Abdulkerim BENLİ

A Ph.D. Thesis

AGU 2023

MATHEMATICAL MODELS FOR PUBLIC TRANSPORTATION PLANNING

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF
ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Abdulkerim BENLİ
December 2023

MATHEMATICAL MODELS FOR PUBLIC TRANSPORTATION PLANNING

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF
ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Abdulkerim BENLİ
December 2023

SCIENTIFIC ETHICS COMPLIANCE

I hereby declare that all information in this document has been obtained in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Abdulkerim BENLİ

REGULATORY COMPLIANCE

Ph.D. thesis titled "**MATHEMATICAL MODELS FOR PUBLIC TRANSPORTATION PLANNING**" has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By	Advisor
Abdulkerim BENLİ	Prof. Dr. İbrahim AKGÜN

Head of the Industrial Engineering Graduate Program

Assoc. Prof. Dr. Ramazan ÜNLÜ

ACCEPTANCE AND APPROVAL

Ph.D. thesis titled “**MATHEMATICAL MODELS FOR PUBLIC TRANSPORTATION PLANNING**” prepared by Abdulkerim BENLİ and has been accepted by the jury in the Industrial Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

……….. /……….. / 2023

JURY:

Advisor: Prof. Dr. İbrahim AKGÜN…………………………………………………

Member: Prof. Dr. Sinan GÜREL…………………………………………………….

Member: Asst. Prof. Dr. Selçuk GÖREN…………………………………………….

Member: Prof. Dr. Emel KIZILKAYA AYDOĞAN…………………………………

Member: Assoc. Prof. Dr. Ramazan ÜNLÜ…………………………………………..

APPROVAL:

The acceptance of this Ph.D. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science, Executive Board dated ….. /….. / ……….. and numbered …………..……. .

… /… / 2024

Graduate School Dean

Prof. Dr. İrfan ALAN

ABSTRACT

MATHEMATICAL MODELS FOR PUBLIC TRANSPORTATION PLANNING

Abdulkerim BENLİ
Ph.D. in Industrial Engineering
Advisor: Prof. Dr. İbrahim AKGÜN

December 2023

In this thesis, we propose mathematical programming models and solution methodologies for the *transit network design problem* (TNDP) and *transit network design and frequency setting problem* (TNDFSP). TNDP aims at designing the routes whereas TNDFSP aims at determining the routes and their frequencies of the routes to satisfy passenger demand in a transit network. The proposed models for TNDP (and TNDFSP) incorporate the features of real-life transit network systems and reflects the views of both passengers and the transit agency by considering in-vehicle travel time, transfers (and waiting times at the boarding and transfer stops, overcrowding and under-utilization of vehicles, and vehicle fleet size). Unlike previous studies that simplify several aspects of TNDP (and TNDFSP), the proposed models are the first to determine the routes (and their frequencies) simultaneously from scratch, i.e., without using a line (and frequency) pool, while considering the aforementioned issues such as transfers. We solve the proposed model for TNDP by the Gurobi solver and a heuristic algorithm based on the Benders decomposition with enhancements including an in-out cut loop scheme, disaggregation, and Pareto-optimal cuts. We solve the proposed model for TNDFSP by Gurobi-based Node Relaxation Heuristics. The proposed models have been validated using the benchmark instances from the literature. We provide the results of what-if analyses conducted using a real-world public bus transport network in the city of Kayseri in Türkiye. The results indicate that the models produce good-quality solutions compared to the state-of-the-art algorithms in the literature and that public transit planners can use the models as a decision aid.

Keywords: Transit network route design and frequency setting problem; urban public transportation; mathematical programming; nonlinear mixed-integer programming; real-world application.

ÖZET

TOPLU TAŞIMA PLANLAMASI İÇİN MATEMATİKSEL MODELLER

Abdulkerim BENLİ
Endüstri Mühendisliği Anabilim Dalı Doktora
Tez Yöneticisi: Prof. Dr. İbrahim AKGÜN

Aralık 2023

Bu tezde, *toplu taşıma ağı tasarım problemi* (TATP) ve *toplu taşıma ağı tasarımı ve frekans ayarlama problemi* (TATFAP) için matematiksel programlama modelleri ve çözüm metodolojileri önerilmektedir. TATP, yolcu talebini karşılamak için bir rota ağı tasarlamayı amaçlarken, TATFAP, rota tasarımına ek olarak bu rotalar için yapılan frekans ayarlanması problemini de ele almaktadır. TATP (ve TATFAP) için önerilen modeller, gerçek hayattaki toplu taşıma ağı sistemlerini gerçekçi bir şekilde modelleyebilmekte ve araç-içi seyahat süresi, aktarma, (ve ilk biniş ve aktarma duraklarındaki bekleme süreleri, araç filosu büyüklüğü, araç kapasite aşımı ve araçların verimsiz kullanımı) gibi bir çok faktörü dikkate alarak, hem yolcuların hem de toplu taşıma kuruluşunun bakış açılarını yansıtabilmektedir. Her iki problem çeşitli şekillerde daha basit ve sade hale getiren literatürdeki çalışmalardan farklı olarak, bu çalışmada önerilen modeler, yukarıda belirtilen aktarma gibi hususları dikkate alan ve bir hat (ve frekans) havuzu kullanmadan, hatları ve (frekansları) sıfırdan belirleyebilmektedir. TATP için önerilen model, Benders ayrıştırmasına dayalı bir sezgisel algoritma ve Gurobi çözücüsü kullanarak çözülmüştür. TATFAP için önerilen model ise Gurobi tabanlı Düğüm Gevşetme Sezgiseli kullanılarak çözülmüştür. Önerilen modellerin geçerliliği, literatürdeki kıyaslama amaçlı kullanılan veri kümeleri esas alınarak doğrulanmıştır. Ayrıca, Türkiye'nin Kayseri şehrindeki halk otobüsü ulaşım ağı esas alınarak oluşturulan senaryolar için modeller kullanılarak elde edilen analiz sonuçları sunulmuştur. Sonuçlar, önerilen modellerin literatürdeki çoğu algoritmadan daha iyi çözümler üretildiğini ve toplu taşıma planlamacıları tarafından bir karar destek mekanizması olarak kullanılabileceğini göstermektedir.

Anahtar kelimeler: Toplu taşıma ağı tasarımı ve frekans ayarlama problemi; matematiksel programlama; doğrusal olmayan karma tam sayılı programlama; gerçek hayat uygulaması.

Acknowledgments

Since starting my graduate studies, my academic advisor, Professor Dr. İbrahim Akgün, has provided me excellent support and guidance through his exceptional academic intellect and skill. He always encourages me to push the boundaries and broaden the scope of my perspective. Besides his academic guidance, he is a kind and understanding person who has assisted me in overcoming my challenges as a graduate student. I would like to express my deepest gratitude to him for making this thesis research possible.

I would also like to thank the members of my thesis monitoring committee, Professors Dr. Sinan Gürel and Dr. Selçuk Gören, for their insightful feedback and guidance during the research and writing of this thesis. I would also want to thank to rest of the thesis defense exam committee, Professors Dr. Emel Kızılkaya Aydoğan and Dr. Ramazan Ünlü, for their generous time and contributions. I am sincerely grateful to my Department of Industrial Engineering colleagues for their friendship and constant support for the last nine years.

I shall be profoundly grateful to my mother and father for the unconditional love and support they have given me. My deepest gratitude goes out to my adored Elif and Aliş. I wish there were some way that I could make up for the time and care that I have to take away from you. All of this work is being done for you.

We acknowledge Kayseri Ulaşım Inc. for their financial support and for providing the necessary data for the research. We also would like to thank Burak Telcioğlu and Feyzullah Gündoğdu for their support and share their expert opinion regarding public transportation problems.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ATT	: Average Travel Time
BD	: Benders Decomposition
BDH	: Benders Decomposition-based Heuristic
FSP	: Frequency Share Procedure
LP	: Linear Program/Problem
LPP	: Line Planning Problem
MIP	: Mixed Integer Program/Problem
MMNIP	: Multiobjective Mixed Nonlinear Integer Program/Problem
MP	: Master Problem
PAP	: Passenger Assignment Problem
PTPM	: Public Transportation Planning Model
SP	: Subproblem
SPP	: Shortest Path Principle
TN	: Transit Network
TNDFSP	: Transit Network Design and Frequency Setting Problem
TNDM	: Transit Network Design Model
TNDP	: Transit Network Design Problem
TTT	: Total Travel Time
UE	: User Equilibrium

Chapter 1

Introduction

Immigration from rural areas and increased mobility of people worldwide have led to rapid urbanization. As of 2018, 55 percent of the world's population resides in urban areas, which is expected to rise to 68 percent by the year 2050 [1]. This unprecedented trend will undoubtedly bring about many issues such as escalating housing costs and forcing residents to relocate from downtown and city center to the outskirts, thereby increasing the amount of daily commuting and traffic [2]. The ever-increasing energy demand by urbanization and the notorious rise in greenhouse gas emissions pose a challenge to sustainable urban development. Providing high-quality and efficient public transportation is likely the most effective solution for cities to circumvent these challenges [3]. However, designing such a public transportation system is difficult because several issues need to be considered simultaneously. Firstly, the system is comprised of various agents with conflicting objectives. On the one hand, every passenger hopes to move from the boarding (i.e., departure) stop at any time to any arrival stop in the shortest amount of time possible while still traveling most comfortably throughout the journey. On the other hand, the transit agency responsible for the system's management may not have the financial resources to make this possible. Secondly, constraints such as the traffic density, the capacity of the vehicle fleet and crew, and the appropriateness of the infrastructure need to be considered to develop a solution. Thirdly, transit agencies should adopt holistic decision-making principles to sustain the system, which is not easy because there are several interlinked and hard problems to solve.

Ceder and Wilson [4] categorize public transport decision problems according to the timeframe into four problems: *transit network design, timetable development, vehicle scheduling,* and *crew scheduling*. *Figure 1.1* depicts the functional relationships between these decision problems. *Transit network design* is a strategic problem that involves planning the routes by taking into account passenger demand and travel time. A route is a simple path connecting terminal stops with intermediate stops. The problem is also addressed in other contexts such as the construction of roads/links [5], the location of stops/stations [6], and the expansion of transit infrastructure [7]. *Timetable development* is a tactical problem that aims at developing a schedule for trips along the routes. The *frequency setting* is a subproblem of timetable development that determines the frequencies of routes (i.e., the number of vehicles serving per time unit for each route) to meet passenger demand. The frequency information of a route helps to determine the route's headway (i.e., the time at which the next bus will arrive at a stop along the route) and the resulting headway can be used to develop the route's timetable. *Vehicle*

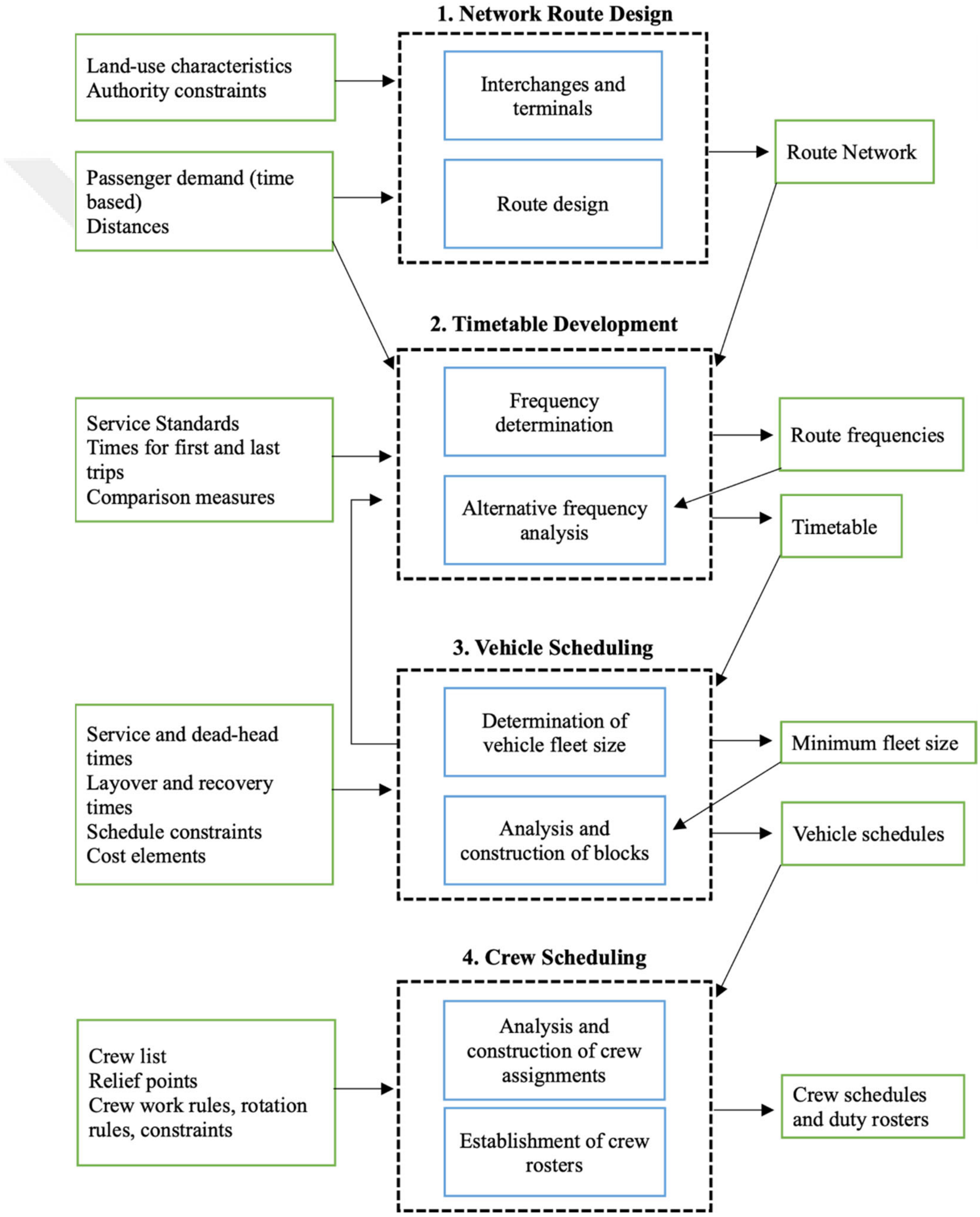


Figure 1.1 The functional relation between the four decision problems of transit operations

scheduling is an operational level problem that assigns vehicle fleet to routes according to the timetable. *Crew scheduling* is another operational problem that creates duty rosters for the crew by taking the final timetable and vehicle schedule into account.

These decision problems are mostly solved separately, and most studies address only certain aspects of the problems. However, there exist studies that attempt to integrate some of these problems. This thesis will address two problems: the *transit network design problem* (TNDP) and the *transit network design and frequency setting problem* (TNDFSP), which integrates TNDP and the *frequency setting* problem.

The *transit network design problem* (TNDP) is to satisfy the passenger demand specified in an Origin-Demand (OD) demand matrix by designing routes considering travel times specified in the OD distance matrix. TNDP is defined under the assumption that the underlying infrastructure for public transport is already established. It is presumed that the network of stops as well as OD demand and distance matrices are already known. OD demand matrix records the number of people who want to travel from a stop (i.e., origin) to another stop (i.e., a destination) in a given time period. OD distance matrix consists of the distances (usually in a time unit) between stop pairs.

The cost function in TNDP studies reflects either perspective of passengers or the transit agency. Passenger-oriented cost is mostly associated with minimizing passengers' travel time including in-vehicle travel time and the number of transfers. Another strategy only focuses on maximizing the number of direct travelers who travel on a single route without transferring. Nevertheless, this strategy generates lengthy journeys resulting in overcrowded vehicles at various points along the route, and numerous empty seats on less-crowded routes, which is inefficient [8]. The alternative is to penalize transfers at a certain threshold rather than imposing the use of a single route to avoid the aforementioned issues while increasing the number of direct travelers. Transit agency-oriented costs refer to variable costs such as the length of routes as well as fixed costs (e.g., investment and maintenance) [9].

TNDP is an NP-hard problem [10]. Therefore, the problem is simplified in various aspects, e.g., passenger transfer is not modeled. However, this can prevent TNDP from making useful implications on real-world applications. For instance, models that do not penalize transfers permit passengers to make many transfers although unnecessary transfers are inconvenient [11]. A few mathematical programming studies treat transfers by expanding the transit network ([10], [12]) even though it increases the complexity of the problem. The construction of routes is also simplified. Most studies design routes selecting from a pool of candidate routes (i.e., line pool) generated by external route construction algorithms employing the shortest path or other heuristic methods [13] or configuring them by route generation algorithms [14]. Even though this simplification may overlook efficient routes, the tradeoff is made against the burden of more complexity. Only a few studies allow the model to build routes from scratch endogenously ([15], [11]).

Transit network design and frequency setting problem (TNDFSP) aims to meet passenger demand by constructing routes and determining frequencies of routes. Routes and associated frequencies are designed considering the passengers' perspective to minimize the total travel time of passengers that includes in-vehicle travel, waiting, and transfer times and to maximize direct-traveler passengers (i.e., minimize transfers). However, considering only the passengers' perspective may be too costly for the transit agency. In this regard, it should also try to minimize the costs, e.g., the costs of trips as well as fixed and operational costs of operating a vehicle fleet. In doing so, the transit agency needs to make several decisions, e.g., the number of routes to operate, the size of the vehicle fleet, the number of stops on a route, and the percentage of direct travelers.

TNDFSP is an NP-hard problem [16] and computationally challenging even for small transit networks with fewer than a few stops. Determining passengers' waiting time requires headway information derived from the inverse of frequency that introduces nonlinearity to TNDFSP. Choosing the stops to include in a route and selecting routes from a line pool require binary decision variables. Furthermore, the problem is multiobjective since passengers and transit agencies have conflicting objectives, especially if the goal is to solve the costs of passengers and the transit agency simultaneously. All these complexities have led researchers to devise (meta)heuristics and analytical methods to solve TNDFSP rather than mathematical programming. However, methods other than mathematical programming use iterative approaches to set frequencies by updating frequencies iteratively according to overcrowding or underutilization of vehicles. It would be ideal if the frequencies were set simultaneously with the route construction [17]. This can be accomplished by defining frequencies as endogenous variables within mathematical programs. Moreover, one cannot be confident about the solution quality unless mathematical programs are used.

Durán-Micco and Vansteenwegen [18] give a comprehensive survey of studies about TNDP and TNDFSP and point out the following issues regarding previous studies.

- Studies primarily focus on simplified networks. For instance, clustering stops to reduce the size of a real transit network [19].
- Since the frequency setting problem is hard to solve with network route design, frequencies are set iteratively [20], i.e., a predefined, initial set of frequencies are updated according to the passenger assignment instead of determining them endogenously in a model. In case of not using an iterative frequency setting, appropriate frequencies are selected out of a finite candidate set [12].
- Passenger assignment is done mainly by ignoring overcrowding in TNDFSP studies. A few TNDFSP studies consider crowding issues and use incremental algorithms in which the frequency of crowded routes is updated iteratively.
- New models and approaches for TNDFSP and TNDP are needed to add more realistic concepts and solve real-world problems. Many assumptions are made before appropriately addressing problems, which results in different studies making different assumptions leading to different problems for which the results cannot be directly compared.
- All TNDP and TNDFSP studies except for a few ([15], [21]) design routes out of a predefined route/line pool ([22], [23]) or route generation algorithms [12] with the risk of overlooking optimal routes.
- (Meta)Heuristic approaches are mostly preferred because of the computationally challenging nature of the problem. Although they can provide good solutions for some benchmark instances, solution quality depends mainly on instance data and requires much tuning effort. Moreover, they are not flexible enough to easily incorporate realistic features and cannot provide information regarding the quality of the generated solutions.
- Case studies stemming from real-world networks simplify passenger demand and deviate from realism by only considering a proportion of OD demand pairs, i.e., a highly-sparse OD demand matrix is used [24].

We are motivated to bridge the gap between theory and practice by developing novel mathematical models and solution methodologies for TNDP and TNDFSP and by solving problems that represent real-world transit network characteristics. Our goals are to solve the models using off-the-shelf commercial solvers, conduct what-if analyses to provide the decision-makers with managerial insights, and assess the flexibility and applicability of the models.

The contribution of this thesis is threefold:

(1) We propose a novel mathematical programming formulation for TNDP for which (meta)heuristics approaches have been used extensively in the literature. The passenger perspective is reflected by considering the in-vehicle travel time and the number of transfers. The transit agency perspective is addressed by minimizing the total length of routes by limiting the number of routes and the maximum number of stops allowed in each route. The model *endogenously determines routes* including their *terminal and intermediate stops.* To our knowledge, *the model is the first to determine routes from scratch*, i.e., without using line pools while realistically modeling transfers. Besides, we develop a heuristic algorithm based on Benders decomposition [25] with additional enhancements to solve large real-world network instances.

(2) We propose a novel mathematical programming formulation based on realistic concepts for TNDFSP. The model reflects passengers' route choice realistically by considering *in-vehicle travel time*, *transfers*, and *waiting times* at boarding (departure) and transfer stops as well as *overcrowding and under-utilization of vehicles* in the passenger assignment. The model also reflects the transit agency's perspective by allowing the transit agency to determine service level by imposing limitations on several parameters such as *vehicle fleet size*. The model *endogenously determines routes* including their *terminal and intermediate stops* as well as their *frequencies*. To our knowledge, *the model is the first to determine routes and their frequencies simultaneously from scratch*, i.e., without using line and frequency pools while considering the aforementioned issues such as transfers and waiting.

(3) We utilize the proposed TNDP and TNDFSP models to conduct *what-if analyses* using a *real-world public bus transport network* in Kayseri, Türkiye. We also present the results of computational tests implemented to validate and verify the model using the benchmark instances in the literature. The results obtained using Gurobi and the

proposed solution methodologies indicate that the models produce good quality solutions compared to the state-of-the-art algorithms in the literature and that public transit planners can use the models as a decision aid.

The remainder of the thesis is organized as follows. Chapter 2 presents a mixed-integer mathematical programming (MIP) formulation for TNDP. A *Benders decomposition-based* heuristic is developed to solve large-scale problems and algorithmic enhancements are implemented to accelerate solution times and improve solution quality. The proposed formulation is initially evaluated using *Mandl* [26], [27] and *Mumford* [32] benchmark instances and then applied to the real-world public transportation network in Kayseri. Chapter 3 proposes a new multiobjective mixed-integer nonlinear mathematical programming (MMNP) formulation for TNDFSP. It gives the results of the computational tests performed for the validation and verification of the proposed model and compares the performance of the proposed model with those of the state-of-art studies in the literature. Moreover, the results of what-if analyses for a real-world network application in Kayseri are discussed by managerial insights and findings. The study in Chapter 3 is published in the journal Mathematics [28]. Chapter 4 concludes the thesis with remarks and a discussion of future research directions.

Chapter 2

Transit Network Design Problem

In this chapter, we address the *transit network design problem (TNDP)* that aims to design routes to satisfy passenger demand in a transit network. TNDP considers the perspectives of two decision-makers: passengers and a transit agency. The objective of passengers is to minimize total travel time including *in-vehicle travel time* and the *number of transfers* while the objective of the transit agency is to reduce the number of routes and the number of stops in routes and hence the total length of routes.

We propose a new mathematical programming model for TNDP. Our approach differs from the related literature in various ways. The proposed model does not depend on a line pool or external route generation algorithms. Preprocessing such as analyzing OD demand pairs and generating candidate routes for a line pool is not required. The model uses endogenous decision variables to construct routes from scratch without predefining any stops on routes beforehand and ensures that the risk of overlooking the optimum routes is avoided. Moreover, the model uses the real transportation network as an input (i.e., it does not use extended extending transit network such as *change & go* [10] and *trajectory* [12] graphs that unnecessarily increase complexity). The *passenger assignment* rule in the proposed model adheres to the *shortest path principle*. Passengers will switch route choices and make transfers if doing so will reduce the total time spent in the transit network.

Metaheuristic approaches are rife in the literature due to the complexity of the TNDP. However, the solution quality of the metaheuristics depends mainly on instance data and requires a lot of tuning effort. We solve the proposed model using Gurobi solver. However, even though Gurobi can solve problems defined on small transit networks successfully, it struggles to solve large-scale problems. Therefore, we propose a heuristic algorithm based on *Benders decomposition* to increase solution performance. Furthermore, the computational capability of the decomposition is improved by employing algorithmic enhancements including *in-out cut loop scheme*, *disaggregation*, and *Pareto-optimal cuts*.

We verify the performance and capability of the proposed model using small- and large-size well-known benchmark instances, namely *Mandl* and *Mumford* instances. Because there are no mathematical programs for TNDP in the literature, we compare the resulting solutions with those obtained by the state-of-art metaheuristic algorithms. We apply the proposed model for a *real-world network with 204 nodes, 455 links, and 13338 OD demand pairs*, which is significantly larger than the problems in the literature.

2.1 Literature Review

Guihaire and Hao [29], Kepaptsoglou and Karlaftis [30], Farahani et al. [16], and recently Durán-Micco and Vansteenwegen [18] provide a detailed review of TNDP studies. That surveys indicate that the scope of studies varies significantly depending on the assumptions regarding objectives, solution approaches, parameters, and network settings. This thesis emphasizes studies that employ mathematical programming approaches; however, (meta)heuristics and analytical formulations will be briefly mentioned as necessary.

The backbone of a generic TNDP study is the methodology used for route design. Generally, routes are designed by selecting out of a line pool [22] or by generating routes via a route generation algorithm [14]. The cost function of TNDP is the most important criterion in the construction of routes. A cost function can be designed from passengers' and transit agencies' points of view. Typically, passenger-oriented costs are related to the amount of time spent traveling in the vehicle and the number of transfers whereas transit agency-oriented costs are related to total length of routes. In the literature, passenger- and transit agency-oriented costs are addressed separately as a single objective. Alternative is modeling TNDP as a multiobjective problem using ε-constraint method [31] and by a weighted sum of the associated costs [32].

The directness, maximum length, number of routes, and operational budgets are the fundamental constraints of TNDP [33]. Besides, design parameters used in TNDP studies range from issues regarding infrastructure to characteristics of passenger demand. Infrastructural parameters are mostly concerned with road network design for public transport (e.g., route and stop spacing [42], [43]). Passenger demand can be *elastic* or *fixed*. *Elastic* demand supposes that passenger demand changes after establishing routes [36] and varies for different periods [35]. Most studies assume the demand is *fixed* so passengers' demand does not depend on the route configuration. Network structure is another design parameter. Mostly routes are assumed to be bidirectional (i.e., having same stops are in both directions on a route), and starting and ending terminal stops of a route are different. However, there are some specific applications for radial [34] or rectangular grid networks [35]. Environmental concerns are also addressed for reducing the carbon footprint of transit networks. Environmental-friendly transit network design has been first studied by Site and Filippi [37] that minimize fuel consumption throughout routes. Beltran et al. [38] investigate subsidizing green routes on which low-emission vehicles serve by an external cost function. In addition to passenger and transit agency costs, Jovanovic et al. [39] consider harmful exhaust gases and the noise level of vehicles on routes. Newer studies focus on the electrification of transit networks (i.e., using electrical vehicles for transport). Iliopoulou et al. [40] provide a heuristic algorithm to construct electrified routes and determine the location of charge stations. Pylarinou et al. [41] redesign an existing transit network by deploying additional electric vehicles assuming that charge stations are located at terminal stops of routes.

Early TNDP studies are mostly solved by *analytical formulations* for very small networks with only a few stops. According to Ceder [42], analytical formulations are useful for carrying out policy analysis on idealized networks but they are not appropriate for designing an entire realistic transit network. Mathematical programming is another option to address TNDP. However, Chakroborty [43] notes that solving mathematical programming models are hard due to the discrete and multiobjective nature of the problem as well as difficulties with modeling complexities such as passenger transfers. As an example for challenging nature of the problem, Vermeir et al. [44] recently could solve only small *Mandl* [26] benchmark instances by an exact algorithm. Besides, some mathematical models attempt to solve particular aspects of TNDP e.g., maximizing the number of direct travelers [45], designing rapid transit networks [46], and for selecting routes out of a line pool [47].

(Meta)Heuristics are employed exhaustively for TNDP in generating routes for the line pool, selecting routes out of the line pool, and route configuration. Routes are mostly generated by shortest path algorithms considering travel times [30] and demand pattern of locations ([32], [48], [58]). An interesting approach for generating routes is using expert knowledge via artificial-based search algorithms ([49], [48]). Selection of routes out of a line pool and route configuration are done by mostly genetic algorithm ([50], [51], [27], [52], [53]). Chakraborty and Dwivedi [32] and Petrelli [54] claim that genetic algorithm is the most effective metaheuristic for TNDP. However, recent studies show that stochastic beam search [55] and hyperheuristics [56] can also find good solutions at least for benchmark instances.

The studies in the literature make many simplifying assumptions about *route design* and *passenger transfers* in handling TNDP and there is a need for new models and approaches that are more realistic in solving real-world problems. To our knowledge, *there does not exist mathematical models that address passenger transfers without a route/line pool in neither bus transit planning setting nor rail transit planning setting.* Most mathematical programming-based studies do not present computational results for benchmark instances or specific transit networks because the models are hard to solve even for small instances. That has led researchers to use metaheuristics primarily [57]. Good quality solutions are obtained for small *Mandl* and *Mumford* instances by evolutionary algorithms of Mumford [27] and John et al. [58], the ad-hoc route generation algorithm of Kiliç and Gök [59], and the hyper-heuristic approach of Ahmed et al. [56]. Moreover, solution quality depends mainly on instance data and requires a lot of tuning effort. They are not flexible enough to easily incorporate realistic features and cannot provide information regarding the quality of the generated solutions. In this regard, we offer a novel mathematical programming model that addresses aforementioned issues. Moreover, we obtain solutions for instances based on a *real-world network with 204 nodes, 455 links, and 13338 OD demand pairs*, which is significantly larger than the problems in the literature, using an off-the-shelf software.

2.2 Problem Description

Consider an undirected network $G = (N, A)$ with node set $N = \{1, \dots, n\}$ representing stops and directed arc set A representing the roads/links between stops. An arc (i, j) between stops i and j exists only if i and j are adjacent and there is a direct road from i and j. Without loss of generality, we assume that the roads between i and j are bidirectional. A subset $S \subseteq N$ ($D \subseteq N$) is distinguished as the set of supply/origin (demand/destination) nodes. A node $i \in S$ generates flows $w_{ij} > 0$ for some $j \in D$, i.e., the number of passengers who would like to go from stop i to stop j. w_{ij} are represented in an origin-destination (OD) demand matrix and specified over a time period (e.g., hour, day, month). It is possible that a node i is in both S and D. For nodes i and j that are both in S and D, it is not necessary that $w_{ij} = w_{ji}$. For a supply node $k \in S$, the total outbound flow is $sup_k = \sum_{j \in D} w_{kj}$.

The passengers desire to go from their origins/boarding stops to their destinations on a single vehicle in the shortest possible time, which requires a route/line between each pair of stops in both directions. However, this is not practical from the perspective of the transit agency as it would be too costly. In this regard, the transit agency is to determine a route set T to provide transport service to passengers. Each route $t \in T$ is a simple path with no repetition of nodes and composed of a *starting node* (*terminal*), an *ending node (terminal)*, a set of *intermediate nodes*, and *a set of arcs* $(i, j) \in A$ connecting the nodes in t.

Passengers departing from i and destined to j may go *directly* on a single route if a route passing through i and j exists. Otherwise, passengers may go *indirectly* to their destination by changing route(s) at *transfer points* where two or more routes coincide. Because making transfers is annoying for passengers, a *transfer discomfort penalty* (in time units) is used to control the number of transfers. Thus, the *total travel time* for passengers to arrive at their destination from their origin is the sum of (1) *in-vehicle travel time* and (2) *total penalty time resulting from transfers*. The lower the total travel time and the number of transfers, the better for passengers. Defining c_{ij} as the travel time (cost) from stop i to stop j, *in-vehicle travel time* is the sum of c_{ij} for all (i, j) on which passengers travel. c_{ij} are known and represented in an OD cost matrix.

The transit agency needs to design a system considering the passengers' perspective. Specifically, the transit agency should attempt to minimize in-vehicle travel and to maximize direct-traveler passengers (i.e., minimize the number of transfers). However, considering only the passengers' perspective may be too costly for the transit agency. In this regard, it should also try to minimize the costs. In doing so, the transit agency needs to make several decisions, e.g., the number of routes to operate, the number of stops on a route, and the percentage of direct travelers. Depending on how several objectives are prioritized, the resulting systems may significantly differ; hence, an analysis of different scenarios needs to be conducted to devise a system balancing both perspectives.

2.3 The Proposed Model

We propose a mathematical programming model, namely, *Transit Network Design Model* (TNDM), to solve TNDP. We define TNDM on an extended network $GT = (NT, AT)$ obtained by adding (1) two dummy nodes to N, namely, a and b that act as a super source node and a super sink node, respectively, and (2) two sets of directed arcs to A, namely, $A_{source} = \{(a, i): i \in N\}$ and $A_{sink} = \{(i, b): i \in N\}$. That is, $NT = N \cup \{a\} \cup \{b\}$ and $AT = E \cup A_{source} \cup A_{sink}$. We assume that all nodes are numbered, with a and b having the smallest and largest numbers, respectively. *Figure 2.1* gives a schematic representation of the extended network $GT = (NT, AT)$.

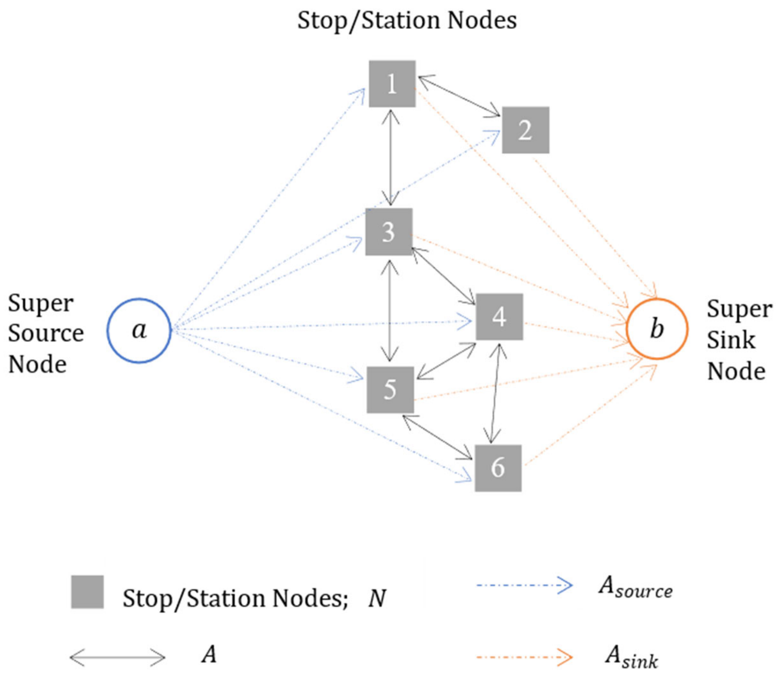


Figure 2.1 Representation of a transit network GT=(NT,AT)

All constructed routes start at node a and end at node b. The model constructs a route $t \in T$ from scratch by selecting an arc $(a, i) \in A_{source}$, an arc $(j, b) \in A_{sink}$, and a set of arcs from A that forms a path from i and j, where nodes i and j are the starting and ending terminals of the route t, respectively. *Figure 2.2* depicts the construction of routes in the model. In the figure, there are two routes: a-4-3-1-2-b and a-1-3-5-6-b. Because a and b are dummy nodes, the routes on the real physical network are 4-3-1-2 and 1-3-5-6. A similar approach that uses dummy nodes for selecting terminal nodes for routes can be found in the work of An and Lo [104].

To construct routes, we define binary decision variables d_{ijt}. Since a route is bidirectional, we define d_{ijt} for arcs $(i,j|i,j \in N \wedge i < j)$ to reduce the number of decision variables. If $d_{ijt} = 1$, then nodes i and j are consecutive stops in route t and passenger flow is allowed in both directions (i,j) and (j,i) on the route. For $d_{ait} = 1$ and $d_{jbt} = 1$, stops i and j are the starting and ending terminals of route t. Another variable set associated with route construction is y_{it} that take on the value of 1 if $i \in N$ is a part of route t and 0, otherwise. It is possible to impose additional requirements on the routes, e.g., the minimum (maximum) number of stops in a route and the maximum distance of a route. The decision variables f_t and h_t represent the frequency and headway of a route t, respectively. In addition to the variables above, we define (1) x_{ijkt} that represent the flow of passengers of origin $k \in S$ in arc (i,j) (i.e., traveling from stop i to stop j) on route t, (2) r_{ijkt} that represent the number of passengers of origin $k \in S$ who transfer at node i to route t with the next stop being node j.

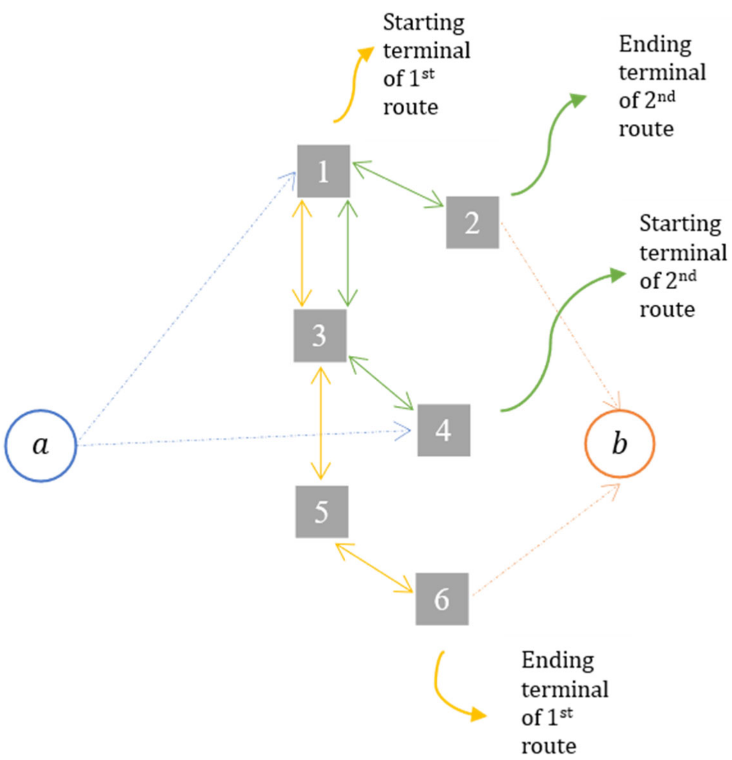


Figure 2.2 Illustration of the route design in a transit network, GT

Below we summarize the sets, parameters, and decision variables used in the model.

Sets and Indices

T — $set\ of\ routes\ (t \in T)$

N — $set\ of\ stops\ (i, j, k \in N)$

A — $set\ of\ arcs\ (i, j)$

S — $set\ of\ departure/supply\ stops\ (S \subseteq N)$

a — $super\ source\ node\ for\ routes$

b — $super\ sink\ node\ for\ routes$

A_{source} — $the\ set\ of\ directed\ arcs\ of\ the\ form\ (a, i), i \in N$

A_{sink} — $the\ set\ of\ directed\ arcs\ of\ the\ form\ (i, b), i \in N$

NT — $node\ set\ of\ extended\ network\ GT = (NT, AT)\ with\ NT = N \cup \{a\} \cup \{b\}$

AT — $arc\ set\ of\ extended\ network\ GT = (NT, AT)\ with\ AT = A \cup A_{source} \cup A_{sink}$

D_k — $set\ of\ arrival\ (destination)\ nodes\ for\ passengers\ of\ origin\ k \in S$

Parameters

sup_k — $the\ number\ of\ passengers\ of\ origin\ \ k \in S$

w_{ik} — $the\ number\ of\ passengers\ of\ origin\ k \in S\ with\ destination\ i \in N$

c_{ij} — $travel\ time\ from\ stop\ i\ \ to\ stop\ j$

$penalty$ — $transfer\ penalty\ (in\ time\ units)$

$topStop$ — $maximum\ number\ of\ stops\ allowed\ for\ a\ route$

$lowStop$ — $minimum\ number\ of\ stops\ allowed\ for\ a\ route$

M — $a\ big\ number\ enough\ to\ allow\ passenger\ flow\ in\ uncapacitated\ edges$

Decision variables

x_{ijkt} — $the\ flow\ of\ passengers\ of\ origin\ k\ who\ travel\ from\ i\ to\ j\ on\ route\ t$

d_{ijt} — $1, if\ arc\ (i, j), i < j, is\ selected\ to\ be\ in\ route\ t;\ \ 0, otherwise$

y_{it} — $1, if\ stop\ i\ is\ in\ route\ t; 0, otherwise$

r_{ijkt} — $the\ number\ of\ passengers\ of\ origin\ k\ who\ transfer\ at\ node\ i\ to\ route\ t$ $with\ next\ stop\ being\ node\ j$

Objective Function Terms:

$$in - vehicle\ travel\ time: \quad z_1 = \sum_{(i,j)\in A} \sum_{k\in S} \sum_{t\in T} (c_{ij} x_{ijkt}) \tag{2.1}$$

$$transfer\ time\ penalty: \quad z_2 = \sum_{(i,j)\in A} \sum_{k\in S} \sum_{t\in T} (penalty\ \ r_{ijkt}) \tag{2.2}$$

$$transit\ agency\ cost: \qquad z_3 = \sum_{\substack{(i,j)\in A;\\ i<j}} \sum_{k\in S} \sum_{t\in T} (c_{ij} d_{ijt}) \tag{2.3}$$

The objective function term (2.1) represents total in-vehicle travel time of passengers. (2.2) finds the total transfer time of passengers who transfer and is used to penalize passenger transfers assuming that each transfer will take a specific time, i.e., penalty time. Along with the TNDP literature [56], the objective term (2.3) is defined for reflecting the costs of the transit agency and sums up the total length (in time unit) of all routes.

Model TNDM: Transit Network Design Model (TNDM)

$$\min Z_{passenger} = z_1 + z_2 \tag{2.4}$$

$$\min Z_{transit\ agency} = z_3 \tag{2.4'}$$

$s.t.$

$$\sum_{i\in N} d_{ait} = 1 \qquad t \in T \tag{2.5}$$

$$\sum_{i\in N} d_{ibt} = 1 \qquad t \in T \tag{2.6}$$

$$\sum_{\substack{j\in N;\\ i<j \wedge (i,j)\in A}} d_{ijt} + \sum_{\substack{j\in N;\\ j<i \wedge (j,i)\in A}} d_{jit} = 2y_{it} \qquad i \in N, t \in T \tag{2.7}$$

$$\sum_{i\in N} \sum_{\substack{j\in N;\\ i<j \wedge (i,j)\in A}} d_{ijt} \le topStop - 1 \qquad t \in T \tag{2.8}$$

$$\sum_{i\in N} \sum_{\substack{j\in N;\\ i<j \wedge (i,j)\in A}} d_{ijt} \ge lowStop - 1 \qquad t \in T \tag{2.9}$$

$$\sum_{i\in N} \sum_{\substack{j\in N;\\ i<j \wedge (i,j)\in A}} d_{ijt} \le |\bar{S}| - 1 \qquad \begin{array}{l} \bar{S} \subset N; \\ 3 \le |\bar{S}| \le topStop - 1, \\ t \in T \end{array} \tag{2.10}$$

$$\sum_{\substack{j\in N;\\ (i,j)\in A}} \sum_{t\in T} x_{ijkt} - \sum_{\substack{j\in N;\\ (j,i)\in A}} \sum_{t\in T} x_{jikt} = \begin{cases} sup_k, & if\ i = k \\ -w_{ki}, & if\ i \in D_k \\ 0, & o.w \end{cases} \qquad i \in N, k \in S \tag{2.11}$$

$$\sum_{k \in S} (x_{ijkt} + x_{jikt}) \leq M d_{ijt} \qquad i,j \in N; i < j \ \wedge (i,j) \in A \quad t \in T \tag{2.12}$$

$$x_{ijkt} = \sum_{\substack{g \in N; \\ g \neq j \wedge (g,i) \in A}} x_{gikt} + r_{ijkt} \qquad i,j \in N;\ (i,j) \in A \quad k \in S;\ i \neq k \quad t \in T \tag{2.13}$$

$$r_{ijkt} \geq 0, \qquad x_{ijkt} \geq 0 \qquad i,j \in N;\ (i,j) \in A \quad k \in S, t \in T \tag{2.14}$$

$$d_{ijt} \in \{0,1\} \qquad (i,j) \in AT \quad t \in T \tag{2.15}$$

$$y_{it} \in \{0,1\} \qquad i \in N, t \in T \tag{2.16}$$

TNDM minimizes the costs of passengers (2.4) and the transit agency cost (2.4'). Constraints (2.5) require each route t to have an arc from the super source node a to a stop $i \in N$, which becomes the starting terminal of that route. Constraints (2.6) require each route t to have an arc from a stop $i \in N$ to the super sink node b with i being the ending terminal of that route. If the starting and ending terminals are known in advance, they can be specified using (2.5) and (2.6). Constraints (2.7) ensure that each stop in a route is connected to two other nodes (stops or super nodes) as shown in *Figure 2.2*.

Constraints (2.8) and (2.9) impose upper and lower limits on the number of stops in a route, respectively. Constraints (2.10) eliminate subtours in the routes using the Dantzig–Fulkerson–Johnson (DFJ) subtour elimination formulation [60] where $\bar{S}$ is a subset of stops with the specified cardinality. The number of such constraints increases exponentially with the cardinality of the node set $|N|$ and hence they cannot be used directly unless $|N|$ is very small. Therefore, we add subtour elimination constraints (2.10) during the solution process as described in *Algorithm 2.1* only when the candidate solutions violate them.

Constraints (2.11) are flow conservation constraints of passengers at the stops and ensure that passengers move from their origins to their destinations through some routes. Constraints (2.12) couple passenger flow variables and arc selection variables and allow passenger flows only when an arc is selected to be in a route.

Constraints (2.13) are flow balance constraints for the transfer of passengers. They state that the number of passengers of origin k on arc (i,j) in route t is equivalent to the number of passengers already traveling on route t and the number of passengers that transfer to route t at node i and move to j. Finally, Constraints (2.14) - (2.16) define the decision variables. Passenger flow and transfer variables are defined to be nonnegative. Because the problem is strategic, fractional values regarding passenger flow and transfers may be accepted.

2.4 Solution Methodology

2.4.1 Subtour Elimination

The number of subtour elimination constraints (2.10) increases exponentially with the cardinality of the node set $|N|$ and hence they cannot be used directly while solving TNDM unless $|N|$ is very small. For this reason, we add constraints (2.10) during the solution process only when candidate solutions violate them. The basic idea is to solve TNDM without constraints (2.10) and add constraints (2.10) that eliminate subtours, i.e., the cuts, whenever there is an integer solution with subtours.

When TNDM is solved without constraints (2.10), subtours may occur in a route $t \in T$. For each integer solution for a route set T, we can check and quickly identify subtours and add specific cuts (2.10) to separate them. We run this intervention procedure within a Lazy Constraints Callback function of *Gurobi*. *Lazy Constraints* are constructed when the user defines a violation for an integer solution. Algorithm 2.1 summarizes the steps of the intervention procedure.

Algorithm 2.1: Subtour Elimination

Step 0: *Start solving TNDM without constraints (2.10) using Gurobi. Due to the formulation consisting of integer decision space, the solver unfolds a branch and bound tree.*

Step 1: *When the solver finds an integer solution in any node of the branch and bound tree, run the Lazy Constraints Callback function defined for detecting subtours.*

Step 2: *If the callback function finds subtour(s) in the integer solution, go to step 3. Otherwise, go to step 4.*

Step 3: *Add corresponding subtour elimination constraints (2.10) for violating integer solution.*

Step 4: *Continue exploring the branch and bound tree nodes.*

2.4.2 Node Relaxation Heuristic

Gurobi solve TNDM and can find good solutions for small networks such as the *Mandl* network. However, it struggles to find a solution on large-scale networks. Therefore, we activate the *Node Relaxation Heuristic* (NoRelHeur) parameter of Gurobi. NoRelHeur is a heuristic algorithm that is especially useful when the root node relaxation is time-consuming in the Branch&Cut solution procedure. NoRelHeur can help the model find and improve the integer upper bound. However, it cannot help to improve the lower bound.

2.4.3 Decomposition of the Transit Network Design Model

To improve the solvability of TNDM, we adopt the *Benders decomposition* (BD) approach [25]. BD has been used in versatile forms for a wide variety of decision problems [61]. The main idea behind BD is to decompose the *original problem* into a *subproblem* (SP) and a *master problem* (MP) and solve these two problems separately by providing necessary information from one to the other. Generally speaking, MP consists of the part with the *complicating variables* out of the original problem. The complicating variables mostly refer to *integer variables*. The SP consists of the *dual* of the residual of the original problem. The reason for using duality in SP is to project out MP's complicating variables onto SP's objective so that information updates regarding the solution quality across MP and SP can be provided. SP becomes a linear program because the complicating variables are found and fixed by MP. MP gives solutions for complicating variables MP and they are checked by SP according to the criteria of *feasibility* and *optimality*. By the duality theorem and given an MP solution;

(1) If SP is *unbounded*, then the primal is *infeasible*. To prevent this infeasibility, we enumerate *extreme rays* out of the unbounded solution of SP and add them as *feasibility cuts* to MP.

(2) If SP is *feasible*, then the primal is *feasible*. So, the *extreme points* of SP can be used to generate cuts which can improve the upper bound. These cuts are called *optimality cuts*.

There are two approaches in implementing BD [95], namely the *classical approach* and *Branch & Benders Cut* (B&BC). In the *classical approach*, MP and SP are solved iteratively within a loop, and they provide primal and dual bounds, respectively. Because MP has integer variables, a new branch and bound (B&B) tree is generated in the solution of MP in each iteration. Theoretically, the primal and dual bounds will converge in a finite number of iterations.

In the B&BC approach, also called the modern approach, MP is solved via a single B&B tree. MP generates solutions along with the tree nodes, and SP generates *feasibility* and *optimality cuts* in return for given solutions by MP. These cuts are added to MP and some portions of the tree are pruned. In the B&BC procedure, MP provides both primal and dual bounds since SP is used only for generating the cuts needed in MP. Once MP is optimally solved, BD is deemed to reach convergence. Recent studies ([62], [63]) show that B&BC procedure can expedite solution times and improve bounds more efficiently compared to the classical iterative approach. We adopt the modern B&BC procedure to exploit these benefits.

We can decompose TNDM into SP and MP by defining the following variables. We obtain SP by taking the dual of constraints in TNDM for fixed values of integer variables.

$New\ Variables$

λ_{ik}	$i \in N, k \in S$	$dual\ variable\ corresponding\ to$ $flow\ conservation\ constraints$ (2.10)
μ_{ijt}	$i \in N, j \in N;$ $i < j$	$dual\ variable\ corresponding\ to$ $linking\ constraints$ $(2.11 - 2.12)$
θ_{ijkt}	$i \in N, j \in N, k \in S,$ $t \in T;\ i! = k$	$dual\ variable\ corresponding\ to$ $transfer\ constraints$ (2.13)
π_{ijt}	$i \in N, j \in N,$ $t \in T;\ i < j$	$a\ slack\ variable\ to\ use\ in\ SP$
η		*a variable for the MP's objective function; an underestimator of the total cost.*

$SubProblem\ (SP)$

$$\max Z = \sum_{i \in N} \sum_{k \in S} (sup_k \lambda_{ik;\, i=k} - w_{ik} \lambda_{ik;\, i \neq k}) + \sum_{i \in N} \sum_{\substack{j \in N; \\ i<j}} \sum_{t \in T} \pi_{ijt} \left(\bar{d}_{ijt} \right) \tag{2.17}$$

$s.t.$

$$\lambda_{ik} - \lambda_{jk} - \sum_{r \in N;\ r \neq i} (\theta_{jrkt}) + \mu_{ijt} \leq c_{ij} \qquad i \in N, j \in N, k \in S, t \in T; \; i = k, j \neq k, i < j \tag{2.18}$$

$$\lambda_{ik} - \lambda_{jk} - \sum_{r \in N;\ r \neq i} (\theta_{jrkt}) + \mu_{jit} \leq c_{ij} \qquad i \in N, j \in N, k \in S, t \in T; \; i = k, j \neq k, j < i \tag{2.19}$$

$$\lambda_{ik} - \lambda_{jk} + \theta_{ijkt} - \sum_{r \in N;\ r \neq i} (\theta_{jrkt}) + \mu_{ijt} \leq c_{ij} \qquad i \in N, j \in N, k \in S, t \in T; \; i \neq k, j \neq k, i < j \tag{2.20}$$

$$\lambda_{ik} - \lambda_{jk} + \theta_{ijkt} - \sum_{r \in N;\ r \neq i} (\theta_{jrkt}) + \mu_{jit} \leq c_{ij} \qquad i \in N, j \in N, k \in S, t \in T; \; i \neq k, j \neq k, j < i \tag{2.21}$$

$$\lambda_{ik} + \theta_{ijkt} + \mu_{ijt} \leq c_{ij} \qquad i \in N, j \in N, k \in S, t \in T; \; i \neq k, j \neq k, i < j \tag{2.22}$$

$$\lambda_{ik} + \theta_{ijkt} + \mu_{jit} \leq c_{ij} \qquad i \in N, j \in N, k \in S, t \in T; \; i \neq k, j \neq k, j < i \tag{2.23}$$

$$-M - \mu_{ijt} + \pi_{ijt} = 0 \qquad i \in N, j \in N, t \in T; \; i < j \tag{2.24}$$

$$-\theta_{ijkt} \leq penalty \qquad i \in N, j \in N, k \in S, t \in T; \; i \neq k \tag{2.25}$$

$$\lambda_{ik}, \ \ free\ variable \qquad i \in N, k \in S \tag{2.26}$$

$$\pi_{ijt}, \ \ free\ variable \qquad i \in N, j \in N, t \in T; \; i < j \tag{2.27}$$

$$\theta_{ijkt} \leq 0 \qquad i \in N, j \in N, k \in S, t \in T; \; i \neq k \tag{2.28}$$

$$\mu_{ijt} \leq 0 \qquad i \in N, j \in N, t \in T; \; i < j \tag{2.29}$$

$Master\ Problem\ (MP)$

$$\min\ \eta \tag{2.30}$$

$s.t.$

$$\sum_{i \in N} d_{ait} = 1 \qquad t \in T \tag{2.31}$$

$$\sum_{i \in N} d_{ibt} = 1 \qquad t \in T \tag{2.32}$$

$$\sum_{\substack{j \in N; \\ i<j \wedge (i,j) \in A}} d_{ijt} + \sum_{\substack{j \in N; \\ j<i \wedge (j,i) \in A}} d_{jit} = 2y_{it} \qquad i \in N, t \in T \qquad \textbf{(2.33)}$$

$$\sum_{i \in N} \sum_{\substack{j \in N; \\ i<j \wedge (i,j) \in A}} d_{ijt} \leq topStop - 1 \qquad t \in T \qquad \textbf{(2.34)}$$

$$\sum_{i \in N} \sum_{\substack{j \in N; \\ i<j \wedge (i,j) \in A}} d_{ijt} \geq lowStop - 1 \qquad t \in T \qquad \textbf{(2.35)}$$

$$\sum_{i \in N} \sum_{\substack{j \in N; \\ i<j \wedge (i,j) \in A}} d_{ijt} \leq |\bar{S}| - 1 \qquad \bar{S} \subset N; \; 3 \leq |\bar{S}| \leq topStop - 1, \; t \in T \qquad \textbf{(2.36)}$$

$$d_{ijt} \in \{0,1\} \qquad i,j \in NT, t \in T \qquad \textbf{(2.37)}$$

$$y_{it} \in \{0,1\} \qquad i \in N, t \in T \qquad \textbf{(2.38)}$$

$$\eta \geq \sum_{i \in N} \sum_{k \in S} (sup_k \lambda^*_{ik;\, i=k} - w_{ik} \lambda^*_{ik;\, i \neq k}) + \sum_{i \in N} \sum_{\substack{j \in N; \\ i<j}} \sum_{t \in T} \pi^*_{ijt} \left(d_{ijt}\right) \qquad \forall \{\lambda^*_{ik}, \pi^*_{ijt}\} \in ext.pts.(SP) \qquad \textbf{(2.39)}$$

$$0 \geq \sum_{i \in N} \sum_{k \in S} (sup^k \hat{\lambda}_{ik;i=k} - w_{ik} \hat{\lambda}_{ik;i \neq k}) + \sum_{i \in N} \sum_{\substack{j \in N; \\ i<j}} \sum_{t \in T} \hat{\pi}_{ijt} \left(d_{ijt}\right) \qquad \forall \{\hat{\lambda}_{ik}, \hat{\pi}_{ijt}\} \in ext.rays(SP) \qquad \textbf{(2.40)}$$

The objective function of MP (2.30) minimizes a surrogate variable η that is an underestimator for the objective function value of SP. MP finds the values of $\boldsymbol{d}$ which are fed into the objective function of SP (2.17) as $\bar{\boldsymbol{d}}$. Constraints (2.39) are optimality cuts added to the master problem when SP is feasible. Constraints (2.40) are feasibility cuts that are added when SP is unbounded. By finding a feasible solution for SP, we can identify its extreme points $\boldsymbol{\lambda}^*$ and $\boldsymbol{\pi}^*$. If SP is unbounded, we can find associated extreme rays $\hat{\boldsymbol{\lambda}}$ and $\hat{\boldsymbol{\pi}}$. Algorithm 2.2 describes the implementation steps of the BD decomposition algorithm.

Algorithm 2.2: Benders Decomposition of TNDM (B&BC Approach)

Step 0: *Start to solve MP; B&B tree will be generated. The solver will drop integrality requirements and relax MP at the root node of the B&B tree.*

Step 1: *Solve the relaxed MP in the root node of the B&B tree and update the lower bound,* $\underline{\eta}$.

Step 2: *Continue to explore the nodes of the tree. Once an incumbent integer solution for* $\boldsymbol{d}$ *is found, run the following subroutine within a Lazy Constraints Callback function.*

Step 2.1. *Run Algorithm 2.1 to detect whether there are any subtour(s). If there are any, add proper subtour elimination cuts to MP.*

Step 2.2. *Solve SP with the incumbent solution* $\overline{\mathbf{d}}$ *(which is fixed with the values of* $\boldsymbol{d}$*). If SP is feasible, then go to step 4. If it is unbounded, go to step 3.*

Step 3: *Add the feasibility cuts to the master problem and continue to the B&B procedure.*

- *If the master problem's lower and upper bounds converge, go to step 4.*
- *If the master problem is infeasible, conclude the algorithm with an error.*

Step 4: *Add the optimality cut to MP and update the incumbent solution. Then, continue the branch and bound procedure.*

- *If MP's lower and upper bounds converge, go to step 4.*
- *If MP is infeasible, conclude the algorithm with an error.*

Step 5: *End the algorithm successfully. The final incumbent solution and the corresponding subproblem solution are optimal.*

Algorithm 2.2 may not converge as desired speed. Therefore, we employ several enhancement strategies to improve the performance of the algorithm. Namely, we employ cut strengthening, disaggregating SP, and Pareto-optimal cut selection.

2.4.4 Benders Decomposition Enhancements

2.4.4.1 Cut Strengthening at the Root Node

Algorithm 2.2 may fail to generate good lower bounds with the default cut generation algorithm that relies upon Kelley[64]'s cut loop scheme. This scheme iteratively finds and adds violated cuts to separate fractional solutions of the relaxed problem and resolve the formulation in a loop sense. It can be ineffective to get good lower bounds when the constraint set is small [65]. This case generally occurs at the start of BD since MP has a small constraint set due to the decomposition. Moreover, if the first iterations do not produce good results, the lower bound may follow an *unstable trajectory*. There are different approaches for *stabilizing* lower bound such as *bundle method* [66] in which subgradient descent algorithm is used. An alternative method for *stabilization* is the *in-out scheme* and its variants are used in various BD implementations ([67], [65], [63]).

In the in-out scheme, three solution points are identified. The first point (d_{in}) is a feasible point in the interior of the convex hull of the decision space. The second point (d_{out}) is chosen at the boundary of the Linear Program (LP) hull. And the third point (d_{sep}) is called the separation point, which lies in the line between d_{in} and d_{out} points. Cuts violated by the d_{sep} are generated to separate it from the LP hull. Repeating this scheme over the resulting polyhedron in each iteration may improve the lower bound more than Kelley's scheme, which is the default cutting plane algorithm in most off-the-shelf solvers (e.g., Gurobi, CPLEX). Experimental results obtained by Fischetti et al. [65] and Zetina et al. [63] support the effectiveness of the *in-out scheme* for improving the lower bound, especially at the root node of the B&B tree. Because have experienced similar slow progress in improving the lower bound with the default cutting plane algorithm, we implement an *in-out scheme* at the root node as defined in *Algorithm 2.3.*

Algorithm 2.3: In-out Scheme Implementation

Step 0: *Obtain the relaxed MP (MP_Rel) by dropping the integrality requirements on the* d.

Step 1: *Solve MP_Rel with the Barrier method to obtain an inner feasible point* d_{in}.

Step 3: *Solve MP_Rel with the Dual simplex (or Simplex) algorithm to obtain an outer feasible point* d_{out}.

- *Update the lower bound,* $\underline{\eta}^{current}$
- If $\left|\underline{\eta}^{current} - \underline{\eta}^{previous}\right| > 0$, *then go to Step 4, otherwise go to Step 6.*

Step 4: *Find the separation point* d_{sep} *that lies on the line between* d_{in} *and* d_{out}.

- *Select a weight parameter* $\alpha \in (0,1)$.
- $d_{sep} = \alpha d_{out} + (1-\alpha) d_{in}$
- Update the inner point $d_{in} = d_{sep}$

Step 5: *Solve SP with fixed* d_{sep} *values and add feasibility (or optimality) cuts to MP.*

Step 6: *End the algorithm by leaving the root node and start to explore the branch nodes.*

In *Algorithm 2.3*, we use the *Barrier method* which is an interior point algorithm to obtain an inner point. Since the *crossover* operation can convert interior points to the boundary of the LP hull, we disable the *crossover* in the Barrier. There is no certain guide for choosing a value for the weight parameter (α) in Step 3 of the algorithm. After testing different values of α, we find out that $\alpha \in [0.4, 0.8]$ give better results. The algorithm ends when MP's lower bound does not improve in the root node.

2.4.4.2 Disaggregation of Subproblem

Solving SP with *Dual Simplex* or *Simplex* algorithms in a reasonable amount of time is not possible for large instances. An alternative approach is to use the *Barrier method*; however, when SP is unbounded, detecting unboundedness may be too time-consuming with the Barrier method. Therefore, we propose disaggregating SP at each stop to reduce its complexity.

To disaggregate SP, we replace π_{ijt} and μ_{ijt} of SP with the decision variables π_{ijkt} for $i \in N, j \in N, k \in S, t \in T;\ i! = k, i < j$ and μ_{ijkt} for $i \in N, j \in N, k \in S;\ i < j$, respectively.

Accordingly, we modify the objective function (2.17) as follows to obtain the disaggregated version (2.41).

$$\max Z^k = \sum_{i \in N} (sup_k \lambda_{ik;\, i=k} - w_{ik} \lambda_{ik;\, i \neq k}) + \sum_{i \in N} \sum_{\substack{j \in N; \\ i<j}} \sum_{t \in T} \pi_{ijkt} \left(\bar{d}_{ijt}\right) \qquad k \in S \tag{2.41}$$

With the disaggregated objective function for SP, optimality and feasibility cuts are defined as follows, respectively.

$$\eta^k \geq \sum_{i \in N} (sup_k \lambda^*_{ik;\, i=k} - w_{ik} \lambda^*_{ik;\, i \neq k}) + \sum_{i \in N} \sum_{\substack{j \in N; \\ i<j}} \sum_{t \in T} \pi^*_{ijkt} \left(d_{ijt}\right) \qquad \forall \{\lambda^*_{ik}, \pi^*_{ijkt}\} \in ext.pts.(SP), k \in S \tag{2.42}$$

$$\mathbf{0} \geq \sum_{i \in N} (sup^k \hat{\lambda}_{ik;i=k} - dem_i^k \hat{\lambda}_{ik;i\neq k}) + \sum_{i \in N} \sum_{\substack{j \in N; \\ i<j}} \sum_{t \in T} \hat{\pi}_{ijkt} \left(d_{ijt}\right) \qquad \forall \{\hat{\lambda}_{ik}, \hat{\pi}_{ijkt}\} \in ext.rays(SP), k \in S \tag{2.43}$$

Objective function of MP is changed to $\sum_{k \in S} \eta^k$.

2.4.4.3 Selection of Pareto-optimal Cuts

SP may have alternative optima; hence the corresponding Benders cuts may slow down the progression of BD and adversely affect the performance of BD. This stems from the primal degeneracy seen in most network design problems [63]. A seminal paper by Magnanti and Wong [68] suggest using *Pareto-optimal* Benders cuts to overcome this problem. An optimality cut can be defined as Pareto-optimal only if it dominates others by the condition if $f(\lambda') + g(\pi') \geq f(\lambda^*) + g(\pi^*)$ where $\eta \geq f(\lambda^*) + g(\pi^*)$ and $\eta \geq f(\lambda') + g(\pi')$ for all (λ^*, π^*).

Pareto-optimal cuts can be identified by a mathematical model using relative interior points of the convex hull of the master problem, namely, core points. We formulate the following mathematical model to find Pareto-optimal cuts for the proposed decomposition. The model is defined on the decision space of each disaggregated subproblem.

Model PRT: Mathematical Model for Selection of Pareto Optimal Cuts

$$\max \; R^k = \sum_{i \in N} (sup_k \lambda_{ik;\, i==k} - w_{ik} \lambda_{ik;\, i\neq k}) + \sum_{i \in N} \sum_{\substack{j \in N; \\ i<j}} \sum_{t \in T} \pi_{ijkt} \left(\mathrm{d}_{ijt}^0\right) \qquad k \in S \tag{2.44}$$

$$\sum_{i \in N} (sup_k \lambda_{ik;\, i==k} - w_{ik} \lambda_{ik;\, i\neq k}) + \sum_{i \in N} \sum_{\substack{j \in N; \\ i<j}} \sum_{t \in T} \pi_{ijkt} \left(\bar{d}_{ijt}\right) = Z^{k^*} \qquad k \in S \tag{2.45}$$

(2.18) – (2.29)

The objective function (2.44) for each k aims to maximize the left-hand-side of the corresponding optimality cut given in (2.42) by using the core points $\boldsymbol{d^0}$. The constraint set for each k contains the constraints (2.18) – (2.29) from the corresponding disaggregated SP and also an equality constraint (2.45) to make sure that obtained solutions must yield corresponding disaggregated SP solution Z^{k^*} (i.e., obtained solutions must be feasible extreme points of corresponding SP). Pareto-optimal cuts are generated similarly to optimality cuts (2.42) except that the values of dual variables λ and π are obtained by PRT.

Core points (d^0) can be generated from the relative interior of the convex hull of MP [68]. Papadokos [69] proves that the core point does not have to be an interior point. Similarly, we observe that points from the integer hull of MP can lead to better Pareto-optimal cuts. We update core points once MP finds a new incumbent solution. This update procedure is done as follows:

- Initialize the core point d^0 with a point in the integer hull of MP.
- Update the core point with a weight parameter (α) when the incumbent solution is updated where the current master solution is $\bar{d}$:

$d^0 = \alpha d^0 + (1 - \alpha)\bar{d}$

Weight parameter α is chosen between 0 and 1. Experiments show that $\alpha \in (0.1, 0.3)$ yields the best results.

2.4.5 Benders Decomposition - based Heuristic

Despite the enhancements in the decomposition algorithm, challenges persist with scaling to larger instances, which is the primary rationale for the initial implementation of decomposition strategy. As the instance size increases, the master problem's size correspondingly increases as well. Consequently, we introduce a Benders decomposition-based heuristic (BDH) approach aimed at diminishing the size of the master problem. This approach reduces the input dimensions -specifically OD distance matrix- for the master problem, thereby curtailing its overall complexity and size. Preprocess in an OD distance matrix is made by the Algorithm 2.4 as follows:

Algorithm 2.4: Preprocess of OD distance matrix.

Step 0: Initialization

- *Set Repetition Counter: Initialize a repetition counter, REP, to control the number of iterations.*

Step 1: Record Arcs

- *Use Algorithm 2.2 (with the enhancements by Algorithm 2.3, disaggregation of subproblem and selection of Pareto-optimal cuts) to solve MP and subsequently solve the disaggregated SPs.*
- *If the solution to the disaggregated SPs is optimal, record the solution for* $\boldsymbol{d}$. *Identify and record all unique arcs* (i, j) *in the solution where* d_{ijt} *equals to* 1.0. *This action is to note the arcs actively utilized in the solution, distinct from the route index. Then, go to Step 2.*
- *If an optimal solution for SPs isn't found, iterate adjustments, and return to solving MP until optimality for SPs is achieved.*

Step 2: Obtain Set of Arcs

- *Compile all recorded unique arcs from each iteration into a set of arcs called 'reducedArcs'.*
- *Repeat the Step 1 until recording unique arcs REP times. Once REP iterations are completed, go to step 3.*

Step 3: *Obtain OD_condensed by refining the OD distance matrix of the instance, which involves nullifying (or emptying) entries that do not correspond to the 'reducedArcs'.*

Algorithm 2.4 effectively generates a condensed version of the (OD) distance matrix, pertinent to the problem instance. This condensed matrix, referred to as OD_condensed, significantly narrows the decision space for the master problem, thereby enhancing computational efficiency and tractability. Original OD distance matrix is replaced by OD_condensed for the Benders decomposition-based heuristic (BDH) that is given in Algorithm 2.5.

Algorithm 2.5: Benders decomposition-based heuristic (BDH)

Step 0: *Disaggregate the SP and generate subproblems (SPs).*

- *Start to solve MP; B&B tree will be generated. The solver will drop integrality requirements and relax MP at the root node of the B&B tree.*

Step 1: *Solve the relaxed MP in the root node of the B&B tree and update the lower bound,* $\underline{\eta}$ *by utilizing in-out scheme of Algorithm 2.3.*

Step 2: *Continue to explore the nodes of the tree. Once an incumbent integer solution for* $\boldsymbol{d}$ *is found, run the following subroutine within a Lazy Constraints Callback function.*

- Step 2.1. *Run Algorithm 2.1 to detect whether there are any subtour(s). If there are any, add proper subtour elimination cuts to MP.*

- Step 2.2. *Solve SPs with the incumbent solution* $\bar{\boldsymbol{d}}$ *(which is fixed with the values of* $\boldsymbol{d}$*). If SP is feasible, then go to step 4. If it is unbounded, go to step 3.*

Step 3: *Add the feasibility cuts to the master problem and continue to the B&B procedure.*

- *If the master problem's lower and upper bounds converge, go to step 4.*
- *If the master problem is infeasible, conclude the algorithm with an error.*

Step 4: *Utilize the model PRT for selecting Pareto-optimal cuts and add Pareto-optimal cuts to MP and update the incumbent solution. Then, continue the branch and bound procedure.*

- *If MP's lower and upper bounds converge, go to step 4.*
- *If MP is infeasible, conclude the algorithm with an error.*

Step 5: *End the algorithm successfully. The final incumbent solution and the corresponding subproblem solution are optimal.*

2.5 Computational Results on Benchmark Instances

We have conducted tests using benchmark instances from the literature, namely Mandl and Mumford instances, and using a real-world transit network in Kayseri. Proposed models are coded with JuMP [70] mathematical modeling package in Julia [71] programming language. Experiments are conducted by Gurobi 9.5 solver.

TNDM solution files for the tests for Mandl, Mumford, and Kayseri204 instances can be found at the following repository. (https://aguedutr-my.sharepoint.com /:f:/g/personal/abdulkerim_benli_agu_edu_tr/Eq4v3KjxFFlAvNb9UOI2lDoBokeWcfpk p4VpbBLnKUOrxw, accessed on December 20, 2023).

2.5.1 Tests for Mandl Network

Mandl network [72] has been obtained by sampling out of the transit network of a Swiss town and consists of 15 nodes/stops and 21 edges. The geographical coordinates of the stops are not known. However, the network layout can be depicted as in *Figure 2.3* by the spectral layout algorithm [73]. The size of a node is proportional to the number of neighborhoods of that node. Features of the instances based on Mandl network are given in *Table 2.1*.

Table 2.1 Features of Mandl Transit Network

Network	Number of nodes/ edges	Number of routes	Number of stops per route (max.)
Mandl	15 / 21	4,6,8,10,12	8, 14
Transfer Penalty (mins.)	**Number of non-zero OD demand pairs**	**Total Passenger Demand**	**Demand period (mins.)**
5	172	15570	60

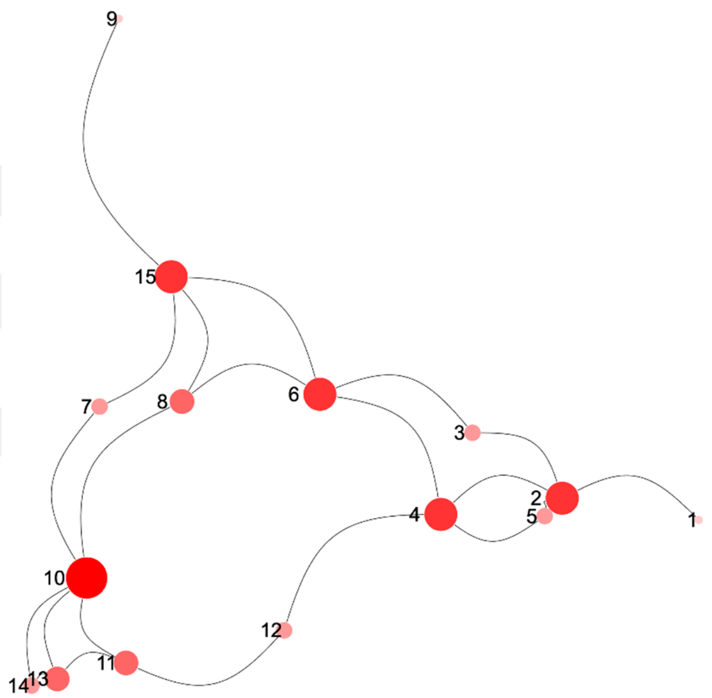


Figure 2.3 Layout of Mandl Transit Network

The OD demand matrix of *Mandl* is given in Table 2.2. Total passenger demand is 15570 and the number of non-zero OD demand pairs is 172. Except for *stop 15*, each stop is a departure point. The OD distance matrix of the network is given in *Table 2.3* as the distance unit is in *minutes*. A distance is recorded in the matrix if only there is a direct connection between two stops.

Table 2.2 Origin – Destination (OD) Demand Matrix Table

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	0	400	200	60	80	150	75	75	30	160	30	25	35	0	0
2	400	0	50	120	20	180	90	90	15	130	20	10	10	5	0
3	200	50	0	40	60	180	90	90	15	45	20	10	10	5	0
4	60	120	40	0	50	100	50	50	15	240	40	25	10	5	0
5	80	20	60	50	0	50	25	25	10	120	20	15	5	0	0
6	150	180	180	100	50	0	100	100	30	880	60	15	15	10	0
7	75	90	90	50	25	100	0	50	15	440	35	10	10	5	0
8	75	90	90	50	25	100	50	0	15	440	35	10	10	5	0
9	30	15	15	15	10	30	15	15	0	140	20	5	0	0	0
10	160	130	45	240	120	880	440	440	140	0	600	250	500	200	0
11	30	20	20	40	20	60	35	35	20	600	0	75	95	15	0
12	25	10	10	25	15	15	10	10	5	250	75	0	70	0	0
13	35	10	10	10	5	15	10	10	0	500	95	70	0	45	0
14	0	5	5	5	0	10	5	5	0	200	15	0	45	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2.3 Origin – Destination (OD) Distance Matrix Table

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	-	8	-	-	-	-	-	-	-	-	-	-	-	-	-
2	8	-	2	3	6	-	-	-	-	-	-	-	-	-	-
3	-	2	-	-	-	3	-	-	-	-	-	-	-	-	-
4	-	3	-	-	4	4	-	-	-	-	-	10	-	-	-
5	-	6	-	4	-	-	-	-	-	-	-	-	-	-	-
6	-	-	3	4	-	-	-	2	-	-	-	-	-	-	3
7	-	-	-	-	-	-	-	-	-	7	-	-	-	-	2
8	-	-	-	-	-	2	-	-	-	8	-	-	-	-	2
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8
10	-	-	-	-	-	-	7	8	-	-	5	-	10	8	-
11	-	-	-	-	-	-	-	-	-	5	-	10	5	-	-
12	-	-	-	10	-	-	-	-	-	-	10	-	-	-	-
13	-	-	-	-	-	-	-	-	-	10	5	-	-	2	-
14	-	-	-	-	-	-	-	-	-	8	-	-	2	-	-
15	-	-	-	-	-	3	2	2	8	-	-	-	-	-	-

Several test instances are generated by changing the number of routes (NR) and maximum number of stops allowed per route (MS). For comparison with the literature, the transfer penalty is taken as 5 minutes in the Mandl tests.

2.5.1.1 Results of Tests

We have used a laptop computer with a core i7@3.30 GHz and 16GB RAM for experiments with *Mandl*. We conduct two sets of experiments by solving TNDM with the *passenger perspective* (i.e., the objective function (2.4)) and the *transit agency perspective* (i.e., the objective function (2.4')), respectively.

The results for the first set of experiments are given in *Table 2.4.* All results are optimal with respect to *total travel time* (TTT). The *average travel time column* is calculated by dividing the TTT by the total passenger demand in the transit system. For instance, by operating with 4 routes, the *average travel time* (ATT) is $\frac{163210}{15570} = 10.482$ minutes. The *direct traveler* (DT) column represents the percentage of passengers who travel on a single route without making any transfer. The *transfer* column represents the percentage of passengers who make a transfer. For instance, only 8.156% of total passengers make a transfer when 4 routes are available and maximum number of stops per route is 8. For all instances, maximum number of transfers is found to be one (i.e., no passenger makes transfers more than one time). The *runtime* column shows the CPU time of the experiment. TNDM can reach optimality for the cases with the number of maximum stops per route is 8. Increasing the number of maximum stops to 14 prevents closing optimality gap, however we know that obtained upper bound values are optimal that are verified by optimal results given by Vermeir et al. [44].

It is worth mentioning that we can find optimal upper bound values with much shorter runtime. For instance, we obtain the optimal solution in 400 seconds for the 4-route and 8-stop case but wait for the remaining 16330 seconds to increase the lower bound to the optimal objective function value.

Table 2.4 Computational Test Results for Mandl Network on Passenger Perspective

MS*	NR	TTT (mins.)	AVT (mins.)	DT (%)	Transfer (%)	Runtime (s)	Gap (%)
8	4	163210	10.482	91.844	8.156	16770	0.0
	6	158500	10.18	97.174	2.826	132000	0.0
	8	156750	10.07	99.230	0.770	167000	0.0
	10	155940	10.02	99.936	0.064	140000	0.0
	12	155820	10.01	99.935	0.065	43000	0.0

14	4	159990	10.28	96.46	3.54	86400	2.63
	6	157040	10.09	99.11	0.89	86400	0.8
	8	156090	10.03	99.88	0.12	86400	0.19
	10	155840	10.01	100	0	86400	0.03
	12	155800	10.01	100	0	86400	0.01
* **MS**: Maximum number of stops per route; **NR**: Number of routes; **TTT**: Total travel time; **ATT**: Average travel time; **DT**: Direct travelers							

The results for the second set of experiments regarding *transit agency perspective* are given in *Table 2.5*. We obtain optimal solutions for all instances. Even though we increase the number of routes, the model does not put additional stops and sticks with the minimum required stops (i.e., just enough to satisfy passenger demand). Therefore, the optimal route length is the same for all cases.

Table 2.5 Computational Test Results for Mandl Network on Transit Agency Perspective

NR	Total length of routes (mins.)	Runtime (s)
4, 6, 8, 10, 12	63	421

The results on Mandl instances show that Gurobi can solve TNDM and find optimal results for small-scale problems. In the following, we present the results of the tests conducted on moderate- to large-scale networks.

2.5.2 Tests for Mumford Networks

Mumford instances consist of four different transit networks: *Mumford0, Mumford1, Mumford2,* and *Mumford3*. The number of nodes/stops (edges) changes from 30 (90) to 127 (425). Other features of the instances are given in *Table 2.6*. The coordinates of the stops and the demand period are unknown. However, layouts of the networks are depicted in *Appendix A*.

Table 2.6 Features of Mumford Transit Networks

Network	Number of nodes/edges	Number of routes	Number of stops per route (min./max.)
Mumford0	30 / 90	12	2 / 15
Mumford1	70 / 210	15	10 / 30
Mumford2	110 / 385	56	10 / 22
Mumford3	127 / 425	60	12 / 25
	Transfer Penalty (mins.)	**Number of non-zero OD demand pairs**	**Total Passenger Demand**
Mumford0	5	870	342160
Mumford1		4820	1926170
Mumford2		11990	4847900
Mumford3		16002	6394950

2.5.2.1 Results of Tests

We have used a workstation computer with 32 core and 256GB RAM for experiments with Mumford. The runtime for all tests is 86400 seconds by reserving 57500 seconds for the Node Relaxation Heuristic (NoRelHeur). *Mumford* instances are hard to solve due to the large number of nodes and non-zero OD demand pairs. There are only a few (meta)heuristic approaches in the literature that have used these networks. The best results for TNDP are obtained by the evolutionary algorithms of Mumford [27] and John et al. [58], the ad-hoc route generation algorithm of Kiliç and Gök [59], and the hyper-heuristic approach of Ahmed et al. [56]. In this regard, we compare the results of TNDM with those of the abovementioned studies/algorithms. The results for the first set of experiments are given in *Table 2.4*.

We conduct two sets of experiments with the *passenger perspective* and the *transit agency perspective*, respectively. The results obtained using Gurobi with Node Relaxation Heuristics (NoRelHeur) and Benders decomposition-based heuristic (BDH) are given for the passenger perspective in *Table 2.7* and for the transit agency perspective in *Table 2.8*. We do not employ BDH for transit agency perspective since we need equality sign in the equation 2.34 during the implementation of Benders decomposition (otherwise the Algorithm 2.5 may terminate early). Handling equation 2.34 with equality sign will increase the transit agency cost unnecessarily that will negatively effect of BDH results.

Table 2.7 Comparison of TNDM (solving by Gurobi with NoRelHeur and BDH) with Other Studies on Mumford Networks on Passenger Perspective

Instance	Model	Average travel time (mins.)	Runtime (s)	Optimality gap (%)
Mumford0	**BDH**	14.33	86400	9.24
	Gurobi with NoRelHeur	14.14	86400	7.97
	Ahmed et al. [56]	14.09	36000	-
	Kılıç and Gök [59]	14.99	-	-
	John et al. [58]	15.40	-	-
	Mumford [27]	16.05	-	-
Mumford1	**BDH**	24.63	8640	21.7
	Gurobi with NoRelHeur	21.91	86400	12.1
	Ahmed et al. [56]	21.70	36000	-
	Kılıç and Gök [59]	23.25	-	-
	John et al. [58]	23.91	-	-
	Mumford [27]	24.79	-	-
Mumford2	**BDH**	29.39	86400	24.57
	Gurobi with NoRelHeur	28.04	86400	20.94
	Ahmed et al. [56]	25.00	36000	-
	Kılıç and Gök [59]	26.82	-	-
	John et al. [58]	27.02	-	-
	Mumford [27]	28.65	-	-
Mumford3	**BDH**	30.88	86400	31.2
	Gurobi with NoRelHeur	29.70	86400	28.46
	Ahmed et al. [56]	27.89	36000	-
	Kılıç and Gök [59]	30.41	28800	-
	John et al. [58]	29.50	158400	-
	Mumford [27]	31.44	-	-

Table 2.8 Comparison of TNDM (solving by Gurobi with NoRelHeur) with Other Studies on Mumford Networks on Transit Agency Perspective

Instance	Model	Total length of routes (mins.)	Runtime (s)	Optimality gap (%)
Mumford0	**Gurobi with NoRelHeur**	94	86400	37.4
	Ahmed et al. [56]	94	36000	-
	Kılıç and Gök [59]	707	-	-
	John et al. [58]	745	-	-
	Mumford [27]	759	-	-
Mumford1	**Gurobi with NoRelHeur**	396	86400	5.3
	Ahmed et al. [56]	408	36000	-
	Kılıç and Gök [59]	1956	-	-
	John et al. [58]	1861	-	-
	Mumford [27]	2038	-	-
Mumford2	**Gurobi with NoRelHeur**	1266	86400	34.4
	Ahmed et al. [56]	1330	36000	-
	Kılıç and Gök [59]	5027	-	-
	John et al. [58]	5461	-	-
	Mumford [27]	5632	-	-
Mumford3	**Gurobi with NoRelHeur**	1805	86400	40
	Ahmed et al. [56]	1746	36000	-
	Kılıç and Gök [59]	5834	-	-
	John et al. [58]	6320	-	-
	Mumford [27]	6665	-	-

Tables 2.7 and *2.8* show that solving TNDM by BDH and Gurobi with NoRelHeur can obtain good-quality solutions. However, we observe that the lower bound does not improve in tests on Mumford instances. Therefore, optimality gap is high for all test cases. Except for Ahmed et al. [56], TNDM can mostly provide better results than other compared approaches considering average travel time and total length of routes. Obtained results on *Mandl* and *Mumford* networks verify TNDM's computational capability.

2.5.3 Tests for Kayseri204 Network

Kayseri204 instances are defined on a real bus transit network in the city of Kayseri, located in the Central Anatolian Region in Türkiye. Kayseri has a population of about 1.5 million. The current public transport system consists of tram and bus transit networks. The tram and bus networks intersect at certain points; however, the tram network passes through a limited number of streets and has limited capacity. The bus network is currently the main public transport system. The number of stops in both directions in the current bus transit system is over 3000, which is high for the size of a city such as Kayseri. In this regard, designing a trunk-and-feeder system is considered an alternative. Our goal in this study is to determine trunk lines/routes on the stops selected by Kayseri Transport Inc., considering demand intensity and location diversity. The features of the instances are given in *Table 2.9*. We assume a transfer penalty of 15 minutes, as in Arbex and da Cunha [97], Borndörfer and Karbstein [107].

Table 2.9 Features for Kayseri204 transit network instances

Number of nodes/ edges	**Number of routes**	**Node limits (min./max.)**	**Transfer Penalty (mins.)**
204 / 405	15, 20, 25	2 / 25	15
Number of non-zero OD demand pairs	**Total Passenger Demand**	**Demand period (mins.)**	
13338	205090	1000	

2.5.3.1 Results of Tests

We conduct tests for Kayseri 204 with Gurobi 9.5 solver on an Intel(R) Xeon(R) Gold 6150@2.70 GHz computer with 256GB RAM. The runtime for all tests is 86400 seconds by reserving 57500 seconds for the Node Relaxation Heuristic (NoRelHeur). The results obtained using Gurobi with NoRelHeur are given for the passenger perspective in *Table 2.10.*

Table 2.10 Results for TNDM (solving by Gurobi with NoRelHeur) on Kayseri204 Network on Transit Agency Perspective

Number of routes	Average travel time (mins.)	Runtime (s)	Optimality gap (%)
15	32.38	86400	36.24
20	31.01	86400	33.44
25	29.86	86400	30.86

2.7 Conclusion

In this chapter, we have addressed the *transit network design problem (TNDP)* and propose a novel mathematical programming model that incorporates features of real-world transit network systems (e.g., passenger transfers). The model is the first to design routes from scratch without utilizing a line pool and attempt to solve large-scale problems. The proposed model is difficult to solve for large-scale networks using Gurobi. Therefore, we develop a solution methodology based on Benders decomposition and Branch&Cut procedure. We employ several enhancement strategies, namely *in-out cut loop scheme, disaggregate the subproblem,* and *Pareto-optimal cuts,* to improve the performance of the solution methodology and finally develop a heuristic based on Benders decomposition.

We conduct experiments using Gurobi solver -with and without the NoRelHeur parameter activation- and Benders decomposition-based heuristic. We verify the performance and capability of the proposed model with small- and large-size well-known benchmark instances (i.e., *Mandl* and *Mumford*) and compare the results obtained by the model to those obtained with the state-of-art studies in the literature. Results show that Benders decomposition-based heuristic can provide solutions for large-scale TNDP whereas the original model cannot unless enabling NoRelHeur parameter of Gurobi. Lastly, we apply the model to a real-world transit network of Kayseri city in Türkiye.

The main insights from the results obtained by the proposed model can be summarized as follows:

- Increasing the number of routes has a marginal effect on the passengers' travel time. This phenomenon holds true across networks of different scales, as exemplified by the Mandl network with its limited number of stops, and the more extensive Kayseri204 network. Such findings suggest that increasing the number of routes is not the most effective strategy for reducing travel times.

- Considering that ineffectual impact, a shift in focus is recommended towards increasing the number of stops in routes. However, the decision to implement new routes should be carefully evaluated against the operational cost implications of lengthier existing routes. The balancing act between the fixed costs associated with the introduction of new routes and the variable costs due to extending number of stops in routes is crucial for optimizing network design.

Furthermore, employing the total length of routes as a proxy for transit agency costs may not fully capture the complex cost elements inherent in public transportation networks. A more sophisticated approach that considers a variety of operational and contextual factors would likely provide a more accurate reflection of the costs that transit agencies encounter.

We remark that the insights may change depending on the values of different parameters. However, the results indicate that the proposed model can be useful as a decision aid for designing a transit network system or evaluating a current system.

Chapter 3

Transit Network Design and Frequency Setting Problem

Transit Network Design and Frequency Setting Problem (TNDFSP) considers *network route design* and *frequency setting problems* simultaneously and aims at designing routes and frequencies of the routes to satisfy passenger demand given as an Origin-Destination (OD) demand matrix. Some literature refers to TNDFSP as the *line planning problem,* especially in railway network settings [13]TNDFSP is an NP-hard problem and computationally challenging even for small networks [16].

Durán-Micco et al. [18] give a recent survey of TNDFSP studies for bus and rail transit settings. Some of their findings are as follows: (1) Due to the complexity of real-world transit network settings, TNDFSP is hard to model for practical applications. Therefore, studies primarily focus on simplified networks [19]. (2) Many assumptions are made before appropriately addressing the problem, which results in different studies making different assumptions leading to different problems for which the results cannot be directly compared. (3) Case studies stemming from real-world networks simplify passenger demand and deviate from realism by only considering a proportion of OD demand pairs, i.e., a highly-sparse OD demand matrix is used [24]. (4) Frequencies are set iteratively, i.e., a predefined, initial set of frequencies are updated according to the passenger assignment rather than determining them endogenously in a model [49]. (5) Passenger assignment is done mainly by ignoring overcrowding. A few studies consider crowding issues and use incremental algorithms in which the frequency of crowded routes is updated iteratively. (6) There is a need for new models and approaches to add more realistic concepts and solve real-world problems.

In addition to these issues pointed out by Durán-Micco et al. [18], all studies except for_Wan and Lo [15] design routes out of a predefined route/line pool or use a route generation algorithm [12] with the risk of overlooking optimal routes. (Meta)Heuristic studies are mostly preferred because of the computationally challenging nature of the problem. Although metaheuristic approaches can provide good solutions for some benchmark instances, solution quality depends mainly on instance data and requires a lot of tuning effort. Moreover, they are not flexible enough to easily incorporate realistic features and cannot provide information regarding the quality of the generated solutions.

Considering the aforementioned issues, we are motivated to develop a novel mathematical model for TNDFSP that represents real-world transit network features. Rather than developing an exact solution methodology for the problem and focusing on computational performance, our goals are to solve the model using off-the-shelf commercial solvers, conduct what-if analyses to provide the decision-makers with managerial insights, and assess the flexibility and applicability of the models. In this regard, we propose a *multi-objective nonlinear mixed-integer programming* (MNMIP) model.

The contribution of this study is twofold: (1) We propose a novel mathematical programming formulation based on realistic concepts. The model reflects passengers' route choice realistically by considering *in-vehicle travel time*, *transfers*, and *waiting times* at boarding (departure) and transfer stops as well as *overcrowding and under-utilization of vehicles* in the passenger assignment. The model also reflects the transit agency's perspective by allowing the transit agency to determine service level by imposing limitations on several parameters such as *vehicle fleet size*. The model *endogenously determines routes* including their *terminal and intermediate stops* as well as their *frequencies*. To our knowledge, *the model is the first to determine routes and their frequencies simultaneously from scratch*, i.e., without using line and frequency pools while considering the aforementioned issues such as transfers and waiting. (2) We provide the results of *what-if analyses* conducted using a *real-world public bus transport network* in the city of Kayseri in Türkiye and the proposed model. We also present the results of computational tests implemented to validate and verify the model using *Mandl benchmark instances from the literature* [72]. The results obtained using Gurobi as the solver indicate that the model produces better solutions than the state-of-the-art algorithms in the literature and that the model can be used by public transit planners as a decision aid.

3.1 Literature Review

Guihaire and Hao [29], Kepaptsoglou and Karlaftis [30], Farahani et al. [16], Ibarra-Rojas et al. [74] and recently Durán-Micco et al. [18] provide a detailed review of TNDP and TNDFSP studies. That surveys indicate that the scope of studies varies significantly depending on the assumptions regarding objectives, solution approaches, parameters, and network settings. This thesis emphasizes studies that employ mathematical programming approaches; however, (meta)heuristics and analytical formulations will be briefly mentioned when necessary.

Mathematical programming-based TNDFSP studies are rare and mostly make assumptions that simplify the realistic aspects of the problem. The models in the literature essentially differ in how they handle (1) *route design*, (2) *frequency setting*, (3) *transfer and waiting at the stops*, (4) *passenger assignment to the routes,* and (5) *vehicle fleet size*. Most studies select the best routes from a *predefined route set* instead of generating routes from scratch using endogenous variables within the models. Marwah [75] generates a candidate route set using an ad-hoc heuristic procedure and proposes a linear mathematical program to determine the best routes out of this route set to minimize the number of transfers. Early studies such as Constantin and Florian [76], Dubois et al. [36], Furth et al. [77], and Lampkin and Saalmans [78] set frequencies as a *sequential step* after determining the route network instead of assigning them simultaneously with the route design. On the other hand, van Nes et al. [17] claim that the sequential solution procedure deviates from reality and propose a model that simultaneously addresses both route design and frequency setting. However, they consider only a set of candidate routes constructed using an algorithm described by Ceder and Wilson [4] and a limited number of frequencies. A different approach by van Oudheusden et al. [79] considers routes as facilities and uses set covering and facility location models to select efficient ones out of a candidate set.

Wan and Lo [15] introduce a mixed integer programming model that designs routes from scratch rather than selecting from a candidate route set. However, their model neither treats *transfers of passengers* nor considers *waiting times*. Guan et al. [80] model passenger transfers by adding expected transfer times to predefined paths for each OD demand pair. Therefore, they assign passengers to lines with minimum in-vehicle travel and transfer time along their path. De-Los-Santos et al. [21] propose a mathematical program formulation that uses decision variables to construct routes considering the walking option to make a transfer at different stops. However, frequency setting is made by parametric analysis rather than letting the model determines them.

Cancela et al. [12] utilize a *trajectory graph* by adding waiting and transfer arcs to the transit network to model transfer and waiting times. Szeto and Jiang [81] utilize a bilevel mathematical program in which the upper level minimizes the number of transfers considering the vehicle fleet size while the lower level assigns passengers to the routes designed in the upper level. Routes are generated between a predetermined set of terminal stops with the help of an artificial bee colony algorithm, and associated frequencies are improved with an iterative approach using a linear program.

Modeling waiting times at boarding and transfer stops affects *passenger assignment* to the routes. Even though there are typically no explicitly capacity limits on routes, setting frequencies for routes will indirectly impose capacity limits. That is, the maximum passenger load that can be carried on a route will depend on the route's frequency. Frequencies also indirectly determine the waiting times of passengers at a stop (via headways). Therefore, route choice of passengers depends on their perception of frequency, and route choice of passengers (i.e., assignment of passengers to routes) can be modeled by the principles of *passenger assignment problem* (PAP). PAP is an implicit problem within TNDFSP. However, there exist limited studies that address this issue. Spiess and Florian [82] suggest assigning passengers to the routes using the *frequency share rule*, which presumes that the probability that a vehicle arrives at a stop first is proportional to the frequency of its route. Hence, assuming passengers get on the first vehicle arriving at the stop, passengers are assigned to routes proportional to their frequencies. The frequency share method can lead to detours from shortest paths and overcrowding or under-utilization of the vehicles, which are eliminated later mainly through iterative frequency setting (e.g., Ahern et al. [31]). Cancela et al. [12] utilize this concept to model waiting times on the *trajectory graph* by associating waiting arcs with a predefined, finite set of frequencies for candidate routes. They test the model using only one of the *Mandl* benchmark instances and solve a simplified bus network with *84 stops* in Uruguay *without allowing transfers*. Alternatives to frequency share are *user equilibrium* and *system optimal* approaches. The *user equilibrium* approach (e.g., Wan and Lo [15]) ignores overcrowding issues since passengers of one OD pair are assigned to the same route(s) by following the shortest paths from origins to destinations. The *system optimal* approach (e.g., Zhou et al. [19]) assigns passengers to the routes to minimize the total travel time of all passengers on the network. When there is a limited number of vehicles, some passengers may be redirected from the shortest paths if it decreases the total travel time in the network.

The *line planning problem* (LPP) is analogous to the TNDFSP; however, it applies to railway transit networks. The LPP presupposes that the infrastructure and OD matrices are provided. It attempts to identify a *line concept* composed of line routes and their corresponding frequencies. LPP is conducted primarily through studies of mathematical programming. Hence, we will review the literature by filtering out decision variables. An early study by Claessens et al. [9] optimize lines, the number of cars and their types, and train frequencies for cost-effectiveness in a subnet of the Dutch railway network. They build a nonlinear integer cost function, then convert it into a linear integer program by using a binary variable, X_{ij}^{rtfc} which takes 1 if the line takes a route r in type t from station i to j with frequency f and cars c. By using that model, Bussieck [83] reduces the size with new strong cuts and data preprocessing, and offers linearization to discard the downside of the huge number of binary variables. Goosens et al. [8] introduce a *multi-type line planning problem* that allows trains not to halt at every station. They define an edge capacity for satisfying traffic load on each line type. Their multicommodity flow-based model can solve that edge capacity problem. Flow variable, F_{ij}^{k} represents the number of passengers of station k who travel between station i and j. Borndörfer et al. [84] try frequency alternatives for each line and claim that tighter bounds are found compared to the models where the frequency is determined by edge capacity constraint.

One of the quality measures for service level is the number of direct travelers. Bussieck et al. [85] construct a model representing direct travelers as an integer variable d_{ijl} for representing the number of travelers between terminal stations i and j on line l. Every traveler must use the shortest path or a combination of shortest paths between terminal stations. Thus, the line pool is generated from the shortest paths between terminals. Börndöfer and Pfetsch [86] introduce a continuous variable, y_{st}^{p} which represents the fraction of the demand between terminal stations $s\ and\ t$ along the path p. Paths and lines are predefined according to combinations of terminals. The compromising objective has two terms: frequency of the lines and weighted travel time of passengers. That model does not determine lines nor explicitly consider transfers. Instead, the model routes passengers along the predetermined lines considering the shortest paths and computes the required frequency level.

None of the LPP models treat transfers until Schöbel and Scholl [10]'s study, which model transfers by the *change & go network* where two nodes are consecutive stations of the same line or are the same station of different lines. Various costs can be assigned to edges in the change & go network, such as counting the number of transfers (i.e., an integer function) or measuring the transfer time (i.e., a continuous time function). They use a binary decision variable, x_{st}^{e} which is 1 if edge e is used on the shortest path along the trip between departure station s and arrival station t, and vice versa. In the objective function, demand from s to t is weighted with the edge cost function when x_{st}^{e} is 1. By this objective term, transfers are penalized (i.e., changing lines overweighted), and total travel time is minimized. However, that model selects lines from a predefined line pool.

Borndörfer et al. [87] define a flow variable y_p which is the number of passengers traveling from s to t on path p. A binary decision variable, x_l is used to replace lines over the paths. Their objective is the minimization of the travel time of passengers and lines' fixed and frequency costs. Borndörfer et al. [11] introduce a multicommodity flow model based on the work of Kim and Barnhart [88]. That model considers passengers as commodities between stations s and t using an arc a that is represented in a continuous decision variable y_a^{st}. Lines are constructed by using a binary line flow variable, z_a^r which represents line r flow passing through arc a. The passenger flow is nonnegative if the binary line flow is active on its path. That model predetermines lines' frequencies and start and end terminals of all line combinations. The advantage of the model is allowing the creation of lines from scratch. However, passenger transfers are ignored. Therefore, Borndörfer and Neumann [89] develop transfer utility within passenger paths by using the decision variable, y_{pk} which defines the number of passengers that travel on path p with at least transferring k times. Nevertheless, their model has some simplifications for transfers, as the number of transfers k is estimated but not optimally determined. In a recent LPP study, Zhou et al. [19] address transfers, waiting times, and frequency setting simultaneously using a line pool and solve a simplified version of the Hong Kong rail network consisting of *44 stations with 52 links*. Line frequencies are not considered in the initial passenger assignment; they are updated iteratively.

For solving LPP mathematical programs, branch and bound ([9], [85]), branch and cut [90], branch and price [11] are the popular approaches. To reduce the size of the problem, column generation and decomposition methods ([87], [91], [89], [10]) are used. For practical issues, variable fixing heuristics [92] and metaheuristics such as cross-entropy [93], and genetic algorithm [94] are used as well.

To sum up, the studies in the literature make many simplifying assumptions about *route design*, *frequency setting*, *transfers*, *waiting*, *passenger assignment*, and *vehicle fleet size* in handling the problem and there is a need for new models and approaches that are more realistic in solving real-world problems. In fact, only a few studies consider all the aforementioned issues. To our knowledge, *there does not exist mathematical models that address all issues simultaneously without a route/line pool in neither bus transit planning setting nor rail transit planning setting.*

Most mathematical programming-based studies do not present computational results for benchmark instances or specific transit networks because the models are hard to solve even for small instances. That has led researchers to use metaheuristics primarily. Iliopoulou et al. [57] and Durán-Micco et al. [18] survey the proposed metaheuristics. Good quality solutions are obtained for small *Mandl* instances that consist of only 15 nodes and 21 links by evolutionary algorithms ([95], [96]), genetic algorithm ([97], [98]), simulated annealing [20], memetic algorithm [99] and hybrid approaches which combine genetic, simulated annealing, tabu search, or greedy algorithms ([33], [100]–[102]). Ahern et al. [20] test their algorithm using large Mumford instances with the number of *nodes changing from 30 to 127* and the number of *links changing from 90 to 425* and obtain satisfactory results. However, like mathematical programming-based studies, metaheuristic-based approaches make many simplifying assumptions and only a few address all the abovementioned issues simultaneously. Moreover, solution quality depends mainly on instance data and requires a lot of tuning effort. They are not flexible enough to easily incorporate realistic features and cannot provide information regarding the quality of the generated solutions. In this regard, we offer an all-encompassing, novel mathematical programming model that addresses all the issues simultaneously. Moreover, we obtain solutions for instances based on a *real-world network with 204 nodes, 455 links,* *and 13338 OD demand pairs*, which is significantly larger than the problems in the literature, using an off-the-shelf software. Table 3.1 presents the properties of this study as well as those of mathematical programming- and heuristic-based studies in the literature.

The following proposed mathematical model for TNDFSP is built on the model TNDM described in Chapter 2. Since we adhere to similar descriptions, notations, and terminology for both models, some portions of the following sections can be the same as the corresponding parts of Chapter 2.

3.2 Problem Description

Consider an undirected network $G = (N, A)$ with node set $N = \{1, \dots, n\}$ representing stops and directed arc set A representing the roads/links between stops. An arc (i, j) between stops i and j exists only if i and j are adjacent and there is a direct road from i and j. Without loss of generality, we assume that the roads between i and j are bidirectional. A subset $S \subseteq N$ ($D \subseteq N$) is distinguished as the set of supply/origin (demand/destination) nodes. A node $i \in S$ generates flows $w_{ij} > 0$ for some $j \in D$, i.e., the number of passengers who would like to go from stop i to stop j. w_{ij} are represented in an origin-destination (OD) demand matrix and specified over a time period (e.g., hour, day, month). It is possible that a node i is in both S and D. For nodes i and j that are both in S and D, it is not necessary that $w_{ij} = w_{ji}$. For a supply node $k \in S$, the total outbound flow is $sup_k = \sum_{j \in D} w_{kj}$.

The passengers desire to go from their origins/boarding stops to their destinations on a single vehicle in the shortest possible time, which requires a route/line between each pair of stops in both directions. However, this is not practical from the perspective of the transit agency as it would be too costly. In this regard, the transit agency is to determine a route set T to provide transport service to passengers. Each route $t \in T$ is a simple path with no repetition of nodes and composed of a *starting node* (*terminal*), an *ending node (terminal)*, a set of *intermediate nodes*, and *a set of arcs* $(i, j) \in A$ connecting the nodes in t. On each route t, several vehicles with a specific capacity (cap_t) operate bidirectionally depending on the frequency (i.e., the number of vehicles per time unit) needed to satisfy the demand.

Table 3.1 Comparison of the proposed TNDFSP model with the studies in the literature

		Wan and Lo [15]	**Cancela et al.** [12]	**Zhou et al.** [19]	**Ahern et al.** [20]	**De-Los-Santos et al.**[21]	***This Thesis's TNDFSP model***
Route design		Endogenous	Line pool	Line pool	Route generation algorithm	Endogenous	Endogenous
Modeling Capability	Transfer Penalty	×	√	√	√	√	√
	Waiting Times	×	√	√	√	√	√
Frequency Setting		Endogenous	Selection out of a finite set	Approximation	Iterative	Parametric analysis	Endogenous
Passenger Assignment Rule		User equilibrium	Frequency share	System optimal	Frequency share	User equilibrium	System optimal
Solution Method		Off-the-shelf solver	Off-the-shelf solver	Off-the-shelf solver	Simulated annealing	Off-the-shelf solver	Off-the-shelf solver
Validation	Transit Type	Bus	Bus	Railway	Bus	Bus	Bus
	Features	A 10-node instance	A *Mandl* instance	A 44-node instance	*Mandl* and *Mumford* instances	10-, 15-, 30-node instances	*Mandl* instances
Real-world Implementation		×	84 nodes with 363 OD demand pairs (Riviera, Uruguay)	×	×	43 nodes with 543 OD demand pairs (Seville, Spain)	204 nodes with 13,338 OD demand pairs (Kayseri, Türkiye)

Passengers departing from *i* and destined to *j* may go *directly* on a single vehicle if a route passing through *i* and *j* exists. Otherwise, passengers may go *indirectly* to their destination by changing route(s) at *transfer points* where two or more routes coincide. Even if it is possible to move from *i* to *j* using a single route, passengers may prefer to transfer depending on *waiting times* at the boarding and transfer points, which are imposed by the *frequencies of the* routes. Because making transfers is annoying for passengers, a *transfer discomfort penalty* (in time units) is used to control the number of transfers. Thus, the *total travel time* for passengers to arrive at their destination from their origin is the sum of (1) *in-vehicle travel time*, (2) *waiting time at the boarding stop*, (3) *waiting time(s) at the transfer stop(s)*, and (4) *total penalty time resulting from transfers*. The lower this total travel time and the number of transfers, the better for passengers.

Let c_{ij} be the travel time (cost) from stop i to stop j, with c_{ij} being represented in a symmetric OD cost matrix as in the literature, i.e., $c_{ij} = c_{ji}$. Thus, *in-vehicle travel time* is the sum of θ_{ij} for all (i, j) on which passengers travel. *Waiting times at boarding and transfer stops are not known* (i.e., variables) and determined by the *headway* of a route. The headway h_t is the reciprocal of the frequency f_t, i.e., $h_t = 1/f_t$, and indicates the time between two consecutive vehicles on a route. For instance, if $f_t = 10$ vehicles per hour, then $h_t = \frac{1}{10}$ hours per vehicle, which is 6 min. Assuming that the arrivals of passengers at boarding and transfer stops are uniformly distributed, the waiting times of passengers for a route are set to half of the headway as an approximation as in Esfeh et al. [103].

The transit agency needs to design a system taking into account the perspective of passengers. Specifically, the transit agency should attempt to minimize the total travel time of passengers, including in-vehicle travel, waiting, and transfer times, and to maximize the number of direct-traveler passengers (i.e., minimize the number of transfers). However, considering only passengers' perspectives may be too costly for the transit agency. In this regard, it should also try to minimize the costs, e.g., the costs of trips as well as the fixed and operational costs of operating a vehicle fleet. In doing so, the transit agency needs to make several decisions, e.g., the number of routes to operate, the size of the vehicle fleet, the number of stops on a route, and the percentage of direct travelers. Depending on how several objectives are prioritized, the resulting systems may significantly differ; hence, an analysis of different scenarios needs to be conducted to devise a system balancing both perspectives.

3.3 The Proposed Model

We propose a mathematical programming model, namely, the *Public Transportation Planning Model* (PTPM), to solve TNDFSP. We define PTPM on an extended network $GT = (NT, AT)$ obtained by adding (1) two dummy nodes to N, namely, a and b that act as a super source node and a super sink node, respectively, and (2) two sets of directed arcs to A, namely, $A_{source} = \{(a, i): i \in N\}$ and $A_{sink} = \{(i, b): i \in N\}$. That is, $NT = N \cup \{a\} \cup \{b\}$ and $AT = E \cup A_{source} \cup A_{sink}$. We assume that all nodes are numbered, with a and b having the smallest and largest numbers, respectively. *Figure 3.1* gives a schematic representation of the extended network $GT = (NT, AT)$.

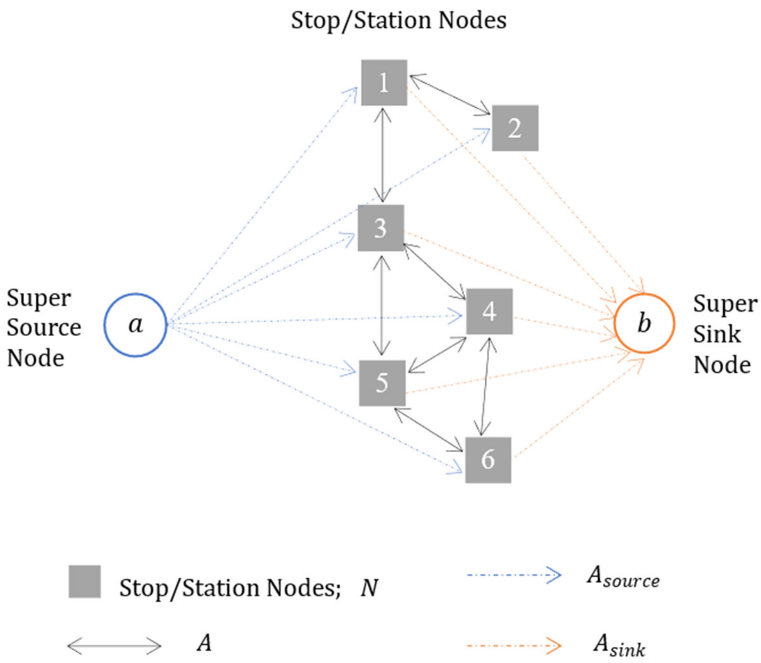


Figure 3.1 Representation of a transit network GT=(NT,AT)

All constructed routes start at node a and end at node b. The model constructs a route $t \in T$ from scratch by selecting an arc $(a, i) \in A_{source}$, an arc $(j, b) \in A_{sink}$, and a set of arcs from A that forms a path from i and j, where nodes i and j are the starting and ending terminals of the route t, respectively. *Figure 3.2* depicts the construction of routes in the model. In the figure, there are two routes: a-4-3-1-2-b and a-1-3-5-6-b. Because a and b are dummy nodes, the routes on the real physical network are 4-3-1-2 and 1-3-5-6. A similar approach that uses dummy nodes for selecting terminal nodes for routes can be found in the work of An and Lo [104].

To construct routes, we define binary decision variables d_{ijt}. Since a route is bidirectional, we define d_{ijt} for arcs $(i,j|i,j \in N \wedge i < j)$ to reduce the number of decision variables. If $d_{ijt} = 1$, then nodes i and j are consecutive stops in route t and passenger flow is allowed in both directions (i,j) and (j,i) on the route. For $d_{ait} = 1$ and $d_{jbt} = 1$, stops i and j are the starting and ending terminals of route t. Another variable set associated with route construction is y_{it} that take on the value of 1 if $i \in N$ is a part of route t and 0, otherwise. It is possible to impose additional requirements on the routes, e.g., the minimum (maximum) number of stops in a route and the maximum distance of a route. The decision variables f_t and h_t represent the frequency and headway of a route t, respectively. In addition to the variables above, we define (1) x_{ijkt} that represent the flow of passengers of origin $k \in S$ in arc (i,j) (i.e., traveling from stop i to stop j) on route t, (2) r_{ijkt} that represent the number of passengers of origin $k \in S$ who transfer at node i to route t with the next stop being node j, and (3) v_t that represent the number of vehicles required for route t.

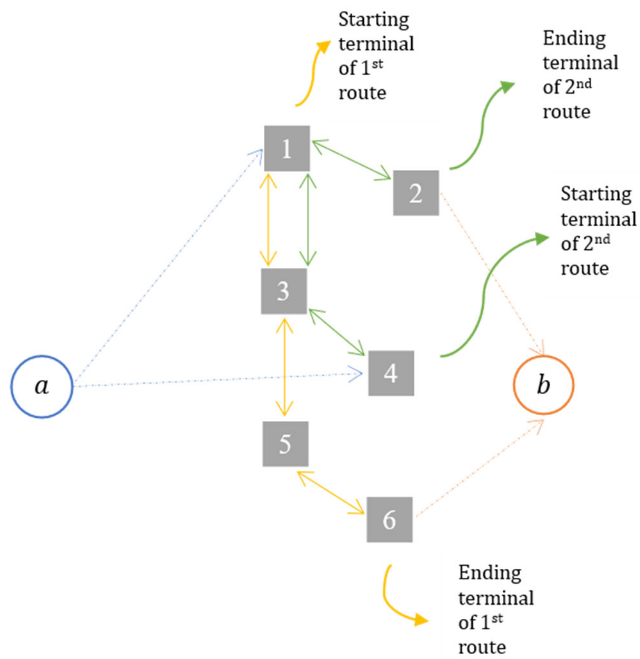


Figure 3.2 Illustration of the route design in a transit network, GT

Below we summarize the sets, parameters, and decision variables used in the model.

Sets and Indices

T — $set\ of\ routes\ (t \in T)$

N — $set\ of\ stops\ (i, j, k \in N)$

A — $set\ of\ arcs\ (i, j)$

S — $set\ of\ departure/supply\ stops\ (S \subseteq N)$

a — $super\ source\ node\ for\ routes$

b — $super\ sink\ node\ for\ routes$

A_{source} — $the\ set\ of\ directed\ arcs\ of\ the\ form\ (a, i), i \in N$

A_{sink} — $the\ set\ of\ directed\ arcs\ of\ the\ form\ (i, b), i \in N$

NT — $node\ set\ of\ extended\ network\ GT = (NT, AT)\ with\ NT = N \cup \{a\} \cup \{b\}$

AT — $arc\ set\ of\ extended\ network\ GT = (NT, AT)\ with\ AT = A \cup A_{source} \cup A_{sink}$

D_k — $set\ of\ arrival\ (destination)\ nodes\ for\ passengers\ of\ origin\ k \in S$

Parameters

cap_t — $capacity\ of\ a\ vehicle\ in\ route\ t \in T$

sup_k — $the\ number\ of\ passengers\ of\ origin\ k \in S$

w_{ik} — $the\ number\ of\ passengers\ of\ origin\ k \in S\ with\ destination\ i \in N$

c_{ij} — $travel\ time\ from\ stop\ i\ to\ stop\ j$

$penalty$ — $transfer\ penalty\ (in\ time\ units)$

$topStop$ — $maximum\ number\ of\ stops\ allowed\ for\ a\ route$

$lowStop$ — $minimum\ number\ of\ stops\ allowed\ for\ a\ route$

$period$ — $time\ period\ for\ which\ OD\ demand\ matrix\ is\ specified$

M — $a\ big\ number\ enough\ to\ allow\ passenger\ flow\ in\ uncapacitated\ edges$

ε — $values\ for\ vehicle\ fleet\ size$

$vehicleF$ — $upper\ limit\ of\ vehicle\ fleet\ size\ of\ the\ transit\ agency$

ϵ — $a\ small\ number\ (\epsilon = 10 \times 10^{-6})$

Decision variables

x_{ijkt} — $the\ flow\ of\ passengers\ of\ origin\ k\ who\ travel\ from\ i\ to\ j\ on\ route\ t$

d_{ijt} — $1, if\ arc\ (i, j), i < j, is\ selected\ to\ be\ in\ route\ t;\ 0, otherwise$

y_{it} — $1, if\ stop\ i\ is\ in\ route\ t; 0, otherwise$

r_{ijkt} — $the\ number\ of\ passengers\ of\ origin\ k\ who\ transfer\ at\ node\ i\ to\ route\ t$ $with\ next\ stop\ being\ node\ j$

f_t — $frequency\ of\ route\ t\ (vehicle\ per\ time\ unit, e.g., hour, minute)$

h_t — $headway\ of\ the\ route\ t\ (i.e., time\ between\ two\ consecutive\ vehicles\ for\ route\ t)$

v_t — $vehicle\ fleet\ required\ for\ route\ t$

Objective Function Terms:

$$in-vehicle\ travel\ time: \quad z_1 = \sum_{(i,j)\in A}\sum_{k\in S}\sum_{t\in T}(c_{ij}x_{ijkt}) \tag{3.1}$$

$$transfer\ time\ penalty: \quad z_2 = \sum_{(i,j)\in A}\sum_{k\in S}\sum_{t\in T}(penalty\ \ r_{ijkt}) \tag{3.2}$$

$$waiting\ time\ at\ boarding: \quad z_3 = \sum_{(i,j)\in A}\sum_{k=i}\sum_{t\in T}((h_t/2)x_{ijkt}) \tag{3.3}$$

$$waiting\ time\ at\ transfer: \quad z_4 = \sum_{(i,j)\in A}\sum_{k\in S}\sum_{t\in T}((h_t/2)r_{ijkt}) \tag{3.4}$$

$$vehicle\ fleet\ size: \quad z_5 = 2\sum_{\substack{(i,j)\in A;\\ i<j}}\sum_{k\in S}\sum_{t\in T}(c_{ij}f_t d_{ijt}) \tag{3.5}$$

The objective function term (3.1) represents total in-vehicle travel time of passengers. (3.2) finds the total transfer time of passengers who transfer and is used to penalize passenger transfers assuming that each transfer will take a specific time, i.e., penalty time. (3.3) and (3.4) compute waiting times of passengers at boarding and transfer stops, respectively. Objective function terms (3.1) through (3.4) represent perspectives of passengers. The objective function term (3.5) computes the total number of vehicles needed by the transit agency for all routes and is used as a proxy cost to denote the costs of the transit agency associated with operating the system. To explain how the fleet size is computed, suppose that the frequency of a route is $16\ vehicles/hour$ and the route length in one direction is $15\ minutes$. Because the route is bidirectional and symmetric, the route length is $2 \times 15\ minutes = 30\ minutes$. Then, the number of vehicles needed on the route is $\left(16\,\frac{vehicles}{hour}\right) \times (0.5\ hours) = 8\ vehicles.$

Model PTPM: Public Transportation Planning Model (PTPM)

$$\min Z_{passenger} = z_1 + z_2 + z_3 + z_4 \tag{3.6}$$
$$\min Z_{transitAgency} = z_5 \tag{3.6'}$$

$s.t.$

$$\sum_{i\in N} d_{ait} = 1 \qquad t \in T \tag{3.7}$$

$$\sum_{i \in N} d_{ibt} = 1 \qquad t \in T \tag{3.8}$$

$$\sum_{\substack{j \in N; \\ i<j \wedge (i,j) \in A}} d_{ijt} + \sum_{\substack{j \in N; \\ j<i \wedge (j,i) \in A}} d_{jit} = 2y_{it} \qquad i \in N, t \in T \tag{3.9}$$

$$\sum_{i \in N} \sum_{\substack{j \in N; \\ i<j \wedge (i,j) \in A}} d_{ijt} \leq topStop - 1 \qquad t \in T \tag{3.10}$$

$$\sum_{i \in N} \sum_{\substack{j \in N; \\ i<j \wedge (i,j) \in A}} d_{ijt} \geq lowStop - 1 \qquad t \in T \tag{3.11}$$

$$\sum_{i \in N} \sum_{\substack{j \in N; \\ i<j \wedge (i,j) \in A}} d_{ijt} \leq |\bar{S}| - 1 \qquad \bar{S} \subset N; \; 3 \leq |\bar{S}| \leq topStop - 1, \; t \in T \tag{3.12}$$

$$\sum_{\substack{j \in N; \\ (i,j) \in A}} \sum_{t \in T} x_{ijkt} - \sum_{\substack{j \in N; \\ (j,i) \in A}} \sum_{t \in T} x_{jikt} = \begin{cases} sup_k, & if\ i = k \\ -w_{ki}, & if\ i \in D_k \\ 0, & o.w \end{cases} \qquad i \in N, k \in S \tag{3.13}$$

$$\sum_{k \in S} (x_{ijkt} + x_{jikt}) \leq M d_{ijt} \qquad i,j \in N; i < j \wedge (i,j) \in A, \; t \in T \tag{3.14}$$

$$\left(\sum_{k \in S} x_{ijkt}\right)/period \leq cap_t f_t \qquad i,j \in N; (i,j) \in A, \; t \in T \tag{3.15}$$

$$x_{ijkt} = \sum_{\substack{g \in N; \\ g \neq j \wedge (g,i) \in A}} x_{gikt} + r_{ijkt} \qquad i,j \in N; \; (i,j) \in A, \; k \in S; \; i \neq k, \; t \in T \tag{3.16}$$

$$2 \sum_{\substack{(i,j) \in A; \\ i<j}} \sum_{k \in S} \sum_{t \in T} \left(c_{ij} f_t d_{ijt}\right) \leq v_t \qquad t \in T \tag{3.17}$$

$$f_t h_t = 1 \qquad t \in T \tag{3.18}$$

$$r_{ijkt} \geq 0, \qquad x_{ijkt} \geq 0 \qquad i,j \in N; \; (i,j) \in A, \; k \in S, t \in T \tag{3.19}$$

$$f_t \geq 0, \qquad h_t \geq 0 + \epsilon \qquad t \in T \tag{3.20}$$

$$v_t \geq 0, \qquad Integer \qquad t \in T \tag{3.21}$$

$$d_{ijt} \in \{0,1\} \qquad (i,j) \in AT, \; t \in T \tag{3.22}$$

$$y_{it} \in \{0,1\} \qquad i \in N, t \in T \tag{3.23}$$

PTPM minimizes the costs of passengers (3.6) and the transit agency (3.6'); hence, it is a multi-objective optimization problem. The objective function (3.6) minimizes the total travel time of all passengers and (3.6') minimizes the transit agency's cost, which is represented as the fleet size required to operate the system. Constraints (3.7) require each route t to have an arc from the super source node a to a stop $i \in N$, which becomes the starting terminal of that route. Constraints (3.8) require each route t to have an arc from a stop $i \in N$ to the super sink node b with i being the ending terminal of that route. If the starting and ending terminals are known in advance, they can be specified using (3.7) and (3.8). Constraints (3.9) ensure that each stop in a route is connected to two other nodes (stops or super nodes) as shown in *Figure 4.2*.

Constraints (3.10) and (3.11) impose upper and lower limits on the number of stops in a route, respectively. Constraints (3.12) eliminate subtours in the routes using the Dantzig–Fulkerson–Johnson (DFJ) subtour elimination formulation [60], where $\bar{S}$ is a subset of stops with the specified cardinality. The number of such constraints increases exponentially with the cardinality of the node set $|N|$ and hence, they cannot be used directly unless $|N|$ is very small. Therefore, we add subtour elimination constraints (3.12) during the solution process as described in *Algorithm 3.1* only when the candidate solutions violate them.

Constraints (3.13) are the flow conservation constraints of passengers at the stops and ensure that passengers move from their origins to their destinations through some routes. Constraints (3.14) couple passenger flow variables and arc selection variables and allow passenger flows only when an arc is selected to be in a route.

Constraints (3.15) ensure that a frequency for each route t, f_t, is determined considering the maximum passenger load on the route, which is equivalent to the maximum value on the left-hand side of (3.15). In other words, f_t can be considered as the number of vehicles needed per time unit to satisfy the demand for route t. Constraints (3.16) are flow balance constraints for the transfer of passengers. They state that the number of passengers of origin k on arc (i, j) in route t is equivalent to the number of passengers already traveling on route t and the number of passengers that transfer to route t at node i and move to j.

Constraints (3.17) compute the number of vehicles for each route. Because routes serve in two directions, there is a multiplier of 2 on the left-hand side. Constraints (3.17) with constraints (3.15) ensure that vehicles are not overcrowded because each route has sufficient vehicles to carry passengers. Constraints (3.18) state that the headway of a route is the inverse of the frequency of that route. Finally, Constraints (3.19) - (3.23) define the variables. Passenger flow and transfer variables, as well as headway variables, are defined to be nonnegative. Because the problem is strategic, fractional values regarding passenger flow and transfers may be accepted. To avoid undefined frequency evaluation due to the constraint set (3.18), we set a lower bound with a small number, ϵ, for the headway variable in constraints (3.20).

3.4 Solution Methodology

3.4.1 Multiobjective Optimization

PTPM is a multi-objective optimization model. In order to solve it using off-the-shelf software and obtain Pareto-optimal solutions, we convert it into a single-objective optimization model using the $\varepsilon -$ constraint method. We move the transit agency's objective function (3.6') to the constraint set as in (3.24), where the left-hand side represents the vehicle fleet size and the right-hand side represents an alternative level for vehicle fleet size. We solve the resulting single-objective optimization model for different levels of vehicle fleet size ε and find Pareto-optimal solutions.

$$\sum_{t \in T} v_t \leq \varepsilon \qquad \varepsilon \leq vehicleF \tag{3.24}$$

It is worth mentioning that by restructuring the PTPM to include an objective function that exclusively focuses on minimizing the number of vehicles, a lower bound for the vehicle fleet size can be determined. The resultant solution derived from this transformation serves as a reliable reference point for determining the minimum requirement of the vehicle fleet.

3.4.2 Subtour Elimination

The number of subtour elimination constraints (3.12) increases exponentially with the cardinality of the node set $|N|$ and hence, they cannot be used directly while solving PTPM unless $|N|$ is very small. For this reason, we add constraints (3.12) during the solution process only when candidate solutions violate them. The basic idea is to solve PTPM without constraints (3.12) and add only those constraints of (3.12) that eliminate subtours, i.e., the cuts, whenever there is an integer solution with subtours.

When PTPM is solved without constraints (3.12), subtours may occur in a route $t \in T$. For each integer solution for a route set T, we can check and quickly identify subtours and add specific cut (3.12) to separate them. We run this intervention procedure within a Lazy Constraints Callback function of *Gurobi*. *Lazy Constraints* are constructed when the user defines a violation for an integer solution. Algorithm 3.1 summarizes the steps of the intervention procedure.

Algorithm 3.1: Subtour Elimination

Step 0: *Start solving PTPM without constraints (3.12) using Gurobi. Due to the formulation consisting of integer decision space, the solver unfolds a branch and bound tree.*

Step 1: *When the solver finds an integer solution in any node of the branch and bound tree, run the Lazy Constraints Callback function defined for detecting subtours.*

Step 2: *If the callback function finds subtour(s) in the integer solution, go to step 3. Otherwise, go to step 4.*

Step 3: *Add corresponding subtour elimination constraints (3.12) for violating integer solution.*

Step 4: *Continue exploring the branch and bound tree nodes.*

3.4.3 Solving Large-Size Instances

We can obtain good solutions for small instances, such as the *Mandl* dataset, in minutes (see Section 3.5) with Gurobi. However, Gurobi cannot find feasible integer solutions for large-size instances such as a real-world application defined in Section 3.6. In this regard, we develop a solution methodology for large-size instances based on essentially providing the solver with a good warm-start solution for route design variables, d_{ijt}. We obtain warm-start solutions by exploiting the *Node Relaxation Heuristic* (NoRelHeur) utility of Gurobi and solving relaxed versions of PTPM. NoRelHeur is an embedded heuristic algorithm of Gurobi and can be used when the root node relaxation consumes too much time during the Branch&Cut (B&C) solution procedure. In applying Gurobi with NoRelHeur, NoRelHeur first tries to find a high-quality feasible solution in the allocated time and then Gurobi implements the branch and cut algorithm with the warm-start feasible solution found by NoRelHeur. The solution procedure is summarized in *Algorithm 3.2*.

In the application of Algorithm 3.2, we solve a relaxed, single-objective version of PTPM, *PTPM_Rel*, obtained by eliminating constraints (3.17), (3.18), (3.20), and (3.21), as well as terms (3.3) and (3.4) in the objective function (3.6). The eliminated constraints and objective function terms are related to vehicle fleet size, as well as to the frequency and headway decisions. PTPM_Rel tries to minimize the sum of in-vehicle travel time and transfer penalty time, i.e., $z_1 + z_2$, with constraints (3.7)-(3.16), (3.19), (3.22), and (3.23) ignoring the objective function (3.6'). While solving PTPM_Rel, we apply *Algorithm 3.1* as necessary to eliminate subtours.

To obtain a warm-start solution for d_{ijt} to use in solving PTPM, we solve PTPM_Rel in two phases. *In the first phase*, we solve PTPM_Rel only for a specific $k \in S$, i.e., *PTPM_Rel_k*. That allows the construction of a route network considering only passengers of origin k. To ensure that the route set is connected and consists of all stops, there should be demand for passengers of origin k at all other nodes, i.e., k should be selected such that $w_{kj} > 0, \ \forall j \in N$. If there does not exist such a k, we select a k with the highest $|dest_k|$ and assign an arbitrary positive value as demand for $j \neq k$ with $w_{kj} = 0$. Let $d_{ijt}^{Rel\_k}$ represent the values of decision variables d_{ijt} obtained after solving PTPM_Rel_k. *In the second phase*, we solve PTPM_Rel with the original OD demand matrix using $d_{ijt}^{Rel\_k}$ as an initial solution for d_{ijt}. Let d_{ijt}^{Rel} represent the values of decision variables d_{ijt} obtained after solving PTPM_Rel. In the final step of *Algorithm 2*, we solve PTPM with the original OD matrix for different vehicle fleet levels using d_{ijt}^{Rel} as the warm-start solution. We use Gurobi with NoRelHeur in solving PTPM_Rel_k and PTPM_Rel while we use Gurobi in solving PTPM.

To obtain a warm-start solution for d_{ijt} to use in solving PTPM, we solve PTPM_Rel in two phases. *In the first phase*, we solve PTPM_Rel only for a specific $k \in S$, i.e., *PTPM_Rel_k*. This allows a route network to be constructed considering only passengers of origin k. To ensure that the route set is connected and consists of all stops, there should be demand for passengers of origin k at all other nodes, i.e., k should be selected such that $w_{kj} > 0, \forall j \in N$. If such a k does not exist, we select a k with the highest $|D_k|$ and assign an arbitrary positive value as a demand for $j \neq k$ with $w_{kj} = 0$. Let $d_{ijt}^{Rel\_k}$ represent the values of decision variables d_{ijt} obtained after solving PTPM_Rel_k. *In the second phase*, we solve PTPM_Rel with the original OD demand matrix using $d_{ijt}^{Rel\_k}$ as an initial solution for d_{ijt}. Let d_{ijt}^{Rel} represent the values of decision variables d_{ijt} obtained after solving PTPM_Rel. In the final step of *Algorithm 3.2*, we solve PTPM with the original OD matrix for different vehicle fleet levels using d_{ijt}^{Rel} as the warm-start solution. We use Gurobi with NoRelHeur in solving PTPM_Rel_k and PTPM_Rel while we use Gurobi in solving PTPM.

Algorithm 3.2: Solution Procedure for Large Instances

Input: A transit network instance with OD demand and distance matrices.

Step 1: *Obtain a route network considering passengers of a specific origin* k.

- Solve PTPM_Rel_k for a specific k using Gurobi with NoRelHeur.
- Save $\mathrm{d}_{\mathrm{ijt}}^{\mathrm{Rel\_k}}$ that represent the values of decision variables for $\mathrm{d}_{\mathrm{ijt}}$ in the solution of PTPM_Rel_k.

Step 2: *Obtain a route network considering the original OD demand matrix.*

- Solve PTPM_Rel setting $\mathrm{d}_{\mathrm{ijt}}^{\mathrm{Rel\_k}}$ as an initial solution using Gurobi with NoRelHeur
- Save $\mathrm{d}_{\mathrm{ijt}}^{\mathrm{Rel}}$ that represent the values of decision variables for $\mathrm{d}_{\mathrm{ijt}}$ in the solution of PTPM_Rel.

Step 3: *Obtain feasible integer solutions for TNDFSP.*

- Solve PTPM setting $\mathrm{d}_{\mathrm{ijt}}^{\mathrm{Rel}}$ as an initial solution and using Gurobi for different vehicle fleet sizes to obtain Pareto-optimal solutions.

Output: Pareto optimal solutions for the transit network for varying vehicle fleet sizes.

3.5 Computational Tests Using Benchmark Instances

In this section, our goal is to show that PTPM works correctly and produces better results than exact and heuristic methods proposed in the literature. We conduct two sets of experiments. In the first set of experiments, we ignore waiting times at boarding and transfer points because the studies with which we compare our results use different assumptions in modeling waiting times; hence, the results are not comparable when waiting times are taken into account. In the second set of experiments, we obtain solutions considering waiting times.

To verify and validate PTPM, we use *Mandl* benchmark instances whose features are given in Section 3.4. We do not use Mumford instances because they are considered inappropriate for TNDFSP tests due to highly unrealistic frequencies [105]. We conduct experiments with Gurobi 9.5 solver using Julia programming language [71] and JuMP modeling language package [70] on an Intel Core i7@3.30 GHz computer with 16GB RAM. We use the default settings of Gurobi parameters except for NonConvex=2, MIPFocus=1, and BranchDir=1. NonConvex=2 tells Gurobi that the model is nonlinear. MIPFocus=1 emphasizes improving primal bound. BranchDir=1 requires up-branch to be explored first when a branching decision is to be made. This setting reduces the number of explored nodes and decreases the computer's RAM load, especially for large-size instances. The runtime for all experiments on Mandl instances is 3600 seconds. Vehicle capacity is 50 with a load factor of 1.25, i.e., $40 \times 1.25 = 50$.

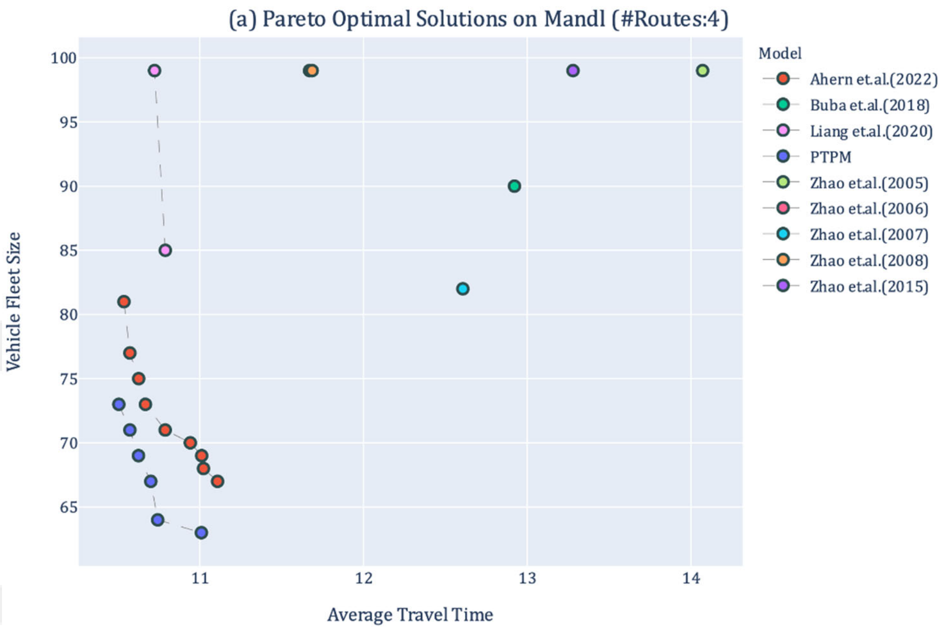
Pareto-optimal solutions are generated with respect to different levels of vehicle fleet size. We start by setting a vehicle fleet size less than or equal to an upper bound of the vehicle fleet and then gradually reduce it step by step. This approach is considered safe because if the number of vehicles is infeasible, the model itself results in infeasibility.

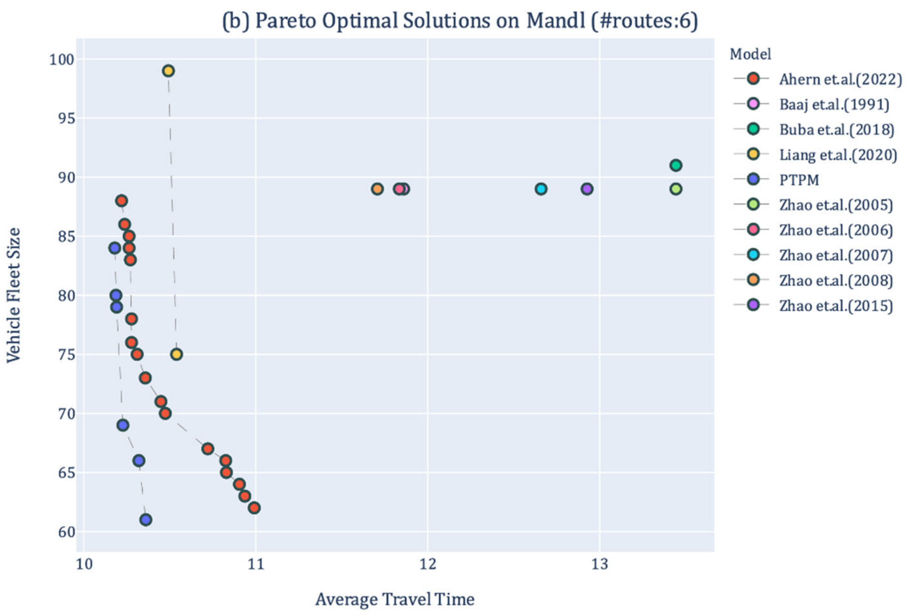
3.5.1 Benchmark Tests without Considering Waiting Times

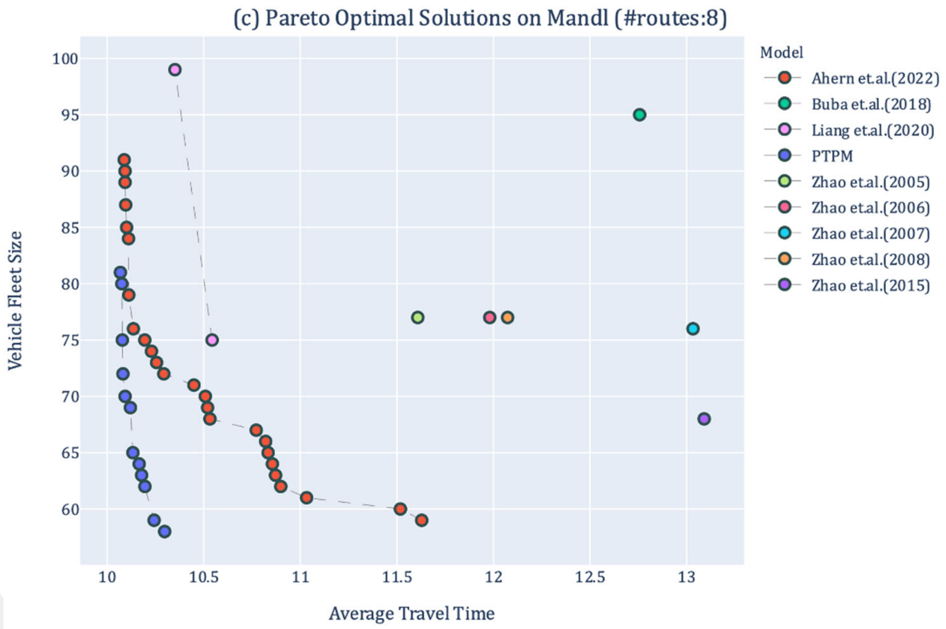
In the literature, few studies model waiting times at boarding and transfer nodes and solve benchmark instances. However, the rules of passenger assignment in these studies differ from our *system optimal* approach, which results in different waiting times in different studies. For example, Ahern et al. [20] use the frequency share method, while Liang et al. [95] assign passengers to routes based on the directness of the trip and the number of transfers. In this regard, the results of different studies are not comparable when waiting times are considered. Therefore, we compare our results with those proposed in the previous studies without considering waiting times, i.e., we solve PTPM with $Z_{passenger} = z_1 + z_2$ by varying vehicle fleet size to obtain Pareto-optimal solutions as described before.

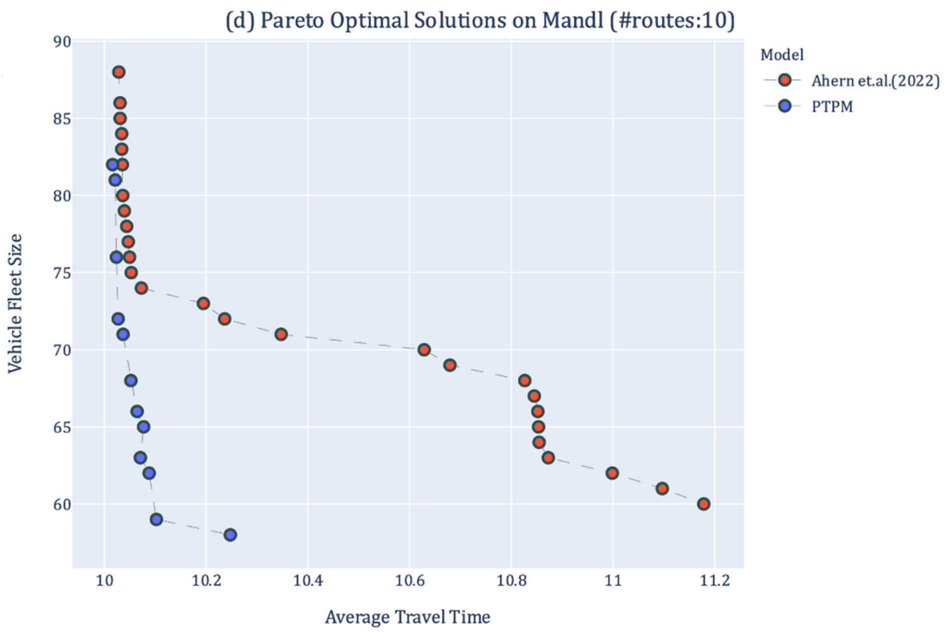
We present the results in the objective space with the x-axis representing the average travel time consisting of in-vehicle travel time and transfer penalty time and the y-axis representing vehicle fleet size. Figure 3.3 indicates the results of PTPM, as well as the results of the studies of Ahern et al. (2022) [20], Arbex and da Cunha [97], Bagloee and Ceder (2011) [98], Baaj and Mahmassani (1991) [49], Buba and Lee (2018) [96], Liang et al. (2020) [106], Zhao et al. (2005) [100], Zhao and Zeng (2006, 2007, 2008)

[33], [101], [102], Zhao et al. (2015) [99] for different number of routes. We show only the results reported in the related studies. We remark that all studies with which we compare our results are (meta)heuristic-based because there does not exist mathematical programming-based studies that attempt to solve benchmark instances with the objective of minimizing the travel time.






(c) Pareto Optimal Solutions on Mandl (#routes:8)
Vehicle Fleet Size
Average Travel Time
Model
Ahern et.al.(2022)
Buba et.al.(2018)
Liang et.al.(2020)
PTPM
Zhao et.al.(2005)
Zhao et.al.(2006)
Zhao et.al.(2007)
Zhao et.al.(2008)
Zhao et.al.(2015)


(d) Pareto Optimal Solutions on Mandl (#routes:10)
Vehicle Fleet Size
Average Travel Time
Model
Ahern et.al.(2022)
PTPM

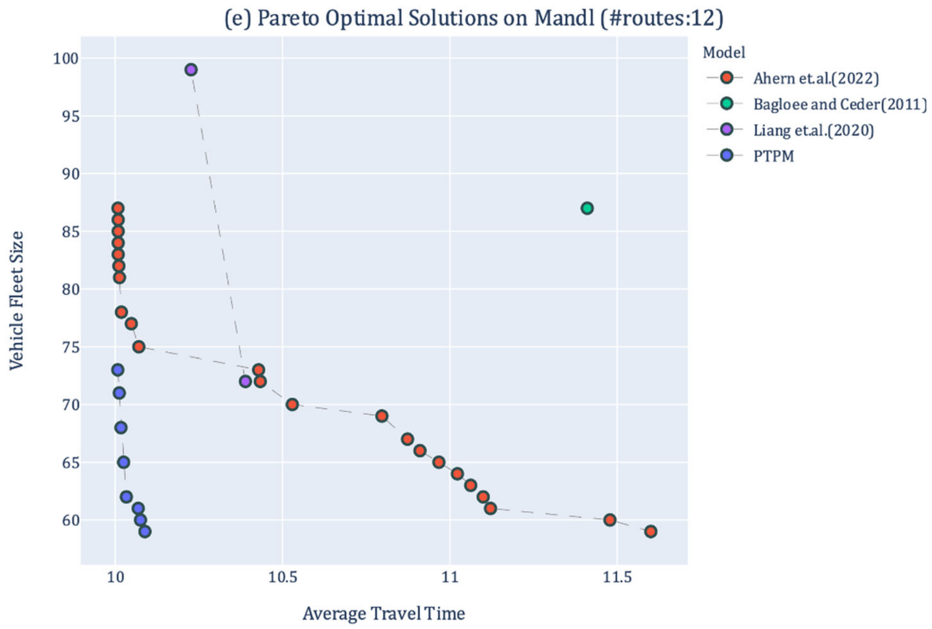


Figure 3.3 Pareto-optimal solutions generated by PTPM and different studies in the literature without considering waiting times.

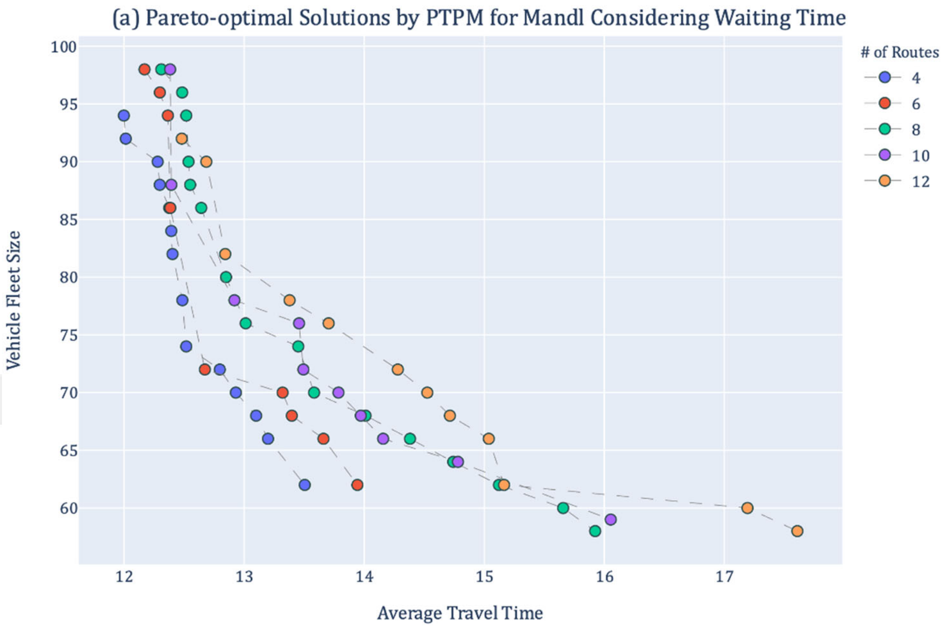
The results indicate that PTPM can provide better Pareto-optimal solutions than other algorithms. The study closest to ours in performance is Ahern et al. [20]. As the fleet size increases, the results of PTPM and Ahern et al. [20] get closer. However, as the fleet size decreases, i.e., when the problems are harder to solve, PTPM produces much better results.

The details of the computational results including the stops, vehicle fleet size, headways and frequencies of routes, average passenger cost and percent of transfers can be accessed through the following repository (https://aguedutr-my.sharepoint.com /:f:/g/personal/abdulkerim_benli_agu_edu_tr/Eq4v3KjxFFlAvNb9UOI2lDoBokeWcfpk p4VpbBLnKUOrxw, accessed on December 20, 2023).

3.5.2 Tests Including Waiting Times

Figure 3.4(a) and *Figure 3.4(b)* present Pareto-optimal solutions generated by PTPM for Mandl instances considering waiting times and without considering waiting times, respectively. The results in *Figure 3.4(a)* indicate that, for a fixed fleet size, the average travel time decreases as the number of routes decreases because more vehicles can be assigned to routes, which decreases waiting times at boarding and transfer stops.

However, as the number of routes decreases, the passenger load gets higher causing more discomfort to passengers. In this regard, there is a need to establish a balance between passenger discomfort resulting from high passenger load and passenger discomfort resulting from long travel times.



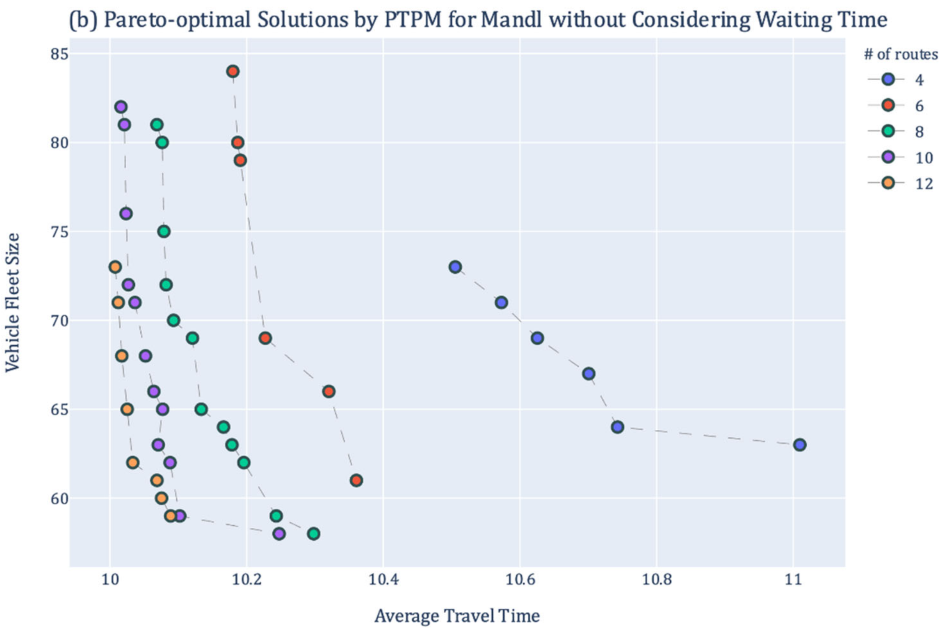


Figure 3.4 Pareto-optimal solutions generated by PTPM for Mandl instances considering waiting times.

The results in *Figure 3.4(b)* indicate that, for a fixed fleet size, the average travel time increases as the number of routes decrease, contrary to the results in *Figure 4.3(a)*. As the number of routes decreases, the number of direct travelers increases; hence, the average in-vehicle travel time increases. The results in *Figure 3.4* emphasize the importance of incorporating waiting times at boarding and transfer points, as the results without waiting times may mislead the transit agency in developing plans.

3.6 A Real-World Application

In this section, we present the results of computational tests conducted for instances defined on a real bus transit network in the city of Kayseri, located in the Central Anatolian Region in Türkiye. Kayseri has a population of about 1.5 million. The current public transport system consists of tram and bus transit networks. The tram and bus networks intersect at certain points; however, the tram network passes through a limited number of streets and has limited capacity. The bus network is currently the main public transport system. The number of stops in both directions in the current bus transit system is over 3000, which is pretty high for the size of a city such as Kayseri. In this regard, designing a trunk-and-feeder system is considered an alternative. Our goal in this study is to determine trunk lines/routes on the stops selected by Kayseri Transport Inc., considering demand intensity and location diversity. The Kayseri204 network consists of 204 nodes and 405 edges. The features of the instances considered are given in *Table 3.2*.

Table 3.2 Features for Kayseri204 transit network instances

Number of nodes/ edges	**Number of routes**	**Node limits (min./max.)**	**Transfer Penalty (mins.)**
204 / 405	15, 20, 25	2 / 25	15
Number of non-zero OD demand pairs	**Total Passenger Demand**	**Demand period (mins.)**	**Vehicle Capacity**
13338	205090	1000	50

We conduct tests with Gurobi 9.5 solver on an Intel(R) Xeon(R) Gold 6150@2.70 GHz computer with 256GB RAM with the parameter settings of Gurobi defined earlier. Since Kayseri204 is a large network, we run Algorithm 2 to find solutions for the instances. In the application of Algorithm 2, we set run times as follows: Step 1: NoRelHeurTime=2000, TimeLimit=3000; Step 2: NoRelHeurTime=50000, TimeLimit=60000; and Step 3: TimeLimit=23400, i.e., the total runtime of *Algorithm 2* is 86400 seconds. We remark that we provide the solution obtained in Steps 1 and 2 as a warm-start solution in Step 3, where we run just the B&C algorithm of Gurobi. We obtain results for 15-, 20-, and 25- route options by setting the number of vehicles to 150, 160, 180, 200, 220, 250, and 300. We assume a transfer penalty of 15 minutes, as in Arbex and da Cunha [97], Borndörfer and Karbstein [107].

OD demand and distance matrices as well as the features of Pareto-optimal solutions, including average travel time, vehicle fleet size, and optimality gap, can be accessed through the following repository (https://aguedutr-my.sharepoint.com/:f:/g/personal/abdulkerim_benli_agu_edu_tr/Eq4v3KjxFFlAvNb9UOI2lDoBokeWcfpkp4VpbBLnKUOrxw, accessed on December 20, 2023).

3.6.1 Travel Time vs. Number of Vehicles

Figure 3.5 gives Pareto-optimal solutions for the different number of routes. The results indicate that the 15-route solution dominates 20- and 25-route solutions with respect to the average travel time per passenger and vehicle fleet size. Like Mandl cases, for a fixed vehicle fleet size, average travel time for a 15-route solution is better than for 20-route and 25-route solutions. That is because more vehicles can be assigned to the routes in the 15-route instance, which decreases waiting times at boarding and transfer points. When the required number of routes increases, more vehicles may be needed to ensure a certain service level.

We remark that feasible solutions cannot be obtained for the vehicle fleet sizes lower than 150, 160, and 180 for the 15-route, 20-route, and 25-route instances, respectively. A sample solution for 15 routes and 150 vehicles is presented in Figure 4.6 on the Kayseri map, where each color corresponds to a different route. Parts of some routes coincide and hence overlap in some segments.

Table 3.3 details the results with respect to average in-vehicle travel time (AIVT), waiting time (AWT), transfer penalty time (ATP), optimality gap (Gap%), average travel time (ATT), and headway (AH) for the different number of routes. The table shows how average in-vehicle travel times, waiting times, and transfer penalty times change with the increasing number of routes and vehicle fleet sizes. As the fleet size increases, all related times get better for a fixed number of routes. As the number of routes increases, average transfer penalty times decrease because the number of direct travelers increases. On the other hand, average waiting times increase because average headways increase as fewer vehicles can be assigned to the routes. Because average in-vehicle travel times are close to each other and the improvements in transfer penalty time are lower than the deterioration in waiting times, average waiting time becomes dominant and generally causes the total average travel time to increase for a fixed fleet size. For a fixed fleet size, increasing the number of routes does not improve service level with respect to total average travel time.

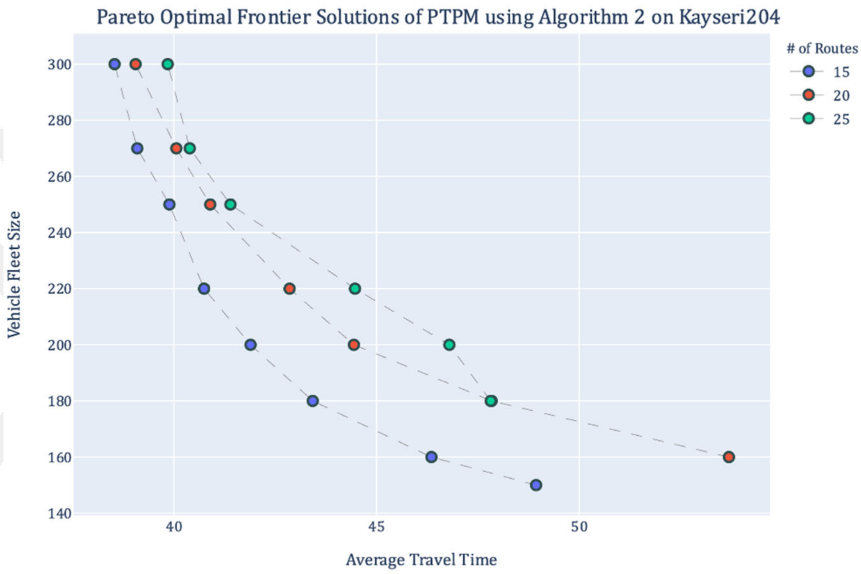


Figure 3.5 Pareto-optimal solutions for Kayseri204 instances

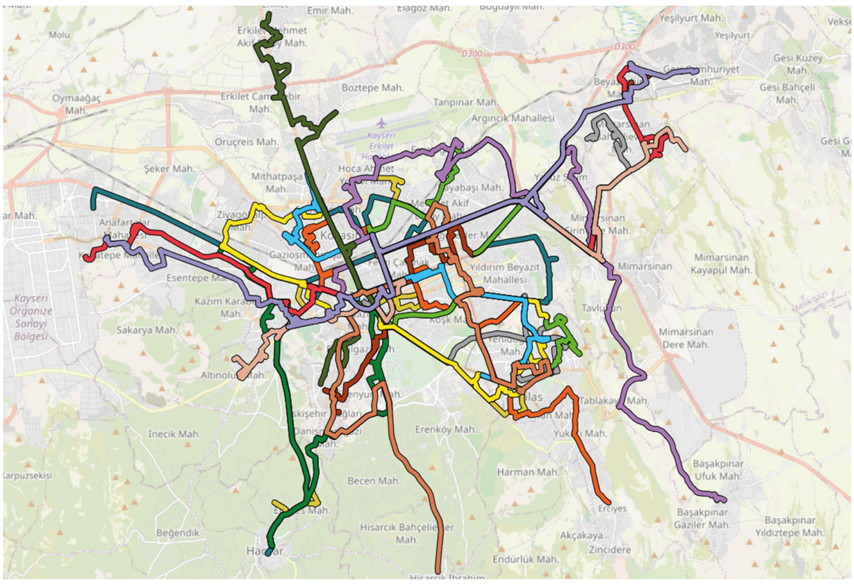


Figure 3.6 Sample solution for 15 routes and 150 vehicles for the Kayseri204 network

Table 3.3 indicates that optimality gaps change from 47.15% to 61.56%. The resulting high gaps occur because Gurobi cannot improve the linear programming relaxation bounds, i.e., the solutions found by Gurobi may actually be much better. To check the quality of solutions found by *Algorithm 3.2*, we have also solved *Mandl benchmark instances* using *Algorithm 3.2*. The results indicate that *Algorithm 3.2* produces solutions that are close to and sometimes better than those found by Gurobi without NoRel, with optimality gaps of about 25% on average. Considering that the solutions found by the proposed model using Gurobi with these high optimality gaps are better than those found by state-of-the-art algorithms, we think that the solutions obtained for the Kayseri204 network may be much better.

We remark that even though the gaps are high, this study is the first to obtain solutions for instances significantly larger than those in the literature using mathematical programming and off-the-shelf software (204 nodes and 13338 OD pairs in comparison to 84 nodes and 363 OD pairs in the literature).

Table 3.3 Average travel times for Kayseri204 network instances

Route	Fleet	AIVT	AWT	ATP	ATT	AH	Gap (%)
15	150	26.69	13.74	8.51	48.94	21.67	57.82
	160	26.31	12.22	7.83	46.35	19.56	55.47
	180	26.07	10.31	7.04	43.43	16.9	52.47
	200	25.87	9.13	6.89	41.89	14.93	50.73
	220	25.96	8.06	6.72	40.75	12.68	49.34
	250	25.84	7.29	6.76	39.89	12.02	48.25
	270	25.68	6.59	6.82	39.10	10.48	47.20
	300	25.69	6.02	6.83	38.54	9.62	46.44
20	160	27.53	17.68	8.48	53.69	39.3	61.56
	180	26.70	14.68	6.46	47.84	31.2	56.85
	200	26.21	12.22	6.01	44.44	26.8	53.55
	220	25.93	10.93	6.00	42.86	22.33	51.83
	250	25.61	9.42	5.87	40.90	20.31	49.53
	270	25.39	8.67	6.00	40.06	18.38	48.47

	300	25.36	7.79	5.91	39.06	15.51	47.15
25	180	26.12	15.83	5.87	47.82	44.27	56.83
	200	26.11	14.48	6.20	46.80	36.39	55.89
	220	25.55	12.96	5.96	44.47	32.21	53.58
	250	25.52	10.72	5.16	41.40	25.17	50.14
	270	25.36	9.91	5.12	40.39	22.47	48.90
	300	25.33	9.32	5.19	39.85	22.05	48.20

* AIVT: Average in-vehicle time, AWT: Average waiting time, ATP: Average transfer penalty, ATT: Average travel time (AIVT+AWT+ATP), AH: Average headway. These units are in minutes.

3.6.2 Utilization of Vehicle Capacity

Analyzing the vehicle capacity utilization for a route helps the transit agency evaluate route crowding and under-utilization issues. The utilization of vehicle capacity for a route set can be computed as follows: Let nv_t represent the number of vehicles assigned to route t, vc the capacity of a vehicle, pl_{ijt} passenger load on the link from i to j on route t, and nl_t the number of links on route t. Utilization of vehicle capacity for the link from i to j on route t is $vcu_{ijt} = \frac{pl_{ijt}}{vc \times nv_t} \times 100$, the average utilization of vehicle capacity for route t is $acu_t = \frac{\sum_{ij} pl_{ijt}}{nl_t}$, and average utilization of vehicle capacity for the whole route set is $u = \frac{\sum_t (acu_t \times nv_t)}{\sum_t nv_t}$.

Figure 3.7 gives the average utilization of vehicle capacity for the different number of routes and vehicle fleet sizes for the Kayseri204 network. For a fixed number of routes, the average utilization of vehicle capacity decreases as the vehicle fleet size increases.

Figure 3.8 depicts an example analysis of the most and least utilized routes for 15-route instance with 150 vehicles. Figures 3.8(a) and 8(b) (8(c) and 8(d)) show the utilization of vehicle capacity on the links of the busiest route (the least busy route), route #5 (route #9), in forward and backward directions, respectively. A bar in the figure indicates vehicle capacity utilization for the link between two consecutive nodes in the x-axis. For instance, the first blue bar in Figure 3.8(a) represents vehicle capacity utilization for the link from 82 to 80. The results indicate that the intensity of passengers and hence the utilization level gets higher towards the middle segment of the routes as expected, reaching 100% for some links. Even though route #9 is the least busy route, the utilization level for some links is 100%.

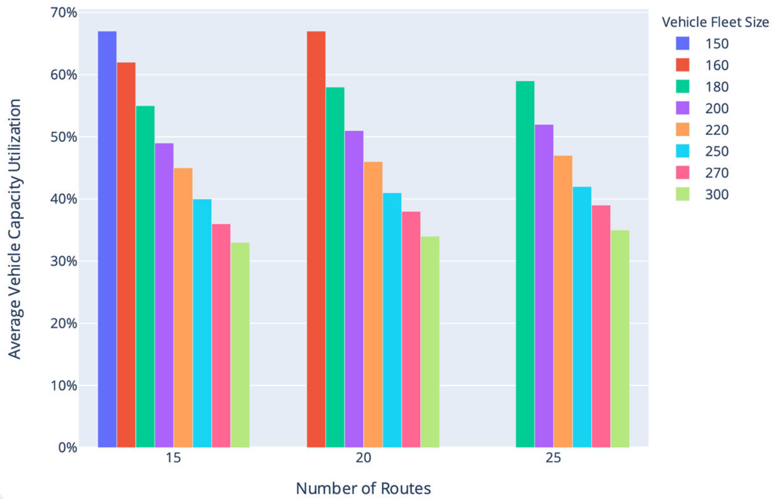
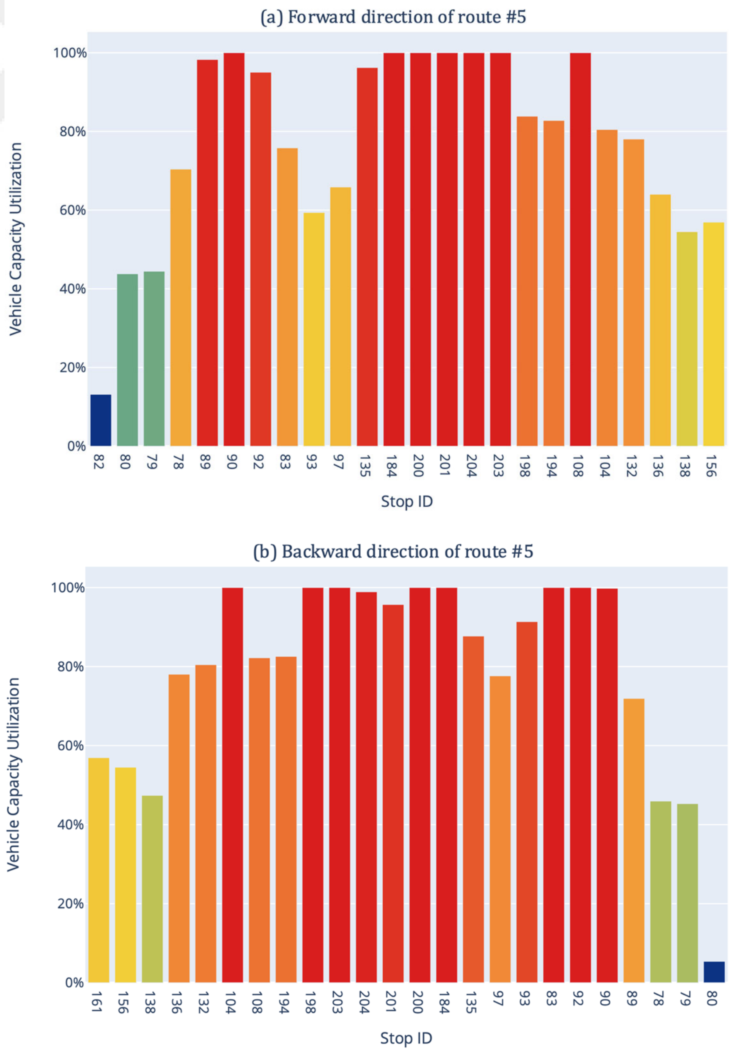


Figure 3.7 Average utilization of vehicles' capacity for Kayseri204 instances



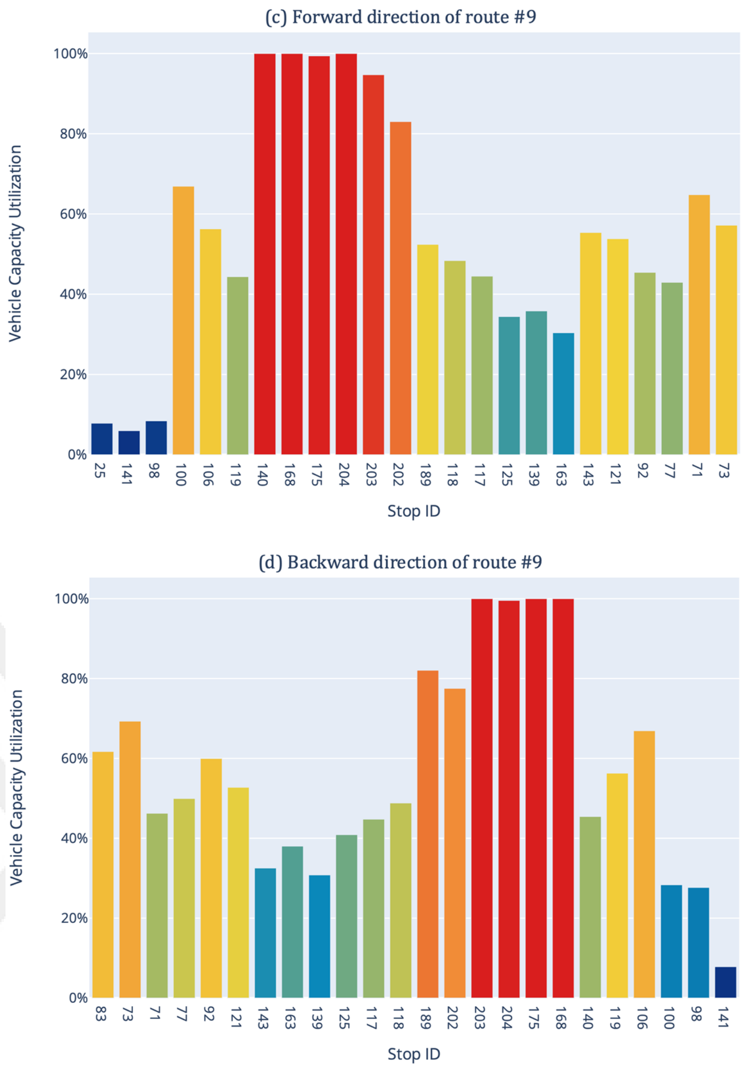


Figure 3.8 Utilization of vehicle capacity for the busiest and the least busy routes for a 15-route solution with 150 vehicles

3.7 Conclusion

We address the *transit network design and frequency setting problem (TNDFSP)* that aims to design the routes and determine the routes' frequencies to satisfy passenger demand in a transit network. We propose a novel mathematical programming model for TNDSFP that incorporates the features of real-life transit network systems. The proposed model reflects the views of both passengers and the transit agency by considering in-vehicle travel time, transfers, waiting times at the boarding and transfer stops, overcrowding and under-utilization of vehicles, and vehicle fleet size. The model is the first to determine routes and their frequencies simultaneously from scratch, i.e., without using line and frequency pools while considering the aforementioned issues, such as transfers and waiting.

We solve the proposed model using Gurobi. We conduct computational tests using Mandl benchmark instances and a real-world transit network. The first group of tests indicates that the proposed model works correctly and produces better Pareto-optimal solutions than state-of-the-art algorithms. The second group of tests shows how what-if analyses may help design a transit network system. The main insights from what-if analyses may be summarized as follows:

- The total average travel time for a fixed fleet size gets better with the decreasing number of routes because more vehicles can be assigned to the routes.
- For a fixed fleet size, average waiting times at boarding and transfer points increase with the increasing number of routes because fewer vehicles can be assigned to the routes.
- For a fixed fleet size, the average transfer penalty time decreases because the number of direct travelers increases with the increasing number of routes.
- For a fixed number of routes, total average travel time, average transfer penalty time, and average waiting times improve with the increasing fleet size.
- The average vehicle capacity utilization decreases with increasing vehicle fleet size.
- The average vehicle capacity utilization may reach up to 100% on some links, even for the least busy routes.
- More vehicles may be needed to ensure a certain service level with respect to the total average time with the increasing number of routes.
- Incorporating waiting times and transfer times as well as vehicle fleet size into the modeling may change the results significantly and hence is of high importance.

We remark that the insights may change depending on the values of different parameters. However, the results indicate that the model can be useful as a decision aid for designing a transit network system or evaluating a current system. This TNDFSP study is the first to obtain solutions for instances significantly larger than those in the literature using mathematical programming and off-the-shelf software (204 nodes and 13338 OD pairs in comparison to 84 nodes and 363 OD pairs in the literature). However, the model is difficult to solve for large-scale problems in designing a system from scratch; hence, there is a need to improve the solvability of the model as a future research direction.

Chapter 4

Conclusions and Future Prospects

4.1 Conclusions

In this thesis, we address the *transit network design problem (TNDP)* that aims to design the routes to satisfy passenger demand in a transit network and the *transit network design and frequency setting problem* (TNDFSP) that considers frequency setting as well as route design. TNDP is an NP-hard problem that is hard to solve for even small networks. Therefore, mathematical programming models are rare in the literature, and they make some simplifications to handle complexity, especially for large transit networks. They design routes by selecting out of a line/route pool. Optimum routes can be overlooked due to that strategy since the line pool may not include optimum ones. Besides, preprocessing data, such as analyzing OD demand pairs and generating candidate routes, is required in such models.

Moreover, they can model passenger transfers with only overextending transit network like *change & go* [10] and *trajectory* [12] graphs that unnecessarily increase the complexity of the problem. Considering the abovementioned issues, we propose a new mathematical programming mixed-integer model regarding TNDP. The proposed mathematical model embraces the complexity, and an off-the-shelf solver can solve it.

The default formulation of the proposed TNDP model works well in small-scale problems. However, it may fail to handle large-scale networks. Therefore, we decompose the original formulation by the *Benders decomposition* (BD) algorithm. The BD is implemented with the Branch and Cut procedure instead of the classical approach, which uses an iterative algorithm. Moreover, some enhancement strategies for the BD are implemented. To increase the dual bound of the master problem, we adopt the *in-out cut loop scheme* replacing *Kelley's scheme*. We disaggregate the subproblem since the aggregated version is hard to solve for large networks, even with modern solvers. We also set a cut selection criterion via an auxiliary mathematical model which finds *Pareto-*

optimal cuts since some Benders optimality cuts can be redundant due to alternative optima solutions of the subproblem. Finally, we develop a heuristic based on Benders decomposition.

TNDFSP is also a complex and NP-hard problem. Therefore, both mathematical programming and metaheuristic studies in the literature make many simplifying assumptions about *route design*, *frequency setting*, *transfers*, *waiting*, *passenger assignment*, and *vehicle fleet size* in handling the problem. We propose an all-encompassing novel mathematical programming model that is flexible enough to incorporate realistic features of a transit network.

Computational experiments are done by Gurobi solver. We verify the performance and capability of the proposed models with the well-known benchmark instances by the comparisons with the state-of-art studies in the literature. There is no mathematical program for TNDP making such verification, and their reported solutions regard only ad-hoc instances. Finally, we apply the proposed models to a real-world transit network consisting *of 204 nodes, 455 links, and 13338 OD demand pairs,* which is significantly larger than the problems in the literature, using off-the-shelf software.

The real-world application is done with what-if analyses that provide insights regarding the number of routes, average travel time, vehicle fleet size, and average vehicle capacity utilization. These insights suggest that the proposed models can be used as a decision aid for designing a transit network system or evaluating a current system.

4.2 Societal Impact and Contribution to Global Sustainability

The application of the proposed mathematical models to real-world transit networks has the potential to mitigate inefficiencies that public transit agencies encounter. Our findings indicate that these models can facilitate a reduction in vehicle fleet size while maintaining, or even enhancing, service quality in terms of passenger travel time. This efficiency gain enables transit agencies to reallocate budgets towards expanding services in underserved areas, thereby optimizing resource utilization.

Furthermore, the strategic route redesign made possible by the proposed models can diminish passenger travel times. This improvement may increase public transportation demand, potentially influencing a modal shift in commuter behavior towards more sustainable options. Such a shift can have immediate and positive environmental implications, including the reduction of carbon emissions and alleviation of traffic congestion.

The research presented in this thesis also aligns with and contributes to the Sustainable Development Goals (SDGs) set forth by the United Nations in 2015 [108], with a particular focus on the eleventh goal: to ensure "sustainable cities and communities". Specifically, the second target of this goal advocates for "affordable and sustainable transportation systems". The efficacy of the proposed models suggests that transit agencies can manage vehicle fleets with greater efficiency, yielding operational cost savings that could translate into lower transportation fees for end-users. Additionally, the adoption of such efficient management practices can enhance the sustainability of transportation systems, underscoring the practical relevance of this research in the pursuit of global sustainability objectives as outlined in the SDGs.

4.3 Future Prospects

Future research directions upon this thesis can focus on employing additional methodological improvements. For the proposed TNDP model, additional accelerating strategies for Benders decomposition can be considered, such as variable fixing, trust region method, and matheuristics. For the proposed TNFSP model, problem-specific cuts, decomposition algorithms, and heuristics may be developed and incorporated into the solution procedure. Moreover, methods to reduce the sizes of the problems, e.g., network analysis or aggregation rules, may also be proposed. Computational experiments show that improving dual bound is hard by the proposed models. So, tighter reformulations are needed with strong valid inequalities.

Aside from methodological improvements, the models can incorporate different aspects of public transportation planning problems. Various externalities can be embedded into cost functions, such as greenhouse emission costs. The electrification of current public transit systems can be studied by modifying the proposed models. Moreover, the design parameter of the problems can be altered with realistic variants. For instance, a heterogeneous vehicle fleet can be easily specified as a parameter change instead of a homogeneous fleet as in the proposed models.

Furthermore, a futuristic aspect of deploying autonomous public transit vehicles can be investigated via a case of redesigning routes by day (or shorter period) according to the real-time prediction of passenger demand. This variant requires solving the proposed models repetitively for each time period. This thesis assumes that the routes are bidirectional (i.e., having the same stops in both directions on a route). However, the proposed models can be transformed to deal with circular or radial routes. Last but not least, analysis of elastic demand has been disregarded in the literature for a long time, even though it can help analyze the redesigned transit networks. Simulation-based optimization approaches can do such post-process analyses.

BIBLIOGRAPHY

[1] United Nations Department of Economic and Social Affairs (UN DESA), “2018 Revision of World Urbanization Prospects,” 2018. [Online]. Available: https://population.un.org/wup/Publications/Files/WUP2018-PressRelease.pdf

[2] T. Li, M. Burke, and J. Dodson, “Transport impacts of government employment decentralization in an Australian city – Testing scenarios using transport simulation,” *Socioecon Plann Sci*, vol. 58, pp. 63–71, Jun. 2017, doi: 10.1016/J.SEPS.2016.10.006.

[3] V. Albino, U. Berardi, and R. M. Dangelico, “Smart Cities: Definitions, Dimensions, Performance, and Initiatives,” *Journal of Urban Technology*, vol. 22, no. 1, pp. 3–21, Jan. 2015, doi: 10.1080/10630732.2014.942092.

[4] A. Ceder and N. H. M. Wilson, “Bus network design,” *Transportation Research Part B: Methodological*, vol. 20, no. 4, pp. 331–344, Aug. 1986, doi: 10.1016/0191-2615(86)90047-0.

[5] G. Laporte, J. A. Mesa, F. A. Ortega, and F. Perea, “Planning rapid transit networks,” *Socioecon Plann Sci*, vol. 45, no. 3, pp. 95–104, Sep. 2011, doi: 10.1016/J.SEPS.2011.02.001.

[6] A. T. Murray, “Strategic analysis of public transport coverage,” *Socioecon Plann Sci*, vol. 35, no. 3, pp. 175–188, Sep. 2001, doi: 10.1016/S0038-0121(01)00004-0.

[7] S. Carrese and S. Gori, “An Urban Bus Network Design Procedure,” *Transportation Planning*, pp. 177–195, Jan. 2002, doi: 10.1007/0-306-48220-7_11.

[8] J. W. Goossens, S. Van Hoesel, and L. Kroon, “On solving multi-type railway line planning problems,” in *European Journal of Operational Research*, Elsevier, Jan. 2006, pp. 403–424. doi: 10.1016/j.ejor.2004.04.036.

[9] M. T. Claessens, N. M. Van Dijk, and P. J. Zwaneveld, “Cost optimal allocation of rail passenger lines,” *Eur J Oper Res*, vol. 110, no. 3, pp. 474–489, Nov. 1998, doi: 10.1016/S0377-2217(97)00271-3.

[10] A. Schöbel and S. Scholl, “Line Planning with Minimal Traveling Time,” *5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS’05)*, 2006, doi: 10.4230/OASIcs.ATMOS.2005.660.

[11] R. Borndörfer, M. Grötschel, and M. E. Pfetsch, "Models for line planning in public transport," in *Computer-aided scheduling of public transport (CASPT). Lecture Notes in Economics and Mathematical Systems*, vol. 600, 2008, pp. 363–378. doi: 10.1007/978-3-540-73312-6_18.

[12] H. Cancela, A. Mauttone, and M. E. Urquhart, "Mathematical programming formulations for transit network design," *Transportation Research Part B: Methodological*, vol. 77, pp. 17–37, Jul. 2015, doi: 10.1016/J.TRB.2015.03.006.

[13] A. Schöbel, "Line planning in public transportation: Models and methods," *OR Spectrum*, vol. 34, no. 3, pp. 491–510, Jul. 2012, doi: 10.1007/s00291-011-0251-6.

[14] A. Mauttone and M. E. Urquhart, "A route set construction algorithm for the transit network design problem," *Comput Oper Res*, vol. 36, no. 8, pp. 2440–2449, Aug. 2009, doi: 10.1016/J.COR.2008.09.014.

[15] Q. K. Wan and H. K. Lo, "A Mixed Integer Formulation for Multiple-Route Transit Network Design," *Journal of Mathematical Modelling and Algorithms*, vol. 2, no. 4, pp. 299–308, 2003, doi: 10.1023/B:JMMA.0000020425.99217.cd.

[16] R. Z. Farahani, E. Miandoabchi, W. Y. Szeto, and H. Rashidi, "A review of urban transportation network design problems," *Eur J Oper Res*, vol. 229, no. 2, pp. 281–302, Sep. 2013, doi: 10.1016/j.ejor.2013.01.001.

[17] R. van Nes, R. Hamerslag, and B. H. Immers, "Design of public transport networks," *Transp Res Rec*, no. 1202, pp. 74–83, 1988, Accessed: Mar. 18, 2022. [Online]. Available: https://trid.trb.org/view.aspx?id=302174

[18] J. Durán-Micco and P. Vansteenwegen, "A survey on the transit network design and frequency setting problem," *Public Transport*, pp. 1–36, Oct. 2021, doi: 10.1007/S12469-021-00284-Y.

[19] Y. Zhou, H. Yang, Y. Wang, and X. Yan, "Integrated line configuration and frequency determination with passenger path assignment in urban rail transit networks," *Transportation Research Part B: Methodological*, vol. 145, pp. 134–151, Mar. 2021, doi: 10.1016/J.TRB.2021.01.002.

[20] Z. Ahern, A. Paz, and P. Corry, "Approximate multi-objective optimization for integrated bus route design and service frequency setting," *Transportation Research Part B: Methodological*, vol. 155, pp. 1–25, Jan. 2022, doi: 10.1016/J.TRB.2021.10.007.

[21] A. De-Los-Santos, D. Canca, and E. Barrena, “Mathematical formulations for the bimodal bus-pedestrian social welfare network design problem,” *Transportation Research Part B: Methodological*, vol. 145, pp. 302–323, Mar. 2021, doi: 10.1016/J.TRB.2021.01.010.

[22] Y. Israeli and A. Ceder, “Transit Route Design Using Scheduling and Multiobjective Programming Techniques,” in *Computer-Aided Transit Scheduling*, J. R. Daduna, I. Branco, and J. M. P. Paixão, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 56–75.

[23] P. Gattermann, J. Harbering, and A. Schöbel, “Line pool generation,” *Public Transport*, vol. 9, no. 1–2, pp. 7–32, Jul. 2017, doi: 10.1007/s12469-016-0127-x.

[24] P. L. Bourbonnais, C. Morency, M. Trépanier, and É. Martel-Poliquin, “Transit network design using a genetic algorithm with integrated road network and disaggregated O–D demand data,” *Transportation (Amst)*, vol. 48, no. 1, pp. 95–130, Feb. 2021, doi: 10.1007/S11116-019-10047-1/FIGURES/16.

[25] J. F. Benders, “Partitioning procedures for solving mixed-variables programming problems,” *Numer Math (Heidelb)*, vol. 4, no. 1, pp. 238–252, Dec. 1962, doi: 10.1007/BF01386316.

[26] C. E. Mandl, *Applied network optimization*. Academic Press, 1979.

[27] C. L. Mumford, “New heuristic and evolutionary operators for the multi-objective urban transit routing problem,” in *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, 2013, pp. 939–946. doi: 10.1109/CEC.2013.6557668.

[28] A. Benli and İ. Akgün, “A Multi-Objective Mathematical Programming Model for Transit Network Design and Frequency Setting Problem,” *Mathematics*, vol. 11, no. 21, 2023, doi: 10.3390/math11214488.

[29] V. Guihaire and J.-K. K. Hao, “Transit network design and scheduling: A global review,” *Transp Res Part A Policy Pract*, vol. 42, no. 10, pp. 1251–1273, Dec. 2008, doi: 10.1016/J.TRA.2008.03.011.

[30] K. Kepaptsoglou and M. Karlaftis, “Transit Route Network Design Problem: Review,” *J Transp Eng*, vol. 135, no. 8, pp. 491–505, Aug. 2009, doi: 10.1061/(ASCE)0733-947X(2009)135:8(491).

[31] L. Fan, C. Mumford, and D. Evans, “A Simple Multi-Objective Optimization Algorithm for the Urban Transit Routing Problem,” in *2009 IEEE Congress on Evolutionary Computation*, Jun. 2009, pp. 1–7. doi: 10.1109/CEC.2009.4982923.

[32] P. Chakroborty and T. Dwivedi, “Optimal Route Network Design for Transit Systems Using Genetic Algorithms,” *Engineering Optimization 34(1):83-100*, 2002, Accessed: Jun. 13, 2020. [Online]. Available: https://www.researchgate.net/publication/228551307_Optimal_Route_Network_Design_for_Transit_Systems_Using_Genetic_Algorithms

[33] F. Zhao and X. Zeng, “Optimization of transit network layout and headway with a combined genetic algorithm and simulated annealing method,” *Engineering Optimization*, vol. 38, no. 6, pp. 701–722, Sep. 2006, doi: 10.1080/03052150600608917.

[34] B. F. Byrne, “Cost minimizing positions, lengths and headways for parallel public transit lines having different speeds,” *Transportation Research*, vol. 10, no. 3, pp. 209–214, 1976, doi: https://doi.org/10.1016/0041-1647(76)90076-9.

[35] V. F. Hurdle, “Minimum Cost Locations for Parallel Public Transit Lines,” *Transportation Science*, vol. 7, no. 4, pp. 340–350, Nov. 1973, doi: 10.1287/trsc.7.4.340.

[36] D. Dubois, G. Bel, and M. Llibre, “A set of methods in transportation network synthesis and analysis,” *Journal of the Operational Research Society*, vol. 30, no. 9, pp. 797–808, 1979, doi: 10.1057/jors.1979.190.

[37] P. D. Site and F. Filippi, “Bus service optimisation with fuel saving objective and various financial constraints,” *Transp Res Part A Policy Pract*, vol. 35, no. 2, pp. 157–176, 2001, doi: 10.1016/S0965-8564(99)00053-1.

[38] B. Beltran, S. Carrese, E. Cipriani, and M. Petrelli, “Transit network design with allocation of green vehicles: A genetic algorithm approach,” *Transp Res Part C Emerg Technol*, vol. 17, no. 5, pp. 475–483, 2009, doi: https://doi.org/10.1016/j.trc.2009.04.008.

[39] A. D. Jovanović, D. S. Pamučar, and S. Pejčić-Tarle, “Green vehicle routing in urban zones – A neuro-fuzzy approach,” *Expert Syst Appl*, vol. 41, no. 7, pp. 3189–3203, 2014, doi: https://doi.org/10.1016/j.eswa.2013.11.015.

[40] C. Iliopoulou, I. Tassopoulos, K. Kepaptsoglou, and G. Beligiannis, “Electric Transit Route Network Design Problem: Model and Application,” *Transp Res Rec*, vol. 2673, no. 8, pp. 264–274, 2019, doi: 10.1177/0361198119838513.

[41] C. Pylarinou, C. Iliopoulou, and K. Kepaptsoglou, “Transit route network redesign under Electrification: Model and application,” *International Journal of*

Transportation Science and Technology, vol. 10, no. 4, pp. 366–379, 2021, doi: https://doi.org/10.1016/j.ijtst.2021.01.001.

[42] A. Ceder, "Operational Objective Functions in Designing Public Transport Routes," *J Adv Transp*, vol. 3, no. 2, pp. 125–144, 2001, doi: 10.1002/atr.5670350205.

[43] P. Chakroborty, "Genetic Algorithms for Optimal Urban Transit Network Design," 2003. Accessed: Jun. 23, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-8667.00309

[44] E. Vermeir, W. Engelen, J. Philips, and P. Vansteenwegen, "An Exact Solution Approach for the Bus Line Planning Problem with Integrated Passenger Routing," *J Adv Transp*, vol. 2021, 2021, doi: 10.1155/2021/6684795.

[45] H. K. Suman and N. B. Bolia, "Improvement in direct bus services through route planning," *Transp Policy (Oxf)*, vol. 81, pp. 263–274, Sep. 2019, doi: 10.1016/J.TRANPOL.2019.07.001.

[46] L. Cadarso, E. Codina, L. F. Escudero, and A. Marín, "Rapid transit network design: Considering recovery robustness and risk aversion measures," *Transportation Research Procedia*, vol. 22, pp. 255–264, 2017, doi: 10.1016/J.TRPRO.2017.03.032.

[47] P. Wu, L. Xu, A. Che, and F. Chu, "A bi-objective decision model and method for the integrated optimization of bus line planning and lane reservation," *J Comb Optim*, 2020, doi: 10.1007/S10878-020-00647-4.

[48] M. H. Baaj and H. S. Mahmassani, "Hybrid route generation heuristic algorithm for the design of transit networks," *Transportation Research Part C*, vol. 3, no. 1, pp. 31–50, Feb. 1995, doi: 10.1016/0968-090X(94)00011-S.

[49] M. H. Baaj and H. S. Mahmassani, "An AI-based approach for transit route system planning and design," *J Adv Transp*, vol. 25, no. 2, pp. 187–209, 1991, doi: 10.1002/atr.5670250205.

[50] M. A. Nayeem, M. K. Rahman, and M. S. Rahman, "Transit network design by genetic algorithm with elitism," *Transp Res Part C Emerg Technol*, vol. 46, pp. 30–45, Sep. 2014, doi: 10.1016/J.TRC.2014.05.002.

[51] X. Feng, X. Zhu, X. Qian, Y. Jie, F. Ma, and X. Niu, "A new transit network design study in consideration of transfer time composition," *Transp Res D Transp Environ*, vol. 66, pp. 85–94, 2019, doi: https://doi.org/10.1016/j.trd.2018.03.019.

[52] M. V. Tom and S. Mohan, "Transit Route Network Design Using Frequency Coded Genetic Algorithm," *J Transp Eng*, vol. 129, no. 2, pp. 186–195, Mar. 2003, doi: 10.1061/(ASCE)0733-947X(2003)129:2(186).

[53] J. S. C. Chew, L. S. Lee, and H. V. Seow, "Genetic algorithm for biobjective urban transit routing problem," *J Appl Math*, vol. 2013, 2013, doi: 10.1155/2013/698645.

[54] M. Petrelli, "A Transit Network Design Model For Urban Areas," *WIT Transactions on The Built Environment*, vol. 75, May 2004, doi: 10.2495/UT040171.

[55] K. A. Islam, I. M. Moosa, J. Mobin, M. A. Nayeem, and M. S. Rahman, "A heuristic aided Stochastic Beam Search algorithm for solving the transit network design problem," *Swarm Evol Comput*, vol. 46, pp. 154–170, May 2019, doi: 10.1016/J.SWEVO.2019.02.007.

[56] L. Ahmed, C. Mumford, and A. Kheiri, "Solving urban transit route design problem using selection hyper-heuristics," *Eur J Oper Res*, vol. 274, no. 2, pp. 545–559, Apr. 2019, doi: 10.1016/j.ejor.2018.10.022.

[57] C. Iliopoulou, K. Kepaptsoglou, and E. Vlahogianni, "Metaheuristics for the transit route network design problem: a review and comparative analysis," *Public Transport*, vol. 11, no. 3, pp. 487–521, Oct. 2019, doi: 10.1007/S12469-019-00211-2.

[58] M. P. John, C. L. Mumford, and R. Lewis, "An Improved Multi-objective Algorithm for the Urban Transit Routing Problem," in *Evolutionary Computation in Combinatorial Optimisation. EvoCOP 2014. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2014, pp. 49–60. doi: 10.1007/978-3-662-44320-0_5.

[59] F. Kiliç and M. Gök, "A demand based route generation algorithm for public transit network design," *Comput Oper Res*, vol. 51, pp. 21–29, Nov. 2014, doi: 10.1016/j.cor.2014.05.001.

[60] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a Large-Scale Traveling-Salesman Problem," *Journal of the Operations Research Society of America*, vol. 2, no. 4, pp. 393–410, Nov. 1954, doi: 10.1287/OPRE.2.4.393.

[61] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei, "The Benders decomposition algorithm: A literature review," *European Journal of Operational Research*, vol. 259, no. 3. Elsevier B.V., pp. 801–817, Jun. 16, 2017. doi: 10.1016/j.ejor.2016.12.005.

[62] C. Matteo Fischetti Laureando and A. Beltramin, “Modern branch-and-cut solvers for Mixed-Integer Linear Programming: a computational comparison,” 2014.

[63] C. A. Zetina, I. Contreras, and J. F. Cordeau, “Exact algorithms based on Benders decomposition for multicommodity uncapacitated fixed-charge network design,” *Comput Oper Res*, vol. 111, pp. 311–324, Nov. 2019, doi: 10.1016/J.COR.2019.07.007.

[64] Jr. J. E. Kelley, “The Cutting-Plane Method for Solving Convex Programs,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 4, pp. 703–712, Jul. 1960, doi: 10.1137/0108053.

[65] M. Fischetti, I. Ljubić, and M. Sinnl, “Redesigning Benders Decomposition for Large-Scale Facility Location,” *Manage Sci*, vol. 63, no. 7, pp. 2146–2162, Jul. 2017, doi: 10.1287/mnsc.2016.2461.

[66] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov, “New variants of bundle methods,” *Mathematical Programming 1995 69:1*, vol. 69, no. 1, pp. 111–147, Jul. 1995, doi: 10.1007/BF01585555.

[67] W. Ben-Ameur and J. Neto, “Acceleration of cutting-plane and column generation algorithms: Applications to network design,” *Networks*, vol. 49, no. 1, pp. 3–17, Jan. 2007, doi: 10.1002/NET.20137.

[68] T. L. Magnanti and R. T. Wong, “Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria,” *https://doi.org/10.1287/opre.29.3.464*, vol. 29, no. 3, pp. 464–484, Jun. 1981, doi: 10.1287/OPRE.29.3.464.

[69] N. Papadakos, “Practical enhancements to the Magnanti–Wong method,” *Operations Research Letters*, vol. 36, no. 4, pp. 444–449, Jul. 2008, doi: 10.1016/J.ORL.2008.01.005.

[70] I. Dunning, J. Huchette, and M. Lubin, “JuMP: A Modeling Language for Mathematical Optimization,” *Society for Industrial and Applied Mathematics*, vol. 59, no. 2, pp. 295–320, 2017, doi: 10.1137/15M1020575.

[71] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, “Julia: A Fresh Approach to Numerical Computing,” *http://dx.doi.org/10.1137/141000671*, vol. 59, no. 1, pp. 65–98, Feb. 2017, doi: 10.1137/141000671.

[72] C. E. Mandl, “Evaluation and optimization of urban public transportation networks,” *Eur J Oper Res*, vol. 5, no. 6, pp. 396–404, Dec. 1980, doi: 10.1016/0377-2217(80)90126-5.

[73] B. Beckman, “Theory of Spectral Graph Layout,” 1994.

[74] O. J. Ibarra-Rojas, F. Delgado, R. Giesen, and J. C. Muñoz, “Planning, operation, and control of bus transport systems: A literature review,” *Transportation Research Part B: Methodological*, vol. 77. Elsevier Ltd, pp. 38–75, Jul. 01, 2015. doi: 10.1016/j.trb.2015.03.002.

[75] B. R. , Marwah, Farokh S. Umrigar, and S. B. Patnaik, “Optimal design of bus routes and frequencies for Ahmedabad,” *Transp Res Rec*, vol. 94, pp. 41–47, 1984.

[76] I. Constantin and M. Florian, “Optimizing frequencies in a transit network: a nonlinear bi-level programming approach,” *International Transactions in Operational Research*, vol. 2, no. 2, pp. 149–164, Apr. 1995, doi: 10.1016/0969-6016(94)00023-M.

[77] P. G. Furth, P. G. Furth, and N. H. M. Wilson, “Setting Frequencies on Bus Routes: Theory and Practice,” *Transp Res Rec*, 1981, Accessed: Jul. 04, 2022. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1068.2594

[78] W. Lampkin and P. D. Saalmans, “The Design of Routes, Service Frequencies, and Schedules for a Municipal Bus Undertaking: A Case Study,” *Journal of the Operational Research Society*, vol. 18, no. 4, pp. 375–397, Dec. 1967, doi: 10.1057/jors.1967.70.

[79] D. L. Van Oudheusden, S. Ranjithan, and K. N. Singh, “The design of bus route systems-An interactive location-allocation approach,” *Transportation (Amst)*, vol. 14, pp. 253–270, 1987.

[80] J. F. Guan, H. Yang, and S. C. Wirasinghe, “Simultaneous optimization of transit line configuration and passenger line assignment,” *Transportation Research Part B: Methodological*, vol. 40, no. 10, pp. 885–902, 2006, doi: 10.1016/j.trb.2005.12.003.

[81] W. Y. Szeto and Y. Jiang, “Transit route and frequency design: Bi-level modeling and hybrid artificial bee colony algorithm approach,” *Transportation Research Part B: Methodological*, vol. 67, pp. 235–263, Sep. 2014, doi: 10.1016/j.trb.2014.05.008.

[82] H. Spiess and M. Florian, “Optimal strategies: A new assignment model for transit networks,” *Transportation Research Part B*, vol. 23, no. 2, pp. 83–102, Apr. 1989, doi: 10.1016/0191-2615(89)90034-9.

[83] M. Bussieck, “Optimal Lines in Public Rail Transport,” Technische Universtat Braunschweig, 1998.

[84] R. Borndörfer, H. Hoppmann, and M. Karbstein, “A configuration model for the line planning problem,” in *OpenAccess Series in Informatics*, Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, 2013, pp. 68–79. doi: 10.4230/OASIcs.ATMOS.2013.68.

[85] M. R. Bussieck, P. Kreuzer, and U. T. Zimmermann, “Optimal lines for railway systems,” *Eur J Oper Res*, vol. 96, no. 1, pp. 54–63, Jan. 1997, doi: 10.1016/0377-2217(95)00367-3.

[86] R. Borndörfer and M. E. Pfetsch, “Routing in Line Planning for Public Transport,” in *Operations Research Proceedings 2005*, Springer Berlin Heidelberg, 2006, pp. 405–410. doi: 10.1007/3-540-32539-5_64.

[87] R. Borndörfer, M. Grötschel, and M. E. Pfetsch, “A column-generation approach to line planning in public transport,” *Transportation Science*, vol. 41, no. 1, pp. 123–132, 2007, doi: 10.1287/trsc.1060.0161.

[88] D. Kim and C. Barnhart, “Transportation Service Network Design: Models and Algorithms,” in *In: Wilson N.H.M. (eds) Computer-Aided Transit Scheduling. Lecture Notes in Economics and Mathematical Systems, vol 471*, 1999, pp. 259–283. doi: 10.1007/978-3-642-85970-0_13.

[89] R. Borndörfer and M. Neumann, “Models for Line Planning with Transfers,” 2010. Accessed: Apr. 23, 2022. [Online]. Available: https://opus4.kobv.de/opus4-zib/files/1174/ZR_10_11.pdf

[90] J. W. Goossens, S. van Hoesel, and L. Kroon, “A branch-and-cut approach for solving railway line-planning problems,” *Transportation Science*, vol. 38, no. 3, pp. 379–393, 2004, doi: 10.1287/trsc.1030.0051.

[91] S. Scholl, “Customer-Oriented Line Planning,” Technische Universitat Kaiserslautern, 2005. Accessed: Jun. 09, 2020. [Online]. Available: http://dnb.ddb.de

[92] M. R. Bussieck, T. Lindner, and M. E. Lübbecke, “A fast algorithm for near cost optimal line plans,” *Mathematical Methods of Operations Research*, vol. 59, no. 2, pp. 205–220, 2004, doi: 10.1007/s001860300332.

[93] M. Kaspi and T. Raviv, “Service-oriented line planning and timetabling for passenger trains,” *Transportation Science*, vol. 47, no. 3, pp. 295–311, 2013, doi: 10.1287/trsc.1120.0424.

[94] S. Burggraeve, S. H. Bull, P. Vansteenwegen, and R. M. Lusby, "Integrating robust timetabling in line plan optimization for railway systems," *Transp Res Part C Emerg Technol*, vol. 77, pp. 134–160, Apr. 2017, doi: 10.1016/j.trc.2017.01.015.

[95] M. Liang, W. Wang, C. Dong, and D. Zhao, "A cooperative coevolutionary optimization design of urban transit network and operating frequencies," *Expert Syst Appl*, vol. 160, p. 113736, Dec. 2020, doi: 10.1016/J.ESWA.2020.113736.

[96] A. T. Buba and L. S. Lee, "A differential evolution for simultaneous transit network design and frequency setting problem," *Expert Syst Appl*, vol. 106, pp. 277–289, Sep. 2018, doi: 10.1016/j.eswa.2018.04.011.

[97] R. O. Arbex and C. B. da Cunha, "Efficient transit network design and frequencies setting multi-objective optimization by alternating objective genetic algorithm," *Transportation Research Part B: Methodological*, vol. 81, pp. 355–376, Nov. 2015, doi: 10.1016/J.TRB.2015.06.014.

[98] S. A. Bagloee and A. A. Ceder, "Transit-network design methodology for actual-size road networks," *Transportation Research Part B: Methodological*, vol. 45, no. 10, pp. 1787–1804, 2011, doi: 10.1016/j.trb.2011.07.005.

[99] H. Zhao, W. Xu, and R. Jiang, "The Memetic algorithm for the optimization of urban transit network," *Expert Syst Appl*, vol. 42, no. 7, pp. 3760–3773, May 2015, doi: 10.1016/J.ESWA.2014.11.056.

[100] F. Zhao, I. Ubaka, and A. Gan, "Transit Network Optimization: Minimizing Transfers and Maximizing Service Coverage with an Integrated Simulated Annealing and Tabu Search Method," *Transportation Research Record: Journal of the Transportation Research Board*, pp. 180–188, 2005, doi: 10.1177/0361198105192300119.

[101] F. Zhao and X. Zeng, "Optimization of User and Operator Cost for Large-Scale Transit Network," *J Transp Eng*, vol. 133, no. 4, pp. 240–251, Apr. 2007, doi: 10.1061/(ASCE)0733-947X(2007)133:4(240).

[102] F. Zhao and X. Zeng, "Optimization of transit route network, vehicle headways and timetables for large-scale transit networks," *Eur J Oper Res*, vol. 186, no. 2, pp. 841–855, Apr. 2008, doi: 10.1016/J.EJOR.2007.02.005.

[103] M. A. Esfeh, S. Saidi, S. C. Wirasinghe, and L. Kattan, "Waiting time and headway modeling considering unreliability in transit service," *Transp Res Part A Policy Pract*, vol. 155, pp. 219–233, Jan. 2022, doi: 10.1016/J.TRA.2021.11.015.

[104] K. An and H. K. Lo, “Two-phase stochastic program for transit network design under demand uncertainty,” *Transportation Research Part B: Methodological*, vol. 84, pp. 157–181, 2016, doi: https://doi.org/10.1016/j.trb.2015.12.009.

[105] J. Yang and Y. Jiang, “Application of Modified NSGA-II to the Transit Network Design Problem,” *J Adv Transp*, vol. 2020, p. 3753601, 2020, doi: 10.1155/2020/3753601.

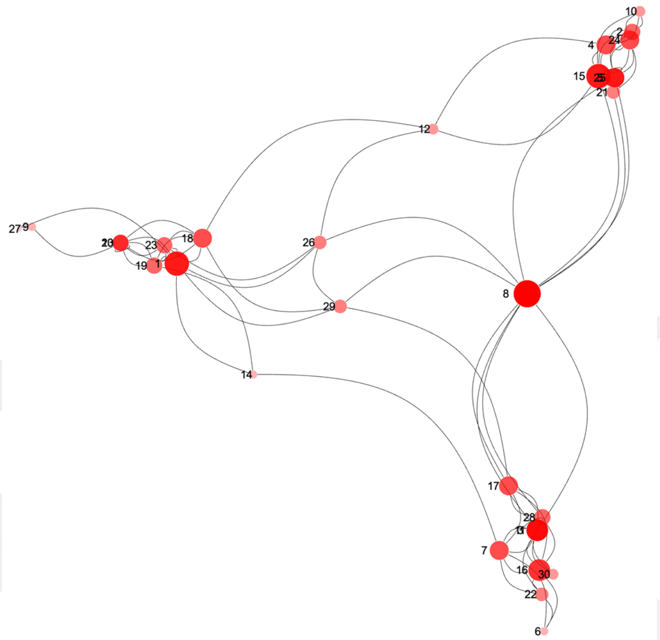
[106] S. Chai and Q. Liang, “An Improved NSGA-II Algorithm for Transit Network Design and Frequency Setting Problem,” *J Adv Transp*, vol. 2020, 2020, doi: 10.1155/2020/2895320.

[107] R. Borndörfer and M. Karbstein, “A direct connection approach to integrated line planning and passenger routing,” in *OpenAccess Series in Informatics*, 2012, pp. 47–57. doi: 10.4230/OASIcs.ATMOS.2012.47.

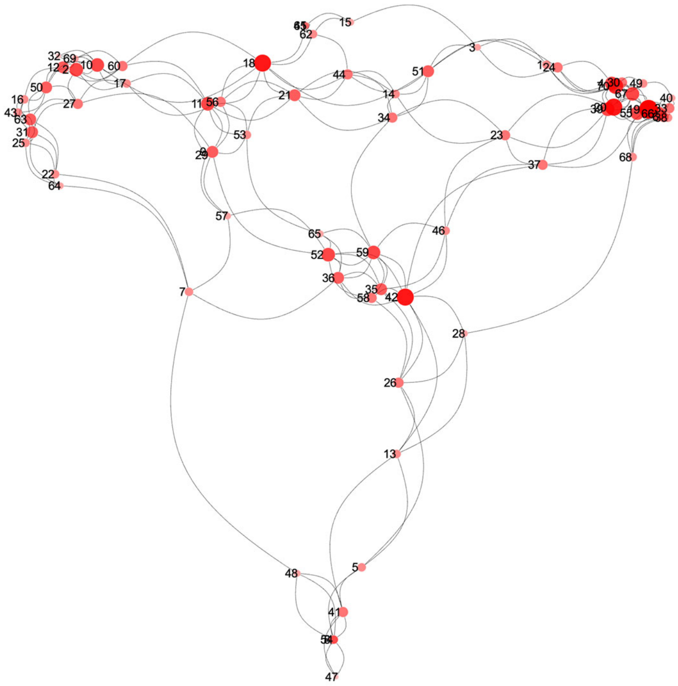
[108] “THE 17 GOALS | Sustainable Development.” Accessed: Dec. 06, 2022. [Online]. Available: https://sdgs.un.org/goals

APPENDIX A

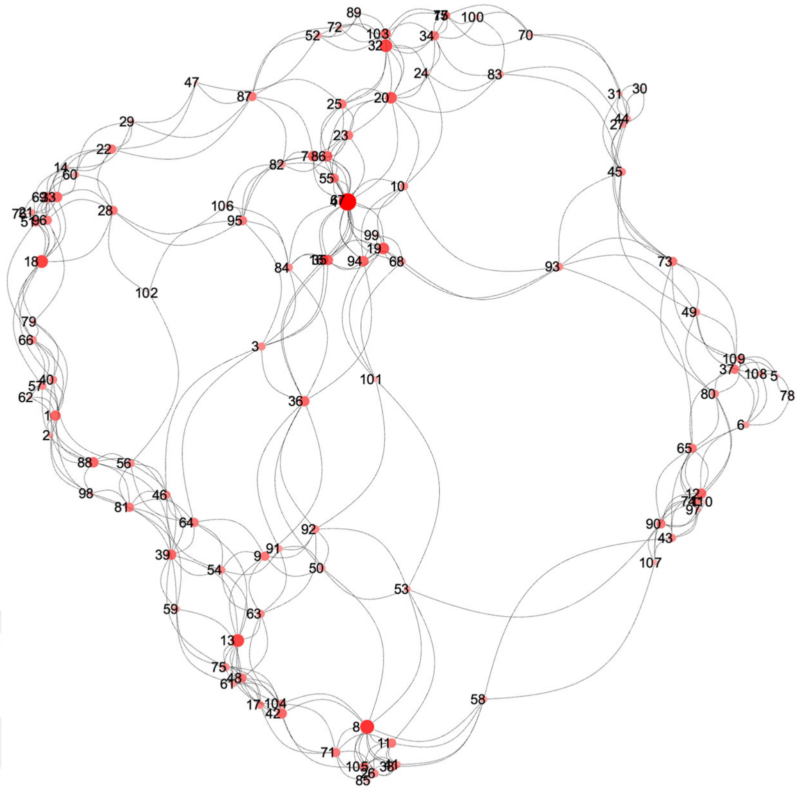
Layout of Mumford Instances



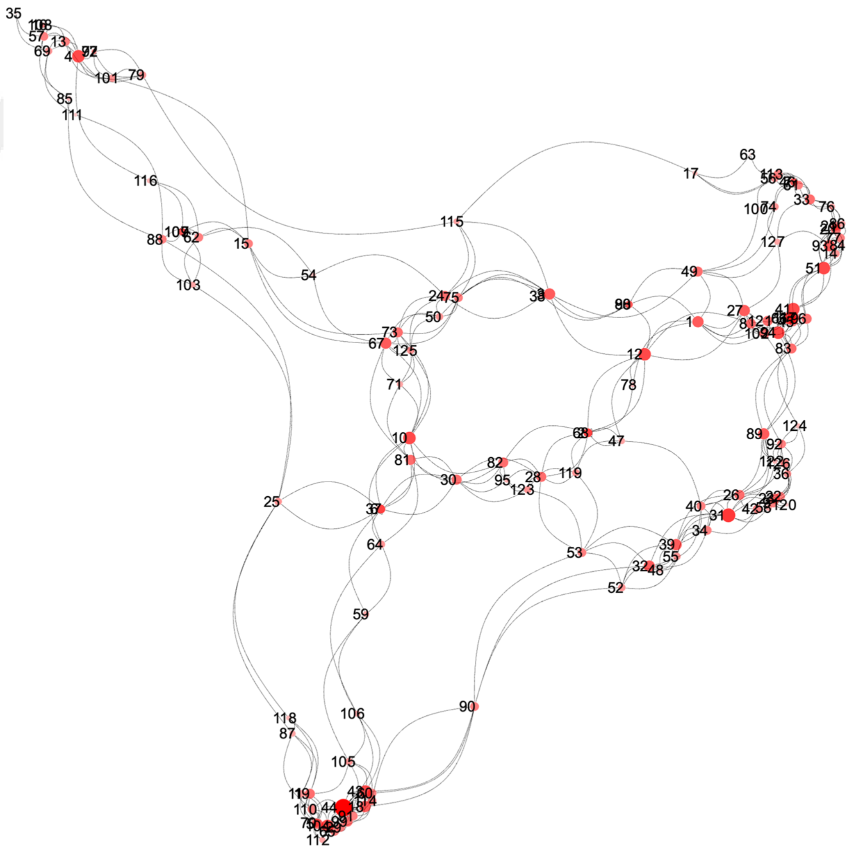
a) Mumford0



b) Mumford1



c) Mumford2



d) Mumford3

Figure A.1 Layout of Mumford Network Instances

CURRICULUM VITAE

2005 - 2011	BSc., Industrial Engineering (Full Scholarship), Istanbul Kultur University, İstanbul, TÜRKİYE
2011 - 2014	Regional Development Specialist, Southern Eagean Development Agency, Denizli, TÜRKİYE
2014 - 2017	MSc., Industrial Engineering, Abdullah Gul University, Kayseri, TÜRKİYE
2017 - 2023	Research Assistant, Department of Industrial Engineering, Abdullah Gul University, Kayseri, TÜRKİYE
2023 - present	Systems Engineer, Decision Support Systems, STM, Ankara, TÜRKİYE

SELECTED PUBLICATIONS AND PRESENTATIONS

J1) A. Benli and İ. Akgün, "A Multi-Objective Mathematical Programming Model for Transit Network Design and Frequency Setting Problem," *Mathematics*, vol. 11, no. 21, 2023, doi: 10.3390/math11214488.

C1) A. Benli and İ. Akgün, "A Mathematical Model for Transit Network Design and Frequency Setting Problem", 42th Operations Research and Industrial Engineering (YAEM) Congress, Pamukkale University, DENİZLİ, TÜRKİYE, October 26, 2022.

C2) A. Benli and İ. Akgün, "Blockchain in Supply Chains", 38th Operations Research and Industrial Engineering (YAEM) Congress, Anadolu University, ESKİŞEHİR, TÜRKİYE, June 28, 2018.

C3) A. Benli and İ. Akgün, "Flow-based p-Hub Median Interdiction Problem", 37th Operations Research and Industrial Engineering (YAEM) Congress, Yıldız Technical University, İSTANBUL, TÜRKİYE, July 06, 2017.

C4) A. Benli and İ. Akgün, "Prediction of Effects of Electricity Shortage on March 31 on Turkish Economy with Inoperability Input-Output Model", 35th Operations Research and

Industrial Engineering (YAEM) Congress, Middle East Technical University, ANKARA, TÜRKİYE, September 09-11, 2015.