are fuzzy numbers.	Not used.	Expected shortest path and α-shortest path concepts are handled. A metaheuristic algorithm integrating genetic algorithm with simulation is proposed.	$O(2^{\|E\|})$, where $\|E\|$ is the number of edges of the graph. The search space is scanned with a genetic algorithm.
Mahdavi et al., 2009	Undirected graph, P topology.	The shhortest path suggestion in graph.	Edge lengths are fuzzy numbers.	Not used	The fuzzy shortest chain problem using a suitable ranking method is used. A dynamic programming approach is applied for solution.	$O(\|V\|^2)$, where $\|V\|$ is the number of vertices of the graph.
Deng et al., 2012	Undirected graph, P topology.	The shortest path suggestion in graph.	Edge lengths are fuzzy numbers.	Not used	The classical Dijkstra algorithm adopted to perform with the graded mean integration representation of fuzzy numbers.	$O(\|V\|^2)$, where $\|V\|$ is the number of vertices of the graph.
Lopez and Lozano, 2014	Various	The shortest path.	Various	Not used	Various techniques used to find shortest paths in multimodal transportation networks are examined.	Various
Nasibov et al, 2016	Directed graph, L' topology.	The shortest route, the least number of transfers, the least number of walking to stops.	Fuzzy stop-stop and stop-line accessibility degrees are proposed.	Smart card data are used	The concept of the highest degree reachable fuzzy optimal route is discussed.	$O(s^2l^3c^4)$, where s - number of PTN stops, l - number of PTN lines, c - number of stops of the longest line.
Bozyiğit et al., 2017	Directed graph, P' topology.	The shortest path and the least number of transfers.	Not used	Not used	Penalty function is added to Dijkstra algorithm. In addition to the shortest route, the minimum number of transfers is also considered	$O(s^2l)$, where s - number of PTN stops, l - number of PTN lines.

Bozyiğit et al., 2018	Directed graph, P' topology.	The shortest path, the least number of transfers, the least number of walking to stops	Not used	Not used	The shortest route, the least number of transfers and the least number of walking stops are considered.	$O(s^2l)$, where s - number of PTN stops, l - number of PTN lines.
Long and Tan, 2020	Undirected graph, P topology.	The shortest path.	An improved genetic algorithm is used to find the optimal route.	Not used	Model constraints such as starting and ending point walking distance, transfer distance, driving distance and cost are calculated. Based on the optimal recursive selection properties of the developed genetic algorithm, the optimal path selection model of the public transportation network is created.	$O(2^{\|E\|})$, where $\|E\|$ is the number of edges of the graph. The search space is scanned with a genetic algorithm.
García-Heredia et al., 2021	An L-space like model approach was used.	The multi-shortest path problem is handled.	A parallelizable metaheuristic approach is proposed for solution.	Data of the Air Traffic Flow Management (ATFM) system are used.	For each network within a collection, the problem of finding the shortest path between two given nodes is defined as an integer programming problem. A metaheuristic approach is used for the solution.	$O(2^{\|E\|})$, where $\|E\|$ is the number of edges of the graph. The search space is scanned with a metaheuristic algorithm.
Nasiboglu, 2021	Directed graph, L' topology.	The shortest path with the highest degree of fuzzy connection.	The fuzzy connection degree of the route is calculated depending on the passenger density in the bus.	Smart card data are used.	A fuzzy optimization model based on passenger density inside the bus is proposed. Minimum aggregation of inter-stop densities was used.	$O(mslc)$, where m - the maximum possible number of connections on the route, s - the number of PTN stops, l - the number of PTN lines, c - the number of stops of the longest line.
Nasiboglu, 2022	Directed graph, P' topology.	The shortest path, least number of transfers, highest degree of fuzzy connection.	The fuzzy connection degree of the route is calculated depending on the passenger density in the bus.	Smart card data are used	An optimization model aiming the shortest route, the least number of transfers and the highest line connection degree is proposed. Aggregated inter-stop intensities is used.	$O(s^2l^2c)$, where s - number of PTN stops, l - number of PTN lines, c - number of stops of the longest line.

6. Conclusion

In this study, various exact and fuzzy/heuristic-based approaches for creating optimal routes on the urban public transportation network were examined. Comparative analyzes of the models were made, taking into account criteria such as network topologies used, optimization criteria, fuzzy or heuristic approaches, use of smart card data, important differences of the models, and computational complexity. When we generally evaluated the models in terms of the solution algorithms used, they were divided into exact algorithms and fuzzy/heuristic algorithms. Apparently, exact algorithms were generally algorithms built on the Dijkstra's algorithm based on dynamic programming. Usually, fuzzy algorithms were also based on the Dijkstra's algorithm and its improvements that can work with fuzzy data. In heuristic algorithms, genetic algorithm and other metaheuristic methods were used to scan the solution space.

In future studies, it may be noteworthy to create more sophisticated fuzzy/metaheuristic optimal route recommendations models based on smart card data and inner-line dynamic indicators.

References

Bagchi, M., White, P.R. (2005). The potential of public transport smartcard data, Transport Policy, 12, 464–474.

Barry, J., Newhouser, R., Rahbee, A., & Sayeda, S. (2002). Origin and destination estimation in New York City with automated fare system data, Transportation Research Record, 1817, 183-187.

Bozyiğit, A., Alankuş, G., & Nasiboğlu, E. (2017). Public transport route planning: Modified dijkstra's algorithm, 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, pp. 502-505, doi: 10.1109/UBMK.2017.8093444.

Bozyigit, A., Alankus, G., & Nasibov, E. (2018). A Public Transport Route Recommender Minimizing the Number of Transfers, Sigma J Eng & Nat Sci, 9(4), 437-446.

Ceder, A. (2007). Public transit planning and operation: theory, modelling and practice, Butterworth-Heinemann, Oxford.

Chen, Y.-Z., Li, N., & He, D.-R. (2007). A study on some urban bus transport networks, Physica A: Statistical Mechanics and Its Applications, 376(1-2), 747–754.

Deng, Y., Chen, Y., Zhang, Y., & Mahadevan, S. (2012). Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment, Applied Soft Computing, 12, 1231–1237.

Dijkstra, E.W. (1959). A note on two problems in connexion with graphs, Numerische Mathematik, 1, 269–271. doi:10.1007/BF01386390.

García-Heredia, D., Molina, E., Laguna, M., & Alonso-Ayuso, A. (2021). A solution method for the shared resource-constrained multi-shortest path problem, Expert Systems with Applications, 182, 115193.

Hui, C., Su, B., Zhou, Y., & He, D. (2007). Assortativity and act degree distribution of some collaboration networks, Physica A: Statistical Mechanics and its Applications, 383(2), 687–702.

Ji, X., Iwamura, K., & Shao, Z. (2007). New models for shortest path problem with fuzzy arc lengths, Appl. Math. Model., 31, 259–269.

Khaing, O., Wai, H.H., Myat, E.E. (2018). Using Dijkstra's Algorithm for Public Transportation System in Yangon Based on GIS, International Journal of Science and Engineering Applications, 7(11), 442-447.

Long, D. -H., Tan, J. (2020). Optimal Path Selection Model of Public Transportation Network Based on Improved Genetic Algorithm, IEEE International Conference on Industrial Application of Artificial Intelligence (IAAI), Harbin, China, pp. 345-350.

López, D., Lozano, A. (2014). Techniques in Multimodal Shortest Path in Public Transport Systems, Transportation Research Procedia, 3, 886-894.

Mahdavi, I., Nourifar, R., Heidarzade, A., & Amiri, N.M. (2009). A dynamic programming approach for finding shortest chains in a fuzzy network, Applied Soft Computing, 9, 503–511.

Marc, B. (2011). Spatial networks. Physics Reports, 499(1–3), 1–101.

Munizaga, M.A., Palma, C. (2012). Estimation of a disaggregate multimodal public transport origin-destination matrix from passive smartcard data from Santiago, Chile, Transportation Research Part C, 24, 9-18.

Nasiboglu, E.N., Kuvvetli, U., Ozkilcik, M., & Eliiyi, U. (2012). Origin-Destination Matrix Generation Using Smart Card Data: Case Study for Izmir, Proceedings of the IV International Conference "Problems of Cybernetics and Informatics" (PCI'2012), Baku, Azerbaijan, 1, 188-191.

Nasiboğlu, R. (2021). Otobüs İçi Yoğunluk Oranını Dikkate Alan Bulanık Optimal Güzergah Öneri Modeli ve Çözüm Algoritması. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 25(2), 432-440.

Nasiboglu, R. (2022). Dijkstra solution algorithm considering fuzzy accessibility degree for patch optimization problem, Applied Soft Computing, 130, 109674

Nasibov, E.N., Diker, A., & Nasibov, E. (2016). A multi criteria route planning model based on fuzzy preference degrees of stops, Applied Soft Computing, 49, 13-26.

Panagiotis, A., Fisk, D. (2006). Large subway systems as complex networks, Physica A: Statistical Mechanics and its Applications, 367(15), 553–558.

Pelletier, M.P., Trépanier, M., & Morency, C. (2011). Smart card data use in public transit: A literature review, Transportation Research Part C: Emerging Technologies, 19, 557-568.

Ray, A., Sharma, H., & Sharma, D. (2022). Analysis and Design of Public Transport Route Planner: Dijkstras Algorithm, International Journal for Research in Applied Science & Engineering Technology (IJRASET), 10(6), 4571-4575.

Schrijver, A. (2012). On the history of the shortest path problem, Documenta Mathematica, Extra Volume ISMP, 155–167.

Sienkiewicz, J., Hołyst, J.A. (2005). Statistical analysis of 22 public transport networks in Poland, Physical Review E, 72(4), 046127.

Sui, Y., Shao, F., Yu, X., Sun, R., & Li, S. (2019). Public transport network model based on layer operations, Physica A: Statistical Mechanics and its Applications, 523(1), 984-995.

Tirastittam, P., Waiyawuththanapoom, P. (2014). Public Transport Planning System by Dijkstra Algorithm: Case Study Bangkok Metropolitan Area, World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering, 8(1), 54-59.

Trepanier, M., Tranchant, N., & Chapleau, R. (2007). Individual trip destination estimation in a transit smart card automated fare collection system, Journal of Intelligent Transportation Systems, 11(1), 1-14.

Vito, L., Marchiori, M. (2001). Efficient behavior of small-world networks, Phys. Rev. Lett. 87(19), 198701.

Von Ferber, C., Holovatch, T., Holovatch, Y., & Palchykov, V. (2007). Network harness: metropolis public transport, Physica A: Statistical Mechanics and Its Applications, 380(1-2), 585–591.

Von Ferber, C., Holovatch, T., Holovatch, Y., & Palchykov, V. (2009). Public transport networks: empirical analysis and modeling, The European Physical Journal B—Condensed Matter and Complex Systems, 68(2), 261–275.

Wilson, N.H., Zhao, J., & Rahbee, A. (2009). The potential impact of automated data collection systems on urban public transportation planning, In Wilson, N.H., Nuzzolo, A., (Eds.). Schedule-based modeling of transportation network, Springer, 75-99.

Zhang, L., Ma, X., Wang, H., Feng, M., & Xue, S. (2013). Modelling and optimisation on bus transport system with graph theory and complex network, International Journal of Computer Applications in Technology, 48(1), 83–92.

Zhang, L., Ma, H., Sun, B., Li, Y., Wang, M., & Xue, S. (2013). Computing and application on complex bus network system, International Journal of Information and Computer Science, 2(7), 127–132.

Zhu, Z.-T., Zhou, J., Li, P., & Chen, X.-G. (2008). An evolutionary model of urban bus transport network based on B-space, Chinese Physics B, 17(8), 2874–2880.