arXiv:2212.03460v2 [math.OC] 12 Oct 2025

Bilevel Optimization and Heuristic Algorithms for Integrating Latent Demand into the Design of Large-Scale Transit Systems

Hongzhao Guan
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology,
hguan7@gatech.edu

Beste Basciftci
Department of Business Analytics, Tippie College of Business, University of Iowa,
beste-basciftci@uiowa.edu

Pascal Van Hentenryck
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology,
pascal.vanhentenryck@isye.gatech.edu

Capturing latent demand has a pivotal role in designing transit services as omitting these riders can lead to poor quality of service and/or additional costs. This paper explores this topic in the design of transit networks by considering the perspectives of both the transit agencies and riders. The paper presents a generic bilevel optimization model, namely the Transit Networks Design with Adoptions (TN-DA), that considers the network design decisions in the leader problem, and routing of the riders in the follower problem under the given network design, while allowing a black-box choice function for representing the adoption behavior of latent demand. The paper then identifies structural properties of the optimal solution of the TN-DA problem, which are desirable for transit agencies for capturing adoption behavior of the riders. The paper further provides guideline metrics for the transit agencies based on these desired adoption properties. Due to the computational complexity of this bilevel problem, the paper proposes five efficient heuristic algorithms to solve large-scale instances, which leverage an iterative procedure by solving a simpler version of the TN-DA problem and integrating the evaluation of rider choices. These algorithms either satisfy the desired properties of the optimal solution or provide fast approximations. The paper presents extensive large-scale case studies on two different transit systems by utilizing real datasets: (i) On-demand Multimodal Transit Systems (ODMTS) and (ii) Scooters-Connected Transit Systems (SCTS). The results demonstrate that the heuristic algorithms can find high-quality solutions much faster than exact approaches over various instances, while satisfying key adoption properties of the optimal solutions.

Key words: bilevel optimization, transportation network design, algorithm design, heuristics, latent demand, on-demand services

1. Introduction

Transit agencies have a critical concern in designing their transportation networks: how to ensure that a transit network design will capture ridership accurately and, in particular, customer adoption. This interaction between transit agencies (or in general, network designers) and latent riders, who represent the potential riders that can adopt the transit system, can be considered as a

game in the context of the transit network design problem. Essentially, this problems involves decision-making by both transit agencies and latent riders with potentially conflicting objectives. The objective of the transit agency revolves around boosting ridership by attracting latent riders while containing the rise in operational expenses. Simultaneously, latent riders expect to access high-quality services, which might incur substantial cost increases for the transit agency.

The decisions made by the transit agencies and riders are usually studied separately. More specifically, transit agencies focus on designing networks with estimated travel demand, a subject extensively explored in literature (e.g., Borndörfer, Grötschel, and Pfetsch (2007), Stiglic et al. (2018), Mahéo, Kilby, and Van Hentenryck (2019), Cats, Gkiotsalitis, and Schöbel (2025)). However, omitting potential riders that can adopt the transit system during the network design can lead to suboptimal designs with lower rider adoption and higher costs. As an example, this phenomena is observed in designing On-Demand Multimodal Transit Systems, which is an emerging form of public transportation system that tightly integrates on-demand shuttles with fixed transit services such as high-frequency buses and urban rail systems. Specifically, the transit agency tends to have higher costs and can provide poor quality of service for certain riders if the behavior of latent riders are omitted during the network design resulting in non-effective utilization of on-demand shuttles (Basciftci and Van Hentenryck 2020, 2023). When it comes to modeling the choices of the riders, research leans towards the utilization of logit and machine learning models for demand analysis (Xie, Lu, and Parkany 2003, Hagenauer and Helbich 2017, Zhao et al. 2020). However, it remains critical to integrate rider choices to the transit network design problems while taking into account the decision making processes of both the transit agency and riders. Consequently, this paper explores the possibilities of integrating the aforementioned two pieces into one single framework through a game theoretic approach. Specifically, the paper proposes a bilevel optimization framework which is called Transit Network Design with Adoptions (TN-DA), with a rider choice modeling integrated as a black-box-like component. The optimal solution of the TN-DA problem can be viewed as the equilibrium point between the transit agency and the latent riders. Furthermore, the TN-DA framework is highly generalized and can be tailored to various network design problems.

Due to the bilevel nature of the problem with the combinatorial choice function of the latent riders, solving TN-DA can be practically challenging, especially for large-scale instances. This paper thus explores whether heuristics can provide fast approximations to the optimal solutions of large-scale instances.

To this end, this paper first identifies the structural properties of the optimal solution of the TN-DA problem, in terms of the adoption behavior. These properties are desirable by the transit agencies in establishing the equilibrium between the transit agency and riders. Then, the paper

presents an iterative solution framework to significantly reduce the problem complexity by splitting the original bilevel framework into two components: (i) a regular transit network design problem without rider adoption and (ii) an evaluation component that uses a choice model to capture the adoption behavior and decides the set of riders and the network to be used in the next iteration. The heuristic algorithms combine these two components by increasing incrementally the sets of riders to be considered in the design and/or by progressively adding more fixed routes in the design, while guaranteeing to satisfy desired adoption properties. Therefore, this paper then proposes five heuristic algorithms to efficiently approximate the optimal solution of the TN-DA problem. The differences among these algorithms mostly occur at the transitions between these two primary steps, e.g, how they reshape the demand data and how they fix certain arcs in the network.

The effectiveness of these five heuristic algorithms is evaluated across two distinct transit network design problems. In addition to the objective value that can be used for evaluating the computational performance of the algorithms, this paper further proposes new metrics that assess the quality of the network designs by measuring the trade-off between the transit agency and the potential new users based on the properties of optimal solutions.

The primary contributions of this study can be summarized as follows:

1. This paper introduces the TN-DA framework, a generic bilevel optimization model that captures latent demand while designing transit network. The paper identifies adoption properties of optimal solutions, which inform heuristic design and new metrics: *False Rejection Rate* and *False Adoption Rate*, offering insights for evaluating and selecting transit network designs.
2. This paper proposes five heuristic algorithms to efficiently approximate optimal solutions of the TN-DA problem that can satisfy desired adoption properties or provide rapid approximations to the optimal solution.
3. This paper demonstrates the benefits of the heuristic algorithms on the TN-DA problem through large-scale case studies using real data on two transit systems: (i) On-demand Multimodal Transit Systems (ODMTS) and (ii) Scooter-Connected Transit Systems (SCTS). The results over various instances show that the heuristics are much more efficient than exact approaches by obtaining high-quality solutions in significantly shorter amounts of time while maintaining key adoption properties.

The remainder of this paper is organized as follows: Section 2 first presents the related work in transit network design and its connection with travel mode choices, followed by a discussion on the application of heuristic algorithms to various mobility problems. Section 3 introduces the generalized TN-DA problem. The five iterative algorithms are introduced in Section 4. Two detailed TN-DA examples, namely ODMTS-DA and SCTS-DA, are fully demonstrated with realistic case studies in Section 5 and 6. The conclusion and future research directions are discussed in Section 7.

2. Related Literature

Many existing methods for addressing transit network design problems are based on the assumption that the passenger demand is known and fixed (Schöbel 2012), whereas fewer number of studies have considered design/planning problems together with latent demand—potential customers who might be attracted by the proposed solutions. Klier and Haase (2008) presented an optimization model without a fixed demand by employing a linear demand function of the expected travel time to model its impact on transit demand. In another study, while maximizing the number of expected passengers, the same authors employed a logit model to estimate the mode choice between the best transit route and an alternative travel mode (Klier and Haase 2015). Additionally, a series of studies presented by Canca et al. comprehensively addressed this problem. Their first study introduced a model that simultaneously determines the network design, line planning, capacity, fleet investment, and passengers mode and route choices (Canca et al. 2016). Later on, two follow-up studies proposed adaptive large neighborhood search metaheuristics to solve variations of this model over large-scale data, respectively (Canca et al. 2017, 2019). Similarly, Laporte et al. (2005) aims to maximize trip coverage while considering rider behavior during the planning of rapid transit networks and Laporte et al. (2011) provide a review on this area. Although these studies highlight the importance of integrating rider behavior during the network design, it remains critical to consider the decision making of both the transit agency and riders within a multi-level formulation.

A bilevel formulation is a mathematical program where certain variables are constrained by the solution of another optimization problem, as described by Kleinert et al. (2021). This approach is valuable for modeling hierarchical decision processes and has extensive applications in transportation. Among the studies that consider latent ridership, Basciftci and Van Hentenryck (2020, 2023) introduced rider adoption awareness to the design of On-Demand Multimodal Transit System (ODMTS). These studies indicate that a bilevel optimization framework needs to be utilized to model the problem to ensure a fair design for the transit system that provides accessible transit services to the areas with existing demand and potential demand. The optimization model includes a personalized choice model which associates adoption choices with the time and cost of trips in the ODMTS. These two studies and a follow-up study (Guan, Basciftci, and Van Hentenryck 2024) provide significant insights into the design of methods that solve the problem under different choice model assumptions. However, these methods still encounter computational challenges in efficiently solving large-scale problems. Building on these studies, this paper presents a generalized bilevel formulation applicable to a broad range of transit network design problems with latent demand by allowing more comprehensive choice and price models, while proposing heuristic algorithms providing desirable properties of the optimal designs with improved computational efficiency.

In general, iterative approaches are widely applied to the field of transportation science, especially for complex large-scale problems. For example, the following studies investigated iterative approaches on problems related to large-scale network design such as designing distribution networks and planning public transit for large urban areas (Gattermann, Schiewe, and Schöbel 2016, Schöbel 2017, Cipriani, Gori, and Petrelli 2012, Liu et al. 2022). Studies presented by Siebert and Goerigk (2013) and Polinder et al. (2022) demonstrate a typical method for employing iterative algorithms in network design, which involves designing a network, establishing timetables, and then iteratively refining the design. When expanding the network design problem from city-wise to country-wise, another example of using iterative approaches to solve large-scale public transit problems is the study presented by Liu et al. (2019). Similar to ODMTS that will be investigated in Section 5, the goal of the problem is to optimize both the operational costs and passenger travel convenience. However, instead of using time to directly measure travel convenience, a conversion from travel time to *operational income* is utilized. Two case studies using the Chinese high-speed rail data presented by the authors demonstrate that iterative algorithms can efficiently solve large-scale public transit planning problems while providing profitable solutions. Furthermore, iterative approaches are utilized to solve more general network design problems such as the *incomplete hub location* problems—selecting hub locations without the restriction of fully-connected networks (Dai et al. 2019) and the *fleet assignment problem*—assigning a fleet while maximizing expected revenue (Dumas, Aithnard, and Soumis 2009). The latter study inspired Kroon, Maróti, and Nielsen (2015), and they proposed a three-step iterative approach to optimize the rolling stock to the disrupted passenger flows. Although these studies do not explicitly address latent demand in network design and omit investigating properties of optimal solutions, they highlight the efficiency of using iterative algorithms in designing large-scale logistic and urban transit networks.

Besides designing public transit networks, iterative approaches are also applied to other topics in the field of transportation science when the original problem has a bilevel nature. Yu, Yang, and Yao (2010) first proposed a bilevel optimization model to determine the optimal bus frequencies to minimize the total passengers' travel time, and an iterative algorithm that consists of a genetic algorithm and a label-marking method is then proposed to solve this model. Tawfik, Gendron, and Limbourg (2022) provided another example of applying iterative algorithms on bilevel problems by continuing on a previous study where a bilevel model formulates service design and pricing for freight networks (Tawfik and Limbourg 2019). Lodi et al. (2016) studied game theoretic frameworks between the transit agency and bus operators for optimizing certain contractual relationships in public transit systems, and employed iterative schemes as part of their solution procedure.

So far, not much attention has been paid to the role of iterative algorithms when integrating public transit planning and passenger mode choices. One of the earliest approaches is presented

by Lee and Vuchic (2005). They consider a transit network design problem with variable transit demand under a given fixed total demand that can be split into transit mode or personal vehicle mode, and this setting is also adopted by Fan and Machemehl (2006) within a genetic algorithm. The approaches used by these studies share two similarities with this paper—(i) the transit demand is variable such that riders can choose between transit services and personally-owned vehicles and (ii) at each iteration, the algorithm reshapes the transit demand then redesigns the transit network. On the other hand, this paper proposes a systematic approach for integrating rider preferences into the transit network design problem by introducing five iterative algorithms and leveraging the bilevel nature of the original problem. Three of these algorithms are driven by modifying trip sets considered in each iteration, whereas the other two are mainly driven by arc selection in the network. Furthermore, beside efficiently solving this large-scale problem, the iterative algorithms in this study also bring novelties by providing network designs with guaranteed adoption and rejection properties with insightful metrics that are practically useful to transit agencies.

3. Transit Network Design with Adoptions Problem (TN-DA)

This section introduces the general formulation of the Transit Network Design with Adoptions (TN-DA) problem. The motivation for the TN-DA problem stems from the desire to establish a transit network design $\mathbf{z}$ that concurrently serves the core riders (existing riders of the transit system) and captures latent demand with their adoption choice characteristics. Table 1 provides a summary of the nomenclatures that will be utilized in the remainder of this paper. Sections 3.1 and 3.2 describes the problem and introduces a generalized bilevel optimization framework for the TN-DA problem, respectively.

3.1. Problem Description

The TN-DA problem revolves around a set of nodes, denoted by N, which represents all the locations involved in the problem. Among these nodes, a subset $H \subseteq N$ represents potential connecting points for network arcs, e.g., high-frequency buses in a transit system. A network is represented by $\mathbf{z} \in \{0,1\}^{|H|\times|H|}$, where the binary variable z_{hl} represents whether an arc connecting nodes h and l in the set H is open or not. For any nodes $i, j \in N$, t^m_{ij} and d^m_{ij} represent the time and distance under travel mode m, respectively, between these nodes, where $m \in M$. The set of modes M can include various transportation modes considered in the problem, such as rail, bus, and walk.

To incorporate preferences of the riders into the planning process, the problem distinguishes between two sets of demand. The first set, denoted as T_{core}, represents riders who rely on transit systems and are guaranteed to use the newly designed system once it is deployed. A key assumption is that T_{core} remains fixed, meaning all individuals within this set are always transit users. The second set, denoted as T_{latent}, consists of riders who currently use other types of transportation

Symbol	Definition
General Sets:	
N	The set of all locations considered in the problem
H	The set of connecting points for the network, $H \subseteq N$
T	The set of all trips
T_{core}	The set of core trips, i.e., existing and fixed transit demand. Assumed to use transit under any conditions.
T_{latent}	The set of latent trips, i.e., latent demand. They are the potential customers of the system.
$\hat{T}$	The set of input trips for the TN-DFD problem. It is the union of T_{core} and a subset of T_{latent}.
M	The set of possible modes, e.g., shuttles, buses, scooters, and walk.
Z	The set of design arcs that are considered in the design problem
$\mathcal{Z}$	The set of all feasible solutions
General Parameters:	
d_{ij}^m	Travel distance between stops $i, j \in N$ under mode m
t_{ij}^m	Travel time between stops $i, j \in N$ under mode m
p^r	The number of riders in trip r
g_π^r	Travel cost for a rider in trip r under path π
φ_π^r	Transit agency's revenue from a rider in trip r under path π
ζ_π^r	Transit agency's net-cost to serve a rider in trip r under path π
t_π^r	Travel time for a rider in trip r under path π
t_{cur}^r	Travel time of a latent trip $r \in T_{latent}$ with its current travel mode
α^r	Adoption factor of a latent trip $r \in T_{latent}$, used in Equation (3)
Decision Variables:	
z_{hl}	A binary variable indicating if a network arc between $h, l \in H$ is open
δ^r	A binary variable indicating adoption decision of trip r, dynamically depending on $\mathbf{z}$
Sets for Algorithms:	
$\overline{T}^k$	The set of trips considered by the algorithms at iteration k
T_{adp}^k	The set of latent trips that adopt network design $\mathbf{z}^k$ at iteration k
C	The set of all adopted trips in previous iterations (Algorithms ρ-GRAD and ρ-GAGR)
T_ρ^k	The set of latent trips that transferred to C at iteration k (Algorithms ρ-GRAD and ρ-GAGR)
R	The set of all rejected trips in previous iterations (Algorithms η-GRRE and ρ-GAGR)
T_m^k	The set of latent trips that transferred to R at iteration k (Algorithms η-GRRE and ρ-GAGR)
D^k	The set of candidate sub-networks at iteration k (Algorithms arc-S1 and arc-S2)
T_{next}^k	The set of latent trips for next iteration's expansion (Algorithms arc-S1 and arc-S2)
Parameters for Algorithms:	
ρ	The amount of trips transferred to C at each iteration (Algorithms ρ-GRAD and ρ-GAGR)
η	The step-size applied to m (Algorithms η-GRRE and ρ-GAGR)
Variables for Algorithms:	
k	An integer that records algorithms' iterations
m	The amount of trips transferred to R at each iteration (Algorithms η-GRRE and ρ-GAGR)
B^k	The upper bound of design objective at iteration k (Algorithms arc-S1 and arc-S2)
p	Stage indicator (Algorithms arc-S1 and arc-S2)
v^r	A customizable value that is used to rank trips (Algorithms ρ-GRAD,η-GRRE, and ρ-GAGR)

Table 1 TN-DA **Notations. (For simplicity and consistency with others, notations such as t_{π^r} is written as t_π^r.)**

and have the option to switch to the new transit system. One can consider an example where a new transit system, denoted as $\mathbf{z}$, is proposed to replace the existing system. Core riders are individuals who currently utilize the existing transit system, and they are expected to continue using public transportation when the new system is implemented. On the other hand, latent riders do not currently rely on the existing transit systems and instead choose to drive their own cars for their travel needs. These individuals have not yet adopted public transportation as their preferred mode of travel. However, the introduction of a newly-designed transit network has the potential to attract and persuade latent riders to adopt public transit. The complete set of trips, denoted by T, is formed by combining T_{core} and T_{latent}. Each trip, $r \in T$, consists of an origin location, $or^r \in N$, a destination location, $de^r \in N$, and a group of p^r riders who travel together, where $p^r \in \mathbb{Z}+$. Within the context of this paper, the term "trip r" denotes a set of p^r riders who undertake a journey

collectively from their origin to destination. This implies that they share the same routes and possess common attributes and characteristics.

A path π^r in the transit network for a given trip r can be dynamically constructed based on the state of the network. The path should start at the trip origin node or^r, end at the trip destination node de^r, and may include zero or multiple intermediate nodes in between. The trip duration, cost, and revenue for a single rider to complete path π^r can be calculated by considering the legs of the path. The duration, cost, and revenue values are represented as t^r_π, g^r_π, φ^r_π, respectively, and the net-cost ζ^r_π incurred by the transit agency for serving the path π^r can be subsequently computed as $\zeta^r_\pi = g^r_\pi - \varphi^r_\pi$. It is important to note that in the given context, the terms *cost*, *revenue*, and *net-cost* do not necessarily refer to the monetary value associated with serving trip r. Instead, they represent customizable measures that can be defined based on specific criteria or considerations, although the most common approach is to use monetary measures.

3.2. Bilevel Optimization Framework

Figure 1 presents a generalized bilevel optimization framework for capturing the travel mode adoption during the transit network design by considering the core riders (T_{core}) and potential riders (T_{latent}) that can adopt the new system. The leader problem (1) corresponds to the network operator, which aims at designing a transit network, i.e., determining which arcs are connected to each other under adoption and problem specific considerations. The objective of the leader problem (1a) minimizes the sum of (i) the investment cost of opening network arcs, which is denoted by a function Invest($\mathbf{z}$), (ii) the net-cost incurred by core riders, and (iii) the net-cost of latent riders that are adopting the new transit system. Constraint (1b) corresponds to the customized restrictions in constructing the transit network design. Additional constraints on the arc variables $\mathbf{z}$ can be introduced based on specific problems; in this section, only the generic form is presented. Constraint (1c) represents the mode choice function of a latent trip depending on path π^r.

Adoption Choice Model In the TN-DA problem, a choice model is employed to capture the adoption decision of latent riders, taking a path π as its input. The binary output of $\mathcal{C}^r$ (i.e., δ^r in (1a) and (1c)) indicates whether all riders in latent trip r will adopt or reject the transit network design $\mathbf{z}$. The choice model $\mathcal{C}^r$ can be arbitrarily complex, which can range from a simple decision model to a sophisticated machine learning-based model. If available, information such as geographical and demographic data of riders can be incorporated into $\mathcal{C}^r$. The crucial aspect is that $\mathcal{C}^r$ is treated as a black-box, which implies the generalizability of the proposed framework.

A simple example of such a model is presented in Equation (3), namely Time-Based. This particular model compares the trip duration t^r_π with the travel duration t^r_{cur} under the current travel mode of the riders, multiplied by a parameter $\alpha^r \geq 1$. Thus, if the travel duration under π^r

$$\min_{\mathbf{z},\delta^r} \quad \textsc{Invest}(\mathbf{z}) + \sum_{r\in T_{core}} p^r(g^r-\varphi^r) + \sum_{r\in T_{latent}} p^r\delta^r(g^r-\varphi^r) \tag{1a}$$

$$\text{s.t.} \quad \mathbf{z}\in\mathcal{Z}\subseteq\{0,1\}^{|H|\times|H|} \tag{1b}$$

$$\delta^r = \mathcal{C}^r(\pi^r) \quad \forall r\in T_{latent} \tag{1c}$$

$$\delta^r\in\{0,1\} \quad \forall r\in T_{latent} \tag{1d}$$

where (π^r, g^r, φ^r) are a solution to the optimization problem OPT-PATH$^r(\mathbf{z})$ as follows:

$$\pi^r \in \arg\min \text{OPT-PATH}^r(\mathbf{z}) \quad \forall r\in T \tag{2a}$$

$$g^r = \text{PATH-COST}(\pi^r) \quad \forall r\in T \tag{2b}$$

$$\varphi^r = \text{PATH-REVENUE}(\pi^r) \quad \forall r\in T \tag{2c}$$

Figure 1 The Generalized Optimization Model for the Transit Network Design with Adoption (TN-DA**) Problem.**

is less than the desired threshold of the riders, then this choice function returns 1, representing adoption of the designed system.

$$\textsc{Time-Based}: \mathcal{C}^r(\pi) \equiv \mathbb{1}(t^r_\pi \le \alpha^r\ t^r_{cur}) \tag{3}$$

Subproblem Each trip r in the TN-DA problem is associated with an abstractly defined subproblem (or follower problem) called OPT-PATH^r. This subproblem takes the solution of the leader problem $\mathbf{z}$ as its input and produces an optimal path π^r specific to trip r. The subproblem OPT-PATH^r can be customized according to specific requirements. A common example of OPT-PATH^r is to find the shortest path from the origin or^r to the destination de^r under the network design $\mathbf{z}$, considering the duration as the primary criterion. It is important to note that the optimal solution of the subproblem (2) may not be unique. Therefore, it is necessary to apply constraints or rules to the subproblem (2) to ensure that the optimal paths do not result in different outcomes from $\mathcal{C}^r$, such as through a lexicographic optimization approach that can account for the choice function within the objective of the subproblem. This could prevent scenarios where two paths have the same objective function value, but one is categorized as a rejecting path while the other is categorized as an adopting path by $\mathcal{C}^r$. Furthermore, the optimal path's cost g^r and revenue φ^r, are computed with customized functions PATH-COST and PATH-REVENUE, respectively, and fed back as inputs to the leader problem, along with π^r. To ensure the existence of π^r, a direct path between the origin and destination using an always available travel mode can be included in the problem. Additionally, subproblem components can be further customized with additional considerations, for instance, to ensure that travel time does not exceed a specified limit.

To this end, the intuition behind the bilevel model can be summarized as follows: The network designer first obtains the design by considering the net-cost of the trips considered. Secondly, the follower problems determine the paths for the riders during operations. The choice function in the leader problem determines, for each trip with a mode choice, whether they would adopt the proposed path or elect to use their previous travel mode. If such riders decide to use the new system, the operator receives a revenue for that trip.

An instance of the TN-DA problem is the On-demand Multimodal Transit System Design with Adoptions (ODMTS-DA) problem, which was examined in previous studies by Basciftci and Van Hentenryck (2020, 2023), Guan, Basciftci, and Van Hentenryck (2024). After introducing the heuristic algorithms in the subsequent Section 4, Section 5 will revisit the ODMTS-DA problem and explore its connection to the algorithms in details. Another example of the TN-DA problem, namely the Scooters-Connected Transit Systems Design with Adoptions (SCTS-DA) problem, will be showcased in Section 6. Both case studies concentrate on emerging technologies in the evolving landscape of public transportation.

4. Heuristic Algorithms and Design Properties

Section 3 presents the generalized TN-DA problem within a bilevel optimization framework. However, the presence of hierarchical decision processes in bilevel optimization problems, combined with the combinatorial nature of the TN-DA problem with rider choices, introduce additional complexities, making it inherently difficult to solve in general. Therefore, this section presents heuristic algorithms to find high-quality solutions of the TN-DA problem in reasonable time, especially for large-scale instances. The heuristic algorithms use, as a key component, the Transit Network Design with Fixed Demand (TN-DFD) problem which is presented in Figure 2. Compared to the TN-DA, the TN-DFD removes the choice function constraint (1c) and the decision variables of the form δ^r that correspond to the user choices. The heuristic algorithms will use TN-DFD with a set of trips $\hat{T} \subseteq T$ that is always constructed from all core trips and a subset of the latent trips.

By leveraging the TN-DFD problem, the overall idea of the heuristics can be summarized in Algorithm 1. All algorithms start their first iteration by solving the TN-DFD problem with the core trip set T_{core} to obtain an initial transit network design, unless specified otherwise. Then, the set $\overline{T}^k$ denotes the set of trips considered by the algorithm at iteration k, i.e., $\hat{T} = \overline{T}^k$ at iteration k, where the latent trips are gradually added to $\overline{T}^k$ over the iterations depending on their adoption behavior under the resulting designs, while the trips in the core trip set remain in $\overline{T}^k$ as k increases. In certain algorithms, network design can be constructed incrementally by identifying a specific set of arcs to be fixed at each iteration.

Five heuristic algorithms are proposed later in this section which can be divided into two categories: (1) trip-based iterative algorithms that present rules for constructing the trip set $\hat{T}$ at

$$\min_{\mathbf{z}} \quad \text{INVEST}(\mathbf{z}) + \sum_{r \in \hat{T}} p^r (g^r - \varphi^r) \tag{4a}$$
$$\text{s.t.} \quad (1b)$$

where (π^r, g^r, φ^r) are a solution to the optimization problem OPT-PATH$^r(\mathbf{z})$ as follows:

$$(2a) - (2c) \quad \forall r \in \hat{T} \tag{5a}$$

Figure 2 **The Generalized Formulation for the Transit Network Design with Fixed Demand $\hat{T}$: TN-DFD($\hat{T}$).**

Algorithm 1 TN-DA Heuristic (Generic Form)

Set $k = 0$, $\overline{T}^0 = T_{core}$, no arcs are fixed, unless other input values are mentioned
while Stop Criteria Not Met **do**
  Solve $\mathbf{z}^k = \text{TN-DFD}(\overline{T}^k)$.
  Evaluate the design $\mathbf{z}^k$ and each latent trip $r \in T_{latent}$'s adoption decision δ^r using $\mathcal{C}^r$.
  Construct $\overline{T}^{k+1}$ using T_{latent} and/or fix a specific set of arcs by applying certain rules while ensuring that $T_{core} \subseteq \overline{T}^{k+1}$.
  Update k.
end while
return $\mathbf{z}^k$ unless another design is more desirable under certain rules.

each iteration and (2) arc-based iterative algorithms that focus on obtaining network designs incrementally by fixing a set of arcs at each iteration. Since the iterative algorithms only provide an approximation to the optimal solution of the TN-DA problem, especially on large-scale instances, two key properties are introduced in this section in order to assess a network design. Ideally, one would like to obtain the design that satisfies the following two properties. Here, the notation $\mathbf{z} = \text{TN-DFD}(\hat{T})$ implies that the optimal solution of the TN-DFD($\hat{T}$) problem is the network design $\mathbf{z}$. The set $T_{latent} \setminus \hat{T}$ represents the set of latent trips that are not considered during the design step, and the set $T_{latent} \cap \hat{T}$ represents the latent trips considered.

Property 1 (Correct Rejection). Let $\mathbf{z} = \text{TN-DFD}(\hat{T})$. Every user $r \in T_{latent} \setminus \hat{T}$ rejects the travel option in $\mathbf{z}$ obtained by OPT-PATH$^r(\mathbf{z})$.

Property 2 (Correct Adoption). Let $\mathbf{z} = \text{TN-DFD}(\hat{T})$. Every user $r \in T_{latent} \cap \hat{T}$ adopts the travel option in $\mathbf{z}$ obtained by OPT-PATH$^r(\mathbf{z})$.

When a design satisfies these two properties, it represents an equilibrium between the transit agency and the latent riders. The following proposition links the optimal solution $\mathbf{z}^*$ of TN-DA and Properties 1 and 2.

Proposition 1. Assume $\mathbf{z}^*$ is the optimal solution of the TN-DA problem (Figure 1). There exists an input trip set $\hat{T}^* \subseteq T$ such that $\mathbf{z}^* = \text{TN-DFD}(\hat{T}^*)$ and $\mathbf{z}^*$ satisfies both Correct Rejection and Correct Adoption properties (Properties 1 and 2).

Proof: Given $\mathbf{z}^*$ and $\boldsymbol{\delta}^*$ as the optimum solution of the TN-DA problem, $\hat{T}^*$ can be defined as $T_{core} \cup \{r \in T_{latent} : (\delta^r)^* = 1\}$. □

The goal of the heuristic algorithms is to find the set $\hat{T}^*$, which is computationally challenging to obtain. Thus, the proposed algorithms aim at achieving one of two different goals: (1) efficiently finding an approximation of $\hat{T}^*$ that satisfies the Correct Rejection property or Correct Adoption property, or (2) rapidly finding an approximation to $\mathbf{z}^*$ regardless of these properties. The rest of this section is organized as follows. Section 4.1 introduces key concepts for evaluating the designs found by the heuristic algorithms. Sections 4.2 and 4.3 describe the trip-based iterative algorithms and arc-based iterative algorithms, respectively. The discussion of the two properties is also presented throughout Sections 4.2 and 4.3 as the iterative algorithms are introduced.

4.1. Key Concepts for Design Evaluation

This section summarizes multiple common features shared by all proposed iterative algorithms. Since the TN-DFD is employed to approximate the optimal solution of the TN-DA, the objective function (1a) of the TN-DA is utilized to evaluate any network design $\mathbf{z}$ produced by the TN-DFD, using Equation (6). Regardless of the set $\hat{T}$ used to obtain the design $\mathbf{z}$, the input trip set to evaluate $\mathbf{z}$ is always the full trip set T. Under the design $\mathbf{z}$, this evaluation is performed using the choice decisions δ^r from $\mathcal{C}^r$ and the optimal path π^r associated with each trip r.

$$\text{Eval}(\mathbf{z}) = \text{Invest}(\mathbf{z}) + \sum_{r \in T_{core}} p^r (g^r_\pi - \varphi^r_\pi) + \sum_{r \in T_{latent}} p^r \delta^r (g^r_\pi - \varphi^r_\pi) \tag{6}$$

To evaluate the quality of the network designs found by the iterative algorithms, two additional metrics, i.e., False Rejection Rate ($\%R_{false}$) and False Adoption Rate ($\%A_{false}$), are introduced to measure the violation of the Correct Rejection and Correct Adoption properties (Properties 1 and 2). These two metrics are defined as follows:

$$\%R_{false} = \frac{|\{r \in T_{latent} \setminus \hat{T} : r \text{ adopts design } \mathbf{z}\}|}{|T_{latent}|} \times \%100$$

$$\%A_{false} = \frac{|\{r \in T_{latent} \cap \hat{T} : r \text{ rejects design } \mathbf{z}\}|}{|T_{latent}|} \times \%100$$

Here, $\%R_{false}$ indicates the percentage of latent trips that are excluded during the network design although they adopt the resulting design, implying false rejection, whereas $\%A_{false}$ corresponds to the percentage of latent trips that are included during the network design although they reject

the resulting design, implying false adoption. Core trips are excluded from the calculation of these metrics, because they are always included in $\hat{T}$ and not associated with a choice function. The subsequent two Remarks highlight that obtaining zero False Rejection Rate or False Adoption Rate is neither a necessary nor a sufficient condition for finding an optimal solution of TN-DA as these conditions depend on the trip set considered in the TN-DFD problem. Nevertheless, they establish desired adoption properties by identifying an equilibrium between the transit agency and latent riders based on the latent riders considered during the network design.

Remark 1. It is possible that the solution of the TN-DFD with a trip set $\hat{T} \neq \hat{T}^*$ is a design $\mathbf{z} \neq \mathbf{z}^*$ that satisfies both Correct Rejection property and Correct Adoption property. Thus, satisfying both properties under an input trip set $\hat{T}$ does not necessarily imply that $\mathbf{z}$ is the optimal solution of TN-DA.

Remark 2. It is possible that the optimal solution of the TN-DFD with a trip set $\hat{T} \neq \hat{T}^*$ is the optimal design $\mathbf{z}^*$ of the TN-DA. In other words, $\mathbf{z}^*$ may be discovered by a trip set $\hat{T} \neq \hat{T}^*$ with $\%R_{false} \neq 0$ or $\%A_{false} \neq 0$ under $\hat{T}$.

4.2. Trip-based Iterative Algorithms

This section introduces the first category of iterative algorithms—the *trip-based algorithms.* In general, each iteration of this type of algorithm can be summarized by three fundamental steps. The first step solves the TN-DFD problem with $\hat{T} = \overline{T}^k$. The second step evaluates the design with Eval($\mathbf{z}$) (Equation (6)). The third step computes the new $\overline{T}^{k+1}$ for the next iteration. All trip-based iterative algorithms have identical first and second steps, with the only difference occurring in the third step. At the third step, the trip-based algorithms need to compute a customizable value for each trip r, denoted as v^r (such as the net-cost $\zeta^r_\pi = g^r_\pi - \varphi^r_\pi$), based on the optimal path of that trip under the given network design. This value is then utilized in constructing the trip set to be considered in the subsequent iterations.

Three trip-based iterative algorithms are proposed. The first two are presented in Sections 4.2.1 and 4.2.2. Another one combining the previous two algorithms is explained in Section 4.2.3, which enables to explore more intermediate network designs, despite of its much longer expected running time. Section 4.2.4 presents some properties of the proposed trip-based algorithms.

4.2.1. Greedy Adoption Algorithm (*ρ-GRAD*) The first trip-based iterative algorithm starts with the core trips and greedily adds adopting trips at each iteration. Algorithm *ρ-GRAD* maintains a set C of trips and transfers latent trips from T_{latent} to C at each iteration. Specifically, at a given iteration k, the solution $\mathbf{z}^k$ identifies a set of adopting trips $T^k_{adp} \in T_{latent}$. It transfers ρ trips from T^k_{adp} to C, choosing those with the least v^r values. This set of ρ trips is denoted by T^k_ρ. The algorithm terminates when there are no latent trips to be absorbed by set C at the end of an iteration, i.e., $|T^k_\rho| = 0$. Algorithm *ρ-GRAD* satisfies the following property.

Algorithm 2 ρ-GRAD

Set $k=0$, $\overline{T}^0 = T_{core}$, $C=\emptyset$, unless other input values are specified.
Require: step-size parameter ρ
while True **do**
  Set $\mathbf{z}^k = \text{TN-DFD}(\overline{T}^k)$.
  Set $T_{check} = T_{latent} \setminus C$.
  for all $r \in T_{check}$ **do**
    Set $\pi^r = \arg\min \text{OPT-PATH}^r(\mathbf{z}^k)$.
    Set $\delta^r = \mathcal{C}^r(\pi^r)$.
    if $\delta^r = 1$ **then**
      Compute v^r for trip r.
    end if
  end for
  Set $T^k_{adp} = \{r \in T_{check} \mid \delta^r = 1\}$.
  Set T^k_ρ as the set of ρ trips from T^k_{adp} with the lowest v^r.
  if $T^k_\rho = \emptyset$ **then**
    break.
  else
    Set $C = C \cup T^k_\rho$.
    Set $\overline{T}^{k+1} = T_{core} \cup C$.
    Set $k = k+1$.
  end if
end while
return $\mathbf{z}^k$

Proposition 2. The last design $\mathbf{z}^k$ found by Algorithm *ρ-GRAD* with trip set $\hat{T} = \overline{T}^k$ satisfies the Correct Rejection property (Property 1), i.e., $\%R_{false} = 0$.

Proof: The last design $\mathbf{z}^k$ is found by solving TN-DFD with $\hat{T} = T_{core} \cup C$. On completion, $|T^k_\rho| = 0$ and no latent trip in $T_{latent} \setminus C$ adopts design $\mathbf{z}^k$. The result follows. □

4.2.2. Greedy Rejection Algorithm (*η-GRRE*) Algorithm *η-GRRE* greedily excludes trips from T_{latent} and introduces a growing set R of rejecting trips. In addition, a variable m is used to indicate how many adopting trips from $T_{latent} \setminus R$ are included in the set $\overline{T}^k$ and m is increased by η at each iteration. Using $T_{latent} \setminus R$ directly would lead to many rejections, so the algorithm proceeds carefully when choosing $\overline{T}^k$ early on. Algorithm *η-GRRE* terminates when it meets the following two conditions at an iteration $k \geq 2$:

1. the design is stable, i.e., $\mathbf{z}^k = \mathbf{z}^{k-1}$;
2. $m - \eta \geq |T^k_{adp}|$.

Note that condition 1 alone is not adequate. Indeed, in early stages, the algorithm may tend to find the same design because m is still relatively small. As a result, condition 2 is firstly employed to guarantee that the algorithms delve into a greater number of iterations. Nevertheless, it is crucial to acknowledge that beyond the point where m exceeds $|T^k_{adp}| + \eta$, the algorithm will commence

Algorithm 3 η-GRRE

Set $k=0$, $m=0$, $R=\emptyset$, $\overline{T}^0=T_{core}$, and $o^*=\infty$, unless other input values are specified.
Require: step-size parameter η
while True **do**
  $\mathbf{z}^k=$ TN-DFD$(\overline{T}^k)$.
  if $\textsc{Eval}(\mathbf{z}^k)<o^*$ **then**
    Set $o^*=\textsc{Eval}(\mathbf{z}^k)$.
    Set $\mathbf{z}_{min}=\mathbf{z}^k$.
  end if
  Set $T_{check}=T_{latent}\setminus R$.
  for all $r\in T_{check}$ **do**
    Set $\pi^r=\arg\min$ OPT-PATH$^r(\mathbf{z}^k)$
    Set $\delta^r=\mathcal{C}^r(\pi^r)$.
    if $\delta^r=1$ **then**
      Compute v^r for trip r.
    else
      Set $R=R\cup\{r\}$.
    end if
  end for
  Set $T^k_{adp}=\{r\in T_{check}\mid\delta^r=1\}$.
  Set $m=m+\eta$.
  Set $T^k_m=$ the set of m trips from $T^k_{adp}$with the lowest v^r.
  if the termination criteria is met **then**
    break.
  else
    Set $\overline{T}^{k+1}=T_{core}\cup T^k_m$;
    Set $k=k+1$.
  end if
end while
return $\mathbf{z}_{min}$

producing designs similar to those from earlier iterations. This is likely due to the decrease in $\hat{T}$ over iterations when many trips are included in R, which might also lead to oscillating results between iterations. Thus, condition 2 serves the purpose of striking a balance between these two effects. The designs found by Algorithm *η-GRRE* may not satisfy either the Correct Rejection property or the Correct Adoption property. Instead, this algorithm is particularly designed to provide a rapid approximation.

Remark 3. Because of possible oscillations in network designs between iterations, the *η-GRRE* algorithm reports the design with the minimal objective instead of the last design. These oscillations occur as the set $\overline{T}^k$ does not monotonically increase. Note that if the last design $\mathbf{z}^k$ is reported, then the algorithm satisfies the Correct Adoption property (Property 2) given that $\mathbf{z}^k=\mathbf{z}^{k-1}$.

4.2.3. Greedy Adoption Algorithm with Greedy Rejection Subproblem (*ρ-GAGR*)

Algorithm *ρ-GAGR* is the same as Algorithm *ρ-GRAD*, except that it uses Algorithm *η-GRRE* as its subroutine instead of solving the TN-DFD problem to obtain a network design. This design

decision makes it possible for Algorithm *ρ-GAGR* to explore many more designs than Algorithm *ρ-GRAD*. Algorithm *ρ-GAGR* shares the same terminating criteria with Algorithm *ρ-GRAD*. The full algorithm is presented in Appendix A.

4.2.4. Additional Properties of Trip-based Algorithms This section analyzes the change in the net-cost of the trips, as different sets of trips are considered during the network design. To this end, recall that $\zeta^r(\mathbf{z}) = g^r(\mathbf{z}) - \varphi^r(\mathbf{z})$ is defined as the optimum net-cost of the trip r under the network design $\mathbf{z}$.

Proposition 3. Let $\mathbf{z}^1 = \text{TN-DFD}(T^1)$ and $\mathbf{z}^2 = \text{TN-DFD}(T^2)$ with $T^1 \subseteq T^2$. Then $\sum_{r \in T^2 \setminus T^1} p^r(\zeta^r(\mathbf{z}^2) - \zeta^r(\mathbf{z}^1)) \leq 0$.

Proof: Note that the optimum objective function value of the network design $\mathbf{z}^2$ under the set of trips T^2 can be written as $\sum_{h,l \in H} \beta_{hl}\mathbf{z}^2_{hl} + \sum_{r \in T^2} p^r\zeta^r(\mathbf{z}^2)$. Since $\mathbf{z}^1$ is a feasible network design for this problem, the following relationship holds:

$$\sum_{h,l \in H} \beta_{hl}\mathbf{z}^2_{hl} + \sum_{r \in T^2} p^r\zeta^r(\mathbf{z}^2) \leq \sum_{h,l \in H} \beta_{hl}\mathbf{z}^1_{hl} + \sum_{r \in T^2} p^r\zeta^r(\mathbf{z}^1)$$
$$\sum_{h,l \in H} \beta_{hl}\mathbf{z}^2_{hl} + \sum_{r \in T^1} p^r\zeta^r(\mathbf{z}^2) + \sum_{r \in T^2 \setminus T^1} p^r\zeta^r(\mathbf{z}^2) \leq \sum_{h,l \in H} \beta_{hl}\mathbf{z}^1_{hl} + \sum_{r \in T^1} p^r\zeta^r(\mathbf{z}^1) + \sum_{r \in T^2 \setminus T^1} p^r\zeta^r(\mathbf{z}^1)$$
$$\sum_{h,l \in H} \beta_{hl}(\mathbf{z}^2_{hl} - \mathbf{z}^1_{hl}) + \sum_{r \in T^1} p^r(\zeta^r(\mathbf{z}^2) - \zeta^r(\mathbf{z}^1)) + \sum_{r \in T^2 \setminus T^1} p^r(\zeta^r(\mathbf{z}^2) - \zeta^r(\mathbf{z}^1)) \leq 0$$

Following an analogous relationship by comparing the network designs under the set of trips T^1 results in

$$\sum_{h,l \in H} \beta_{hl}\mathbf{z}^2_{hl} + \sum_{r \in T^1} p^r\zeta^r(\mathbf{z}^2) \geq \sum_{h,l \in H} \beta_{hl}\mathbf{z}^1_{hl} + \sum_{r \in T^1} p^r\zeta^r(\mathbf{z}^1)$$
$$\sum_{h,l \in H} \beta_{hl}(\mathbf{z}^2_{hl} - \mathbf{z}^1_{hl}) + \sum_{r \in T^1} p^r(\zeta^r(\mathbf{z}^2) - \zeta^r(\mathbf{z}^1)) \geq 0$$

Combining the above derivations, it follows that $\sum_{r \in T^2 \setminus T^1} p^r(\zeta^r(\mathbf{z}^2) - \zeta^r(\mathbf{z}^1)) \leq 0$. □

Corollary 1. If the set $T^2 \setminus T^1$ is a singleton $\{r\}$, then $\zeta^r(\mathbf{z}^2) \leq \zeta^r(\mathbf{z}^1)$.

These results have interesting consequences for Algorithms *ρ-GRAD* and *ρ-GAGR*. With a step size of 1, the added trip will not have a net-cost in the last design worse than those in the prior iteration. With larger step size, at least one of the added trips will satisfy this property. However, this property is not guaranteed for all the added trips in that iteration. Note that, as opposed to Algorithms *ρ-GRAD* and *ρ-GAGR*, this property does not necessarily hold for Algorithm *η-GRRE* as $\overline{T}^k \subseteq \overline{T}^{k+1}$ is not ensured in any iteration k.

4.3. Arc-based Iterative Algorithms

This section introduces the second type of iterative algorithms—the *arc-based iterative algorithms*. These algorithms begin with a pre-defined transit network design, denoted as $\mathbf{z}_{fixed}$, which can be either an empty network or an existing network such as a well-developed rail system. At the beginning of each iteration, these algorithms solve the TN-DFD problem and obtain a network design building on the open arcs in $\mathbf{z}_{fixed}$. Then, a sub-network from the resulting design is incrementally added in each iteration to enhance the overall network. The input trip set considered for the TN-DFD problem is then expanded, considering the adoption decisions of the riders. These steps are iterated until no improving sub-network can be added to the design. Since these algorithms select the best sub-network with respect to the improvement in the overall objective, arc-based algorithms produce a sequence of network designs whose evaluated objectives are monotonically decreasing. As a result, they find primal solutions quickly, which is not guaranteed in trip-based algorithms. In summary, arc-based algorithms satisfy the following properties:

1. The design objective, computed by function $\textsc{Eval}(\mathbf{z})$ (Equation (6)), monotonically decreases.
2. The transit network design $\mathbf{z}_{fixed}$ monotonically increases.

This paper proposes two arc-based algorithms. Section 4.3.1 presents the Arc-based Greedy Algorithm *arc-S1*, and Section 4.3.2 presents the algorithm *arc-S2* that extends *arc-S1* to overcome the potential challenges faced by *arc-S1* in terms of run time and solution quality. Both algorithms are parametrized by one or more functions that expand the set of users at each iteration. Section 4.3.3 introduces expansion rules to construct these Expand functions used by the algorithms.

4.3.1. Arc-based Greedy Algorithm (*arc-S1*) Algorithm *arc-S1* describes the arc-based greedy algorithm. It maintains an upper bound B^k that represents the best design objective found up to iteration k. Algorithm *arc-S1* also uses a generalization of the TN-DFD, where the second parameter represents the set of arcs that must be fixed to 1 in the TN-DFD solution (line 4).

During each iteration, after solving such a generalized TN-DFD, the resulting design $\mathbf{z}^k_{temp}$ is decomposed into two parts: $\mathbf{z}_{fixed}$ and $\mathbf{z}^k_{unfixed}$ (line 5). The partial design $\mathbf{z}_{fixed}$ denotes all the arcs which were constrained to be opened in the TN-DFD, while $\mathbf{z}^k_{unfixed}$ is the remaining assignment in $\mathbf{z}^k_{temp}$. Then, the sub-network candidate pool D^k needs to be carefully determined while ensuring that the addition of sub-networks does not lead to infeasibility in future iterations by considering the specific characteristics and requirements of the problem at hand. In particular, each iteration attempts to add a new sub-network $\mathbf{d}^k \in D^k$ to the fixed design $\mathbf{z}_{fixed}$. Algorithm *arc-S1* does this greedily, adding the sub-network $\mathbf{d}^k$ which provides the greatest decrease in B^k (lines 10–18). Therefore, the series $B^0, \ldots, B^k$ decreases monotonically over the iterations, while the design $\mathbf{z}_{fixed}$ expands (line 21), where the computational time of each iteration tends to decrease as the

Algorithm 4 arc-S1

1: Set $k=0$, $\mathbf{z}_{fixed}=\vec{0}$, $\overline{T}^0=T_{core}$, $B^0=\infty$.
2: **Require**: EXPAND function for increasing the set of users considered.
3: **while** True **do**
4: Set $\mathbf{z}^k_{temp}$ = TN-DFD($\overline{T}^k$,$\{(h,l) \mid z_{fixed_{hl}}=1 \quad h,l \in H\}$).
5: Set $\mathbf{z}^k_{unfixed}=\mathbf{z}^k_{temp}-\mathbf{z}_{fixed}$.
6: Set D^k to the set of all candidate sub-networks in $\mathbf{z}^k_{unfixed}$.
7: **if** $D^k=\emptyset$ **then**
8: break.
9: **end if**
10: **for all** $\mathbf{d} \in D^k$ **do**
11: **if** $(\mathbf{z}_{fixed}+\mathbf{d}) \in \mathcal{Z}$ (i.e. if the design is feasible) **then**
12: Set $obj_{\mathbf{d}}=$ EVAL$(\mathbf{z}_{fixed}+\mathbf{d})$.
13: **end if**
14: **end for**
15: Set $obj^k=\min\limits_{\mathbf{d}\in D^k}\{obj_{\mathbf{d}}\}$ and $\mathbf{d}^k=\arg\min\limits_{\mathbf{d}\in D^k}\{obj_{\mathbf{d}}\}$.
16: **if** $obj^k<B^k$ **then**
17: Set $B^{k+1}=obj^k$.
18: **else**
19: break.
20: **end if**
21: Set $\mathbf{z}_{fixed}=\mathbf{z}_{fixed}+\mathbf{d}^k$.
22: Set $\mathbf{z}^k=\mathbf{z}_{fixed}$
23: Set $T^k_{next}=$ EXPAND$(T_{latent},\mathbf{z}^k)$.
24: Set $\overline{T}^{k+1}=\overline{T}^k \cup T^k_{next}$.
25: Set $k=k+1$.
26: **end while**
27: **return** $\mathbf{z}^k$

number of variables decreases. Algorithm *arc-S1* also expands the trips to consider in the TN-DFD (lines 23–24). The algorithm terminates when the design objective bound is not decreasing, i.e., $B^k=B^{k+1}$, which can happen under the following two cases:

1. D^k is empty, i.e., it does not contain any new candidate sub-networks.
2. No sub-networks in D^k provides a better objective value.

4.3.2. Arc-based Extended Greedy Algorithm (*arc-S2*) The EXPAND function used in Algorithm 4 can have a significant impact on the performance of the algorithm, as it provides the balance in avoiding the set of trips in $\overline{T}^k$ from growing too quickly or too slowly over the iterations. A rapid expansion of the set $\overline{T}^k$ can introduce computational challenges when solving the TN-DFD problem. Conversely, a slow expansion of $\overline{T}^k$ may lead to premature termination of the algorithm, resulting in poor solutions. To overcome this difficulty, *arc-S2* implements a two-phase extension of *arc-S1* by using two EXPAND functions. Initially, *arc-S2* expands the set of trips slowly until convergence using function EXPAND$_1$. Then *arc-S2* enters its second phase where the trip set

increases faster. Both phases implement *arc-S1*, except that the second phase starts with the fixed design and the set of trips produced by phase 1. The full algorithm is presented in Appendix A.

4.3.3. Trip Expansion Rules The arc-based algorithms have been examined with various expansion rules to build the EXPAND functions over different transit systems. The initial rule, referred to as the "Full Adoption Expansion Rule", considers *all latent trips that adopt* $\mathbf{z}^k$ to be included in the subsequent iteration. This rule is intuitive and can be generally applied to various types of transit systems. In the following Sections 5 and 6, the paper will further demonstrate how expansion rules can be carefully designed within a specific context by leveraging the underlying problem. This will showcase the flexibility of the approach in adapting to different transit system scenarios while maintaining the underlying principles discussed earlier. When using this expansion rule, Algorithm *arc-S1* satisfies the CORRECT REJECTION property. This result extends to Algorithm *arc-S2*.

Proposition 4. The transit network design returned by Algorithm *arc-S1* with Full Adoption Expansion Rule satisfies the CORRECT REJECTION property (Property 1), i.e., $\%R_{false} = 0$.

Proof: When Algorithm *arc-S1* terminates at iteration k, the design $\mathbf{z}_{fixed}$ does not change. That means that the TN-DFD was executed with $\hat{T}^k = \overline{T}^{k-1} \cup T^{k-1}_{next}$. Moreover, T^{k-1}_{next} is the set of all latent trips that adopt $\mathbf{z}_{fixed}$. Hence, all latent trips that are not in $\hat{T}^k$ reject design $\mathbf{z}^k$ and Property 1 holds. □

Corollary 2. The transit network design returned by Algorithm *arc-S2* with the Full Adoption Expansion Rule in the second phase satisfies the CORRECT REJECTION property (Property 1).

5. TN-DA Example 1: ODMTS Design with Adoptions Problem

This section presents a detailed example of the TN-DA problem with three realistic case studies, focusing on a specific type of transit network known as On-demand Multimodal Transit Systems (ODMTS). Therefore, within this particular setting, the TN-DA problem is referred to as ODMTS-DA (ODMTS Design with Adoptions) Problem.

As depicted in Figure 3, ODMTS are operated around a number of hubs that serve as stops for buses and rail. The on-demand shuttles are primarily employed as the feeders to the hubs and represent an effective solution to the first-and-last-mile problem faced by most of the transit agencies. Several simulation studies have demonstrated that ODMTS can provide considerable cost and convenience benefits in different cities under diverse circumstances (Mahéo, Kilby, and Van Hentenryck 2019, Dalmeijer and Van Hentenryck 2020, Auad et al. 2021). In practice, ODMTS were also successfully deployed in the cities of Atlanta (Van Hentenryck et al. 2023) and Savannah (Chatham Area Transit 2024) in 2022 and 2024—2025, respectively. Using the ODMTS-DA as

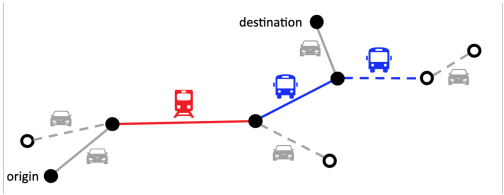


Figure 3 **Picture that illustrates the On-demand Multimodal Transit Systems (ODMTS). The solid circles represent transit hubs. The solid line represents a typical ODMTS path offered by the transit agency.**

an example of the TN-DA, the goal of this section is to demonstrate the practical advantages and outcomes of the heuristic algorithms in a real-world transit network design problem.

In the context of ODMTS-DA, two previous studies addressed this problem by modeling it with bilevel optimization frameworks (Basciftci and Van Hentenryck 2020, 2023). In particular, the agency proposes an ODMTS service and optimal paths for each user under the current design by considering both their duration and cost; the potential riders then evaluate the convenience of these paths, and decide whether to adopt the ODMTS service. It is important to note that this is a fundamental distinction between ODMTS and conventional transit systems, as the transit agencies provide the designated paths for riders by using on-demand shuttles and transit network, and potential riders make their adoption decisions accordingly.

The remainder of this section is structured as follows: Sections 5.1—5.3 study ODMTS-DA problem using the bilevel optimization framework and explain the employment of heuristic algorithms to solve ODMTS-DA. Section 5.4 provides an overview of the three real-world case studies, which are categorized by size as medium, large, and extra-large. Finally, Sections 5.5—5.7 present the results over the corresponding three cases.

5.1. ODMTS-DA Bilevel Formulation

The analogy between ODMTS-DA and the generalized framework TN-DA presented in Section 3 can be summarized as follows:

- Investment Function INVEST: Sums of cost of opening each design arc, where the cost of opening z_{hl} is denoted as β_{hl}. The unit of β_{hl} should be consistent with g^r and φ^r.
- Subproblem OPT-PATH^r: Finds the shortest path π^r for trip r from its origin to destination in terms of the cost g^r. It is important to emphasize that this subproblem aims to emulate the functionality of a rider mobile app (a realistic rider app example can be found in the Appendix of Van Hentenryck et al. (2023)), suggesting an ODMTS path by considering a convex combination of the agency's monetary cost and the rider's travel time. As a result, the solution to OPT-PATH^r is not necessarily the shortest path solely in terms of travel duration for trip r. Furthermore, the path can be multimodal, such as shuttle → bus → shuttle, or

simply a single shuttle leg, namely *direct shuttle*. However, ODMTS does not allow a path with only two shuttle legs involved, i.e., shuttle → shuttle. Note that the existence of direct shuttle trips will guarantee the subproblem is always feasible.

– Function PATH-COST: Computes the weighted cost g^r of path π^r, based on the travel time and on-demand shuttle operating cost.
– Function PATH-REVENUE: Returns a constant revenue value φ^r, because the ticket price is the same for all paths in ODMTS in practice.

The detailed bilevel formulation of ODMTS-DA can be found in Appendix B.

5.2. Existing Methods for Solving ODMTS-DA

To solve the ODMTS-DA problem, Basciftci and Van Hentenryck (2023) proposed an exact decomposition algorithm whose resulting design can be considered as the equilibrium point of the game between the transit agency and the potential transit users. The effectiveness of this exact algorithm was demonstrated in a case study conducted in Ann Arbor and Ypsilanti area in Michigan, USA. Nevertheless, this exact algorithm becomes computationally challenging on large-scale instances. Furthermore, due to the complexity of the exact algorithm, it is even practically challenging to obtain high-quality sub-optimal solutions on large-scale problems, demonstrating the need to develop effective heuristics for providing fast approximations to the optimal solutions of large-scale instances with desired adoption properties of the resulting network designs.

As an alternative approach, another method, known as P-Path, reformulates the corresponding bilevel problem into a single-level Mixed-Integer Programming (MIP) problem (Guan, Basciftci, and Van Hentenryck 2024). P-Path is capable of finding optimal solutions efficiently for medium-sized ODMTS-DA instances and within a reasonable time frame for large-scale instances. However, its computational resource requirements (e.g., memory) can be substantial because of the size of the resulting MIPs. Consequently, heuristic algorithms are beneficial, as they not only provide computational efficiency but might also offer a high-quality initial solution for P-Path.

5.3. Heuristic Algorithms for ODMTS-DA

This section illustrates how heuristic algorithms can be utilized to solve the ODMTS-DA problem and presents additional technical details to be utilized as part of the arc-based algorithms tailored to this problem setting. Building upon the general concepts discussed in Section 4, the heuristic algorithms make use of the ODMTS Design with Fixed Demand (ODMTS-DFD) problem and the choice model separately. The ODMTS-DFD problem is a specific form of the TN-DFD problem that is tailored to the ODMTS settings. The full formulation for ODMTS-DFD can be found in Appendix B. To evaluate a design found by ODMTS-DFD, the evaluation function Eval($\mathbf{z}$) is

adopted from the objective function of ODMTS-DA where π^r and δ^r can all be deduced from $\mathbf{z}$, and the revenue term φ^r for core trips are dropped because it remains as a constant value:

$$\text{EVAL-ODMTS}(\mathbf{z}) = \sum_{h,l \in H} \beta_{hl} z_{hl} + \sum_{r \in T_{core}} p^r g^r_\pi + \sum_{r \in T_{latent}} p^r \delta^r (g^r_\pi - \varphi^r) \tag{7}$$

To apply trip-based algorithms to ODMTS—Algorithms *ρ-GRAD*, *η-GRRE*, and *ρ-GAGR*, the customizable value v^r should be clearly defined in order to rank trips. The net profit of serving p^r riders of each trip r under a given network design $\mathbf{z}$ constitutes a critical factor in determining which latent trips should be considered. To this end, $fee(\pi^r)$ corresponds to the monetary shuttle operating cost of the trip r under the path π^r, and its monetary net-cost value can be defined as $\textit{net-spending}(\pi^r) = \textit{fee}(\pi^r) - \textit{income}(\pi^r)$ considering the income of utilizing the ODMTS under path π^r (which is a constant in this particular setting). It is important to reemphasize that *net-spending*, *fee*, and *income* represent monetary values, while cost g^r_π and revenue φ^r in the optimization frameworks are weighted values between monetary cost and convenience.

For arc-based algorithms—Algorithms *arc-S1* and *arc-S2*, the set of all candidate sub-networks D^k at each iteration k is restricted to include only directed bus cycles. Given that $\mathbf{z}^k_{temp}$, $\mathbf{z}^k_{unfixed}$, and $\mathbf{z}_{fixed}$ all satisfy constraint (10b), the algorithm of Johnson (1975) can be used to find the set D^k of directed cycles (lines 6 and 7 in Algorithms *arc-S1* and *arc-S2*, respectively). The arc-based algorithms have been tested with a number of expansion rules to construct the EXPAND based on the specific nature of ODMTS. The expansion rules used at iteration k includes:

(**a**) All latent trips that adopt $\mathbf{z}^k$. This is the Full Adoption Expansion Rule.

(**b**) All latent trips that adopt $\mathbf{z}^k$ and are not served by direct shuttle, i.e., they use fixed-routes.

(**c**) All latent trips that adopt $\mathbf{z}^k$ and satisfies $\text{UB}^r \leq \alpha^r \cdot t^r_{cur}$, where the term UB^r is an upper bound on ODMTS travel time. The derivation of this bound can be found in Appendix B.

The definition of rule (**c**) may seem overly complex; however, it leads to a useful proposition presented below, with its proof also provided in Appendix B.

Proposition 5. When using TIME-BASED as Choice Model $\mathcal{C}^r$ (Equation (3)) in ODMTS-DA, the ODMTS design found by Algorithm *arc-S1* with expansion rule (**c**) satisfies the Correct Adoption property (Property 2), i.e., $\%A_{false} = 0$.

Additionally, when applying Algorithm *arc-S2* in practice, it is worth noticing that the first rule should not be rule (**a**) as this rule includes all adopting latent trips, covering the trips considered in other expansion rules.

5.4. Overview on ODMTS-DA Case Studies

This section summarizes the three ODMTS-DA case studies. Table 2 provides a comparison of the settings across these cases by providing instance and solution approach information.

Size	Location	$\|N\|$	$\|H\|$	$\|Z\|$	$\|T_{core}\|$	$\|T_{latent}\|$	Congestion	Choice Model $\mathcal{C}^r$	Methods
Medium	Ypsilanti	1267	10	90	937	566	No	Time-Based	Exact Alg., P-Path, Heuristics
Large	Atlanta	2426	58	828	15,478	36,283	No	Time-Transfer-Based	P-Path, Heuristics
Extra Large	Atlanta	2426	66	2424	15,478	36,283	Yes	Time-Based	Exact Alg., Heuristics

Table 2 A summary on the three ODMTS-DA case studies

The first case study utilizes a real dataset from AAATA (https://www.theride.org Last Visited Date: September 30, 2025), the transit agency serving the greater Ypsilanti and Ann Arbor region of Michigan, USA. This case study assumes that the agency will replace the existing transit system entirely by implementing an ODMTS. Since the exact algorithm and P-Path can both successfully find the optimal design in a reasonable amount of time on this regular-sized dataset, this case study aims at demonstrating the workability of the heuristics proposed in Section 4 by comparing computational results obtained from different methods in terms of solution quality, adoption properties, and run time. Additional information on the dataset and experimental settings can be found in Appendix D.

The second case study conducted with real travel data represents a regular workday in Atlanta, Georgia, to evaluate the performance of the heuristic algorithms on large-scale instances. This case study assumes that the agency will retain the rail system, remove the existing bus services, and integrate the rail system with newly designed ODMTS. As a result, all existing four rail lines are preserved, and all 38 rail stations are designated as ODMTS hubs (set H). Visualizations and data processing details are available in Appendix E. As shown by Guan, Basciftci, and Van Hentenryck (2024), P-Path can efficiently find the optimal solution for large-scale ODMTS-DA instances within a reasonable time frame and clearly outperforms the exact algorithm in computational efficiency. Therefore, in this case study, only P-Path and the heuristic algorithms are compared. Furthermore, to showcase that ODMTS-DA can take the choice model $\mathcal{C}^r$ as a black-box, another choice model, namely Time-Transfer-Based, is applied. In this choice model, a path π is evaluated based on both travel duration and the number of required transfers, which is denoted by l^r_π. For instance, a "shuttle → rail → bus" path involves 2 transfers. Riders also have a transfer tolerance (l^r_{ub}), which is typically a low number and is set to 3 for all riders in this case study. The α value for all latent riders is set to 1.5.

$$\textsc{Time-Transfer-Based}: \mathcal{C}^r(\pi) \equiv \mathbb{1}(t^r_\pi \leq \alpha^r \ t^r_{cur}) \wedge \mathbb{1}(l^r_\pi \leq l^r_{ub}) \tag{8}$$

The third case study uses the same data as the second case study, however it incorporates a larger set for Z by considering more arcs between hubs (see details in Appendix E). Preliminary investigation indicates that P-Path cannot handle this setting due to its massive memory requirement, exceeding 500 GB with commercial solver Gurobi 9.5. However, the exact algorithm, which

is based on a decomposition approach, can partially solve this instance. Therefore, in this case, the exact algorithm is chosen as the baseline for evaluating the heuristic algorithms. The choice model is switched back to Time-Based to showcase the connection with Proposition 5. Additionally, this case study incorporates congestion modeling to enhance its realism.

The design objectives reported in the following subsections are based on Eval-ODMTS. Note that, although only a subset of trips is used to obtain the design in all algorithms, the design evaluation is computed with all core trips and latent trips to account for all riders considered.

5.5. Results: Medium Instance in Ypsilanti

Table 3 compares the computational performances of all solution methods along with the quality of their resulting network designs. The column '# Total Itr.' reports the number of iterations, corresponding to the number of ODMTS-DFD problems solved during the runs. The column '# Outer Itr.' indicates the number of iterations for *ρ-GAGR*, where its corresponding total iterations represent the number of iterations of *η-GRRE* (its subproblem). First, it can be seen that Algorithm *ρ-GAGR* requires significantly more computational time, whereas Algorithm *η-GRRE* takes about the same amount of computational time as Algorithm *ρ-GRAD*. Note that a practical approach to reduce the running time is to increase the step sizes of the trip-based algorithms. On the other hand, arc-based algorithms take less time compared to the trip-based algorithms. Secondly, four out of seven heuristic runs find the optimal solution, whereas Algorithm *arc-S1* (with both rule (**a**) and rule (**c**)) and Algorithm *arc-S2* (with rules (**c**) and (**a**)) report solutions with small optimality gaps. Despite having the same gap, the two arc-based algorithms report different results, due to their different expansion rules. Moreover, Propositions 2, 4, and 5 can be related to the results on False Rejection Rate and False Adoption Rate. In general, for transit agencies, small $\%R_{false}$ or $\%A_{false}$ values are indications of advantageous ODMTS designs when the optimal solution is unavailable. Three additional experiments with different parameter settings are provided in Appendix D, demonstrating the performances of the proposed algorithms while providing network designs with desired adoption properties.

5.6. Results: Large Instance in Atlanta

Table 4 presents the results from heuristic algorithms on a large-scale instance, using the optimal solution from the P-Path model as a benchmark. For trip-based iterative algorithms, three step sizes, i.e., 1000, 2000, and 3000, were tested, and the step size associated with the best design objective is reported. The results show that while the heuristics do not reach the optimal solution, they still produce high-quality designs in a very short amount of time, whereas the optimal solution requires nearly five days of computation. Note that due to the structure of the ODMTS-DFD problem, the heuristics can utilize multi-threading techniques; here, 12 threads are employed.

Method	Step Size	Trip Rule	# Outer Itr.	# Total Itr.	Run Time (min)	Design Obj	% Opt Gap.	% Gap	$\%R_{false}$	$\%A_{false}$
Exact Alg.	-	-	-	-	360.29	29962.79	1.85	-	0.00	0.00
P-Path	-	-	-	-	3.34	29962.79	-	0.00	0.00	0.00
Alg. *ρ-GRAD*	10	-	1	26	2.48	29962.79	-	0.00	0.00	0.53
Alg. *η-GRRE*	10	-	1	26	2.49	29962.79	-	0.00	11.13	0.53
Alg. *ρ-GAGR*	10	-	26	357	47.14	29962.79	-	0.00	0.00	1.41
Alg. *arc-S1*	-	(**a**)	1	8	1.05	30019.14	-	0.19	0.00	34.28
Alg. *arc-S1*	-	(**c**)	1	7	1.13	30020.26	-	0.19	13.78	0.00
Alg. *arc-S2*	-	(**b**), (**a**)	1	8	1.06	29962.79	-	0.00	0.00	2.65
Alg. *arc-S2*	-	(**c**), (**a**)	1	8	0.95	30020.26	-	0.19	0.00	0.00

Table 3 Performance comparison on the medium instance. The exact algorithm reports optimality gap after 6 hours. The heuristics report the gap between themselves and the optimal solution (P-Path**).**

Additionally, using the solution from Algorithm *arc-S2* as a warm-start for P-Path (because it provides the best solution in a shorter amount of time) can help reduce overall computation time. However, as pointed out by Guan, Basciftci, and Van Hentenryck (2024), starting P-Path with a heuristic solution does not necessarily guarantee a faster path to the optimal solution.

One may wonder why heuristics are necessary if an optimal solution is available for instances up to a certain size, even though it is time-consuming to obtain. One of the reasons is that a transit agency may need to quickly develop a transit plan for special events, and not all locations have access to powerful computing resources capable of solving a large-scale P-Path model. In these circumstances, the heuristics can be useful. Furthermore, as noted by Guan, Basciftci, and Van Hentenryck (2024), P-Path may spend a substantial amount of computing time on closing the final 1% of the optimality gap during the branch-and-bound processes. The solution reported here required 73% of its total computation time (around 85 out of the 116.48 hours) to address this last 1%, while the heuristics can achieve a better solution within just a few minutes.

Method	Best Step Size	Trip Rule	# Outer Itr.	# Total Itr.	Run. Time (hour)	Design Obj	%Gap	$\%R_{false}$	$\%A_{false}$
P-PATH	-	-	-	-	116.48	214463.71	0.00	0.00	0.00
P-PATH Warm	-	-	-	-	88.19	214463.71	0.00	0.00	0.00
Alg. *ρ-GRAD*	3000	-	1	8	0.07	214906.96	0.21	0.00	7.90
Alg. *η-GRRE*	1000	-	1	17	0.15	221265.96	3.17	4.72	0.84
Alg. *ρ-GAGR*	2000	-	10	69	0.66	214906.96	0.21	0.40	6.70
Alg. *arc-S1*	-	(**a**)	1	20	0.72	214846.58	0.18	0.00	14.37
Alg. *arc-S1*	-	(**c**)	1	13	0.31	242229.26	12.95	31.43	0.00
Alg. *arc-S2*	-	(**b**),(**a**)	1	22	0.60	214846.58	0.18	0.00	7.29
Alg. *arc-S2*	-	(**c**),(**a**)	1	22	0.50	214906.96	0.21	0.00	11.60

Table 4 A comparison between P-PATH **and the heuristics on the large instance.** P-PATH **is solved to optimal. Warm stands for warm-start, the reported time includes both the heuristic and the** P-Path **computation time.**

5.7. Results: Extra-Large Instance in Atlanta

This section evaluates the computational results of the extra-large case study from various viewpoints along with the analyses on the resulting network designs. Additional discussions on results such as adoptions, operating cost, travel time, and travel distance can be found in Appendix F.

Algorithmic Performances Table 5 compares the performances of all algorithms. The most obvious finding to emerge from the analysis is that the heuristic algorithms can outperform the exact algorithm in a reasonable amount of time. Moreover, after 48 hours, the exact algorithm finds a solution with objective 213828.87, which is only 1.61% better than the 24-hour running time result. Thus, the proposed iterative algorithms can efficiently provide high-quality solutions to the ODMTS-DA problem. The only exception is Algorithm *arc-S1* with rule (**c**), whose greater objective value can be attributed to the strict nature of rule (**c**).

Method	Best Step Size	Trip Rule	# Outer Itr.	# Total Itr.	Run. Time (hour)	Design Obj.	% Gap	$\%R_{false}$	$\%A_{false}$
Exact Alg.	-	-	-	-	26.20	220557.59	-	-	-
Alg. *ρ-GRAD*	3000	-	1	9	1.38	209548.18	-4.99	0.00	7.18
Alg. *η-GRRE*	3000	-	1	11	1.66	213409.00	-3.24	5.46	0.49
Alg. *ρ-GAGR*	3000	-	8	53	9.72	209412.04	-5.05	1.41	4.07
Alg. *arc-S1*	-	(**a**)	1	20	5.58	208664.07	-5.39	0.00	20.16
Alg. *arc-S1*	-	(**c**)	1	11	0.74	246561.82	11.79	63.78	0.00
Alg. *arc-S2*	-	(**b**), (**a**)	1	25	3.69	211253.00	-4.22	0.00	11.48
Alg. *arc-S2*	-	(**c**), (**a**)	1	27	4.75	208294.23	-5.56	0.00	18.48

Table 5 Performance comparison between all algorithms on the extra-large instance. The exact algorithm reports the best feasible solution within a 24 hours running-time limit. '% Gap' shows the gap between the objective value reported by the heuristics and the objective reported by the exact algorithm.

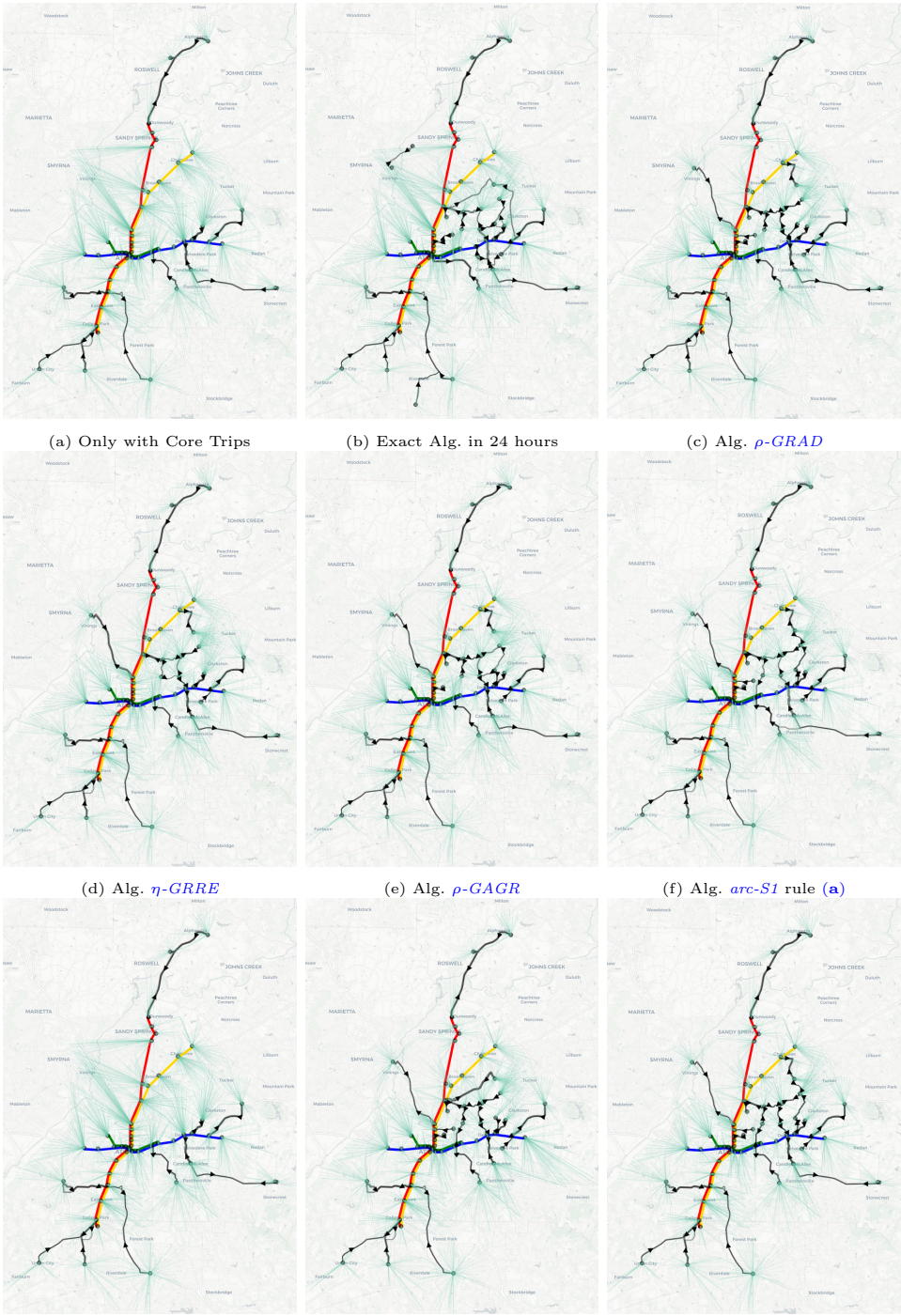
The table also shows that Algorithms *ρ-GRAD* and *η-GRRE* achieve the second and third fastest termination among all iterative algorithms, although they fail to discover a better solution compared to the other approaches. These performances are likely to be related to the adoption set C of Algorithm *ρ-GRAD* and the rejecting set R of Algorithm *η-GRRE*. These sets are designed to speed up the design searching processes, yet it leads the algorithm to overlook multiple intermediate ODMTS designs. Note that multi-threading is disabled for this instance in order to produce a fair comparison on all algorithms. However, when the multi-threading techniques for subproblems in ODMTS-DFD are allowed, the heuristics can be massively sped up. For instance, with 18 threads and a step size of 3000, Algorithm *η-GRRE* can terminate in 0.24 hours. On the other hand, the combined trip-based iterative algorithm can indeed explore more ODMTS designs and provide a lower objective: algorithm *ρ-GAGR* reports a better objective than Algorithm *η-GRRE*. However, it does not outperform Algorithm *ρ-GRAD*. A possible explanation for this result can be the choices of step sizes.

In this particular case study, Algorithm *ρ-GRAD* produces a better balance between running time and design objective among all the trip-based iterative algorithms. The reported $\%R_{false}$ value is consistent with Proposition 2. In other words, with ODMTS design with $\%R_{false} = 0$, all riders excluded during the designing procedure reject the system. However, there will be $\%A_{false}$ riders who reject the system even though they were included in the design procedure, e.g., 7.18% latent trips reject the design when using Algorithm *ρ-GRAD* although they are in $\hat{T}$. Thus, this type of ODMTS designs can be treated as an upper limit when agencies plan to expand their transit network to serve more passengers.

For the arc-based iterative algorithms, Algorithm *arc-S1* has consistent $\%R_{false}$ and $\%A_{false}$ values with Propositions 4 and 5. Algorithm *arc-S1* with rule (**c**) is the only case that guarantees $\%A_{false} = 0$ in this study. That is, all riders used to design the ODMTS service will adopt it. Although this is a relatively conservative approach, the transit agency can start to expand the services based on this ODMTS design. Secondly, the extended version of Algorithm *arc-S1*, i.e., Algorithm *arc-S2*, also found high-quality solutions with a considerable drop in running time and more iterations compared to Algorithm *arc-S1* with trip rule (**a**). This result can be explained by the fact that the expansion rule of Algorithm *arc-S1* is significantly looser than the first rule employed by Algorithm *arc-S2*. Thus, the size of the $\hat{T}$ in Algorithm *arc-S2* for each iteration is substantially smaller, resulting in shorter running time per iteration. Closer inspection of the table shows that the solution found by Algorithm *arc-S2* (with rules (**c**) and (**a**)) outperforms all other algorithms and successfully establishes the smallest upper bound on the design objective in a relatively short duration. Furthermore, the results found by Algorithm *arc-S2* are also consistent with Corollary 2.

In summary, the results compare the algorithmic performances of all heuristic algorithms. A few practical recommendations for choosing the algorithms might then be summarized: (i) use Algorithms *ρ-GRAD* or *η-GRRE* with relatively large step size to conduct a fast exploration process, (ii) employ Algorithm *arc-S1* or Algorithm *arc-S2* with certain trip expansion rules in order to build the upper bound of the problem, and (iii) apply either Algorithm *ρ-GRAD* or Algorithm *arc-S2* when the balance between solution quality and running time is demanded.

ODMTS Designs The ODMTS designs reported by the exact algorithm and all iterative algorithms are illustrated in Figure 4. In particular, Figure 4a presents the design when solely using the core trip set T_{core}, i.e., $\hat{T} = T_{core}$. It is thus the ODMTS design discovered by all trip-based heuristic algorithms at the first iteration. The other figures show the ODMTS designs obtained from the algorithms. The non-direct shuttle legs are on-demand shuttle services that connect a hub and a rider's origin stop or destination stop. For clearer interpretation of the maps, the main concentrations of latent trips in this case study are in the East Atlanta region (east of the red



(a) Only with Core Trips (b) Exact Alg. in 24 hours (c) Alg. *ρ-GRAD*

(d) Alg. *η-GRRE* (e) Alg. *ρ-GAGR* (f) Alg. *arc-S1* rule (**a**)

(g) Alg. *arc-S1* rule (**c**) (h) Alg. *arc-S2* rules (**b**) (**a**) (i) Alg. *arc-S2* rules (**c**) (**a**)

Figure 4 **The ODMTS designs found by the Algorithms for the extra-large case study. Black polylines and green straight lines are utilized to represent opened bus arcs and opened non-direct shuttle legs, respectively. Thickness of the black polylines indicates the number of users using the arcs. The rail systems are in red, yellow, dark green, and blue.**

and yellow rail lines and north of the blue rail line), which is also where the primary differences in the designs appear. Additional details can be found in the latent demand heatmaps provided in Appendix E.2. Compared to the design in Figure 4a, it can be seen that additional trips in $\hat{T}$ generally lead to a more complex network in the East Atlanta region. Closer inspection of all ODMTS designs show that the majority of bus arcs are used to deliver riders to the rail system. This might be explained by the fact that the rail system operates at a high frequency and connects the major districts such as Downtown (where all 4 rail lines intersect), Midtown (just north of Downtown, where the red and yellow lines overlap), and the city of Sandy Springs (the northernmost section of the red line). The simplest design among Figure 4 is found by Algorithm *arc-S1* with rule (**c**), illustrated in Figure 4g, which is identical to the design with only core trips as input. The designs found by arc-based algorithms with looser expansion rule can be seen in Figures 4f, 4h, and 4i, and these designs are relatively more convoluted as expected. Additionally, Figure 4b presents the design reported by the exact algorithm after 24 hours running time. Comparing to the other designs, this design shows a few differences in the areas with latent riders, e.g., in the South and Northwest regions. This can be explained by the 1% or 8 hours terminating criteria for the master problem in the ODMTS-DA problem (Figure 6).

6. TN-DA Example 2: SCTS Design with Adoptions Problem

In order to show that TN-DA can be tailored for different transit network design problems, this section introduces another example, namely Scooters-Connected Transit Systems Design with Adoptions (SCTS-DA). As pointed out by McQueen and Clifton (2022) and Yan et al. (2023), for connecting to transit lines, e-scooters started to emerge as a solution to the first and last mile challenges. Different to the ODMTS example, the SCTS-DA problem assumes there are two independent service operators involved: (1) a transit agency responsible for operating high-frequency buses and rails, and (2) a Transportation Network Company (TNC) that offers e-scooter services. This problem assumes a relatively simplified scenario where there is a sufficient number of e-scooters available throughout the city, and the TNC is responsible for managing and relocating the e-scooters as needed. In this context, the primary goal of the problem is to design an efficient high-frequency bus network that can be seamlessly connected with the e-scooter services while taking into account rider adoption considerations.

6.1. SCTS-DA Bilevel Formulation

The analogy between SCTS-DA and the generalized framework TN-DA, can be summarized as follows:

– Investment Function Invest: Sums of cost of opening each design arc, where the cost of opening z_{hl} is denoted as β_{hl}. The unit of β_{hl} should be consistent with g^r and φ^r.

– Subproblem OPT-PATH^r: Finds the shortest path π^r for trip r in terms of the cost g^r. Assume that the riders' primary goal is minimizing travel time; this is equivalent to the conventional shortest path based on travel time. A direct scooter trip between the origin and destination ensures that the subproblem is feasible.
– Function PATH-COST: Computes the weighted cost g^r of path π^r. The cost g^r is weighted based on π^r's travel time of path t^r_π.
– Function PATH-REVENUE: Returns constant revenue value φ^r because transit ticket price is flat for all users. Scooter fees are not considered as part of the revenue as the scooter services are not offered by the transit agency.

The detailed bilevel formulation of SCTS-DA can be found in Appendix C.

The decision model employed in this case study is referred to as Time-Based-Scooter (Equation (9)). It takes into account three factors: firstly, the travel duration within the transit system is required to be smaller than or equal to an α^r value multiplied by the travel duration of the current mode of transportation. Secondly, riders will only adopt the transit system if they actually utilize buses or trains, which will incur income for the transit agency. Lastly, riders only use the transit system if they travel on scooter less than a threshold value d^r_{ub} given that scooters are not an efficient mode for longer distances. In the following equation, x_{hl} and y_{ij} denote if trip r use design arc (h,l) and scooter arc (i,j). The travel distance between locations i and j is d_{ij}.

$$\textsc{Time-Based-Scooter}: \mathcal{C}^r(\pi) \equiv \mathbb{1}(t^r_\pi \leq \alpha^r \ t^r_{cur}) \wedge \mathbb{1}\left(\sum_{h,l\in H} x^r_{hl} > 0\right) \wedge \mathbb{1}\left(\sum_{i,j\in N} y^r_{ij} \cdot d_{ij} \leq d^r_{ub}\right) \quad (9)$$

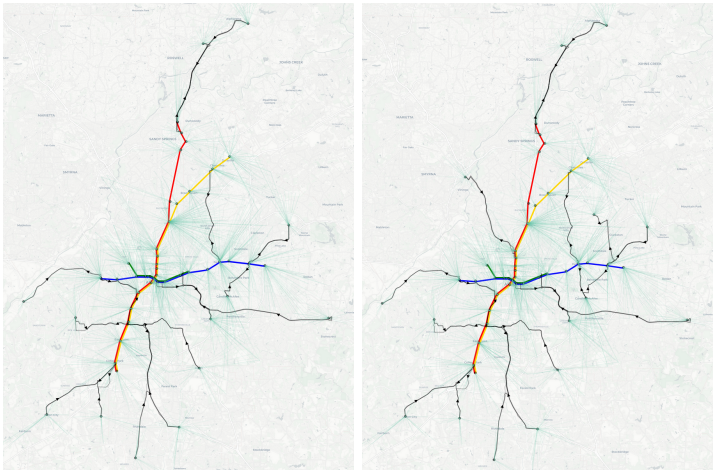
6.2. Heuristic Algorithms for SCTS-DA

This section presents the heuristic settings for SCTS-DA, the formulations and the evaluation function are documented in Appendix C.

In the case of the trip-based iterative algorithms, the value used to rank trips, denoted as v^r, is determined based on the time traveled by e-scooters since e-scooter services typically charge riders based on the amount of time they utilize the scooters. For arc-based algorithms, the expansion rules used at iteration k includes:

(i) All latent trips that adopt $\mathbf{z}^k$. This is the Full Adoption Expansion Rule.
(ii) All latent trips that adopt $\mathbf{z}^k$ and use the backbone transit lines. Backbone refers to arcs that are preserved in the system without requiring design through solving TN-DA, e.g., rail system in Atlanta.

The first rule considers an expansive case by including all latent riders adopting, whereas second rule considers a subset of these trips by reflecting a common scenario where scooters are used for first-and last-mile connections to primary transit corridors, typically rail lines. For instance, these transit stations are usually equipped with convenient parking areas for scooters in Atlanta.



(a) Trip-Based Algorithms (b) Arc-Based Algorithms and Modified P-Path

Figure 5 The Designs Found by the Heuristic Algorithms and modified P-Path **for the SCTS-DA problem in Atlanta. Light green lines represent scooter connections.**

6.3. Results: SCTS-DA in Atlanta

Similar to the large-scale ODMTS-DA case studies explained in the previous section for Atlanta, this case study employs the same dataset and parameters for buses and rails (see Appendix E.5). For scooters, the average speed is assumed to be 16.1 km/h (10 mph) when traveling in Atlanta. Due to the similar structure of this problem to ODMTS-DA, a modified version P-Path is derived to seek the optimal solution of SCTS-DA, and treated as a benchmark for this setting.

The computational results are reported in Table 6. Here, the two arc-based algorithms can find a solution that have zero False Rejection Rate and False Adoption Rate, which are the desired properties of an optimal solution. Although Algorithms *ρ-GRAD* and *η-GRRE* find the solution with higher objective, they terminate in a short duration. On the other hand, Algorithm *ρ-GAGR* is ineffective in producing advantageous results for this case study. In Figure 5, it can be observed that the designs obtained by the three trip-based heuristic algorithms (Figure 5a) are identical, as do the designs discovered by the two arc-based algorithms (Figure 5b). Furthermore, it is worth mentioning that the designs resulting from the trip-based algorithms are quite similar to the design derived exclusively from core trips.

7. Conclusion

This study aims at capturing the latent demand for the transit agency such that the transit services can concurrently provide high-quality services to existing riders and extend the services to current underserved areas. The decision-making processes of both the transit agency and the latent riders form a game-like interaction. To model this dynamic interaction, this paper first introduces a bilevel optimization framework called Transit Network Design with Adoptions (TN-DA). The

Method	step size	trip rule	# outer itr.	# total itr.	time (hours)	design obj	% Gap	$\%R_{false}$	$\%A_{false}$
Modified P-PATH	-	-	-	-	4.58	1051031.55	0.00	0.00	0.00
Alg. *ρ-GRAD*	500	-	1	12	0.11	1055265.30	0.40	11.61	0.00
Alg. *η-GRRE*	500	-	1	12	0.10	1055265.30	0.40	11.61	0.00
Alg. *ρ-GAGR*	500	-	12	79	0.70	1055265.30	0.40	11.61	0.00
Alg. *arc-S1*	-	(i)	1	15	0.55	1051031.55	0.00	0.00	0.00
Alg. *arc-S2*	-	(ii), (i)	1	15	0.57	1051031.55	0.00	0.00	0.00

Table 6 Performance comparison between all algorithms on the Large-scale experiment.

framework essentially models a network design problem with an incorporated black-box-like mode choice model. Achieving success in solving the TN-DA can identify the equilibrium point between the agency and the latent riders. However, even with an exact solution algorithm, due to the bilevel and combinatorial nature of the TN-DA, practically solving it can lead to computational challenges, particularly for large-scale instances.

This paper then attempts to address this limitation and proposes five heuristic algorithms to efficiently approximate the optimal solution of the TN-DA problem. The fundamental idea of these iterative algorithms is to separate the previous bilevel framework into two individual but simpler components, i.e., a regular network design problem with no adoption awareness (the TN-DFD problem in Figure 2) and an evaluation procedure based on a mode choice function. These heuristic algorithms can be further categorized into two general classes based on their algorithmic designs: (1) trip-based algorithms and (2) arc-based algorithms. In general, at each iteration, the trip-based algorithms reconstruct the trip sets for the TN-DFD problem and the arc-based algorithms attempt to fix a certain amount of bus arcs. The advantages and limitations of each algorithm are discussed in this paper, and several theoretical properties are derived from the optimal solutions in order to provide more guidelines for the iterative algorithms under varied circumstances. In particular, two key properties, correct rejection (Property 1) and correct adoption (Property 2), are introduced. Then, the FALSE REJECTION RATE and the FALSE ADOPTION RATE are proposed as metrics to evaluate the ODMTS designs, providing additional insights on the performance of the heuristic algorithms when an optimal solution becomes unavailable for large-scale instances.

This paper further presents comprehensive case studies on two different transit systems to validate the heuristics. For the first transit system—On-Demand Multimodal Transit Systems (ODMTS), one case study is carried out with a normal-sized transit dataset collected by the transit agency in the Ann Arbor and Ypsilanti area in the state of Michigan. The results demonstrate that the heuristic algorithms can discover the optimal ODMTS design. Another case study is based on multi-sourced data and includes a very large numbers of stops and trips in Atlanta in the state of Georgia. The main goal of this case study is to show the advantages of the heuristic algorithms on large-scale instances. Compared to the exact algorithm, the heuristic algorithms achieve

better objective values in considerably less amount of time. The second transit system—Scooters-Connected Transit Systems (SCTS) are discussed in Section 6. These case studies serve as evidence that the TN-DA framework can be readily tailored to various transit systems and exhibit strong performance on large-scale scenarios while providing practical insights.

Acknowledgments

This work is supported by NSF Leap-HI Grant 1854684, NSF CMMI Grant 2334302, and T-SCORE Grant 69A3552047141 from the Department of Transportation.

References

Auad R, Dalmeijer K, Riley C, Santanam T, Trasatti A, Van Hentenryck P, Zhang H, 2021 *Resiliency of on-demand multimodal transit systems during a pandemic. Transportation Research Part C: Emerging Technologies* 133.

Basciftci B, Van Hentenryck P, 2020 *Bilevel optimization for on-demand multimodal transit systems.* Hebrard E, Musliu N, eds., *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, 52–68 (Springer International Publishing).

Basciftci B, Van Hentenryck P, 2023 *Capturing travel mode adoption in designing on-demand multimodal transit systems. Transportation Science* 57(2):351–375.

Borndörfer R, Grötschel M, Pfetsch ME, 2007 *A column-generation approach to line planning in public transport. Transportation Science* 41(1):123–132.

Canca D, De-Los-Santos A, Laporte G, Mesa JA, 2016 *A general rapid network design, line planning and fleet investment integrated model. Annals of Operations Research* 246(1):127–144.

Canca D, De-Los-Santos A, Laporte G, Mesa JA, 2017 *An adaptive neighborhood search metaheuristic for the integrated railway rapid transit network design and line planning problem. Computers & Operations Research* 78:1–14.

Canca D, De-Los-Santos A, Laporte G, Mesa JA, 2019 *Integrated railway rapid transit network design and line planning problem with maximum profit. Transportation Research Part E: Logistics and Transportation Review* 127:1–30.

Cats O, Gkiotsalitis K, Schöbel A, 2025 *50 years of operations research in public transport. EURO Journal on Transportation and Logistics* 100160.

Chatham Area Transit, 2024 *CAT Smart Grant 2024*. https://www.catchacat.org/smart/, URL https://www.catchacat.org/smart/.

Cipriani E, Gori S, Petrelli M, 2012 *Transit network design: A procedure and an application to a large urban area. Transportation Research Part C: Emerging Technologies* 20(1):3–14.

Dai W, Zhang J, Sun X, Wandelt S, 2019 *Hubbi: Iterative network design for incomplete hub location problems. Computers & Operations Research* 104:394–414.

Dalmeijer K, Van Hentenryck P, 2020 *Transfer-expanded graphs for on-demand multimodal transit systems. International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, 167–175 (Springer).

Dumas J, Aithnard F, Soumis F, 2009 *Improving the objective function of the fleet assignment problem. Transportation Research Part B: Methodological* 43(4):466–475.

Fan W, Machemehl RB, 2006 *Optimal transit route network design problem with variable transit demand: genetic algorithm approach. Journal of transportation engineering* 132(1):40–51.

Gattermann P, Schiewe A, Schöbel A, 2016 *An iterative approach for integrated planning in public transportation. 9th Triennial Symposium on Transportation Analysis.*

Guan H, Basciftci B, Van Hentenryck P, 2024 *Path-based formulations for the design of on-demand multimodal transit systems with adoption awareness. INFORMS Journal on Computing* 36(6):1359–1756.

Hagenauer J, Helbich M, 2017 *A comparative study of machine learning classifiers for modeling travel mode choice. Expert Systems with Applications* 78:273–282.

Johnson DB, 1975 *Finding all the elementary circuits of a directed graph. SIAM Journal on Computing* 4(1):77–84.

Kleinert T, Labbé M, Ljubić I, Schmidt M, 2021 *A survey on mixed-integer programming techniques in bilevel optimization. EURO Journal on Computational Optimization* 9:100007.

Klier MJ, Haase K, 2008 *Line optimization in public transport systems. Operations Research Proceedings 2007*, 473–478 (Springer).

Klier MJ, Haase K, 2015 *Urban public transit network optimization with flexible demand. Or Spectrum* 37(1):195–215.

Kroon L, Maróti G, Nielsen L, 2015 *Rescheduling of railway rolling stock with dynamic passenger flows. Transportation Science* 49(2):165–184.

Laporte G, Mesa J, Ortega F, Perea F, 2011 *Planning rapid transit networks. Socio-Economic Planning Sciences* 45(3):95 – 104.

Laporte G, Mesa J, Ortega F, Sevillano I, 2005 *Maximizing trip coverage in the location of a single rapid transit alignment. Annals of Operations Research* 136:49–63.

Lee YJ, Vuchic VR, 2005 *Transit network design with variable demand. Journal of Transportation Engineering* 131(1):1–10.

Liu D, Vansteenwegen P, Lu G, Peng Q, 2019 *An iterative approach for profit-oriented railway line planning. Linköping University Electronic Press* 806–825.

Liu J, Chen W, Yang J, Xiong H, Chen C, 2022 *Iterative prediction-and-optimization for e-logistics distribution network design. INFORMS Journal on Computing* 34(2):769–789.

Lodi A, Malaguti E, Stier-Moses NE, Bonino T, 2016 *Design and control of public-service contracts and an application to public transportation systems. Management Science* 62(4):1165–1187.

Lu J, Trasatti A, Guan H, Dalmeijer K, Van Hentenryck P, 2024 *The impact of congestion and dedicated lanes on on-demand multimodal transit systems. Travel behaviour and society* 36:100772.

Mahéo A, Kilby P, Van Hentenryck P, 2019 *Benders decomposition for the design of a hub and shuttle public transit system. Transportation Science* 53(1):77–88.

McQueen M, Clifton KJ, 2022 *Assessing the perception of e-scooters as a practical and equitable first-mile/last-mile solution. Transportation research part A: policy and practice* 165:395–418.

Polinder GJ, Cacchiani V, Schmidt M, Huisman D, 2022 *An iterative heuristic for passenger-centric train timetabling with integrated adaption times. Computers & Operations Research* 142:105740.

Schöbel A, 2012 *Line planning in public transportation: models and methods. OR spectrum* 34(3):491–510.

Schöbel A, 2017 *An eigenmodel for iterative line planning, timetabling and vehicle scheduling in public transportation. Transportation Research Part C: Emerging Technologies* 74:348–365.

Siebert M, Goerigk M, 2013 *An experimental comparison of periodic timetabling models. Computers & Operations Research* 40(10):2251–2259.

Stiglic M, Agatz N, Savelsbergh M, Gradisar M, 2018 *Enhancing urban mobility: Integrating ride-sharing and public transit. Computers & Operations Research* 90:12–21.

Sutlive C, Tomlinson CS, 2019 *2019 annual report and audit: Planning for the forward motion of metro atlanta.* https://atltransit.ga.gov/wp-content/uploads/2019/12/ATL_ARA-Final-11-25-19.pdf.

Tawfik C, Gendron B, Limbourg S, 2022 *An iterative two-stage heuristic algorithm for a bilevel service network design and pricing model. European Journal of Operational Research* 300(2):512–526.

Tawfik C, Limbourg S, 2019 *A bilevel model for network design and pricing based on a level-of-service assessment. Transportation Science* 53(6):1609–1626.

Van Hentenryck P, Riley C, Trasatti A, Guan H, Santanam T, Huertas JA, Dalmeijer K, Watkins K, Drake J, Baskin S, 2023 *Marta reach: Piloting an on-demand multimodal transit system in atlanta. arXiv preprint arXiv:2308.02681* .

Xie C, Lu J, Parkany E, 2003 *Work travel mode choice modeling with data mining: decision trees and neural networks. Transportation Research Record* 1854(1):50–61.

Yan X, Zhao X, Broaddus A, Johnson J, Srinivasan S, 2023 *Evaluating shared e-scooters' potential to enhance public transit and reduce driving. Transportation research part D: transport and environment* 117:103640.

Yu B, Yang Z, Yao J, 2010 *Genetic algorithm for bus frequency optimization. Journal of Transportation Engineering* 136(6):576–583.

Zhao X, Yan X, Yu A, Van Hentenryck P, 2020 *Prediction and behavioral analysis of travel mode choice: A comparison of machine learning and logit models. Travel behaviour and society* 20:22–35.

Appendix A: Additional Information For Heuristic Algorithms

This section completes Section 4 with algorithms. The corresponding propositions are represented in this section again for enhancing readability.

A.1. Greedy Adoption Algorithm with Greedy Rejection Subproblem

This section presents Algorithm *ρ-GAGR*.

Algorithm 5 ρ-GAGR

Set $k = 0$, $C = \emptyset$, $\overline{T}^0 = T_{core} \cup C$, and $o^* = \infty$.
Require: step-size parameters ρ and η
while True **do**
  Set $\mathbf{z}^k = \eta$-*GRRE* (Algorithm η-*GRRE* starts from $\overline{T}^k$ instead of T_{core})
  if $\textsc{Eval}(\mathbf{z}^k) < o^*$ **then**
    Set $o^* = \textsc{Eval}(\mathbf{z}^k)$.
    Set $\mathbf{z}_{min} = \mathbf{z}^k$.
  end if
  Carry out steps 5—21 of *ρ-GRAD*
end while
return $\mathbf{z}_{min}$

A.2. Arc-based Extended Greedy Algorithm

This section presents Algorithm *arc-S2*.

Appendix B: Additional Information on ODMTS-DA

This section completes Section 5 by presenting the full formulations for ODMTS-DA and ODMTS-DFD and their modeling details with omitted proofs.

B.1. ODMTS-DA Bilevel Formulation

This section presents the bilevel formulation of ODMTS-DA, where extensive discussions on this formulation can be found in Basciftci and Van Hentenryck (2023), Guan, Basciftci, and Van Hentenryck (2024). Under the context of ODMTS-DA, the set N represents stops for shuttles to pick-up and drop-off passengers. The set $H \subseteq N$ corresponds to the hubs that can be stops for high-frequency buses or rail services. In order to maintain conciseness and avoid complexity, this section focuses on the formalization of buses, as establishing new rail lines in practice can be challenging. However, existing rail lines or similar services (e.g., Bus rapid transit) can be preserved as *backbone arcs* and integrated into ODMTS-DA by utilizing a set $Z_{backbone}$. To have a weighted objective considering both convenience and monetary cost aspects of the ODMTS design, a parameter $\theta \in [0,1]$ is defined such that the trip duration associated with convenience is multiplied by θ and monetary spending is multiplied by $1-\theta$. In particular, the transit agency invests a weighted cost $\beta_{hl} = (1-\theta) b_{dist}\, n\, d^{bus}_{hl}$ if the leg between hubs $h, l \in H$ is open, where b_{dist} is the cost of using a bus per kilometer and n is the number of buses operating in each open leg within the planning horizon. The weighted investment β_{hl} can alternatively be modeled as $\beta_{hl} = (1-\theta) b_{time}\, n\, t^{bus}_{hl}$ by the transit agency, where b_{time} stands for the monetary cost of operating a bus per hour. The parameter n is determined in advance prior to

Algorithm 6 arc-S2

Set $k=0$, $\mathbf{z}_{fixed}=\vec{0}$, $\overline{T}^0=T_{core}$, $B^0=\infty$, $p=1$
Require: EXPAND_i functions for increasing the set of users considered $(i=1,2)$
while $p\leq 2$ **do**
  while True **do**
    Step 4—22 from Algorithm 4.
    Set $T^k_{next}=\text{EXPAND}_p(T_{latent},\mathbf{z}^k)$.
    Step 24—25 from Algorithm 4.
  end while
  Set $p=p+1$.
end while
return $\mathbf{z}^k$

solving the problem, and a common approach is to compute it based on the headway of buses. For backbone arcs in $Z_{backbone}$, β_{hl} can be simply set to 0 since they are preserved from the existing transit system. The bilevel optimization model uses binary variable z_{hl} whose value is 1 if the leg between the hubs $h,l\in H$ is open.

For each trip $r\in T$, the model uses binary variables x^r_{hl} and y^r_{ij} to determine whether this trip utilizes the leg between the hubs $h,l\in H$, and the shuttle leg between the stops $i,j\in N$, respectively. By combining the selected x^r_{hl} and y^r_{ij} variables, a path π^r is provided to riders of trip r from their origin to destination. To this end, this section interchangeably uses π^r and vectors $(\mathbf{x^r},\mathbf{y^r})$ to denote the paths of the riders. The transit agency incurs a weighted inconvenience cost $\tau^r_{hl}=\theta(t^{bus}_{hl}+t^{wait}_{hl})$ for the parts of the trips traveled by buses, where t^{wait}_{hl} is the average waiting time of a passenger for the bus between hubs h and l. A similar equation can be applied if there exists backbone connection between hubs h and l, e.g., $\tau^r_{hl}=\theta(t^{rail}_{hl}+t^{wait}_{hl})$. For the legs of the trips traveled by on-demand shuttles, the transit agency incurs a weighted cost and inconvenience cost $\gamma^r_{ij}=(1-\theta)\omega\, d^{shuttle}_{ij}+\theta t^{shuttle}_{ij}$ for each trip $r\in T$, where ω is the monetary cost of using a shuttle per kilometer. Core riders are assumed to utilize the ODMTS once the system is built, following the general assumptions in TN-DA. This is justified by the case study in Basciftci and Van Hentenryck (2023) that demonstrated the improvement in convenience of the suggested paths compared to the existing transit system. On the other hand, the latent riders can decide whether to adopt the ODMTS, based on the choice function $\mathcal{C}^r(\mathbf{x^r},\mathbf{y^r})$. The transit agency charges riders a fixed fare ϕ to use the ODMTS, irrespective of their paths. Thus, the fixed value $\varphi=(1-\theta)\phi$ becomes an additional revenue to the transit agency for riders adopting the ODMTS. Additionally, as compared to the general formulation of TN-DA, the objective function of ODMTS-DA can exclude the revenue from core riders since it remains as a constant (see the second term in the objective from both TN-DA and ODMTS-DA).

Figure 6 presents the bilevel formulation of ODMTS-DA. The leader problem (10) designs the network between the hubs for the ODMTS, whereas the follower problem (11) obtains paths for each trip $r\in T$ by considering the legs of the network design and the on-demand shuttles to serve the first and last miles of the trips. The objective of the leader problem (10a) minimizes the sum of (i) the investment cost of opening bus legs, (ii) cost of the trips of the existing riders, and (iii) the net-cost of the riders with choice that are adopting the ODMTS. Constraint (10b) ensures weak connectivity between the hubs with the same number

$$\min_{z_{hl},\delta^r} \quad \sum_{h,l\in H} \beta_{hl} z_{hl} + \sum_{r\in T_{core}} p^r(g^r - \varphi^r) + \sum_{r\in T_{latent}} p^r \delta^r (g^r - \varphi^r) \tag{10a}$$

$$\text{s.t.} \quad \sum_{l\in H} z_{hl} = \sum_{l\in H} z_{lh} \quad \forall h \in H \tag{10b}$$

$$z_{hl} = 1 \quad \forall (h,l) \in Z_{backbone} \tag{10c}$$

$$\delta^r = \mathcal{C}^r(\mathbf{x^r}, \mathbf{y^r}) \quad \forall r \in T_{latent} \tag{10d}$$

$$z_{hl} \in \{0,1\} \quad \forall h,l \in H \tag{10e}$$

$$\delta^r \in \{0,1\} \quad \forall r \in T_{latent} \tag{10f}$$

where $(\mathbf{x^r}, \mathbf{y^r}, g^r)$ are a solution to the optimization problem

$$\underset{x^r_{hl}, y^r_{ij}, g^r, t^r}{\text{lex-min}} \quad \langle g^r, t^r \rangle \tag{11a}$$

$$\text{s.t.} \quad g^r = \sum_{h,l\in H} \tau^r_{hl} x^r_{hl} + \sum_{i,j\in N} \gamma^r_{ij} y^r_{ij} \tag{11b}$$

$$t^r = \sum_{h,l\in H} (t^{bus/rail}_{hl} + t^{wait}_{hl}) x^r_{hl} + \sum_{i,j\in N} t^{shuttle}_{ij} y^r_{ij} \tag{11c}$$

$$\sum_{\substack{h\in H \\ \text{if } i\in H}} (x^r_{ih} - x^r_{hi}) + \sum_{i,j\in N} (y^r_{ij} - y^r_{ji}) = \begin{cases} 1 & \text{if } i = or^r \\ -1 & \text{if } i = de^r \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N \tag{11d}$$

$$x^r_{hl} \le z_{hl} \quad \forall h,l \in H \tag{11e}$$

$$x^r_{hl} \in \{0,1\} \quad \forall h,l \in H, y^r_{ij} \in \{0,1\} \quad \forall i,j \in N. \tag{11f}$$

Figure 6 The Bilevel Optimization Model for ODMTS Design with Adoptions (ODMTS-DA**).**

of incoming and outgoing open legs. Constraint (10c) assures all preserved arcs are open. Constraint (10d) represents the mode choice of the riders in T_{latent} depending on the ODMTS paths. Since the key of this entire section is to demonstrate the bilevel framework and heuristic algorithms on a particular transit network design problem, without loss of generality, this section assumes that ODMTS-DA only considers a unique bus frequency for potential bus arcs z_{hl}. However, incorporating multiple frequencies can be achieved by introducing a frequency variable for each potential bus arc and adjusting Constraint (10b) to ensure flow balance when considering multiple frequencies.

For each trip r, the objective of the follower problem (11) minimizes the lexicographic objective function $\langle g^r, t^r \rangle$, where g^r corresponds to the cost of trip r and t^r breaks potential ties between the solutions with the same value of g^r by returning the most convenient path for the rider of trip r. Once again, when solving the shortest-path problem based on the cost metric g^r, the transit agency may not always provide the most convenient path to the riders, as their decisions are influenced by spending considerations as well. Since sub-objective g^r contains both monetary spending and duration of trip r, g^r includes sub-objective t^r multiplied

$$\min_{z_{hl}} \quad \sum_{h,l \in H} \beta_{hl} z_{hl} + \sum_{r \in \hat{T}} p^r (g^r - \varphi^r) \tag{12a}$$

$$\text{s.t.} \quad (10b), (10c), (10e) \tag{12b}$$

where $(\mathbf{x^r}, \mathbf{y^r}, g^r)$ are a solution to the optimization problem (11).

Figure 7 The Optimization Model for ODMTS Design with Fixed Demand $\hat{T}$: ODMTS-DFD**($\hat{T}$).**

by θ. As presented in Basciftci and Van Hentenryck (2023), for any network design, a lexicographic minimizer of the follower problem (11) exists, and the lexicographic minimum is unique. Thus, once a network design $\mathbf{z}$ is given, the path of the riders in trip r and their choices, i.e., x^r_{hl}, y^r_{ij}, and δ^r, can all be deduced. This aligns with the requirements in TN-DA, as it ensures that the subproblem offers a unique solution. Constraint (11d) guarantees flow conservation used in trip r for origin, destination and each of the intermediate points utilized. Constraint (11e) ensures that the path only takes into account the open legs between the hubs. As pointed out by Basciftci and Van Hentenryck (2023), the follower problem has a totally unimodular constraint matrix; it can be solved as a linear program with an objective of the form $M\ g^r + t^r$ for a suitably large value of M. This can be achieved by replacing τ^r_{hl} and γ^r_{ij} values with $\hat{\tau}^r_{hl} := M\tau^r_{hl} + t'_{hl} + t^{wait}_{hl}$ and $\hat{\gamma}^r_{ij} := M\gamma^r_{ij} + t_{ij}$, respectively, where the value of big-M needs to be chosen according to the studied instances. To adjust the objective function value in (10a), β_{hl} and φ values can be further replaced with $\hat{\beta}_{hl} := M\beta_{hl}$ and $\hat{\varphi} := M\varphi$, respectively. Furthermore, to limit the maximum number of transfers in each suggested path, a constraint can be added to the follower problem. As the addition of this constraint violates the totally unimodularity property, the underlying network can be remodelled through a transfer expanded graph that captures the transfer limit by design (Dalmeijer and Van Hentenryck 2020). Note that the term *direct shuttle trips* indicates ODMTS trips that are directly served by on-demand shuttles without the involvement of buses or rails.

B.2. ODMTS-DFD

Figure 7 presents the formulation of ODMTS-DFD.

B.3. Proof for Proposition 5

The term UB^r for each trip r is defined in the following Proposition, which is proved by Basciftci and Van Hentenryck (2023).

Proposition 7: Consider an ODMTS design $\mathbf{z}^1$, and any ODMTS design $\mathbf{z}^2 \geq \mathbf{z}^1$. Let the ODMTS travel time for trip r under $\mathbf{z}^1$ and $\mathbf{z}^2$ be denoted as t^1 and t^2. Then t^2 is bounded by an upper bound UB^r defined as follows:

$$\text{UB}^r = \begin{cases} t^1 + \frac{(1-\theta)}{\theta}\omega \left(d^{shuttle}_{or^r m} + d^{shuttle}_{nde^r} - \min_{h,l \in H} \{ d^{shuttle}_{or^r h} + d^{shuttle}_{lde^r} \} \right), \\ \text{if trip } r \text{ is assigned to a multimodal path under design } \mathbf{z}^1 \text{ such that} \\ m \text{ and } n \text{ are the first and last hub in the path, respectively.} \\ \\ \max \left\{ t^1, t^1 + \frac{(1-\theta)}{\theta}\omega \left(d^{shuttle}_{or^r de^r} - \min_{h,l \in H} \{ d^{shuttle}_{or^r h} + d^{shuttle}_{lde^r} \} \right) \right\}, \\ \text{if trip } r \text{ is assigned to a direct shuttle path under design } \mathbf{z}^1 \end{cases}$$

Proposition 5: The ODMTS design found by Algorithm *arc-S1* with expansion rule (c) satisfies the Correct Adoption property (Property 2), i.e., $\%A_{false} = 0$.

Proof: Consider the series of designs $\mathbf{z}^0, \mathbf{z}^1 ..., \mathbf{z}^k$ found by Algorithm *arc-S1* using expansion rule (c); then there are $\mathbf{z}^0 \leq \mathbf{z}^1 \leq ... \leq \mathbf{z}^k$ by construction. Moreover, $\mathbf{z}^0$ satisfies the Correct Adoption property since there are no latent trips included in $\hat{T} = \overline{T}^0$. For an arbitrary design $\mathbf{z}^i$ where $0 < i \leq k$, it follows that all latent trips included in $\hat{T} = \overline{T}^i = T_{core} \cup \left(\bigcup_{j=0,\ldots,i-1} {T_{latent}}^j_{next} \right)$ adopt $\mathbf{z}^i$ by definition of expansion rule (c) and because trips in ${T_{latent}}^j_{next}$ for any $j = 0, 1, ..., i-1$ adopt $\mathbf{z}^i$, which is greater than any $\mathbf{z}^j$. Thus, all of these designs satisfy the Correct Adoption property. □

Appendix C: Additional Information on SCTS-DA

This section completes Section 6 by providing the full formulations for SCTS-DA and SCTS-DFD and their modeling details.

$$\min_{z_{hl}, \delta^r} \quad \sum_{h,l \in H} \beta_{hl} z_{hl} + \sum_{r \in T_{core}} p^r (g^r - \varphi^r) + \sum_{r \in T_{latent}} p^r \delta^r (g^r - \varphi^r) \tag{13a}$$

$$\text{s.t.} \quad (10b) - (10f)$$

where $(\mathbf{x^r}, \mathbf{y^r}, g^r)$ are a solution to the optimization problem

$$\text{lex-min}_{x^r_{hl}, y^r_{ij}, g^r, t^r, s^r} \quad \langle t^r, s^r \rangle \tag{14a}$$

$$\text{s.t.} \quad g^r = \sum_{h,l \in H} \theta (t^{bus/rail}_{hl} + t^{wait}_{hl}) x^r_{hl} + \sum_{i,j \in N} \theta t^{scooter}_{ij} y^r_{ij} \tag{14b}$$

$$t^r = \sum_{h,l \in H} (t^{bus/rail}_{hl} + t^{wait}_{hl}) x^r_{hl} + \sum_{i,j \in N} t^{scooter}_{ij} y^r_{ij} \tag{14c}$$

$$s^r = \mathbb{1}\left(\sum_{h,l \in H} x^r_{hl} = 0 \right) \tag{14d}$$

$$s^r \in \{0, 1\} \tag{14e}$$

$$(11d) - (11f)$$

Figure 8 The Bilevel Formulation for Scooters-Connected Transit Systems Designs with Adoptions (SCTS-DA**).**

The set N represents the locations where scooters are available, while the set $H \subseteq N$ represents the hubs used for connecting rails and high-frequency buses. Similar to the ODMTS-DA problem in Section 5, the focus in SCTS-DA is on designing the high-frequency bus network, while the rail system is preserved in the set $Z_{backbone}$. The variables x^r_{hl} represent whether a trip r utilizes the hub-to-hub arc (h, l) in its path, while y^r_{ij} indicate the utilization of the scooter leg (i, j) in the journey.

The bilevel formulation of the SCTS-DA problem is presented in Figure 8. Its leader problem is identical to ODMTS-DA because the agency also aims to design a transit network. Unlike in ODMTS-DA, where

$$\min_{z_{hl}} \quad \sum_{h,l \in H} \beta_{hl} z_{hl} + \sum_{r \in \hat{T}} p^r (g^r - \varphi^r) \tag{15a}$$

$$\text{s.t.} \quad (10b), (10c), (10e) \tag{15b}$$

where $(\mathbf{x^r}, \mathbf{y^r}, g^r)$ are a solution to the optimization problem (14).

Figure 9 The Optimization Model for scooter Design with Fixed Demand $\hat{T}$: SCTS-DFD($\hat{T}$).

the transit agency provides paths to riders considering both operating cost and travel duration, in SCTS-DA, riders are assumed to choose their own paths based on the fastest option. Therefore, for each trip r, the subproblem is formulated as a shortest path problem based on travel duration t^r. Binary variable s^r indicates if riders exclusively use scooters and do not utilize the bus or rail network. To unify the objective in the objective function (13a), the cost term g^r is defined as $t^r\theta$. The lexicographic objective (14a) prioritizes paths that use only scooters in case of ties in travel time.

Following the construction of TN-DFD, the SCTS-DFD problem is presented in Figure 9, and the Evaluation function for design $\mathbf{z}$ is:

$$\text{EVAL-SCTS}(\mathbf{z}) = \sum_{h,l \in H} \beta_{hl} z_{hl} + \sum_{r \in T_{core}} p^r g^r_\pi + \sum_{r \in T_{latent}} p^r \delta^r (g^r_\pi - \varphi^r) \tag{16}$$

Appendix D: Additional Information for ODMTS-DA Instance in Ypsilanti, MI

This section provides the supplementary information and additional computational studies over the medium-sized ODMTS-DA instance (see Section 5.5) in Ypsilanti, Michigan, USA.

D.1. Experimental Settings

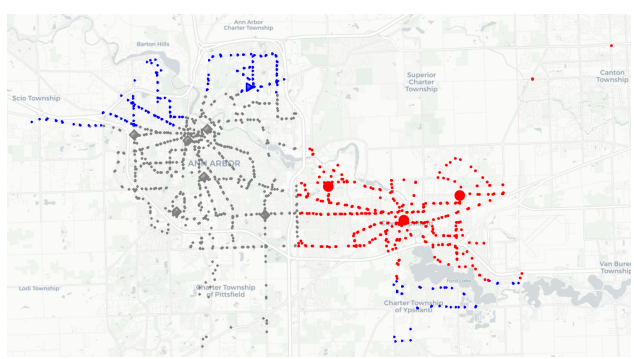


Figure 10 The 1267 bus stops used in this case study. The enlarged 10 stops are the designated hubs. The low-income, middle-income, and high-income stops are represented by red circles, grey squares, and blue triangles respectively.

This case study is based on the AAATA transit system that operates over 1,267 stops. In order to design an ODMTS, 10 stops are designated as ODMTS hubs, and the other stops are only accessible to on-demand shuttles. There are 1,503 trips for a total of 5,792 riders, and each trip is represented by an origin stop and

a destination stop. The time horizon for this case study is between 6 pm and 10 pm. Therefore, this dataset primarily consists of commuting trips from work locations to home. The mode preference of riders depends on their income levels, i.e, a rider with higher income level has lower tolerance on the amount of ODMTS travel time. Thus, the whole 1,503 trips were first divided into three different categories: low-income, middle-income, and high-income, based on the destination of the trips as it is associated with the home location of the riders. Figure 10 visualizes the locations of the stops and their corresponding income levels. If a trip is assigned to a certain income-level class, all riders belonging to this trip are then assumed to have the same income-level. Following this procedure, there are 476 low-income, 819 middle-income, and 208 high-income trips with 1,754, 3,316, and 722 riders respectively. Once the classification is carried out, an α value associated to the model Time-Based (Equation (3)) is assigned to each class. It must be noted that all trips in the low-income class are treated as members of the core trips set T_{core}; hence, no α value is required for them. For the middle-income and the high-income classes, 2.0 and 1.5 are employed as the α values, reflecting an assumption that the medium-income class has higher tolerance for travel time on transit.

In order to evaluate the performances of all five iterative algorithms with different configurations, two values for each of the two parameters—rider multiplier and core trips percentage are selected, resulting in overall four experiments as described in Table 7. More specifically, the rider multiplier parameter multiplies the number of riders for each trip. The core trips percentage parameter is utilized to divide the dataset into core trips and latent trips by varied partitions. Note that Experiment 4 is already presented in Section 5.5.

The on-demand shuttle price is fixed at \$1 per kilometer and each shuttle is assumed to only serve one passenger. Shuttles between bus hubs are not allowed in this case study. For buses, the operating fee is \$3.38 per kilometer. Four buses per hour is the unique bus frequency considered here; hence, each opened bus leg has an average waiting time (t_{hl}^{wait} in (11c)) of 7.5 minutes. This assumes that passengers arrive at the bus stop randomly rather than attempting to synchronize with the bus schedule. For passengers, there is no limit on the number of transfers. A constant ticket price that is consistent with the existing AAATA system is applied. Thus, a full ODMTS service charges \$2.5 for each individual regardless of the travel length. Inconvenience is measured in seconds, and the inconvenience and cost parameter θ is fixed at 0.001.

The iterative algorithms and the exact algorithm are programmed with Python 3.7. The exact algorithm is set to terminate if it reaches a 0.1% optimality gap; otherwise, it reports the upper bound as the "optimal" solution after a 6 hour time limit. The difference between the lower bound and upper bound identified by the exact algorithm when it terminates is denoted as '% Opt. Gap'. The computational experiments are carried out using Gurobi 9.5 without multi-threading on subproblems. P-Path can obtain optimal solution in short amount of time, and it was programmed with Python 3.9 with the support of Gurobi 9.5. For the trip-based iterative algorithms, the ρ value for Algorithm *ρ-GRAD* and Algorithm *ρ-GAGR*, and the η value for Algorithms *η-GRRE* and *ρ-GAGR* are both set to 10. For arc-based iterative algorithms, two experimental runs are designed to test Algorithm *arc-S1* with trip expansion rules (**a**) and (**c**). Algorithm *arc-S2* employs rules (**b**) and (**a**) and rules (**c**) and (**a**) for the two stages. When solving the ODMTS-DFD problem at each iteration, all iterative algorithms employ a stand-alone solving tool developed by Dalmeijer and Van Hentenryck (2020) that leverages Benders decomposition algorithm. This tool is based on the solver CPLEX 12.9 and allows multi-threading when solving subproblems in ODMTS-DFD. However, in this case study, multi-threading techniques are disabled in order to fairly compare the algorithmic performances.

	ridership	low income core trips			medium income core trips			high income core trips		
		% trips	# trips	# riders	% trips	# trips	# riders	% trips	# trips	# riders
Exp.1	regular	100%	476	1754	75%	614	2842	50%	104	434
Exp.2	regular	100%	476	1754	50%	409	2262	25%	52	258
Exp.3	doubled	100%	476	3508	75%	614	5684	50%	104	868
Exp.4	doubled	100%	476	3508	50%	409	4524	25%	52	516

Table 7 The experimental setups for the four experiments. For doubled ridership, the number of riders for each O-D pair is multiplied by 2. The core trips percentages for each income level are [100%, 75%, 50%] and [100%, 50%, 25%] for low, medium, and high income trips, respectively. Experiment 4 is already presented in Section 5.5.

D.2. Additional Results for Experiments 1, 2, and 3

Method	Step Size	Trip Rule	# Outer Itr.	# Total Itr.	Run Time (min)	Design Obj.	% Opt Gap.	% Gap	$\%R_{false}$	$\%A_{false}$
Exact Alg.	-	-	-	-	364.10	19012.91	0.65	-	0.00	0.00
P-Path	-	-	-	-	2.19	19012.91	-	0.00	0.00	0.00
Alg. *ρ-GRAD*	10	-	1	17	1.66	19012.91	-	0.00	0.00	0.00
Alg. *η-GRRE*	10	-	1	17	1.67	19012.91	-	0.00	0.00	0.00
Alg. *ρ-GAGR*	10	-	17	154	14.52	19012.91	-	0.00	0.00	0.00
Alg. *arc-S1*	-	(a)	1	5	0.37	19012.91	-	0.00	0.00	29.13
Alg. *arc-S1*	-	(c)	1	5	0.38	19012.91	-	0.00	19.42	0.00
Alg. *arc-S2*	-	(b), (a)	1	6	0.43	19012.91	-	0.00	0.00	2.27
Alg. *arc-S2*	-	(c), (a)	1	6	0.43	19012.91	-	0.00	0.00	0.00

Table 8 Performance comparison on Experiment 1. The exact algorithm reports optimality gap after 6 hours. The heuristics report the gap between themselves and the optimal solution (P-PATH).

Method	Step Size	Trip Rule	# Outer Itr.	# Total Itr.	Run Time (min)	Design Obj.	% Opt Gap.	% Gap	$\%R_{false}$	$\%A_{false}$
Exact Alg.	-	-	-	-	367.52	16635.73	2.48	-	0.00	0.00
P-Path.	-	-	-	-	7.80	16635.73	-	0.00	0.00	0.00
Alg. *ρ-GRAD*	10	-	1	33	4.28	16635.73	-	0.00	0.00	5.12
Alg. *η-GRRE*	10	-	1	30	4.18	16635.73	-	0.00	12.90	5.12
Alg. *ρ-GAGR*	10	-	30	480	54.78	16635.73	-	0.00	0.00	5.30
Alg. *arc-S1*	-	(a)	1	5	0.59	16635.73	-	0.00	0.00	26.50
Alg. *arc-S1*	-	(c)	1	4	0.41	16983.87	-	2.09	33.57	0.00
Alg. *arc-S2*	-	(b), (a)	1	6	0.76	16635.73	-	0.00	0.00	14.66
Alg. *arc-S2*	-	(c), (a)	1	6	0.61	16635.73	-	0.00	0.00	13.60

Table 9 Performance comparison on Experiment 2. The exact algorithm reports optimality gap after 6 hours. The heuristics report the gap between themselves and the optimal solution (P-PATH).

Table 8 presents an overview of the algorithmic performances of all algorithms on Experiment 1. All heuristic algorithms achieve the optimal solution in a short duration. Table 9 compares the computational results obtained from all algorithms on Experiment 2. Similar to Experiment 4 (see Section 5.5), the ratio of core trips to the set of all trips is significantly lower in this experiment; hence, more iterations are conducted in most of the algorithms. Furthermore, $\%R_{false}$ and $\%A_{false}$ are not necessarily equal to 0 although the optimal design $\mathbf{z}^*$ is found by the iterative algorithms. This result can be explained by the statement summarized in Remark 2. Under this setting, most of the iterative algorithms successfully obtain the optimal design

found by the exact algorithm and P-PATH, except Algorithm *arc-S1* with rule (**c**). This exception could be attributed to the design of rule (**c**) which strictly selects trips for the following iterations, resulting in an earlier termination. The algorithmic performances of Experiment 3 are summarized in Table 10. Under this setting, none of the proposed algorithms successfully discover the optimal solution. The trip set $\hat{T}^*$ constructed by the optimal solution $\mathbf{z}^*$ includes a few trips that reject all ODMTS designs explored by the heuristic algorithms. For trip-based algorithms, this can be a result of the expansion rules as all of them only select candidates from adopting trips at certain iterations, and it is difficult to justify the addition of currently rejected trips. For arc-based algorithms, these results are likely to be related to their greedy searching rules. The arc-based algorithms only fix the cycle that provides the greatest design objective decrements, leading to local, but not global, optimality. Although the proposed algorithms may not discover the optimal solution, these observations may support the hypothesis that the proposed iterative algorithms can find a high-quality solution in short duration. Indeed, the reported designs only have a 0.03% optimality gap. Furthermore, some iterative algorithms achieve zero FALSE REJECTION RATE and zero FALSE ADOPTION RATE while not obtaining the optimal solution, as discussed in Remark 1.

Method	Step Size	Trip Rule	# Outer Itr.	# Total Itr.	Run Time (min)	Design Obj.	% Opt Gap.	% Gap	$\%R_{false}$	$\%A_{false}$
Exact Alg.	-	-	-	-	363.22	34732.09	0.99	-	0.00	0.00
P-Path	-	-	-	-	2.62	34732.09	-	0.00	0.00	0.00
Alg. *ρ-GRAD*	10	-	1	15	2.12	34743.29	-	0.03	0.00	0.00
Alg. *η-GRRE*	10	-	1	15	2.15	34743.29	-	0.03	0.00	0.00
Alg. *ρ-GAGR*	10	-	15	121	17.10	34743.29	-	0.03	0.00	0.00
Alg. *arc-S1*	-	(**a**)	1	8	0.91	34743.29	-	0.03	0.00	42.39
Alg. *arc-S1*	-	(**c**)	1	8	0.80	34743.29	-	0.03	12.94	0.00
Alg. *arc-S2*	-	(**b**), (**a**)	1	9	0.86	34743.29	-	0.03	0.00	5.50
Alg. *arc-S2*	-	(**c**), (**a**)	1	9	0.86	34743.29	-	0.03	0.00	0.00

Table 10 Performance comparison on Experiment 3. The exact algorithm reports optimality gap after 6 hours. The heuristics report the gap between themselves and the optimal solution (P-PATH).

Appendix E: Data and Experimental Settings for Atlanta Case Studies

This section introduces the procedures to construct the dataset used in the large-scale case studies, both ODMTS-DA and SCTS-DA. This dataset focuses on the 6 am–10 am morning peak of a regular weekday. ODMTS or SCTS operate across 2,426 locations. For hubs, 58 and 66 are selected in different case studies: ODMTS-DA large instance and SCTS-DA have 58 hubs, while ODMTS-DA extra-large instance has 66 hubs. Among the hubs, 38 of them are adopted from rail stations. There are 15,478 core trips, 36,283 latent trips with 55,871 existing riders, and 54,902 riders with choices. As morning rush hours are considered, congestion is considered in computing the travel times of on-demand shuttles in case studies.

E.1. Core Demand

The core demand of this case study is based on two realistic datasets provided by the transit agency *Metropolitan Atlanta Rapid Transit Authority* (MARTA). One dataset includes the transit pass transaction records collected by the Automated Fare Collection (AFC) system, and the other dataset is constructed using Automated Passenger Counter (APC) system in the MARTA buses. These two datasets are combined to generate

a set of trips that represent a regular weekday in March 2018. For practical reasons, all MARTA stops within a 1500 feet (457.2 m) radius are first clustered into 1,585 *cluster stops*. The clustering process initially guarantees the reservation of all rail stations as cluster stops. Subsequently, it progressively incorporates individual bus stops into clusters or designates them as new clusters. In cases where a bus stop is eligible for inclusion in multiple clusters, it is allocated to the closest cluster regarding to the haversine distance. The core trip set T_{core} is then constructed after narrowing the time window to the morning peak and applying trip chaining techniques, which are identical to the data employed by Auad et al. (2021) for the baseline scenarios.

E.2. Latent Demand

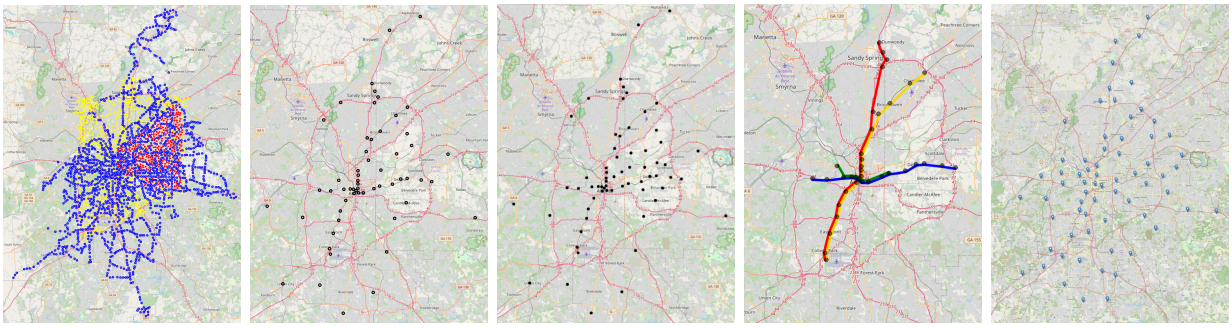
This section introduces the source of the latent demand for this case study. The ABM-ARC dataset is simulated by the Atlanta Regional Commission (ARC) based on the ARC Regional Household Travel Survey conducted in 2011 following a rigorous activity-based model (ABM) (http://abmfiles.atlantaregional.com/ Last Visited Date: September 30, 2025). In this dataset, there are more than 4.5 million simulated agents who are characterized by demographic data corresponding to themselves and their households. It also contains more than 13 million individual tours that are completed in the Atlanta metropolitan area in a typical weekday. All tours are described by detailed travel information including origin-destination (O-D) locations, travel period, travel distance, travel mode, and purposes. In particular, the data with horizon year 2020 is used for this case study to align with the 2018 core demand.

The latent demand for this case study consists of all individual tours that satisfy the following five conditions: (i) drive alone tours, (ii) commuting tours, (iii) the tours start between 6 am and 10 am, (iv) tour agents whose households are located inside the *East Atlanta* region—a large and generally underserved residential area, and (v) tour agents whose work locations located inside or around the I-285 interstate highway loop or inside the city of Sandy Springs—both are MARTA's primary service regions. The first two conditions regulate tour mode and tour purposes such that commuters who drive alone form the potential market of an ODMTS service. Particularly, the second condition regulates tour purpose because a commuter's household and work location are not regularly changed; hence, these tours are more representative of a regular weekday. The third condition further refines the dataset by narrowing down the time window to the morning peak. The last two conditions control the origin and destination locations. It is worth pointing out that this study keeps the tour destination rather than the first trip-leg destination in order to generalize the home-to-work demand for people residing in East Atlanta. Once the valid tours are isolated from the ABM-ARC dataset, their tour origins and tour destinations are then extracted to form the latent demand for this case study.

All the geographical information in the ABM-ARC dataset are at Traffic Analysis Zone (TAZ) level; however, instead of directly utilizing the TAZ centroids to represent the exact locations, origins and destinations are further sampled to lower geographical levels to construct a more representative dataset. In particular, a TAZ can be decomposed into multiple Census Blocks that belong to or intersect with it. A Census Block consists of one or several adjacent neighbourhoods and records the number of residents. Since all selected trips originate from the commuters' home locations, each origin is sampled from its corresponding TAZ to a Census Block by the population distribution of the census blocks. Similarly, the destinations are sampled

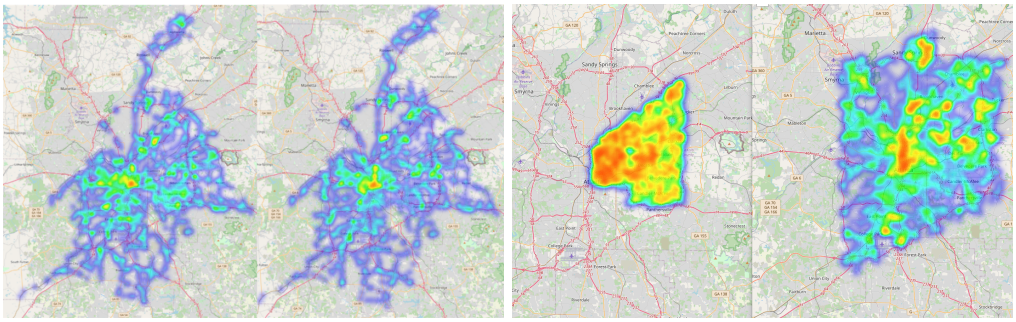
from TAZ level to Points of Interest (POI) level. Points of interest are specific locations such as restaurants, schools, and factories. POIs are associated with a TAZ and their area size is also recorded; thus, in this study, each destination is sampled from its corresponding TAZ to a POI by the area distribution of the POIs.

E.3. Available Locations, Transit Hubs, O-D Pairs, and Congestion Modeling



(a) All Locations (set N) (b) 58 Transit Hubs (set H) (c) 66 Transit Hubs (set H) (d) MARTA Rail Lines (e) Congestion Query Point

Figure 11 **The 2,426 locations are shown in (a). The blue circles, red triangles, and yellow triangles denote the locations of the MARTA cluster stops, stops adopted from Census Block centers, and stops adopted from Points of Interest centers, respectively. The ODMTS hubs and the MARTA rail system are visualized in (b), (c), and (d) respectively. All 38 MARTA rail stations are reserved as ODMTS hubs. The 68 congestion query points are shown in (e).**



(a) O-Ds in Core Trip Set T_{core} (b) O-Ds in Latent Trip Set T_{latent}

Figure 12 **The heat-maps in (a) visualize the origin and destination stops of the core trip set T_{core}. All origins and destinations are MARTA cluster stops. The heat-maps in (b) visualize the origin and destination stops of the Latent Trip set T_{latent}. All origins are from the East Atlanta area, and all destinations are either inside or around the I-285 interstate or inside the city of Sandy Springs. The origins and destinations are first sampled to lower geographical level, then mapped to their nearest ODMTS stops.**

The 2,426 stops (set N) are allocated on the basis of three sources: (i) the 1,585 MARTA cluster stops, (ii) the centers of Census Blocks in East Atlanta, (iii) the centers of POIs inside or around the I-285 interstate highway loop or inside the city of Sandy Springs. Specifically, all 1,585 MARTA cluster stops are first reserved as the ODMTS stops. To improve the granularity of stops such that the maximum haversine distance between

a stop and its nearest stop is 457.2 m (1500 feet), 247 Census Block centers and 594 POI centers are then appointed as the ODMTS stops. All accessible locations are illustrated in Figure 11a.

Following the allocation of all stops, for each trip in the latent demand, its origin and destination are mapped to their nearest location with regards to haversine distance. At this stage, the Latent Trip set T_{latent} is constructed. It should be recalled that the trips in the core trip set T_{core} originate from and arrive at MARTA cluster stops, which are already reserved in set N. The origins and destinations of T_{core} and T_{latent} are illustrated as heat-maps in Figure 12a and Figure 12b, respectively.

As shown in Figure 11b and 11c, 58 and 66 stops are designated as transit hubs, used for different case studies. Among these hubs, 38 of them are rail stations and the rail system is illustrated in 11d. The other 28 stops are appointed as hubs following these four rules: (1) the stops are on major roads, (2) there are a considerable amount of riders who depart from or arrive at this stop, (3) the haversine distance between two hubs is at least 1 km, and (4) a higher density of hubs should be allocated in the East Atlanta region.

To estimate congested shuttle travel time during the morning peak, the most convenient method would be to query the Google Maps Directions API (https://developers.google.com/maps/documentation/directions/overview Last Visited Date: September 30, 2025). However, due to the high cost associated with this approach, this study follows the methodology proposed by Lu et al. (2024) using query points as illustrated in Figure 11e. The 68 query points include all rail stations and have a significant overlap with the transit hubs. To estimate the congested travel time between two stops, i and j, Equation (17) is utilized. In this equation, q_i and q_j correspond to the nearest query points to stops i and j in terms of haversine distance, respectively. If stops i and j are mapped to the same query point, then q_i and its nearest query point are selected. The congested travel time between any two query points is obtained by querying the Google Map Directions API, while the free flow travel time and travel distance between any two stops are queried from GraphHopper (https://www.graphhopper.com/ Last Visited Date: September 30, 2025), which is constructed based on OpenStreetMap data. Here, the same travel time and distance are applied to both buses and shuttles. Since buses run only on the main corridors (typically major roads or multi-lane highways), the difference in travel time between the two modes is relatively small, unlike on smaller local roads.

$$t_{i,j}^{shuttle} = t_{i,j}^{shuttle,free} \times \frac{t_{q_i,q_j}^{shuttle,congested}}{t_{q_i,q_j}^{shuttle,free}} \tag{17}$$

E.4. Experimental Setting for ODMTS-DA in Atlanta

This section presents the experimental settings for the two ODMTS-DA case studies in Atlanta. The MARTA rail system, which is the rail system of the Atlanta Metropolitan area, is completely preserved in this experiment and the passenger travel time between any two rail stations is derived from GTFS data. Therefore, these decision variables form the set $Z_{backbone}$ in ODMTS-DA (Figure 6) and ODMTS-DFD (Figure 7). The unique bus frequency considered in this experiment is six buses per hour; hence, the average waiting time (t_{hl}^{wait} in (11b)) is five minutes. Unlike the Ypsilanti case study, shuttles are allowed to move between hubs in this case study. Additionally, bus arcs are restricted from overlapping rail arcs, and for each hub, a bus arc can only connect the hub to its three closest neighboring hubs in terms of travel time to

simplify the problem setting. For the extra-large instance, this number is increased to 10 to add complexity to the problem. For the adoption models TIME-BASED and TIME-TRANSFER-BASED, $\alpha = 1.5$ is applied to all latent trips. The transfer tolerance term l^r_{ub} in TIME-TRANSFER-BASED is set to 3 for all latent trips. A \$2.5 ticket is charged for each ODMTS rider and this constant price is identical to the current service provided by MARTA. The operating fees for on-demand shuttle and buses are fixed at \$0.62 per km (\$1 per mile) and \$72.15 per hour, respectively. The inconvenience and cost parameter θ is fixed at $7.25/(60+7.25)$. This case study employs minute as the unit of time in computation. These parameter values are directly adopted from an Atlanta-based ODMTS study presented in Auad et al. (2021). Congested travel time is only applied to the extra-large case study.

For P-PATH, it ran until it reaches the optimal solution. For the exact algorithm, the running time is limited at 24 hours. Due to the complexity of the master problem in the ODMTS-DA problem (Figure 6), for each iteration of the exact algorithm, the master problem terminates when reaching to 1% MIP gap or an 8 hours time limit. For the trip-based iterative algorithms, the iterative step sizes ρ and η are tested with three values—1000, 2000, and 3000. The step size that leads to the best design objective is reported together with the best discovered ODMTS design. In particular, Algorithm *ρ-GRAD* and Algorithm *η-GRRE* only demand the ρ value and the η value respectively, and η is set to equal to ρ for Algorithm *ρ-GAGR*. The combined trip-based iterative algorithm, i.e., Algorithm *ρ-GAGR*, is also terminated after 24 hours of running time. For the arc-based iterative algorithms, expansion rule (**a**) and (**c**) are separately utilized by Algorithm *arc-S1*. The two-stage Algorithm *arc-S2* employs two groups of rules—(i) rules (**b**) and (**a**) and (ii) rules (**c**) and (**a**). Furthermore, at the starting point, $\mathbf{z}_{fixed}$ represents the MARTA rail system. Lastly, the same set of computational tools and solvers are used as in the Ypsilanti case study (see Appendix D).

E.5. Experimental Setting for SCTS-DA in Atlanta

Since the transit agency will not receive any fare if riders only use the scooters, and the scooters are relatively slow, the theta value is chosen to be 0.6, to encourage riders use high-frequency services such as the rail. A \$2.5 constant fare is incurred to the transit agency for each rider.

The iterative step sizes ρ and η are tested as 500. The step size that leads to the best design objective is reported together with the best discovered SCTS design. In particular, Algorithm *ρ-GRAD* and Algorithm *η-GRRE* only demand the ρ value and the η value respectively, and η is set to equal to ρ for Algorithm *ρ-GAGR*.

For the arc-based heuristic algorithms, to expand the network design over iterations, the type of sub-network considered by the algorithms is the cycles formed by bus arcs. Expansion rule (i) and (ii) are separately utilized by Algorithm *arc-S1*. The two-stage Algorithm *arc-S2* employs a group of rules—rules (ii) and (i). Furthermore, at the starting point, $\mathbf{z}_{fixed}$ represents the MARTA rail system, which is already included in the set $Z_{backbone}$.

Appendix F: The Benefits of ODMTS-DA

In this section, the benefits of the ODMTS-DA are presented under realistic assumptions, providing a complementary perspective to the content covered in Section 5.7.

Algorithm	Trip Rule	Adoption		Only Use Shuttle		Use Bus / Rail		Profitable Adoption	
Exact Alg.	-	31421	57.23%	15920	50.67%	15501	49.33%	19919	63.39%
Alg. *ρ-GRAD*	-	30657	55.84%	15228	49.67%	15429	50.33%	20617	67.25%
Alg. *η-GRRE*	-	33080	60.25%	16481	49.82%	16599	50.18%	21338	64.50%
Alg. *ρ-GAGR*	-	30674	55.87%	15456	50.39%	15218	49.61%	20507	66.85%
Alg. *arc-S1*	(**a**)	30487	55.53%	15286	50.14%	15201	49.86%	20643	67.71%
Alg. *arc-S1*	(**c**)	39460	71.87%	20784	52.67%	18676	47.33%	20408	51.72%
Alg. *arc-S2*	(**b**), (**a**)	31043	56.54%	15235	49.08%	15808	50.92%	20990	67.62%
Alg. *arc-S2*	(**c**), (**a**)	30187	54.98%	14758	48.89%	15429	51.11%	20779	68.83%

Table 11 Adoption results of the reported ODMTS designs. To compute % only use shuttle, % Use Bus or Rail, and % Profitable adoptions, the denominator is # adopted riders.

F.1. Latent Demand Adoption

Table 11 summarizes the adoption results of the ODMTS designs. The first observation is that all ODMTS designs provide at least a 54.98% adoption rate, which is more than half of the riders with choice. Among these adopted riders, significant percentage of them only use the on-demand shuttle services. That is, the on-demand shuttles pick up the rider at the origin stop and drops her at the exact destination stop. There are several possible explanations for this result. First, there are many local trips inside the study area; thus, a direct on-demand shuttle is the best solution when considering both of cost and convenience. A rider always adopts the direct on-demand shuttle service since its travel time is assumed to be identical to the direct driving time. Secondly, the θ value might exaggerate the importance of cost. A rider is then assigned to an ODMTS path with relatively high inconvenience, resulting in rejection. Lastly, the choice function used in this study might be too strict for some riders. For example, a rider rejects a 12 minutes long ODMTS service when the direct driving time is only 5 minutes. Note that this challenge can be overcome by considering more flexible choice functions. Although the proportion of direct shuttle users is high, there is still a massive adoption value in terms of bus or rail users. Recalling that there are 55,871 existing riders in the system; thus, these newly adopted bus riders and rail riders can solely increase the number of public transit users by 27.2%–33.4%. These outcomes can be ascribed to the realistic assumptions on congested travel time. Given that major corridors experience congestion during peak hours, ODMTS emerges as a more favorable commuting alternative. It guides individuals toward high-frequency fixed routes, particularly the rail systems, which remain slightly affected or totally unaffected by congestion. Furthermore, the shuttle aspect of ODMTS trips is usually marginally impacted by congestion, primarily because they involve local trip legs.

Inspections on both Table 5 (see Section 5.7) and Table 11 show that the design with the smallest objective generally has a lower additional demand adoption rate. This might be explained by the nature of ODMTS-DFD (Figure 7) since it tries to provide an optimized network with the input trip set $\hat{T}$ regardless of the existence of the latent trips, resulting in a relatively inconvenient network but with lower objective. Once the ODMTS-DFD problem is solved, the evaluation procedure is carried out using the entire trip set T, and some of the latent trips reject the discovered ODMTS although they were involved in $\hat{T}$. Another interesting observation from Table 11 is at least 51.72% percent of the newly adopted riders can bring profit to the

Algorithm	Trip Rule	Car Dist. after ODMTS (km)	Car Dist. Reduced (km)	% Car Dist. Reduced	Bus Dist. (km)
Exact Alg.	-	434.43k	147.58k	25.36	8.50k
Alg. *ρ-GRAD*	-	428.30k	153.71k	26.41	8.02k
Alg. *η-GRRE*	-	423.62k	158.39k	27.21	7.35k
Alg. *ρ-GAGR*	-	429.58k	152.44k	26.19	7.85k
Alg. *arc-S1*	(**a**)	429.28k	152.73k	26.24	8.30k
Alg. *arc-S1*	(**c**)	414.93k	167.08k	28.71	5.46k
Alg. *arc-S2*	(**b**), (**a**)	421.89k	160.12k	27.51	7.62k
Alg. *arc-S2*	(**c**), (**a**)	427.89k	154.13k	26.48	8.58k

Table 12 Travel Distance of additional riders. The car distance before ODMTS aggregates the travel distance covered by the additional riders when they drive to commute. On the other hand, the car distance after ODMTS sums the travel distance on shuttles of the adopting riders and the travel distance of the non-adopting riders.

		Riders Adopting ODMTS		Core Riders		Riders not adopting ODMTS	
Algorithm	Trip Rule	ODMTS	Direct	ODMTS	Direct	ODMTS	Direct
Exact Alg.	-	18.86	15.64	24.00	20.94	38.96	20.17
Alg. *ρ-GRAD*	-	18.91	15.55	24.03	20.94	38.46	20.15
Alg. *η-GRRE*	-	18.62	15.44	23.99	20.94	39.50	20.83
Alg. *ρ-GAGR*	-	18.84	15.52	24.04	20.94	38.51	20.19
Alg. *arc-S1*	(**a**)	18.79	15.47	24.01	20.94	38.33	20.21
Alg. *arc-S1*	(**c**)	19.06	16.14	23.83	20.94	39.68	21.25
Alg. *arc-S2*	(**b**), (**a**)	19.16	15.73	23.87	20.94	37.99	19.98
Alg. *arc-S2*	(**c**), (**a**)	18.97	15.55	24.02	20.94	37.81	20.05

Table 13 Average Travel Time of Different Groups.

transit agency. It is worth pointing out that only the shuttle operating cost and the $2.5 revenue from the transit ticket are involved in the computation.

F.2. Travel Distance, Travel Time, and Operating Cost

This section further analyzes the ODMTS design using other performance measures. Table 12 summarizes the impact of the ODMTS on travelled distance by car. Note that the core riders are existing transit users; hence, they are not included in this table. What stands out of the table is the substantial reduction in travel distance after a group of self-driving commuters adopts the ODMTS, which is replaced by the bus or rail services. Table 12 also reports the travel distance of the ODMTS buses. As shown in Figure 11a, the existing bus system in Atlanta exhibits a dense cluster of stops, resulting in significant emissions and more congested traffic due to the regular pull in and pull out movements within traffic. This situation is further exacerbated by the utilization of diesel fuels. It's important to note that this illustration showcases clustered bus stops; the original stop network is even more extensive. In contrast, ODMTS only operate buses on a few key corridors that connect more widely spaced stops, offering a solution to alleviate congestion and emissions. The observations above together suggest that ODMTS can provide a positive impact on morning peak traffic by reducing congestion and pollution.

Table 13 presents the ODMTS travel time of three different groups: (i) riders with choice who adopt the system, (ii) core riders, and (iii) riders with choice who continue to drive alone. For each group, the travel times of two modes, ODMTS and direct driving, are compared. For the riders who adopt ODMTS, the average ODMTS travel time is only slightly greater than the average direct driving time. This can be explained by

Algorithm	Trip Rule	Revenue ($)	Bus Investment ($)	Shuttle Opt. Cost ($)	Rail Opt. Cost ($)	NP/rider ($)
Exact Alg.	-	218.23k	13.64k	136.51k	24.87k	0.49
Alg. *ρ-GRAD*	-	216.32k	12.75k	124.51k	24.87k	0.63
Alg. *η-GRRE*	-	222.38k	10.97k	133.58k	24.87k	0.60
Alg. *ρ-GAGR*	-	216.36k	12.26k	125.01k	24.87k	0.63
Alg. *arc-S1*	(**a**)	215.90k	12.98k	123.58k	24.87k	0.63
Alg. *arc-S1*	(**c**)	238.33k	7.59k	177.91k	24.87k	0.29
Alg. *arc-S2*	(**b**), (**a**)	217.28k	12.36k	127.21k	24.87k	0.61
Alg. *arc-S2*	(**c**), (**a**)	215.15k	13.90k	121.48k	24.87k	0.64

Table 14 Revenue and Cost from the Agency Perspective. Opt. and NP stand for Operating and Net Profit.

recalling the direct shuttle percentage reported in Table 11. On the other hand, the non-adopting commuters experience a larger time gap, where the average duration of the routes suggested by ODMTS is still less than two times of the average direct trip duration. For core riders, ODMTS can offer a level of service nearly comparable to driving a car, despite their lack of personal vehicles. It is also important to compare the travel time of the existing system for the core riders. As reported in Auad et al. (2021), when evaluating on the same data (core riders in this study) with the existing system, on average, the core riders spend 40 minutes on the transit system if they are assumed to only use the bus services, and this value drops to 26 minutes when rail system is available. Thus, from the core riders' perspective, all of the ODMTS systems found by the iterative algorithms outperform the existing transit system. The advantages of ODMTS stem from its effective utilization of the rail system, where many shuttle legs serve as feeders to hubs, thereby circumventing congestion issues. In contrast, driving alone often entails contending with traffic congestion, especially during the morning peak of a workday.

Finally, the cost analysis from the perspective of the transit agency is shown in Table 14. Ticket is the only source of revenue considered in this table, and *ticket revenue* stands for the revenue collected from the core riders and the adopting riders. The *bus investment* and *shuttle operating cost* are directly computed from the equations explained in Section 3. This study assumes that the ODMTS is operated by MARTA; thus, a fixed *rail operating cost* between 6 am—10 am is also considered in this table. The rail operating cost is estimated based on the MARTA rail schedule and the approximated $240 operating cost per revenue hour is reported in fiscal year 2018 (shown in Section 4.7.1 in Sutlive and Tomlinson (2019)). After including the 24.87k rail operating cost, the *net profit per rider* is computed. Table 14 shows that the ODMTS system is currently earning between $0.29-$0.64 when serving a customer, depending on the selected design. This is a result of the on-demand shuttles and the rapid bus services efficiently replacing the traditional buses, especially the buses that operate in regions with low ridership. In addition, the current potential riders only reside in one specific area, i.e., East Atlanta. Given that the rail operating cost is fixed, a case study with an augmented additional demand region might increase the number of adopting riders, resulting in a citywide profitable system.