

## Real-time re-optimization for generalized ridesharing feeder service with mixed scheduled and on-demand riders<sup>☆</sup>

Yanjie Yi<sup>a</sup>, Zheyong Bian<sup>a,b,\*</sup>, Bijun Wang<sup>c</sup>

<sup>a</sup> Department of Industrial Engineering, University of Houston, 4730 Calhoun Road, Houston, TX 77004, USA

<sup>b</sup> Department of Construction Management, University of Houston, 4730 Calhoun Road, Houston, TX 77004, USA

<sup>c</sup> Department of Industrial and Systems Engineering, University of Nevada, Reno, 1664 N. Virginia St, Reno, NV 89557, USA

### ARTICLE INFO

#### Keywords:

Ridesharing  
Shared mobility  
Dynamic optimization  
First mile

### ABSTRACT

Ridesharing revolutionizes feeder services by seamlessly connecting dispersed passengers to common destinations, such as train stations, bus terminals, airports, special event locations, and post-disaster shelters. This paper develops real-time re-optimization methodologies for generalized dynamic ridesharing feeder service operation. Such ridesharing feeder service typically encompasses a mix of scheduled and on-demand service, in which riders can schedule the request early in advance or send on-demand request for immediate service, and both types of riders are possible to share the ride. The developed methodologies enable the system to continuously re-optimize and re-adjust the vehicle routing plan that achieves the primary objective of maximizing the total number of served riders and the secondary objective of minimizing the total vehicle miles or hours traveled, simultaneously accounting for mixed types of riders' mobility requirements. This paper develops a re-optimization model implemented by a rolling horizon planning approach. An efficient heuristic algorithm, namely Large Neighborhood Search by Tabu Search algorithm (LNS-TS), is developed to solve large-scale problems. To validate the methodology, a simulation is developed to model rider activity, as well as the ridesharing process. This paper presents two case studies in Houston, Texas: a first-mile ridesharing service improving transit connectivity and a post-disaster cooling shelter access service addressing emergency transport under extreme heat. These cases represent two distinct operational contexts—everyday urban mobility and emergency disaster response—demonstrating the model's adaptability across diverse scenarios. The results generated by the re-optimization methodologies are compared with those from a periodic optimization approach, where matching and routing optimizations are conducted in isolation at different time intervals. The simulation results demonstrate that the routing plan obtained by the real-time re-optimization methodologies can serve more riders and save more vehicle miles or hours traveled compared with the periodic optimization methods. The proposed real-time, dynamic re-optimization approach for ridesharing feeder services not only demonstrates practical benefits but also holds promise for applications in autonomous vehicle systems.

### 1. Introduction

Ridesharing feeder services represent a transformative approach to urban and regional mobility, addressing critical gaps in transportation networks by connecting dispersed populations to centralized hubs (Rodrigue, 2020). Examples of ridesharing feeder services include first-mile ridesharing (Kumar and Khani, 2021), shared transportation to special event locations (Covelli Garrido et al., 2023), carpooling for commutes (Benita, 2020), and accessing shelters after disasters or extreme weather (Woo et al., 2021). From a transportation geography

perspective, these services mediate spatial mismatches between demand (e.g., suburban commuters, disaster-affected communities) and infrastructure (e.g., transit stations, cooling shelters), while navigating constraints like travel time, accessibility, and equity (Giuliano and Hanson, 2017). Traditional transportation geography research emphasizes the spatial organization of mobility (Wen et al., 2025; Yu et al., 2025), where inefficiencies arise from disparities between where people live, work, and access services. First-mile ridesharing, for instance, tackles the "last-mile problem" in reverse, bridging gaps in low-density suburban areas with poor transit coverage (Berrojo Romeyro Mascarenhas

<sup>☆</sup> This article is part of a Special issue entitled: 'New Mobility' published in Journal of Transport Geography.

\* Corresponding author.

E-mail addresses: [yyi7@cougarnet.uh.edu](mailto:yyi7@cougarnet.uh.edu) (Y. Yi), [zbian2@central.uh.edu](mailto:zbian2@central.uh.edu) (Z. Bian), [Wangbij@gmail.com](mailto:Wangbij@gmail.com) (B. Wang).

et al., 2023; Mohiuddin, 2021). Similarly, disaster-response ridesharing addresses spatial inequities in emergency access, where vulnerable populations often cluster in underserved neighborhoods (Hendricks and Van Zandt, 2021). Our work advances these geographic themes by optimizing dynamic routing that adapts to real-time spatial demand patterns.

Ridesharing feeder services operate under dynamic and mixed demand conditions by serving both scheduled and on-demand riders, combining the predictability of pre-booked trips with the flexibility to accommodate real-time ride requests. For instance, in a first-mile service connecting residential areas to a transit hub, some passengers may book their rides in advance, specifying a pickup time that aligns with their train or bus schedules. Meanwhile, others may request rides on-demand when unexpected circumstances arise, such as missing their usual ride or needing to catch an earlier train. While most research on ridesharing systems focuses exclusively on either scheduled or on-demand ride-sharing, limited attention has been given to systems that integrate both rider types, leaving a critical gap in understanding hybrid models. Moreover, ridesharing feeder services must address specific rider requirements, such as the latest acceptable arrival times and numbers of shared riders (Masoud et al., 2017; Bian and Liu, 2019; Bian et al., 2020). For instance, a traveler heading to a train station has a hard deadline determined by the train's departure time and must arrive punctually. Additionally, the traveler may prefer a limited number of co-riders due to factors like carrying luggage and extensive detour, which affect comfort. These preferences significantly influence the service's matching and routing strategies, requiring a balance between individual convenience and operational efficiency. This study addresses these requirements to enable ridesharing feeder services to deliver tailored, time-sensitive solutions while ensuring high vehicle utilization and passenger satisfaction.

This research advances the field by introducing a mixed scheduled and on-demand service model that incorporates mobility requirements. A real-time re-optimization methodology is proposed to address the dynamic nature of ridesharing feeder services. The primary objective of the developed methodology is to maximize the number of served riders, with the secondary goal of minimizing total vehicle miles traveled (VMT) or vehicle hours traveled (VHT), contributing to sustainable and efficient ridesharing operations. To handle large-scale problems effectively, the study develops a Large Neighborhood Search with Tabu Search (LNS-TS) algorithm featuring seven neighborhood structures and implements it within a rolling horizon planning framework.

To evaluate the practical applicability of the proposed approach, the developed methodologies are applied to two case studies to demonstrate their effectiveness. The first case study focuses on a first-mile ridesharing service designed to connect passengers to the Houston Amtrak Station in Texas, United States. This service addresses the critical challenge of improving access to major transit hubs, enabling more efficient and seamless integration with broader transportation networks. It represents an everyday mobility context, where both scheduled and on-demand riders must be served in a coordinated and efficient manner. The second case study examines a non-routine ridesharing service aimed at improving accessibility to cooling shelters in the aftermath of Hurricane Beryl, which struck Houston in July 2024. This hurricane caused widespread power outages across the Houston area, leaving a significant portion of the population in urgent need of transportation to designated cooling centers to escape the hazardous heat. The case study emphasizes the importance of ridesharing services in disaster response scenarios, particularly for vulnerable populations who may lack other means of transportation. In such situations, the demand from some vulnerable groups can be identified and scheduled in advance by government agencies or other organizations, while other requests may arise spontaneously. This comprehensive approach highlights the versatility of the methodologies in addressing both everyday mobility challenges and critical emergency needs. These scenarios also reflect key concerns in transportation geography, including spatial accessibility, disaster-

response mobility, and equitable access to essential services. Furthermore, the developed methodology is not only applicable to conventional ridesharing feeder services, but its ability to dynamically respond to rider demand and optimize operations in real time also makes it highly relevant for shared autonomous vehicle (SAV) systems.

The rest of the paper is structured as follows. Section 2 reviews previous work on ridesharing and the relevant optimization models. Section 3 introduces the methodologies developed in this study. Section 4 describes the simulation and case study results and Section 5 presents concluding remarks, respectively.

## 2. Literature review

Technology-enabled mobility services play a vital role in addressing contemporary urban transportation challenges. In recent years, a substantial and growing body of research has focused on the design of demand-responsive ridesharing feeder systems (e.g., Lee and Savelsbergh, 2017; Ma et al., 2019; Mendes et al., 2021; Wang, 2019; Wang et al., 2023). These services can be broadly categorized into scheduled and on-demand models, each presenting unique advantages and optimization challenges. While both aim to improve mobility and accessibility, their operational strategies and real-world applications differ considerably. The following subsections first present representative application scenarios to highlight the relevance of each model across various transportation contexts, followed by a detailed review of methodologies in handling optimization of scheduled and on-demand ridesharing feeder service.

### 2.1. Applications of ridesharing feeder services

Ridesharing feeder services have been implemented across a wide range of operational contexts, reflecting their flexibility in addressing both routine and emergency mobility needs. Prior studies have identified several representative application scenarios, including first-mile ridesharing service, shared transportation for special events, carpooling for commutes, and accessing shelters during disaster or extreme weather, each highlighting the practical value and diverse use cases of such systems.

**First-mile ridesharing service:** A first-mile ridesharing service is a transportation solution to help passengers travel from their starting location (such as their home, workplace, or another point of origin) to a transit hub, such as a train station, bus stop, or metro terminal. Encouraging the use of public transportation is increasingly regarded as an effective strategy to mitigate the negative externalities that may be generated by conventional passenger vehicles, such as traffic congestion, road fatalities, and transport-related air pollution. Technology-enabled ridesharing has attracted wide attention in the literature due to its possible impact on promoting nationwide public transit ridership (Bian et al., 2020; Kumar and Khani, 2021; Lee and Savelsbergh, 2017; Ma et al., 2019; Ma et al., 2019). This research could improve the real-time integration of ridesharing services with public transit systems.

**Shared transportation for special events:** Ridesharing services are increasingly utilized to improve accessibility to stadiums, concert venues, and conference centers by providing shared transportation options for attendees. These services consolidate riders from nearby areas into single vehicles, reducing the need for individual car trips and alleviating common challenges such as limited parking, traffic congestion, and logistical inefficiencies. The benefits of this ridesharing include cost savings for riders, mitigated parking pressure at event locations, reduced environmental impact through fewer vehicles on the road, and improved traffic flow near event locations (Robbins et al., 2007).

**Carpooling for commutes:** Carpooling for commutes involves individuals traveling to workplaces, sharing a single vehicle instead of driving separately. Carpooling can reduce transportation costs, minimize wear and tear on personal vehicles, decrease traffic congestion, lower greenhouse gas emissions, and reduce demand for parking spaces.

By optimizing vehicle use, carpooling not only saves money and time but also supports broader environmental and urban mobility goals (Li et al., 2018; Xia et al., 2019).

Accessing shelters during disaster or extreme weather: Ridesharing provides crucial access to shelters during disaster or extreme weather by offering a flexible and reliable transportation option for individuals who may lack personal vehicles or face mobility challenges. These services can dynamically match riders in need with shared rides to designated shelters, ensuring timely and efficient transport. Ridesharing enhances accessibility for vulnerable populations, such as the elderly, low-income individuals, or those with disabilities, who are disproportionately affected by disaster or extreme weather. By consolidating trips, ride-sharing reduces road congestion near shelters, allowing emergency services to operate more effectively. Moreover, it supports community resilience by fostering a collaborative response to extreme weather, ensuring that critical resources like shelters are accessible to those who need them most (Wong et al., 2021; Yoon et al., 2022).

## 2.2. Scheduled ridesharing feeder service

Scheduled ridesharing feeder services focus on optimizing transportation systems where all ride requests are known in advance. Static optimization methodologies dominate this domain, leveraging pre-defined data to achieve cost-efficiency, minimize travel distances, and maximize ride utilization. These methods often employ mathematical programming models, such as mixed-integer linear programming (MILP) and heuristic approaches.

Most research on scheduled ridesharing feeder services primarily focuses on optimizing first-mile connections to public transit hubs. For instance, Cao et al. (2021) proposed a genetic algorithm-based model to optimize ride-hailing routes in urban networks, targeting high-efficiency ride-hailing operations during peak hours and considering vehicle capacity, route rationality, and time windows. In addition, Bian and Liu (2019) proposed a mechanism to satisfy scheduled passengers' personalized requirements on different inconvenience attributes of the service. It helps determine the optimal vehicle-passenger matching and vehicle routing plan, and it also customizes a pricing scheme that can be adapted to different scenarios for the first-mile ridesharing problem. Lyu et al. (2019) built a new taxi-sharing system to provide flexible and personalized taxi-sharing services considering the nearby alternative pick-up/drop-off locations and scheduling a flexible sharing route to reduce passengers' walking distance. Chen et al. (2020) developed a mixed integer linear programming (MILP) model to optimize the first-mile ridesharing problem using autonomous vehicles, focusing on pre-scheduled passenger requests to enhance fleet management efficiency and reduce operational costs. Kumar and Khani (2021) proposed an algorithm for integrating pre-scheduled transit systems with peer-to-peer ridesharing to address the first-mile/last-mile problem, utilizing a schedule-based approach to optimize rider-driver matches based on fixed transit schedules and improving operational efficiency for suburban commuters facing limited transit coverage. Additionally, Dai et al. (2022) designed an autonomous taxi ride-sharing system optimized for commuting trips, integrating taxi schedules, depot locations, and fleet size into a unified model to minimize energy consumption and vehicle usage.

Methodologies for scheduled ridesharing optimization are often static, primarily designed to address pre-scheduled or known requests. These methods typically focus on maximizing efficiency based on fixed parameters, such as rider distribution and vehicle availability. They are not well-equipped to handle dynamically occurring requests that arise in real time. When new ride requests come in unexpectedly, these static models struggle to quickly adjust the routing and allocation of resources to accommodate these changes, leading to suboptimal solutions. This limitation highlights the need for more adaptive methodologies that can dynamically adjust to fluctuating demand and changing conditions in real time.

## 2.3. On-demand ridesharing feeder service

In contrast, on-demand ridesharing feeder services address real-time transportation needs, dynamically matching passengers with vehicles as requests arise. Real-time optimization methodologies underpin these services, relying on algorithms capable of rapid decision-making and adaptive routing. Many algorithms were reported in the literature to address optimization in different scenarios. Cordeau (2006) applied a mixed-integer programming formulation and a branch-and-cut algorithm to solve the dial-a-ride problem by randomly generating small to medium-scale instances. Lotfi et al. (2019) presented a modeling framework for the operations of on-demand mobility services while providing ridesharing and transfer services by a modified version of the column generation algorithm. Kang and Levin (2021) developed a max-pressure dispatch policy to maximize passenger throughput using a model predictive algorithm. Ma et al. (2019) adopted a queueing-theoretic framework for vehicle dispatching and developed the idle vehicle relocation algorithms for the coexistence of ridesharing and public transit services. Bian et al. (2020) adopted the mechanism design theory that accounts for individual-tailored mobility preferences to develop optimization models and proposed an efficient heuristic algorithm (i.e., solution pooling approach) for large-scale matching and routing problems. A possible extension of this work can be related to truly focusing on the integration of dynamic ridesharing and public transport systems. Gu and Liang (2024) proposed a centralized system for integrating ridesharing and public transit, featuring ILP-based optimization and hypergraph algorithms, including LP-rounding and weighted set packing, to prioritize reducing commuting times while improving vehicle occupancy. Chen et al. (2021) developed an improved Monte Carlo Tree Search (MCTS) approach for efficient dispatching in on-demand ride services, incorporating a tree policy and branch reduction to solve multi-period sequential dispatching problems. Nitter et al. (2024) developed a bi-objective mixed integer programming (MIP) model for the Static Ridesharing Routing Problem with Flexible Locations (SRRPFL), focusing on maximizing the number of passengers served while minimizing travel times. To address real-life scenarios, they introduced an Adaptive Large Neighborhood Search (ALNS) heuristic, further enhanced by a local search and a set partitioning method (Route Combination Problem) to improve route optimization.

While plenty of research has addressed on-demand ridesharing challenges, the majority of studies focus on optimizing ridesharing operations in isolation, often assuming that the sequence of passenger pick-ups and drop-offs remains fixed once established by the optimization algorithm. This simplification overlooks potential dynamic adjustments to the sequence that could further enhance efficiency and adaptability in real-time scenarios. Limited research has addressed real-time "re-optimization" for dynamic on-demand ridesharing service, which updates matches between waiting riders and idle or en-route vehicles. Existing research mainly relies on simple insertion heuristics (Ma et al., 2013; Xu et al., 2020; Haferkamp and Ehmke, 2020; Herbawi and Weber, 2012) and reinforcement learning (Al-Abbas et al., 2019; Haliem et al., 2021; Abdelmoumène et al., 2024; Meneses-Cime et al., 2022; Feng et al., 2022).

## 2.4. Knowledge gaps and contributions

Despite significant advancements, existing research primarily focuses on either scheduled or on-demand ridesharing services in isolation. Scheduled methods tend to be static, lacking the flexibility to accommodate dynamically occurring ride requests. On the other hand, on-demand systems often overlook opportunities to integrate scheduled and real-time on-demand requests, which limits their efficiency and scalability. In addition, integrating scheduled and on-demand requests presents major challenges in modeling and optimization, as it involves real-time processing of differences in request timing and location, coordination of diverse service constraints, and management of varying

rider statuses and vehicle limitations. Moreover, many existing re-planning methods rely on simple insertion heuristics or reinforcement learning models. These methods typically focus on incorporating newly arriving requests without adjusting previously assigned matches. As a result, they may fail to capture the full system dynamics in mixed-demand environments. As a result, the challenge of real-time re-optimization for mixed scheduled and on-demand ridesharing feeder systems remains inadequately addressed, with current methodologies proving insufficient to fully resolve these issues.

To address this challenge, this study develops innovative and efficient methodologies. The complex re-optimization for ridesharing feeder system is modeled as mixed integer programming, which can solve small-scale problems. An efficient heuristic algorithm is designed for large-scale scenarios. Both approaches integrate scheduled and on-demand ridesharing within a unified framework. Unlike prior methods that focus on partial updates, the proposed approach re-optimizes all active requests and vehicle routes based on real-time system status, aiming to maximize the number of served riders and minimize total vehicle miles or hours traveled, while meeting rider mobility requirements.

### 3. Real-time dynamic re-optimization methodology

#### 3.1. Description of the system

In ridesharing feeder services, riders may preschedule the ridesharing or send on-demand ridesharing requests to a common destination. In addition to the destination, riders can report two mobility requirements, the latest arrival time at the destination ( $\alpha_i^{AD}$ ) and the maximum acceptable number of shared riders ( $\alpha_i^{NR}$ ). The event consequence of the dynamic ridesharing feeder service can be described as follows:

- 1) Prescheduled request information customization: riders submit pre-scheduled request information in advance of the service, including departure location and mobility requirements.
- 2) Processing prescheduled requests: as the service is approaching, the system will consolidate requests with close latest arrival times at the destination and perform an initial optimization to match and route prescheduled requests. After an initial optimization, riders will be notified of the estimated pick-up time, and vehicle drivers will also be notified of the pick-up sequences.
- 3) On-demand request information customization: On-demand requests are defined as those that riders need to take the ridesharing service instantly, meaning that riders are ready to be picked up right away. Thus, these riders have a response time before which the rider must be notified of the estimated pickup time. Thus, before such latest response time, the system must finish the decision so that updated routing plan can be sent to the driver, and the riders can also receive the estimated pickup time promptly.
- 4) On-demand request processing and routing re-optimization: as the closest one of all pending riders' latest response times (e.g., 2 min after the request is submitted) is approaching, the system synthesizes all riders' service requests submitted within the time interval between the first rider's requesting time and his/her latest response time. The routes would be re-optimized with these new inputs. The service requests involving too critical requirements that cannot be satisfied in general (e.g., too close required arrival time) might be rejected.
- 5) Re-optimization in the following time slice: after the above time interval, the system receives new requests, and more vehicles become available. The vehicle routing and rider pick-up sequence will be re-optimized continuously.

#### 3.2. Rolling horizon planning approach

Rolling Horizon Planning (RHP), as a method extensively used in

dynamic ridesharing problems (Bian et al., 2022; Homsi et al., 2024; Lu et al., 2016) and other planning systems (Herding and Mönch, 2024; Madridano et al., 2021), is employed for the ridesharing feeder system in this study. Fig. 1 presents a simplified demonstration of RHP using first-mile ridesharing as an example. Four passenger requests, which are Passenger 1, Passenger 4, Passenger 5, and Passenger 7, were pre-scheduled in advance. As the service time is approaching, the system conducts an initial optimization to match prescheduled passengers and route available vehicles. The optimal routing plans of two vehicles after this initial optimization are "V1 → P1 → P4 → D" and "V2 → P7 → P5 → D", respectively (V: vehicle, P: passenger, D: drop-off location). After the initial optimization for prescheduled service is completed, the system will receive on-demand requests. Suppose that passenger request 2 is the first request after starting receiving on-demand requests and let the request time be  $t_{r2}$ . Request 2 has a latest response time ( $t_{e2}$ ), indicating that before  $t_{e2}$ , the system must obtain a re-optimized result and send the updated routing plan to vehicle drivers. Between Request 2's request time  $t_{r2}$  and the latest response time  $t_{e2}$ , the system receives three additional requests, passenger requests 3, 6, and 8. The system needs to account for all these four requests and conduct a round of re-optimization for re-matching and re-routing. As Fig. 1 shows, the updated re-matching and re-routing plan with four new passenger requests is represented by solid lines: "V1 → P2 → P1 → P4 → P3 → D" and "V2 → P6 → P7 → P8 → P5 → D". After the first-round re-optimization is conducted, the second round of re-optimization will continue after new requests are received. The RHP will never stop re-optimization until no new requests occur.

#### 3.3. Initial optimization modeling

This section builds a mathematical model for the initial optimization to match and route prescheduled riders. This paper accounts for two major scenarios regarding the supply of vehicles. One is that sufficient vehicles are serving all or most of the on-demand passengers that were commonly investigated in literature (Sun et al., 2018). In the other case, available vehicles are limited and cannot provide abundant services for all riders. As a result, the optimization model accounts for two levels of objectives. The primary objective is to maximize the number of served riders. The secondary objective is to minimize the total vehicle miles traveled (VMT) or vehicle hours traveled (VHT) given that the first objective is achieved. Formula (1) can achieve these two levels of objective function. The model is formulated by Formulas (1–12) whose notation is presented in Table 1 where  $M$  is a sufficiently large number and  $VMT(X)$  or  $VHT(X)$  is the total vehicle miles or hours traveled given the solution  $X$ .  $VMT(X)$  or  $VHT(X)$  can be formulated by Formula (2)

$$VMT(X) \text{ or } VHT(X) = \sum_{k \in V} \sum_{i \in P} \sum_{j \in P} x_{ijk} c_{ij} + \sum_{k \in V} \sum_{i \in P} y_{ki} d c_{ki} + \sum_{k \in V} \sum_{i \in P} z_{ki} ch_i \quad (2)$$

Subject to

$$y_{kj} + \sum_{i \in P} x_{ijk} = w_{kj} \quad \forall k \in V, j \in P \quad (3)$$

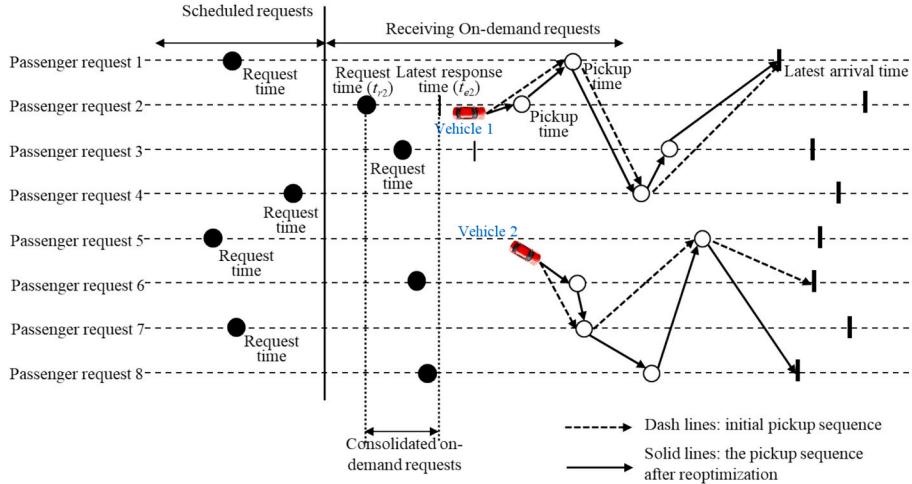
$$z_{ki} + \sum_{j \in P} x_{ijk} = w_{ki} \quad \forall k \in V, i \in P \quad (4)$$

$$\sum_{k \in V} w_{ki} \leq 1 \quad \forall i \in P \quad (5)$$

$$\sum_{i \in P} y_{ki} \leq 1 \quad \forall k \in V \quad (6)$$

$$\sum_{i \in P} w_{ki} n p_i \leq Q_k \quad \forall k \in V \quad (7)$$

$$IVT_i \geq IVT_j + t_{ij} - M \left( 1 - \sum_{k \in V} x_{ijk} \right) \quad \forall i, j \in P \quad (8)$$



**Fig. 1.** Implementation of the dynamic re-optimization algorithm with rolling horizon planning (RHP).

**Table 1**

Notation of the initial optimization for prescheduled service.

$$\max \sum_{k \in V} \sum_{i \in P} w_{ki} M - VMT(X) \text{ or } \max \sum_{k \in V} \sum_{i \in P} w_{ki} M - VHT(X) \quad (1)$$

Sets	
$P$	Set of $n$ prescheduled requests, $P = \{1, 2, \dots, n\}$ .
$V$	Set of available vehicles, $V = \{n + 1, n + 2, \dots, n + m\}$ .
Variables	
$x_{ijk}$	$\begin{cases} 1 & \text{Vehicle } k \text{ travels from Rider(s) } i \text{ to Rider(s) } j \\ 0 & \text{Otherwise} \end{cases} \quad k \in V, i, j \in P$
$y_{ki}$	$\begin{cases} 1 & \text{Rider(s) } i \text{ is the first to be picked up by Vehicle } k \\ 0 & \text{Otherwise} \end{cases} \quad k \in V, i \in P$
$z_{ki}$	$\begin{cases} 1 & \text{Rider(s) } i \text{ is the last to be picked up by Vehicle } k \text{ before heading to the destination} \\ 0 & \text{Otherwise} \end{cases} \quad k \in V, i \in P$
$w_{ki}$	$\begin{cases} 1 & \text{Rider(s) } i \text{ is picked up by Vehicle } k \\ 0 & \text{Otherwise} \end{cases} \quad k \in V, i \in P$
Parameters	
$dc_{ki}$	The travel distance or time from Vehicle $k$ to Rider(s) $i$ 's location ( $i \in P$ )
$ch_i$	The travel distance or travel time from Rider(s) $i$ 's location to the drop-off location ( $i \in P$ )
$Q_k$	Vehicle $k$ 's seat capacity.
$\alpha_i^{AD}$	Rider(s) $i$ 's latest arrival time at the drop-off location ( $i \in P$ ).
$\alpha_i^{NR}$	Rider(s) $i$ 's acceptable maximum number of co-riders ( $i \in P$ ).

$$IVT_i \geq t_{i0} \quad \forall i \in P \quad (9)$$

riders reach their drop-off points by the designated latest arrival time. Formula (12) specifies that certain variables take binary values.

$$\sum_{j \in P} w_{kj} np_j \leq w_{ki} (\alpha_i^{NR} + np_i) + (1 - w_{ki})M \quad \forall i \in P, k \in V \quad (10)$$

$$\sum_{i \in P} \sum_{j \in P} x_{ijk} t_{ij} + \sum_{i \in P} y_{ki} t_{ki} + \sum_{i \in P} z_{ki} t_{i0} \leq w_{kg} \alpha_g^{AD} + (1 - w_{kg})M \quad \forall g \in P, k \in V \quad (11)$$

$$x_{ijk}, y_{ki}, z_{ki}, w_{ki} = \{0, 1\} \quad \forall i, j \in P, k \in V \quad (12)$$

Formula (3) and (model the relationship between the decision variables  $x_{ijk}$  and  $w_{ki}$ . They ensure that if a rider's request is accepted and will be served by Vehicle  $k$ , then Vehicle  $k$  must go through this rider's location; otherwise, Vehicle  $k$  will not go through this location. Formula (5) models that a rider's request can be served at most once. Formula (6) ensures that within each round of re-optimization, a vehicle can be assigned at most once to prevent multiple dispatches within the same time slice. Formula (7) formulates the vehicle capacity constraint. Formula (8) aims to eliminate illegal subtours, while it can also model a rider's in-vehicle travel time along with Formula (9). Formula (10) ensures that the number of shared riders for all pre-scheduled riders remains within their acceptable limits. Formula (11) guarantees that

### 3.4. Re-optimization modeling based on RHP

Once the initial optimization for prescheduled service is completed and the system receives the on-demand requests, the system needs to conduct real-time re-optimization to update the matching plan and routing sequence. To address the differences between scheduled and on-demand requests in terms of time, location, response time, and service constraints, we simultaneously consider scheduled and on-demand riders, classify rider and vehicle statuses, and design various constraints to ensure feasibility and fairness of service. This section introduces the real-time re-optimization methodology based on mixed integer programming for dynamic ridesharing. Each re-optimization makes no difference, and thus as long as a versatile model is developed, it is applicable for all re-optimization rounds. We introduce how each re-optimization can be formulated. Riders can be classified into three categories based on their status in each time slice, within which rider requests will be consolidated for this round of re-optimization. Type-1 riders are defined as the ones whose requests are emerging, and sent after the last optimization and before the current re-optimization. Type-

2 riders pre-schedule the service or send the requests before the last re-optimization and the requests are already assigned to specific vehicles, but they have not been picked up yet. Type-3 riders are those who send requests before the last re-optimization and have already been picked up by a vehicle. Moreover, two categories of vehicles are also differentiated in this study, in which Type-1 vehicles are emerging and available or will be available soon when the current re-optimization is approaching, and Type-2 vehicles are already scheduled or dispatched based on previous optimization results. The real-time re-optimization model is formulated by Formulas (13–27). Table 2 presents the model notation.

$VMT(X)$  or  $VHT(X)$  can be formulated by Formula (14)

$$\begin{aligned} VMT(X) \text{ or } VHT(X) = & \sum_{k \in V^t} \sum_{i \in P^t} \sum_{j \in P^t} x_{ijk} c_{ij} + \sum_{k \in V^t} \sum_{i \in P^t} y_{ki} d_{ci} + \sum_{k \in V^t} \sum_{i \in P^t} z_{ki} ch_i \\ & + \sum_{k \in V^t} cv_k \left( 1 - \sum_{i \in P^t} z_{ki} \right) \end{aligned} \quad (14)$$

**Table 2**  
Notation of the real-time re-optimization model.

$$\max \sum_{k \in V^t} \sum_{i \in P^t} w_{ki} M - VMT(X) \text{ or } \max \sum_{k \in V^t} \sum_{i \in P^t} w_{ki} M - VHT(X) \quad (13)$$

Sets	
$P^t$	Set of rider requests for $t$ th re-optimization. $P^t$ involves Type-1 and Type-2 riders but does not include Type-3 riders.
$V^t$	Set of vehicles to for $t$ th re-optimization when any Type-1 rider's latest response time is approaching.
Variables	
$x_{ijk}^t$	$\begin{cases} 1 & \text{Vehicle } k \text{ travels from Rider(s) } i \text{ to Rider(s) } j \\ 0 & \text{Otherwise} \end{cases} \quad k \in V^t, i, j \in P^t$
$y_{ki}^t$	$\begin{cases} 1 & \text{Vehicle } k \text{ first picks up Rider(s) } i \\ 0 & \text{Otherwise} \end{cases} \quad k \in V^t, i \in P^t$
$z_{ki}^t$	$\begin{cases} 1 & \text{Rider(s) } i \text{ is the last to be picked up by Vehicle } k \\ 0 & \text{Otherwise} \end{cases} \quad k \in V^t, i \in P^t$
$w_{ki}^t$	$\begin{cases} 1 & \text{Rider(s) } i \text{ is picked up by Vehicle } k \\ 0 & \text{Otherwise} \end{cases} \quad k \in V^t, i \in P^t$
Parameters	
$dc_{ki}$	The distance or time from Vehicle $k$ to Rider(s) $i$ 's location ( $i \in P^t$ )
$ch_i$	The distance or time from Rider(s) $i$ 's location to the drop-off location ( $i \in P^t$ )
$cv_k$	If Vehicle $k$ ( $k \in V^t$ ) is a Type-2 vehicle, then $cv_k$ equals the distance or time from this vehicle's real-time location to the drop-off location. If it is Type-1 vehicle, then $cv_k = 0$ .
$RQ_k$	The remaining capacity defined by the number of seats of Vehicle $k$ ( $k \in V^t$ ) before the re-optimization.
	The time when Vehicle $k$ ( $k \in V^t$ ) will be available. Some vehicles are empty and have not been dispatched yet. These vehicles' available time is equal to the current time. Some vehicles are finishing their ongoing drop-off tasks.
$AT_k$	These vehicles' available time is the time when they will finish their tasks. Some vehicles are waiting at the requested location for the riders to show up. These vehicles' available time is the estimated time when the riders are picked up.
$DLS_k$	The latest arrival time of the Type-3 riders in Vehicle $k$ ( $k \in V^t$ ). For example, Vehicle $k$ already has three riders in it. If these three riders' latest arrival times are $d_{l1}, d_{l2}$ , and $d_{l3}$ , then, $DLS_k = \min(d_{l1}, d_{l2}, d_{l3})$ .
$RNR_k$	Number of additional riders that Vehicle $k$ ( $k \in V^t$ ) can pick up. This parameter is determined by the number of Type-3 riders in the vehicle and these riders' requirement on the number of shared riders. For example, if one rider (Type-3 rider) has been already picked up by Vehicle $k$ and another Type-2 rider must be picked up by the same vehicle but has not been picked up yet. The Type-3 rider's acceptable number of co-riders is 3. Then, the vehicle can pick up two more riders at most ( $RNR_k = 2$ ). Note that the Type-2 rider may also have requirements regarding the number of shared riders, but this requirement does not influence the parameter $RNR_k$ because we will use another constraint (Formula 22) to satisfy Type-2 riders' requirements. The number of riders (Type-3 riders) who have been already picked up by Vehicle $k$ ( $k \in V^t$ ).
$NPIV_k$	An indicator parameter: if Vehicle $k$ ( $k \in V^t$ ) is already dispatched to pick up rider(s) $i$ ( $i \in P^t$ ) based upon previous re-optimization results, then $\delta_{ki} = 1$ ; otherwise $\delta_{ki} = 0$ .
$\alpha_i^{AD}$	Rider(s) $i$ 's latest arrival time at the drop-off location ( $i \in P^t$ ).
$\alpha_i^{NR}$	Rider(s) $i$ 's acceptable maximum number of shared riders ( $i \in P^t$ ).

Subject to

$$y_{kj} + \sum_{i \in P^t} x_{ijk} = w_{kj} \quad \forall k \in V^t, j \in P^t \quad (15)$$

$$z_{ki} + \sum_{j \in P^t} x_{ijk} = w_{ki} \quad \forall k \in V^t, i \in P^t \quad (16)$$

$$\sum_{k \in V^t} w_{ki} \leq 1 \quad \forall i \in P^t \quad (17)$$

$$\sum_{i \in P^t} y_{ki} \leq 1 \quad \forall k \in V^t \quad (18)$$

$$\sum_{i \in P^t} w_{ki} np_i \leq RQ_k \quad \forall k \in V^t \quad (19)$$

$$IVT_i \geq IVT_j + t_{ij} - M \left( 1 - \sum_{k \in V^t} x_{ijk} \right) \quad \forall i, j \in P^t \quad (20)$$

$$IVT_i \geq t_{i0} \quad \forall i \in P^t \quad (21)$$

$$NPIV_k + \sum_{j \in P^t} w_{kj} np_j \leq w_{ki} (\alpha_i^{NR} + np_i) + (1 - w_{ki}) M \quad \forall i \in P^t, k \in V^t \quad (22)$$

$$\sum_{i \in P^t} w_{ki} np_i \leq RNR_k + \sum_{i \in P^t} \delta_{ki} np_i \quad \forall k \in V^t \quad (23)$$

$$\sum_{i \in P^t} \sum_{j \in P^t} x_{ijk} t_{ij} + \sum_{i \in P^t} y_{ki} t_{ki} + \sum_{i \in P^t} z_{ki} t_{i0} \leq w_{kg} (\alpha_g^{AD} - AT_k) + (1 - w_{kg}) M \quad \forall g \in P^t, k \in V^t \quad (24)$$

$$\sum_{i \in P^t} \sum_{j \in P^t} x_{ijk} t_{ij} + \sum_{i \in P^t} y_{ki} t_{ki} + \sum_{i \in P^t} z_{ki} t_{i0} \leq DLS_k - AT_k \quad \forall k \in V^t \quad (25)$$

$$w_{ki} \geq \delta_{ki} \quad \forall i \in P^t, k \in V^t \quad (26)$$

$$x_{ijk}, y_{ki}, z_{ki}, w_{ki} = \{0, 1\} \quad \forall i, j \in P^t, k \in V^t \quad (27)$$

Formula (15–18) achieves similar constraints with Formulas (3–6) in the initial optimization for pre-scheduled service. Formula (19) formulates the constraint that the number of riders who are assigned to the vehicle but have not been picked up yet cannot exceed the vehicle's remaining capacity. Formulas (20–21) are similar to Formulas (8–9). Formula (22) models the constraint that all Type-1 and Type-2 riders' acceptable number of shared riders cannot be exceeded. Formula (23) ensures that Type-3 riders' requirement on the number of shared riders must be satisfied. Formula (24) ensures that Type-1 and Type-2 riders must arrive at the drop-off location before their latest arrival time. Formula (25) models the constraints of Type-3 riders' latest arrival time. Formula (26) indicates that Type-2 riders will not transfer to other vehicles. Formula (27) represents that all decision variables are binary.

While the proposed model is mixed-integer programming (MIP), which can be exactly solved using the CPLEX solver for small-scale problems, the fundamental challenges in serving both request types stem from their distinct temporal characteristics and service requirements. On-demand requests typically have much tighter time constraints, often requiring response within 1–3 min compared to pre-scheduled trips. This creates complex trade-offs in vehicle capacity allocation, where we must balance guaranteed service for pre-scheduled riders against the immediate needs of on-demand users. Moreover, each new on-demand request may invalidate previously established constraints for pre-scheduled riders, such as their maximum acceptable number of co-riders or latest arrival times, necessitating comprehensive re-evaluation of the entire routing plan. This paper develops methodological framework to address these challenges.

Our approach distinguishes from prior sequential methods. Unlike

traditional methods that rely on incremental insertion of new requests into existing schedules or conduct periodic optimization, our approach performs full re-optimization of the entire routing plan at each decision point. This complete reassessment enables simultaneous consideration of all active pre-scheduled and on-demand requests, allowing the system to fundamentally restructure vehicle routes when beneficial rather than being constrained by previous scheduling decisions. The constraint propagation mechanism in the MIP model forms a critical component of this integration, where pre-scheduled riders' service parameters actively shape the feasibility space for on-demand assignments. Key parameters like  $RNR_k$  (remaining capacity for new riders) in Table 2 dynamically update based on both the number of pre-scheduled riders already in vehicles and their individual co-rider preferences ( $\alpha_i^{NR}$ ). This creates a bidirectional relationship where pre-scheduled riders influence on-demand assignment possibilities, while new on-demand requests may trigger re-evaluation of pre-scheduled routes. The system maintains this interdependence through continuous updates to vehicle states (e.g.,  $NPIV_k$ ,  $DLS_k$ ) and comprehensive constraint checking during each re-optimization cycle.

### 3.5. Solution approach to large-scale problems

As solving the re-optimization model is NP hard, involving a large number of riders and vehicles, the CPLEX solver may be difficult in finding a solution with the problem size increasing. Thus, this paper developed an efficient heuristic algorithm, namely Large Neighborhood Search by Tabu Search algorithm (LNS-TS) to solve large-scale problems. We developed seven neighborhood structures for the LNS-TS algorithm that specifically accounts for interplay between request types. The detailed neighborhood structures are described below and illustrated in Fig. 2. Note that all solutions generated by the seven neighborhood structures must be feasible and satisfy all constraints.

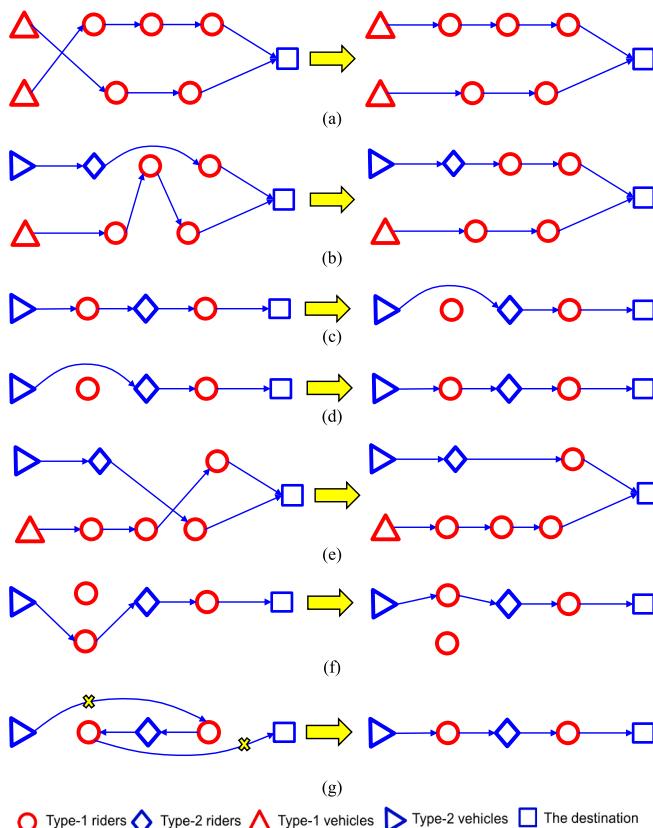


Fig. 2. Neighborhood Structures in the LNS-TS Algorithm.

- Neighborhood 1 (Fig. 2. a): Randomly select two Type-1 vehicles and then exchange riders to be picked up between the two vehicles.
- Neighborhood 2 (Fig. 2. b): Randomly select two vehicles. Randomly select one of the Type-1 riders who will be picked up by one vehicle based on the current solution and then assign this rider to the other selected vehicle.
- Neighborhood 3 (Fig. 2. c): Randomly select a vehicle that will pick up at least one Type-1 rider and then remove this rider from this vehicle's pick-up list, leaving the rider unserved.
- Neighborhood 4 (Fig. 2. d): Randomly select a rider not in any vehicles' pick-up list. Let this rider be picked up by a vehicle, which is also randomly selected.
- Neighborhood 5 (Fig. 2. e): Randomly select two vehicles. Randomly select two Type-1 riders from the two vehicles' pick-up lists, respectively. Then exchange the two riders between the two vehicles' pick-up lists.
- Neighborhood 6 (Fig. 2. f): Randomly select one vehicle. Randomly remove a Type-1 rider from this vehicle's pick-up list and randomly select a Type-1 rider who is not in any vehicle's pick-up list to replace the removed rider in this vehicle's pick-up list.
- Neighborhood 7 (Fig. 2. g): Within a vehicle's routing sequence, randomly cut the two links, reverse the routing sequence between the two links, and reconnect the links. This neighborhood is also known as the famous 2-opt neighborhood structure.

Table 3 presents the pseudocode of the LNS-TS algorithm incorporating the seven neighborhood structures.

The algorithm employs an adaptive parameter-setting mechanism that adjusts based on problem scale, characteristics, and the quality of currently explored solutions. Key parameters in the LNS-TS algorithm include the total number of iterations, the number of candidate neighborhood solutions per iteration, and the probability of selecting each neighborhood structure.

Both the total number of iterations ( $NI$ ) and the number of candidate solutions ( $CN$ ) increase with the number of rider requests ( $|P^t|$ ) and the number of vehicles ( $|V^t|$ ) increasing, as defined in Formulas (28) and (29), respectively.

The selection probabilities for each neighborhood structure must adhere to the following rules:

1. The sum of the probabilities for all neighborhood structures must equal 100 %.
2. When there are no rider requests assigned to vehicles in  $X_{current}$ , the probabilities of neighbors 1, 2, 3, 5, 6, and 7 should be all zero, and the probability of neighbor 4 should be 100 %.
3. When all rider requests are assigned to vehicles in  $X_{current}$ , the probabilities of neighbors 4 and 6 should be zero, while the probabilities of neighbors 1, 2, 3, 5, and 7 should be all greater than zero.

Based on the three rules, we develop Formula (30–33) to define the probability of adopting each neighborhood structure.

$$NI = \lceil 500(|P^t| \times |V^t|)^{\frac{1}{4}} \rceil \quad (28)$$

$$CN = \lceil 15(|P^t| \times |V^t|)^{0.3} \rceil \quad (29)$$

$$\begin{aligned} \Pr(\text{Neighbor } 1, \gamma) &= \Pr(\text{Neighbor } 2, \gamma) = \Pr(\text{Neighbor } 5, \gamma) \\ &= 3\gamma^2 / 50 + 6\gamma / 25 \end{aligned} \quad (30)$$

$$\Pr(\text{Neighbor } 3, \gamma) = \Pr(\text{Neighbor } 7, \gamma) = \gamma^2 / 100 + \gamma / 25 \quad (31)$$

$$\Pr(\text{Neighbor } 4, \gamma) = \gamma^2 - 2\gamma + 1 \quad (32)$$

$$\Pr(\text{Neighbor } 6, \gamma) = -6\gamma^2 / 5 + 6\gamma / 5 \quad (33)$$

**Table 3**

Pseudocode of large neighborhood search by tabu search algorithm (LNS-TS).

```

Input all parameters;
Randomly generate an initial solution  $X_0$ ;
Set  $X_c = X_0$ ; %  $X_c$  records the current solution.
Set  $f_c = f(X_c)$ ; % “ $f()$ ” gets the objective function value of the solution in the bracket.  $f_c$  is the
objective function value of  $X_c$  (Formula 1).
Set  $X_{best} = X_0$ ; %  $X_{best}$  is the best solution ever found.
Set  $f_{best} = f(X_{best})$ ;
Set  $TL = \{X_0, X_0, \dots, X_0\}$ ;
For  $ni = 1:NI$  %  $NI$  is the total number of iterations and  $ni$  is the index of current iteration
    Generate  $CN$   $X_{current}$ 's neighborhood solutions  $\mathbf{X} = \{X_1, X_2, \dots, X_{CN}\}$  based on the seven
    neighborhood structures; %  $CN$  is the number of generated neighborhood solutions
     $X_{opt} = \text{argmax}(f(\mathbf{X}))$ ;
    If  $f(X_{opt}) > f_{best}$ 
         $f_{best} = f(X_{opt})$ ;
         $X_{best} = X_{opt}$ ;
         $f_c = f(X_{opt})$ ;
         $X_c = X_{opt}$ ;
         $TL = [TL, X_{opt}]$ ;  $TL = TL \setminus TL(1)$ ; % Put  $X_{opt}$  into the tabu list as the last element and
        remove the first element from the tabu list.
    Else
        While  $X_{opt} \in TL$ 
             $\mathbf{X} = \mathbf{X} \setminus X_{opt}$ ;
             $X_{opt} = \text{argmax}(f(\mathbf{X}))$ ;
        End while
         $f_c = f(X_{opt})$ ;
         $X_c = X_{opt}$ ;
         $TL = [TL, X_{opt}]$ ;  $TL = TL \setminus TL(1)$ ; % Put  $X_{opt}$  into the tabu list as the last element and
        remove the first element from the tabu list.
    End if
End for
Output  $X_{best}$ ;

```

where  $\gamma$  is the ratio of the number of rider requests already assigned to vehicles to the total number of rider requests, in the current solution ( $X_{current}$ ).

#### 4. Case studies and simulation

This section presents two case studies to demonstrate the versatility and effectiveness of the developed methodologies for optimizing ride-sharing feeder services and to interpret the results in practical contexts. The first case focuses on first-mile ridesharing, connecting passengers to a transit hub, using the Houston Amtrak Train Station as the testbed. The second case is regarding cooling shelter accessibility under extreme heat in Houston, following widespread power outages caused by Hurricane Beryl on July 8, 2024. All models and codes run on a computer with a processor of 12th Gen Intel(R) Core(TM) i9-12900HK 2.50 GHz, with installed RAM of 32.0 GB, and with a 64-bit operating system and x64-based processor.

##### 4.1. First-mile ridesharing

The first-mile ridesharing study conducts two small-scale simulation experiments, two medium-scale simulation experiments, and two large-scale simulation examples with considerations of both sufficient and insufficient available vehicles. These simulation examples aim to present corresponding tests on proposed optimization methodologies, which can also be adapted for practical application.

###### 4.1.1. Small-scale simulations

In small-scale numerical examples, 10 prescheduled passenger requests and 80 on-demand passenger requests are simulated near the

Houston Amtrak Train Station. Passenger request locations are randomly and uniformly generated that are at least 0.5 miles and at most 5 miles away from the train station. All passengers will catch the same train at the Houston Amtrak Train Station. The train departure time is set at 9:05 a.m. Prescheduled passengers sent requests before 8:00 a.m. After 8:00 a.m., the platform starts to receive on-demand passengers' requests. Passengers' request time windows vary with their latest required arrival times at the transit hub (denoted as  $\alpha_i^{AD}$ ) and factors in the minimum necessary travel and pickup constraints. Specifically, the time window for passenger  $i$  is defined as  $[8:00 \text{ a.m.}, \alpha_i^{AD} - t_{i0} - 2 \text{ (min.)} - 15 \text{ (min.)}]$ , where

- $\alpha_i^{AD}$  represents the latest time by which the passenger must arrive at the transit hub.  $\alpha_i^{AD}$  is uniformly generated between 8:40 a.m. and 9:00 a.m. with train departure time of 9:05 a.m..
- $t_{i0}$  is the shortest travel time required to transport the passenger from their pickup location to the transit hub.
- **2 min** is an estimated time required for the pickup process (e.g., boarding the vehicle).
- **15 min** is a built-in buffer time to ensure timely arrival and account for potential delays.

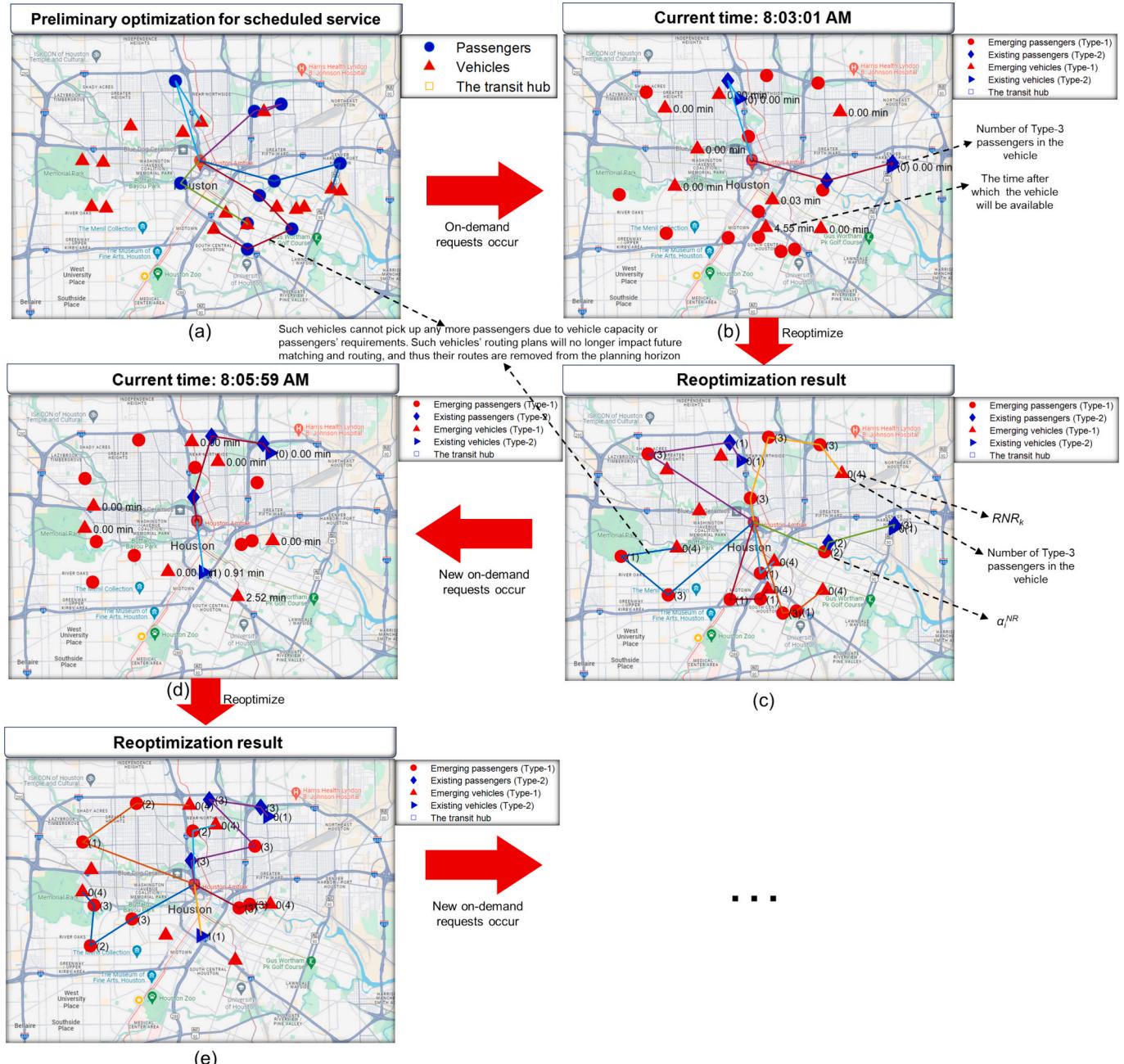
This ensures that all ride requests occur within a meaningful time-frame, preventing last-minute requests that would make timely arrival impossible.

In the scenario with sufficient vehicles, the number of Type-1 vehicles is  $m = \max(\lceil n/2 \times (1 + \text{rand}) \rceil, 4)$ , where  $n$  is the number of Type-1 passengers, “rand” is a random number uniformly distributed between 0 and 1, and “ $\lceil \cdot \rceil$ ” rounds up the number in the bracket to an integer. In the scenario with insufficient vehicles, the only difference is that the number of Type-1 vehicles is  $m = \max(\lceil n/8 \times (1 + \text{rand}) \rceil, 2)$ . The number

of Type-1 vehicles that are not immediately available is  $[rand \times 0.5 \times m]$  (available within  $5 \times rand$  minutes). All other Type-1 vehicles are immediately available.

**Fig. 3** displays the simulation results for the ridesharing services in the rolling horizon. The following clarifications are presented in this figure. The results based on the dynamic optimization methodology include matching plans and routing sequences. **Fig. 3.a** presents the initial optimization for scheduled service. **Fig. 3.b** presents existing routing plans of two vehicles (Type-2) and 12 emerging passenger requests (Type-1) sent within the first time slice between 8:00:00 a.m. and 8:03:01 a.m., as well as 8 newly available vehicles (Type-1 vehicles). The system re-optimizes the matching and routing plan at 8:03:01 a.m. as one of the passengers' latest response times is approaching. Based on the re-optimization results, five new vehicles (Type-1) have participated in the ridesharing services, which will pick up 10 emerging passengers

(Type-1). The other two Type-1 passengers are picked up by the two Type-2 vehicles, whose original routing plans are adjusted. After the first re-optimization, all vehicles will adopt the routing plan based on the re-optimization results. The system continues receiving passengers' requests and conducting re-optimization based on real-time information of passenger requests and vehicles. Specifically, at 8:05:59 a.m., which reaches one passenger's response deadline in the second time slice, the system conducts re-optimization with available vehicles and pending passengers with service requirements (**Fig. 3. d**). Note that five of the seven Type-2 existing vehicles cannot serve more passengers because the vehicle capacity is reached or passengers' maximum tolerable number of shared riders is reached, and thus they are removed from the rolling horizon planning. The remaining two Type-2 vehicles can still provide ridesharing services for waiting passengers. The results after re-optimization in the second time slices are shown in **Fig. 3. e**.



**Fig. 3.** Results of dynamic optimization methodology on the rolling horizon (a) Initial optimization for scheduled service; (b) Before 1st re-optimization for on-demand service; (c) After 1st re-optimization; (d) Before 2nd re-optimization; and (e) After 2nd re-optimization.

**Table 4** (a) and (b) summarize the small-scale simulation results with sufficient vehicles and insufficient vehicles, respectively. The two tables summarize the time slice duration, number of requests submitted by passengers, number of passenger requests served, total vehicle miles traveled (VMT), and computational cost. The re-optimization Mixed Integer Programming (RMIP) by CPLEX solver and LNS-TS are compared in the two tables. We also compare the real-time dynamic re-optimization approaches to the periodic optimization approaches, including periodic mixed integer programming (PMIP) and solution pooling approach (SPA), proposed by [Bian et al. \(2020\)](#). The periodic optimization methodologies, PMIP and SPA, focus solely on optimizing newly incoming ride requests and newly available vehicles in the current round. It does not re-optimize previously assigned vehicle-passenger matching plans or vehicle routes, nor does it account for the integrated optimization of prescheduled and on-demand service. Several findings can be drawn from [Table 4](#):

- (1) Taking the number of passenger requests served and VMT as serviceability metrics, the periodic optimization approaches, PMIP and SPA ([Bian et al., 2020](#)), have worse performances compared to dynamic re-optimization methodologies (e.g., RMIP and LNS-TS). In scenarios with sufficient vehicles, only two of 90 passenger requests are rejected in both re-optimization methodologies, RMIP and LNS-TS, while both periodic optimization approaches, PMIP and SPA, have 3 requests to be rejected. In scenarios with insufficient vehicles, the difference between dynamic re-optimization methodologies and periodic optimization methodologies in the number of passengers served is more significant: both re-optimization methodologies rejected 14 requests, while both periodic optimization methodologies have 19 requests to be rejected. There are two possible reasons behind the missing passenger requests. Firstly, the passenger preferences may be so strict that they cannot be satisfied by the practical vehicle routing plans. Secondly, the available vehicles are insufficient to satisfy all requests sent by passengers.
- (2) Regardless of whether the supply of vehicles is sufficient or insufficient, the gross VMTs obtained by the re-optimization methodologies are smaller than those by the periodic optimization approaches. In particular, the VMTs obtained by the dynamic re-optimization methodologies (277.99 miles by RMIP and 281.80 miles by LNS-TS) are around 14 % lower than those of the periodic optimization methodologies (326.01 miles by PMIP and SPA) in the scenario with sufficient vehicles. In the scenario with insufficient vehicles, the VMTs obtained by re-optimization methodologies (264.18 miles in RMIP and 264.18 miles in LNS-TS) are also lower than those of the periodic optimization methodologies (276.28 miles by PMIP and 276.28 miles by SPA).
- (3) LNS-TS have close but not exact values in simulation results with RMIP. RMIP and LNS-TS have identical numbers of served requests in both two scenarios. Regarding the VMT, RMIP and LNS-TS differ in only one round of re-optimization in the scenario of sufficient vehicle with time slice length equal to 2.05 min. In this case, the difference is only 3.78 miles (35.07 miles for RMIP vs. 38.85 miles for LNS-TS). For all other re-optimization rounds across both scenarios, the VMT values for RMIP and LNS-TS are identical.
- (4) Although LNS-TS and RMIP have slight differences in the routing plans, the computational costs of these two methodologies have significant distinctions. The gross computing times of RMIP with sufficient vehicles and insufficient vehicles are 230.85 s and 116.41 s, while the total computing times of LNS-TS are only 7.51 s and 5.83 s, respectively. Take an individual time slice as an example, RMIP needs over 133 s to process 12 passenger requests, while LNS-TS only takes less than 1 s when sufficient vehicles are dispatched in the ridesharing. This computing time may have already exceeded the length of one time slice, which is quite

impractical in real-time dynamic optimization. If a greater number of passengers are served, the computing time is expected to be considerable. These findings suggest that computation time is a critical factor when deploying real-time ridesharing systems, especially across areas with diverse spatial layouts, demand intensities, and service needs.

- (5) The length of the time slice declines as the time approaches the train departure time (e.g., 9:05 a.m. in this simulation study). This is in line with the rule of determining the time slice length. When passengers are more urgent to arrive at the transit hub and to catch the train, the response time should be close enough so that these passengers can have less waiting time, leading to a smaller time slice, and vice versa.

#### 4.1.2. Medium-scale simulations

From the small-scale simulations, we observe that LNS-TS achieves results very close to those of the RMIP approach in terms of both objective functions: the number of requests served and vehicle miles traveled (VMT). This similarity may stem from the limited problem scale, as both RMIP and LNS-TS can efficiently find near-optimal solutions in such scenarios. To further investigate the performance differences between the two methods, we conduct two medium-scale simulations under different vehicle supply conditions: one with sufficient vehicles and another with insufficient vehicles. In both cases, we set the number of scheduled passenger requests to 30 and on-demand requests to 240, maintaining all other problem parameters and settings consistent with the small-scale simulations. This allows for a direct comparison of algorithmic performance under more challenging and realistic conditions. Both the re-optimization approaches RMIP and LNS-TS are compared with the periodic optimization approaches, PMIP and SPA by [Bian et al. \(2020\)](#). Due to the excessive time required by CPLEX to reach an optimal solution, we imposed a 3600-s time limit on the RMIP and PMIP. The comparison results are summarized in [Table 5](#).

[Table 5\(a\)](#) shows results for the medium-scale case with sufficient vehicles. In terms of the comparison between re-optimization approaches and periodic optimization approaches, RMIP and LNS-TS maintain the same service level (270 requests) as PMIP and SPA but save travel distances significantly—RMIP reduces VMT by 9.14 % to 682.19 miles versus PMIP's 750.84 miles, while LNS-TS achieves 656.22 miles, an 8.28 % improvement over SPA's 715.48 miles. In terms of the comparison between RMIP and LNS-TS, we observe that both RMIP and LNS-TS achieve the same primary objective—maximizing the number of served rider requests—in all re-optimization rounds. Since all rider requests can be satisfied in this scenario, optimizing the secondary objective becomes more important. LNS-TS outperforms RMIP in minimizing Vehicle Miles Traveled (VMT), with total VMTs of 656.22 and 682.18, respectively. Additionally, LNS-TS demonstrates significantly higher computational efficiency, requiring less than 7 s for all re-optimization rounds, while RMIP often exceeds the 3600-s time limit, making it impractical for this application.

[Table 5\(b\)](#) shows the results for the medium-scale case with insufficient vehicles. In terms of the comparison between re-optimization approaches and periodic optimization approaches, RMIP and LNS-TS maintain slightly higher service levels (201 served requests for RMIP and 199 served requests for LNS-TS) than PMIP and SPA (196 for PMIP and 193 for SPA), and also RMIP and LNS-TS save VMT compared to PMIP and SPA: 525.01 miles for RMIP and 497.51 miles for LNS-TS vs. 543.25 miles for PMIP and 502.94 miles for SPA. In terms of the comparison between RMIP and LNS-TS, LNS-TS consistently matches or is slightly outperformed by RMIP in the number of requests served (199 vs. 201 in total). However, the differences are marginal, highlighting the inherent constraints of insufficient resources. In terms of the secondary objective, LNS-TS again excels in minimizing VMT, achieving a total of 497.52 miles compared to MIP's 525.02 miles. This suggests that LNS-TS not only handles resource constraints effectively but also optimizes routing efficiency. Computationally, LNS-TS remains far more efficient,

**Table 4**

Summary of optimization results in small-scale simulations.

(a) With sufficient vehicles															
Time Slice Duration (min)	Number of Passenger Requests	Number of Requests Served					Vehicle Miles Traveled (VMT)					Computing Time (second)			
		RMIP	Gap 1 (%) <sup>*</sup>	LNS-TS	PMIP	SPA	RMIP	Gap 2 (%) <sup>**</sup>	LNS-TS	PMIP	SPA	RMIP	LNS-TS	PMIP	SPA
Scheduled	10	10	0.00	10	10	10	30.15	0.00	30.15	30.15	30.15	45.26	0.82	35.77	0.76
	3.00	12	0.00	12	12	12	36.88	0.00	36.91	39.01	39.01	133.59	0.92	57.31	1.15
	2.73	9	0.00	9	9	9	24.62	0.00	24.62	24.99	24.99	5.20	0.67	5.73	0.93
	2.49	7	0.00	7	7	7	19.05	0.00	19.05	23.02	23.02	1.37	0.59	0.86	0.71
	2.28	7	0.00	7	7	7	26.59	0.00	26.59	29.80	29.80	1.34	0.37	1.02	0.58
	2.05	12	0.00	12	12	12	35.07	0.00	38.85	39.07	39.07	34.82	0.66	87.29	1.12
	1.86	5	0.00	5	5	5	16.24	0.00	16.24	24.56	24.56	1.16	0.34	0.63	0.60
	1.70	4	0.00	4	4	4	11.57	0.00	11.57	16.09	16.09	1.14	0.46	0.63	0.28
	1.51	7	0.00	7	7	7	17.22	0.00	17.22	22.35	22.35	1.23	0.42	0.86	0.61
	1.38	4	0.00	4	4	4	23.15	0.00	23.15	23.15	23.15	1.25	0.25	0.72	0.38
On-demand	1.27	3	0.00	3	2	2	15.44	0.00	15.44	13.87	13.87	1.14	0.29	0.62	0.22
	1.13	3	0.00	2	2	2	10.69	0.00	10.69	10.69	10.69	0.31	0.36	0.62	0.27
	1.00	1	0.00	1	1	1	0.61	0.00	0.61	7.95	7.95	0.31	0.38	0.73	0.15
	0.85	3	0.00	2	2	2	0.32	0.00	0.32	4.25	4.25	1.26	0.45	0.62	0.33
	0.69	1	0.00	1	1	1	1.88	0.00	1.88	4.63	4.63	0.29	0.26	0.62	0.16
	0.55	2	0.00	2	2	2	8.51	0.00	8.51	12.43	12.43	1.18	0.27	0.63	0.42
	Total	90	88	88	87	87	277.99		281.80	326.01	326.01	230.85	7.51	194.66	8.67

(b) With insufficient vehicles															
Time Slice Duration (min)	Number of Passenger Requests	Number of Requests Served					Vehicle Miles Traveled (VMT)					Computing Time (second)			
		RMIP	Gap 1 (%) <sup>*</sup>	LNS-TS	PMIP	SPA	RMIP	Gap 2 (%) <sup>**</sup>	LNS-TS	PMIP	SPA	RMIP	LNS-TS	PMIP	SPA
Scheduled	10	10	0.00	10	10	10	34.20	0.00	34.20	34.20	34.20	37.60	0.70	30.53	0.65
	3.00	16	0.00	12	10	10	32.47	0.00	32.47	29.12	29.12	60.76	0.61	9.84	0.70
	2.72	10	0.00	6	6	6	21.62	0.00	21.62	21.62	21.62	1.52	0.29	1.07	0.43
	2.48	7	0.00	6	6	6	21.41	0.00	21.41	21.41	21.41	1.12	0.25	0.76	0.37
	2.27	10	0.00	8	7	7	26.40	0.00	26.40	21.56	21.56	5.48	0.38	5.24	0.45
	2.07	5	0.00	4	4	4	19.69	0.00	19.69	20.04	20.04	1.12	0.22	0.62	0.29
	1.87	9	0.00	8	7	7	19.81	0.00	19.81	19.55	19.55	1.63	0.40	0.94	0.42
	1.68	2	0.00	2	2	2	7.33	0.00	7.33	13.53	13.53	0.23	0.29	0.62	0.28
	1.54	4	0.00	3	3	3	11.31	0.00	11.31	11.31	11.31	1.13	0.29	0.65	0.43
	1.40	3	0.00	3	3	3	4.93	0.00	4.93	13.92	13.92	1.14	0.22	0.62	0.20
On-demand	1.23	1	0.00	1	1	1	5.99	0.00	5.99	5.99	5.99	0.23	0.24	0.62	0.13
	1.13	2	0.00	2	2	2	9.21	0.00	9.21	9.21	9.21	1.14	0.32	0.64	0.28
	0.99	4	0.00	4	3	3	18.98	0.00	18.98	17.32	17.32	1.17	0.29	0.77	0.26
	0.88	3	0.00	3	3	3	11.87	0.00	11.87	14.42	14.42	1.15	0.30	0.63	0.21
	0.75	1	0.00	1	1	1	5.24	0.00	5.24	5.24	5.24	0.25	0.29	0.63	0.11
	0.63	1	0.00	1	1	1	2.18	0.00	2.18	4.00	4.00	0.25	0.24	0.71	0.12
	0.50	1	0.00	1	1	1	8.55	0.00	8.55	8.55	8.55	0.24	0.22	0.62	0.12
	0.50	1	0.00	1	1	1	2.98	0.00	2.98	5.27	5.27	0.24	0.28	0.64	0.12
	Total	90	76	76	71	71	264.17		264.17	276.26	276.26	116.40	5.83	56.16	5.57

\* Gap 1 represents the maximum relative difference between CPLEX's solution and the theoretical upper bound in optimizing the primary objective function value, i.e., the number of requests served.

\*\* Gap 2 represents the maximum relative difference between CPLEX's solution and the theoretical lower bound in optimizing the secondary objective function, specifically the vehicle miles traveled (VMT). Note that Gap 2 is only meaningful when Gap 1 is zero—if the primary objective function has not reached its theoretical optimum, the secondary objective function cannot be accurately evaluated.

**Table 5**

Summary of optimization results in medium-scale simulations.

(a) With sufficient vehicles															
Time Slice Duration (min)	Number of Passenger Requests	Number of Requests Served					Vehicle Miles Traveled (VMT)					Computing Time (second)			
		RMIP	Gap 1 (%) <sup>*</sup>	LNS-TS	PMIP	SPA	RMIP	Gap 2 (%) <sup>**</sup>	LNS-TS	PMIP	SPA	RMIP	LNS-TS	PMIP	SPA
Scheduled	30	30	0.00	30	30	30	70.12	11.24	66.93	69.23	66.93	3602.32	5.23	3601.11	5.11
3.00	39	39	7.14	39	39	39	95.42	–	91.03	116.86	93.75	3608.24	6.64	3605.75	6.61
2.74	18	18	0.00	18	18	18	41.77	50.37	41.77	45.47	44.30	3604.29	3.46	3603.79	3.15
2.52	32	32	3.03	32	32	32	71.98	–	63.32	67.89	65.67	3605.73	6.00	3611.95	5.58
2.30	25	25	7.41	25	25	25	68.85	–	65.86	65.78	65.03	3612.68	4.08	3612.11	3.90
2.11	22	22	4.35	22	22	22	54.85	–	51.31	57.53	57.58	3605.72	4.05	3604.50	3.69
1.92	23	23	4.17	23	23	23	60.34	–	57.30	64.25	58.05	3604.56	4.61	3604.09	4.35
1.74	20	20	0.00	20	20	20	43.91	29.66	43.75	48.61	48.95	3604.62	4.25	3603.93	3.83
1.59	8	8	0.00	8	8	8	22.92	0.01	22.92	25.11	25.11	1.07	1.80	0.85	1.48
1.45	9	9	0.00	9	9	9	18.54	0.06	18.54	21.57	21.57	10.50	2.09	5.50	1.57
1.33	7	7	0.00	7	7	7	18.55	0.00	18.55	25.21	25.21	0.75	1.54	0.96	1.04
On-demand	1.19	7	7	0.00	7	7	15.22	0.00	15.22	18.70	18.70	0.84	1.65	0.95	1.29
1.02	7	7	0.00	7	7	7	32.68	0.00	32.68	33.59	33.59	0.74	1.44	0.84	0.89
0.93	7	7	0.00	7	7	7	19.12	0.00	19.12	24.28	24.28	0.75	1.73	0.75	1.24
0.84	6	6	0.00	6	6	6	23.84	0.02	23.84	26.45	26.45	0.75	1.41	0.75	0.93
0.71	4	4	0.00	4	4	4	5.34	0.00	5.34	12.43	12.43	0.63	1.24	0.63	0.69
0.63	1	1	0.00	1	1	1	0.36	0.00	0.36	3.55	3.55	0.63	0.45	0.73	0.39
0.54	1	1	0.00	1	1	1	4.00	0.00	4.00	4.00	4.00	0.63	0.41	0.87	0.33
0.5	1	1	0.00	1	1	1	1.82	0.00	1.82	4.69	4.69	0.63	0.42	0.72	0.33
0.5	1	1	0.00	1	1	1	5.19	0.00	5.19	5.19	5.19	0.72	0.39	0.63	0.37
0.5	1	1	0.00	1	1	1	1.54	0.00	1.54	4.62	4.62	0.64	0.41	0.73	0.38
0.5	1	1	0.00	1	1	1	5.83	0.00	5.83	5.83	5.83	0.66	0.46	0.63	0.33
Total	270	270	270	270	270	270	682.19	656.22	750.84	715.48	28,868.10	53.76	28,862.77	47.48	
(b) With insufficient vehicles															
Time Slice Duration (min)	Number of Passenger Requests	Number of Requests Served					Vehicle Miles Traveled (VMT)					Computing Time (second)			
		RMIP	Gap 1 (%) <sup>*</sup>	LNS-TS	PMIP	SPA	RMIP	Gap 2 (%) <sup>**</sup>	LNS-TS	PMIP	SPA	RMIP	LNS-TS	PMIP	SPA
Scheduled	30	30	0.00	30	30	30	70.55	11.11	67.45	70.55	67.45	3602.28	3.37	3601.25	3.54
3.00	32	28	15.15	28	25	25	73.57	–	68.49	72.71	60.69	3604.67	6.00	3604.34	6.97
2.74	30	21	12.50	20	21	20	55.71	–	49.67	55.71	49.52	3608.28	4.67	3609.89	4.97
2.51	25	19	13.64	19	19	18	53.47	–	51.19	51.19	43.19	3604.39	5.13	3604.59	4.79
2.29	24	15	16.67	14	15	14	40.24	–	29.24	40.24	29.24	3605.25	4.09	3605.59	4.45
2.10	20	13	0.00	13	13	13	27.93	0.07	27.93	27.93	27.93	529.69	3.46	527.63	3.88
1.91	19	12	0.00	12	12	12	30.56	0.06	30.56	30.56	30.56	601.11	2.91	605.22	3.97
1.72	22	11	8.33	11	11	11	30.20	–	30.20	30.20	30.20	3603.79	4.03	3603.88	4.70
1.57	13	5	0.00	5	5	5	10.24	0.00	10.24	10.24	10.24	1.29	2.14	1.07	1.26
1.44	12	9	0.00	9	9	9	22.28	0.00	22.28	22.28	22.28	1.32	1.28	1.28	1.40
1.30	9	8	0.00	8	8	8	19.52	0.00	19.52	22.32	22.32	1.41	1.58	1.18	1.28
1.19	8	7	0.00	7	6	6	12.09	0.00	12.09	14.24	14.24	0.83	1.07	0.74	0.81
On-demand	1.06	5	3	0.00	3	3	13.72	0.00	13.72	13.72	13.72	0.62	0.86	0.64	0.70
0.96	2	2	0.00	2	2	2	9.38	0.00	9.38	9.38	9.38	0.64	0.84	0.64	0.48
0.88	5	5	0.00	5	4	4	17.81	0.00	17.81	19.24	19.24	0.63	1.28	0.63	0.75
0.79	3	2	0.00	2	2	2	4.89	0.00	4.89	4.89	4.89	0.63	1.01	0.71	0.50
0.68	3	3	0.00	3	3	3	7.11	0.00	7.11	10.34	10.34	0.63	0.93	0.71	0.49
0.55	2	2	0.00	2	2	2	8.43	0.00	8.43	8.43	8.43	0.70	0.89	0.63	0.46
0.50	2	2	0.00	2	2	2	3.72	0.00	3.72	3.72	3.72	0.71	0.79	0.63	0.42
0.50	1	1	0.00	1	1	1	8.29	0.00	8.29	8.29	8.29	0.63	0.73	0.63	0.34
0.50	1	1	0.00	1	1	1	0.31	0.00	0.31	6.69	6.69	0.63	0.75	0.63	0.33
0.50	1	1	0.00	1	1	1	4.93	0.00	4.93	4.93	4.93	0.63	0.79	0.63	0.29
0.50	1	1	0.00	1	1	1	0.06	0.00	0.06	5.45	5.45	0.64	0.75	0.64	0.30
Total	270	201	199	196	193	193	525.01	497.51	543.25	502.94	22,771.40	49.35	22,773.78	47.08	

\* Gap 1 represents the maximum relative difference between CPLEX's solution and the theoretical upper bound in optimizing the primary objective function value, i.e., the number of requests served.

\*\* Gap 2 represents the maximum relative difference between CPLEX's solution and the theoretical lower bound in optimizing the secondary objective function, specifically the vehicle miles traveled (VMT). Note that Gap 2 is only meaningful when Gap 1 is zero—if the primary objective function has not reached its theoretical optimum, the secondary objective function cannot be accurately evaluated.

completing all re-optimization rounds within under 50 s total, while MIP frequently hits one-hour time limits. This solidifies LNS-TS as a more scalable and practical solution, particularly in time-sensitive scenarios where real-time decision-making is crucial.

#### 4.1.3. Large-scale simulations

Large-scale simulation examples are investigated in this section to further explore the performance of the proposed LNS-TS algorithm in a more practical case. In this one-hour studied period, it addresses 60

scheduled passenger requests and 480 on-demand requests with sufficient vehicles and insufficient vehicles, respectively. The small-scale and medium-scale simulations have proven that RMIP may suffer from considerably inefficient computation costs when no less than 10 passenger requests need to be addressed in one time slice. Thus, this section only focuses on the simulations with LNS-TS. Note that the periodic optimization approach SPA by [Bian et al. \(2020\)](#) is also compared with the re-optimization approach of LNS-TS.

In the simulation with sufficient vehicles (see [Table 6\(a\)](#)), 539 out of

**Table 6**  
Summary of optimization results in large-scale simulations.

(a) With Sufficient Vehicles							
Time Slice Duration (min)	Number of Passenger Requests	Number of Requests Served		Vehicle Miles Traveled (VMT)		Computing Time (second)	
		LNS-TS	SPA	LNS-TS	SPA	LNS-TS	SPA
Scheduled	60	60	60	113.11	113.11	9.57	8.02
On-demand	3.00	60	60	113.55	116.76	9.30	8.94
	2.74	57	57	117.92	120.24	8.53	8.36
	2.51	58	58	116.08	118.26	9.34	8.69
	2.29	53	53	101.04	106.87	9.34	8.81
	2.10	56	56	108.14	110.62	8.73	8.11
	1.90	37	37	80.77	82.18	7.24	6.69
	1.74	32	32	65.23	66.45	6.59	6.14
	1.59	30	30	60.15	63.46	5.78	4.64
	1.45	21	21	49.38	51.15	4.05	3.25
	1.33	7	7	21.11	23.15	1.60	1.11
	1.20	14	14	32.49	36.56	3.17	2.37
	1.09	12	12	25.81	29.27	2.78	1.97
	0.99	7	7	9.44	27.49	1.84	0.99
	0.90	5	5	27.70	27.70	1.32	0.66
	0.80	7	7	18.21	36.33	1.92	0.98
	0.72	4	4	11.56	11.74	1.43	0.65
	0.65	6	6	22.88	26.43	1.64	0.76
	0.60	4	4	8.38	12.40	1.38	0.61
	0.53	1	1	4.54	4.54	0.57	0.32
	0.50	2	1	2.22	8.48	0.96	0.38
	0.50	2	2	5.77	6.66	0.91	0.39
	0.50	2	2	1.37	4.60	0.93	0.40
	0.50	1	1	8.84	8.84	0.61	0.36
	0.50	1	1	4.04	4.04	0.54	0.33
	0.50	1	1	0.15	2.94	0.64	0.32
	540	539	539	1129.88	1220.27	100.71	84.25
(b) With Insufficient Vehicles							
Time Slice Duration (min)	Number of Passenger Requests	Number of Requests Served		Vehicle Miles Traveled (VMT)		Computing Time (second)	
		LNS-TS	SPA	LNS-TS	SPA	LNS-TS	SPA
Scheduled	60	60	60	118.55	118.55	8.57	8.00
On-demand	3.00	78	51	81.69	80.08	9.61	7.03
	2.74	55	37	67.48	64.42	5.66	5.26
	2.51	48	29	56.43	66.37	4.85	4.59
	2.30	56	28	53.49	54.53	5.14	4.69
	2.11	42	34	63.16	68.98	5.40	5.21
	1.93	38	31	62.37	57.78	5.17	4.48
	1.76	27	17	38.36	38.36	2.82	2.61
	1.61	23	17	42.28	40.18	3.05	2.56
	1.47	20	15	30.31	30.31	2.65	2.02
	1.34	13	9	21.49	19.12	1.96	1.14
	1.22	17	9	18.64	18.64	1.91	1.28
	1.12	10	7	17.81	17.81	1.65	1.02
	1.01	8	5	16.92	11.45	1.52	0.69
	0.93	10	5	10.10	9.33	1.72	0.75
	0.81	8	6	13.87	11.65	1.87	0.66
	0.73	4	4	17.36	17.36	1.21	0.56
	0.67	5	4	9.84	9.84	1.34	0.61
	0.59	5	3	12.10	13.04	1.49	0.66
	0.52	4	3	13.51	11.03	1.38	0.53
	0.50	2	2	2.25	8.87	1.01	0.47
	0.50	2	2	11.12	13.88	0.98	0.51
	0.50	3	2	6.37	13.57	1.18	0.54
	0.50	2	2	0.99	4.51	0.98	0.42
Total	540	384	369	786.49	799.66	73.12	56.29

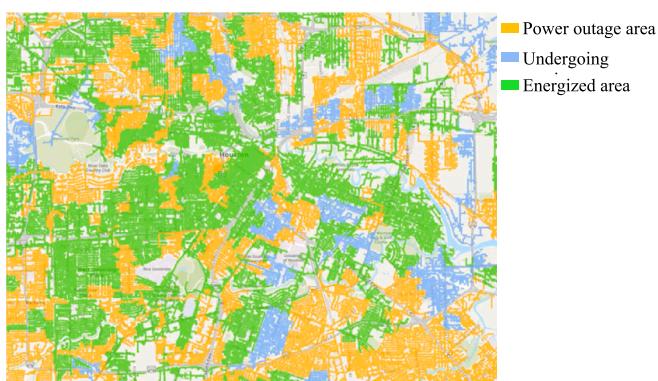
540 passenger requests are served based on the results of both dynamic re-optimization and periodic optimization methods. However, LNS-TS is never outperformed by SPA in terms of VMT in all rounds of optimization. The total travel cost, noted as VMT, in LNS-TS (1129.88 miles) is 7.41 % lower than that in the periodic optimization SPA (1220.27 miles), which indicates that the routing plan obtained by LNS-TS can provide a more efficient ridesharing service with less travel cost than the periodic optimization.

With insufficient vehicles (see Table 6(b)), the dynamic-re-optimization-based routing plan by LNS-TS can serve more passengers' requests (384) than the periodic SPA (369). Even though, the gross VMT by LNS-TS (786.49) is still slightly lower than that in the periodic SPA (799.66). Overall, the proposed dynamic re-optimization methodology can provide ridesharing services with more passengers and with less travel cost. Regarding the computational cost, no matter vehicles are sufficient or insufficient, the LNS-TS only takes less than 10 s for each round of optimization in each time slice, which is prompt for practical implementation.

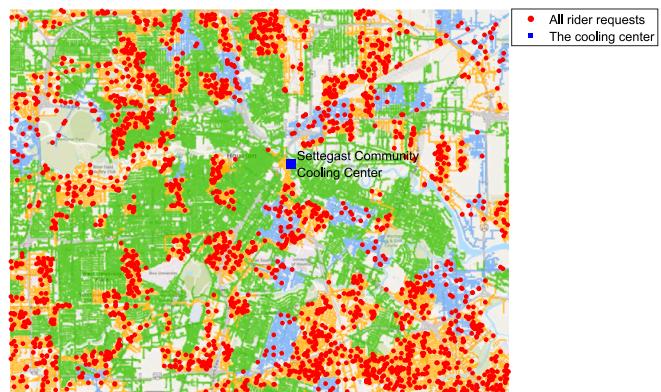
#### 4.2. Cooling shelter accessibility under extreme heat

The second case study explores a ridesharing service designed to enhance access to cooling shelters. The case study focuses on the Houston area, Texas, United States, where was impacted by Hurricane Beryl on July 8, 2024. The storm led to extensive power outages across the region, creating an urgent need for transportation to cooling centers to protect residents from extreme heat. Fig. 4 presents the power outage map on July 10, 2024, 8:42 a.m., two days after Hurricane Beryl, when more than 1.2 million CenterPoint Energy customers were out of power. This section applies this map to simulate ride requests directed to the Settegast Community Cooling Center, which has regained power. Given the widespread outages across the Houston area at the time, the simulation models a substantial population, 2400 ride requests, from affected regions—including those undergoing power restoration—requesting transportation to the Settegast Cooling Center. These simulated requests are illustrated in Fig. 5. The simulation model also simulates a fleet of vehicles that are assumed to be dispatched to transport vulnerable population to the cooling shelter during the extreme heat after the disaster.

In this case, scheduled requests refer to pre-arranged transportation for vulnerable groups, such as the elderly and disabled individuals, while on-demand requests represent spontaneous ride needs from other individuals. Suppose that Settegast Community Cooling Center opens at 9:00 a.m. in the morning. The 2400 ride requests are simulated between 9:00 a.m. and 12:00 p.m.. Note that unlike first-mile ridesharing services, where passengers typically have similar latest arrival times and can be grouped for a single round of preliminary optimization, the riders requesting access to cooling shelters may have widely varying latest



**Fig. 4.** Power outage map in houston (partial area) on July 10, 2024, 8:42 a.m. (Sarah, 2024).



**Fig. 5.** Simulated ride requests directed to the cooling center.

arrival times. In this study, we assume that scheduled riders' latest arrival times are randomly distributed between 9:00 a.m. and 12:00 p.m. The system initiates preliminary optimization only when a scheduled rider's arrival time is approaching—specifically, before one hour of their latest arrival time. This optimization includes scheduled requests and may also incorporate on-demand requests. Because these optimizations can occur in multiple rounds as arrival times approach, there is no clear cut-off or single timeline for separating scheduled and on-demand requests. The key distinction is that scheduled requests made well in advance of the latest arrival time are prioritized for service. Thus, the results in Table 7 do not present scheduled service as the first-mile ridesharing case study does.

The developed methodology, LNS-TS, is utilized to solve each real-time re-optimization problem. Note that the RMIP model is not tested in this case study because the problem scale is so large that RMIP is unable to return a solution within a reasonable amount of time. Fig. 6 illustrates the real-time re-optimization process for ridesharing routes to the cooling shelter with mixed pre-scheduled and on-demand requests. This re-optimization is performed five minutes after a new ride request is received, following the previous round of re-optimization. In this study, the secondary objective function focuses on minimizing vehicle hours traveled (VHT) rather than vehicle miles traveled (VMT). This prioritization reflects the urgency of disaster response scenarios, where reducing travel time is critical to ensuring prompt to serve affected populations. The LNS-TS re-optimization approach is compared with the periodic optimization approach SPA in terms of total number of riders served and vehicle hours traveled (VHT). The simulation and comparison results are presented in Table 7.

All 2400 rider requests during the observed time slices were successfully served using both the re-optimization LNS-TS and the periodic optimization approach SPA, demonstrating the system's efficiency in meeting demand. The LNS-TS method consistently resulted in lower total VHT (237.1 h) compared to the independent optimization approach (244.2 h). Also, the LNS-TS method achieved lower VHT compared to the periodic optimization approach in most time slices, showcasing its advantage in adapting to real-time changes. The LNS-TS method requires a total of 630.36 s of computing time across all time slices. The computing times across all rounds of re-optimization are less than 50 s, which is prompt for this large-scale optimization in an on-demand condition. In summary, the LNS-TS method demonstrates superior efficiency in minimizing vehicle hours traveled, particularly in high-demand scenarios, at the cost of slightly higher computing time. Its dynamic re-optimization approach effectively balances real-time responsiveness with overall system performance.

#### 4.3. Case study summary and transport geography implications

This section provides a summary of all case studies in Table 8 and provides transport geography implications.

**Table 7**

Summary of optimization results for enhancing ridesharing accessibility to cooling shelters.

Time Slice Start Time	Time Slice End Time (Re-optimization time)	Number of Passenger Requests	Number of Requests Served		Vehicle Hours Traveled (VHT)		Computing Time (second)	
			LNS-TS	SPA	LNS-TS	SPA	LNS-TS	SPA
9:00:02 a.m.	9:05:02 a.m.	89	89	89	8.78	9.16	23.47	19.47
9:05:04 a.m.	9:10:04 a.m.	90	90	90	8.18	8.35	24.87	23.21
9:10:06 a.m.	9:15:06 a.m.	86	86	86	8.22	7.76	25.94	21.00
9:15:08 a.m.	9:20:08 a.m.	75	75	75	7.63	8.20	15.40	14.41
9:20:10 a.m.	9:25:10 a.m.	79	79	79	8.39	8.42	18.53	19.42
9:25:10 a.m.	9:30:10 a.m.	90	90	90	8.98	9.24	28.17	29.26
9:30:11 a.m.	9:35:11 a.m.	89	89	89	8.59	8.93	22.20	20.53
9:35:13 a.m.	9:40:13 a.m.	90	90	90	9.23	9.79	20.65	21.61
9:40:15 a.m.	9:45:15 a.m.	98	98	98	9.34	9.53	37.27	35.15
9:45:15 a.m.	9:50:15 a.m.	92	92	92	9.31	9.49	33.98	28.69
9:50:17 a.m.	9:55:17 a.m.	89	89	89	8.07	8.72	18.94	18.99
9:55:18 a.m.	10:00:18 a.m.	80	80	80	8.64	9.05	21.80	20.57
10:00:21 a.m.	10:05:21 a.m.	77	77	77	7.40	7.78	17.60	15.41
10:05:26 a.m.	10:10:26 a.m.	90	90	90	9.85	10.52	28.06	29.19
10:10:28 a.m.	10:15:28 a.m.	68	68	68	6.44	6.60	19.08	16.18
10:15:28 a.m.	10:20:28 a.m.	84	84	84	9.04	9.06	12.19	10.12
10:20:39 a.m.	10:25:39 a.m.	104	104	104	10.87	11.11	49.43	47.62
10:25:40 a.m.	10:30:40 a.m.	89	89	89	9.34	9.91	17.40	16.22
10:30:40 a.m.	10:35:40 a.m.	96	96	96	9.47	8.83	38.05	33.27
10:35:48 a.m.	10:40:48 a.m.	81	81	81	7.42	7.95	19.71	17.35
10:40:51 a.m.	10:45:51 a.m.	75	75	75	7.55	7.86	11.49	9.92
10:45:51 a.m.	10:50:51 a.m.	86	86	86	8.12	8.29	24.22	25.28
10:50:52 a.m.	10:55:52 a.m.	86	86	86	8.66	8.57	20.74	21.34
10:55:53 a.m.	11:00:53 a.m.	87	87	87	7.97	8.27	17.14	13.82
11:00:59 a.m.	11:05:59 a.m.	68	68	68	6.84	7.09	10.54	8.61
11:06:04 a.m.	11:11:04 a.m.	85	85	85	8.35	8.05	20.12	18.18
11:11:08 a.m.	11:16:08 a.m.	80	80	80	7.19	7.60	25.76	26.05
11:16:39 a.m.	11:21:39 a.m.	54	54	54	5.31	5.39	4.90	3.98
11:21:59 a.m.	11:26:59 a.m.	29	29	29	2.52	2.67	1.75	1.46
11:28:12 a.m.	11:33:12 a.m.	13	13	13	1.19	1.83	0.66	0.64
11:34:44 a.m.	11:39:44 a.m.	1	1	1	0.23	0.23	0.30	0.27
Total		2400	2400	2400	237.12	244.25	630.36	587.22

#### 4.3.1. First-mile ridesharing

Small-sale simulation: In scenarios with sufficient vehicles, the proposed re-optimization methodologies—RMIP and LNS-TS—successfully serve 88 out of 90 passenger requests, slightly outperforming the 87 requests served by the periodic optimization methods (PMIP and SPA) proposed by [Bian et al. \(2020\)](#). More notably, dynamic re-optimization reduces vehicle miles traveled (VMT) by over 14 % compared to periodic optimization: 277.99 miles (RMIP) and 281.80 miles (LNS-TS) vs. 326.01 miles (PMIP and SPA). Under limited vehicle supply, RMIP and LNS-TS serve 7.04 % more passengers than periodic methods (76 vs. 71 requests) while still achieving 10.89 % lower VMT: 246.17 miles (RMIP & LNS-TS) vs. 276.26 miles (PMIP & SPA). In addition, LNS-TS exhibits high computational efficiency compared to RMIP by saving 93.12 % and 94.99 % computing time in scenarios of sufficient vehicles and insufficient vehicles, respectively.

Medium-scale simulation: In scenarios with sufficient vehicle supply, the re-optimization methodologies significantly reduce vehicle miles traveled (VMT) compared to periodic optimization methods while serving the same number of passenger requests (270). Specifically, RMIP achieves 682.19 miles, a 9.14 % reduction compared to PMIP's 750.84 miles, while LNS-TS reaches 656.22 miles, an 8.28 % reduction compared to SPA's 715.48 miles. Under insufficient vehicle supply, RMIP and LNS-TS serve more requests (201 and 199, respectively) than periodic methods (196 by PMIP and 193 by SPA), while also achieving 3.36 % and 1.08 % lower VMTs (525.01 miles for RMIP and 497.51 miles for LNS-TS, compared to 543.25 miles for PMIP and 502.94 miles for SPA). Notably, LNS-TS maintains computational efficiency, reducing computing time by 99.81 % and 99.78 % compared with RMIP, respectively in scenarios with sufficient and insufficient vehicle supply: 53.76 s vs. RMIP's 28,868.10 s in the scenario of sufficient vehicle supply and 49.35 s vs. RMIP's 22,771.40 s in the scenario of insufficient vehicle supply.

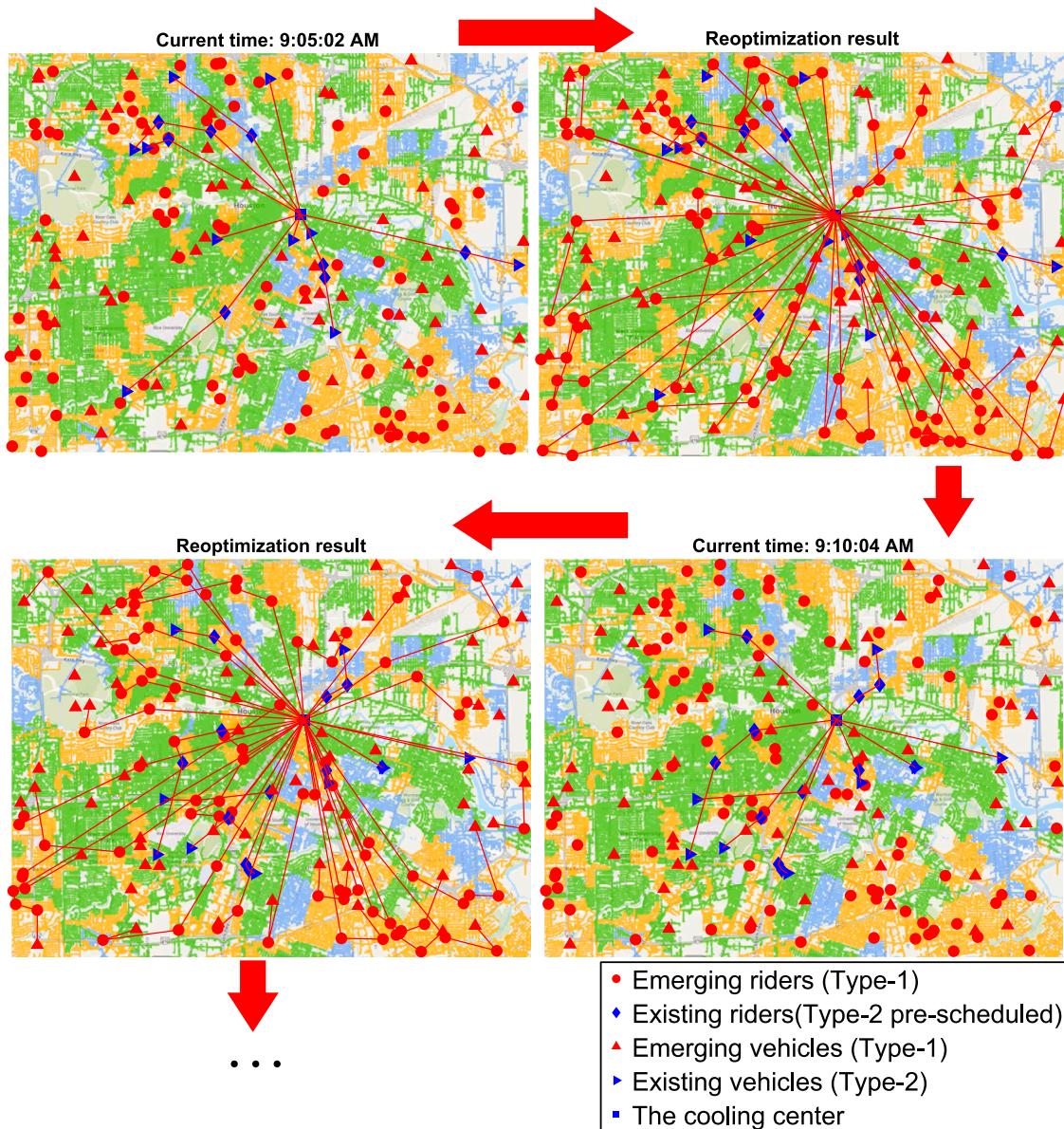
Large-scale simulation: In scenarios with sufficient vehicles, the re-optimization methodology LNS-TS reduces VMT by 7.41 % compared to the periodic optimization method SPA while serving the same number of passenger requests (539): 1129.88 miles for LNS-TS versus 1220.27 miles for SPA. Under insufficient vehicle supply, LNS-TS serves 4.07 % more passenger requests than SPA while still achieving a 1.67 % reduction in VMT.

#### 4.3.2. Cooling center access under extreme heat

Both the re-optimization LNS-TS and periodic optimization SPA successfully fulfilled all 2400 rider requests during the observed time intervals, confirming the system's capability to handle peak demand. However, the LNS-TS method outperformed the SPA by reducing 2.91 % total vehicle hours traveled (VHT): 237.1 h for LNS-TS, compared to 244.2 h for SPA. In terms of computational efficiency, LNS-TS requires a total of 630.4 s across all re-optimization rounds, with each iteration completed in under 50 s—a reasonable processing time for large-scale, on-demand operations.

Based on the summary of all case studies, we can conclude that the proposed dynamic ridesharing re-optimization approaches—Re-optimization Mixed Integer Programming (RMIP) and Large Neighborhood Search-Tabu Search (LNS-TS)—address key transport geography concerns by improving first-mile connectivity, reducing vehicle miles traveled (VMT), minimizing vehicle hours traveled (VHT), and enhancing prompt response for large-scale operations.

**4.3.2.1. Enhancing first-mile connectivity and accessibility.** First-mile transportation remains a critical challenge in urban and suburban mobility, particularly in low-density areas with limited transit options. The simulations demonstrate that dynamic re-optimization (RMIP and LNS-TS) outperforms periodic optimization (PMIP and SPA) in serving more passenger requests while reducing unnecessary travel:



**Fig. 6.** Illustration of Real-time Re-optimization of Ridesharing Routing to the Cooling Shelter.

- Small-scale: Under limited vehicle supply, RMIP and LNS-TS serve 7.04 % more passengers (76 vs. 71) while reducing VMT by 10.89 %.
- Medium-scale: With insufficient vehicles, RMIP and LNS-TS serve 201 and 199 requests (vs. 196 by PMIP and 193 by SPA), improving accessibility for riders in underserved zones.
- Large-scale: LNS-TS serves 4.07 % more passengers than SPA under limited vehicle supply, ensuring better spatial coverage.

These findings highlight how dynamic re-optimization enhances transport equity by maximizing service availability in areas where demand may otherwise go unmet due to rigid periodic scheduling.

**4.3.2.2. Reducing vehicle miles traveled (VMT) and environmental impact.** Transport Geography emphasizes sustainable mobility by minimizing congestion and emissions. The results show that RMIP and LNS-TS significantly reduce VMT compared to periodic methods:

- Small-scale: 14 % VMT reduction (277.99 miles for RMIP and 281.80 miles vs. 326.01 miles for PMIP/SPA).

- Medium-scale: 9.14 % reduction with RMIP compared to PMIP (682.19 miles vs. 750.84 miles) and 8.28 % with LNS-TS compared to SPA (656.22 miles vs. 715.48 miles).
- Large-scale: LNS-TS achieves 7.41 % lower VMT compared to SPA (1129.88 miles vs. 1220.27 miles).

This reduction in VMT translates to lower fuel consumption, reduced emissions, and decreased road wear, aligning with sustainability goals in transport planning.

**4.3.2.3. Minimizing vehicle hours traveled (VHT) for efficient resource use.** In scenarios such as cooling center access during extreme heat, where timely service is critical, LNS-TS reduces total VHT by 2.91 % (237.1 h vs. 244.2 h for SPA). This efficiency ensures faster response times for riders in need, lower operational costs for service providers, and improved network fluidity, reducing delays in high-demand periods.

**4.3.2.4. Prompt response for scalable urban mobility solutions.** Transport Geography also considers system adaptability and prompt

**Table 8**  
Summary of all case studies.

Case studies	Re-optimization methodologies (Proposed by this paper)						Periodic optimization methodologies (Bian et al., 2020)						
	RMIP			LNS-TS			PMIP			SPA			
	Number of served requests	VMT or VHT <sup>*</sup>	Computing time (second)	Number of served requests	VMT or VHT <sup>*</sup>	Computing time (second)	Number of served requests	VMT or VHT <sup>*</sup>	Computing time (second)	Number of served requests	VMT or VHT <sup>*</sup>	Computing time (second)	
First-mile ridesharing	Sufficient vehicles	88	277.99	230.85	88	281.80	7.51	87	326.01	194.66	87	326.01	8.67
	Insufficient vehicles	76	264.17	116.40	76	264.17	5.83	71	276.26	56.16	71	276.26	5.57
	Sufficient vehicles	270	682.19	28,868.10	270	656.22	53.76	270	750.84	28,862.77	270	715.48	47.48
	Insufficient vehicles	201	525.01	22,771.40	199	497.51	49.35	196	543.25	22,773.78	196	502.94	47.08
	Sufficient vehicles	-**	-	-	539	1129.88	100.71	-	-	-	539	1220.27	84.25
	Insufficient vehicles	-	-	-	384	786.49	73.12	-	-	-	369	799.66	56.29
Cooling center access under extreme heat	Large-scale simulation	-	-	-	2400	237.12	630.36	-	-	-	2400	244.25	587.22

\* VMT for first-mile ridesharing and VHT for cooling center access.

\*\* Represents that no results are obtained due to too long computing time.

responsiveness in real-time operations. LNS-TS demonstrates exceptional computational performance:

- Small-scale: 93–95 % computing time (i.e. response time) reduction compared with RMIP.
- Medium-scale: more than 99.78 % faster than RMIP.
- Large-scale & emergency scenarios: Processes requests in under 50 s per iteration, making it viable for real-world, on-demand ridesharing systems.

This efficiency ensures that dynamic ridesharing can be deployed in megacities and emergency scenarios without prohibitive computational delays.

In conclusion, the dynamic ridesharing re-optimization approaches—RMIP and LNS-TS—offer significant transport geography benefits by improving first-mile accessibility, reducing vehicle miles traveled (VMT) and vehicle hours traveled (VHT), and enhancing computational efficiency for scalable urban mobility solutions. These methods outperform periodic optimization by serving more passenger requests, particularly in underserved areas, while substantially lowering environmental and operational costs through reduced travel distances and faster processing times. The efficiency of LNS-TS, especially in large-scale and emergency scenarios, demonstrates its potential for real-world deployment, supporting sustainable, equitable, and responsive transportation networks that adapt dynamically to fluctuating demand. Future research could further optimize these systems by integrating them with public transit networks and incorporating equity-based demand allocation strategies.

## 5. Conclusions

This paper proposed a real-time, dynamic re-optimization approach to ridesharing services, and considered a mixed type of riders. The primary objective is to maximize the number of served riders and the secondary objective is to minimize the travel cost, measured by the total vehicle miles traveled (VMT) or vehicle hours traveled (VHT), given that the first objective is achieved. The dynamic re-optimization approach is achieved by a rolling horizon planning approach with the aid of a mixed integer programming model (MIP). A Large Neighborhood Search by Tabu Search algorithm (LNS-TS) was designed in this paper to solve large-scale ridesharing problems, in which seven neighborhood structures are developed to generate solutions satisfying all practical constraints. In the first-mile ridesharing case study, the small-scale simulations proved that regardless of whether vehicles are sufficient or not, the proposed dynamic re-optimization method is able to provide ridesharing services for more requested passengers with fewer vehicle miles traveled in total than the independent optimization method. Compared with the MIP by CPLEX solver, LNS-TS demonstrates almost identical serviceability but significantly higher efficiency. The feasibility and practicability of the proposed approach in large-scale ridesharing problems were also manifested with better performance in route planning in real-time, which serves as a novel methodology for academia and practice in the domain of shared mobility. The second case study, focused on cooling shelter accessibility during extreme heat following a disaster, demonstrates the versatility of the developed methodology in addressing critical emergency scenarios. Additionally, it highlights the method's efficiency in optimizing large-scale problems involving a substantial number of riders seeking transportation to cooling centers.

Although the proposed method effectively integrates scheduled and on-demand ridesharing requests, several aspects still need further investigation. First, the proposed framework depends on timely and accurate real-time data, which may be delayed or uncertain in practice. Second, the method demonstrates good performance in large-scale simulations, but its scalability to more complex or higher-density urban environments may still require improvement. Third, while the model considers passenger preferences, real-world behaviors may be more complex, diverse, and influenced by dynamic contextual factors.

In the future, research can focus on improving the method's resilience to real-time data uncertainty, enhancing algorithm scalability through advanced computation, and incorporating adaptive passenger preferences to better reflect real-world conditions. The proposed algorithm can also be extended to account for road interruptions or infrastructure damage (e.g., those caused by hurricanes), enabling more adaptive routing in disaster-response scenarios. Furthermore, as cities evolve into smart and connected environments, the proposed framework can be integrated with shared autonomous vehicle (SAV) systems to support real-time decision-making in autonomous mobility services.

### CRediT authorship contribution statement

**Yanjie Yi:** Writing – original draft, Validation, Formal analysis, Data curation, Methodology. **Zheyong Bian:** Writing – review & editing, Visualization, Methodology, Conceptualization, Software, Supervision. **Bijun Wang:** Investigation, Visualization, Writing – review & editing.

### Data availability

Data will be made available on request.

### References

- Abdelmoumène, H., Bencheriet, C.E., Belleili, H., Touati, I., Zemouli, C., 2024. Dynamic matching optimization in ridesharing system based on reinforcement learning. *IEEE Access* 12, 29525–29535.
- Al-Abbasi, A.O., Ghosh, A., Aggarwal, V., 2019. Deeppool: distributed model-free algorithm for ride-sharing using deep reinforcement learning. *IEEE Trans. Intell. Transp. Syst.* 20 (12), 4714–4727.
- Benita, F., 2020. Carpool to work: determinants at the county-level in the United States. *J. Transp. Geogr.* 87, 102791.
- Berrojo Romeyro Mascarenhas, Guillermo, Joost Krämer, 2023. Reaching Your Destination: A Qualitative Study on Ridesharing as a Solution for First and Last-Mile Transportation Challenges.
- Bian, Z., Liu, X., 2019. Mechanism design for first-mile ridesharing based on personalized requirements part I: theoretical analysis in generalized scenarios. *Transp. Res. B Methodol.* 120, 147–171.
- Bian, Z., Liu, X., Bai, Y., 2020. Mechanism design for on-demand first-mile ridesharing. *Transp. Res. B Methodol.* 138, 77–117.
- Bian, Z., Bai, Y., Liu, X., Wang, B., 2022. An online hybrid mechanism for dynamic first-mile ridesharing service. *Transp. Res. Part C Emerg. Technol.* 138, 103585.
- Cao, Y., Wang, S., Li, J., 2021. The optimization model of ride-sharing route for ride hailing considering both system optimization and user fairness. *Sustainability* 13 (2), 902.
- Chen, S., Wang, H., Meng, Q., 2020. Solving the first-mile ridesharing problem using autonomous vehicles. *Comput. Aided Civ. Inf. Eng.* 35 (1), 45–60.
- Chen, X.M., Chen, X., Zheng, H., Xiao, F., 2021. Efficient dispatching for on-demand ride services: systematic optimization via Monte-Carlo tree search. *Transp. Res. Part C Emerg. Technol.* 127, 103156.
- Cordeau, J.-F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. *Oper. Res.* 54 (3), 573–586.
- Covelli Garrido, C.I., Giovannini, A., Mangone, A., Silvestri, F., 2023. Managing urban mobility during big events through living lab approach. *Sustainability* 15 (19), 14566.
- Dai, R., Ding, C., Gao, J., Wu, X., Yu, B., 2022. Optimization and evaluation for autonomous taxi ride-sharing schedule and depot location from the perspective of energy consumption. *Appl. Energy* 308, 118388.
- Feng, Siyuan, et al., 2022. Coordinating ride-sourcing and public transport services with a reinforcement learning approach. *Transp. Res. Part C Emerg. Technol.* 138, 103611.
- Giuliano, G.E.N.E.V.I.E.V.E., Hanson, S., 2017. Looking to the future. *Geogr. Urban Transp.* 359–388.
- Gu, Q.-P., Liang, J.L., 2024. Algorithms and computational study on a transportation system integrating public transit and ridesharing of personal vehicles. *Comput. Oper. Res.* 164, 106529.
- Haferkamp, J., Ehmke, J.F., 2020. An efficient insertion heuristic for on-demand ridesharing services. *Transp. Res. Procedia* 47, 107–114.
- Haliem, M., Mani, G., Aggarwal, V., Bhargava, B., 2021. A distributed model-free ride-sharing approach for joint matching, pricing, and dispatching using deep reinforcement learning. *IEEE Trans. Intell. Transp. Syst.* 22 (12), 7931–7942.
- Hendricks, Marcus D., Van Zandt, Shannon, 2021. Unequal protection revisited: planning for environmental justice, hazard vulnerability, and critical infrastructure in communities of color. *Environ. Justice* 14 (2), 87–97.
- Herbawi, W.M., Weber, M., 2012 July. A genetic and insertion heuristic algorithm for solving the dynamic ridematching problem with time windows. In: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, pp. 385–392.
- Herding, R., Mönch, L., 2024. A rolling horizon planning approach for short-term demand supply matching. *CEJOR* 32 (3), 865–896.
- Homsi, G., Gendron, B., Jena, S.D., 2024. Rolling horizon strategies for a dynamic and stochastic ridesharing problem with rematches. *Discret. Appl. Math.* 343, 191–207.
- Kang, D., Levin, M.W., 2021. Maximum-stability dispatch policy for shared autonomous vehicles. *Transp. Res. B Methodol.* 148, 132–151.
- Kumar, P., Khani, A., 2021. An algorithm for integrating peer-to-peer ridesharing and schedule-based transit system for first mile/last mile access. *Transp. Res. Part C Emerg. Technol.* 122, 102891.
- Lee, A., Savelsbergh, M., 2017. An extended demand responsive connector. *Euro J. Transp. Logist.* 6 (1), 25–50.
- Li, R., Liu, Z., Zhang, R., 2018. Studying the benefits of carpooling in an urban area using automatic vehicle identification data. *Transp. Res. Part C Emerg. Technol.* 93, 367–380.
- Lotti, S., Abdelghany, K., Hashemi, H., 2019. Modeling framework and decomposition scheme for on-demand mobility services with ridesharing and transfer. *Comput. Aided Civ. Inf. Eng.* 34 (1), 21–37.
- Lu, C.-C., Ying, K.-C., Chen, H.-J., 2016. Real-time relief distribution in the aftermath of disasters—a rolling horizon approach. *Transp. Res. Part E Logist. Transp. Rev.* 93, 1–20.
- Lyu, Y., Lee, V.C., Ng, J.K.-Y., Lim, B.Y., Liu, K., Chen, C., 2019. Flexi-sharing: a flexible and personalized taxi-sharing system. *IEEE Trans. Veh. Technol.* 68 (10), 9399–9413.
- Ma, S., Zheng, Y., Wolfson, O., 2013. T-share: a large-scale dynamic taxi ridesharing service. In: 2013 IEEE 29th International Conference on Data Engineering (ICDE), pp. 410–421.
- Ma, T.-Y., Rasulkhani, S., Chow, J.Y., Klein, S., 2019. A dynamic ridesharing dispatch and idle vehicle repositioning strategy with integrated transit transfers. *Transp. Res. Part E Logist. Transp. Rev.* 128, 417–442.
- Madridano, Á., Al-Kaff, A., Martín, D., De La Escalera, A., 2021. Trajectory planning for multi-robot systems: methods and applications. *Expert Syst. Appl.* 173, 114660.
- Masoud, N., Nam, D., Yu, J., Jayakrishnan, R., 2017. Promoting peer-to-peer ridesharing services as transit system feeders. *Transp. Res. Rec.* 2650 (1), 74–83.
- Mendes, R.S., Lush, V., Wanner, E.F., Martins, F.V., Sarubbi, J.F., Deb, K., 2021. Online clustering reduction based on parametric and non-parametric correlation for a many-objective vehicle routing problem with demand responsive transport. *Expert Syst. Appl.* 170, 114467.
- Meneses-Cime, K., Akşun Guvenc, B., Guvenc, L., 2022. Optimization of on-demand shared autonomous vehicle deployments utilizing reinforcement learning. *Sensors* 22 (21), 8317.
- Mohiuddin, H., 2021. Planning for the first and last mile: a review of practices at selected transit agencies in the United States. *Sustainability* 13 (4), 2222.
- Nitter, J., Yang, S., Fagerholt, K., Ormevik, A.B., 2024. The static ridesharing routing problem with flexible locations: a Norwegian case study. *Comput. Oper. Res.* 167, 106669.
- Robbins, D., Dickinson, J., Calver, S., 2007. Planning transport for special events: a conceptual framework and future agenda for research. *Int. J. Tour. Res.* 9 (5), 303–314.
- Rodrigue, J.P., 2020. *The Geography of Transport Systems*. Routledge.
- Sarah, 2024 July 10. Houston Power Outages: CenterPoint Map. Houston Public Media. Retrieved from <https://www.houstonpublicmedia.org/articles/infrastructure/2024/07/10/493056/houston-power-outages-centerpoint-map-beryl-wednesday/>.
- Sun, X., Wandelt, S., Stumpf, E., 2018. Competitiveness of on-demand air taxis regarding door-to-door travel time: a race through Europe. *Transp. Res. Part E Logist. Transp. Rev.* 119, 1–18.
- Wang, H., 2019. Routing and scheduling for a last-mile transportation system. *Transp. Sci.* 53 (1), 131–147.
- Wang, H., Li, J., Wang, P., Teng, J., Loo, B.P., 2023. Adaptability analysis methods of demand responsive transit: a review and future directions. *Transp. Rev.* 43 (4), 676–697.
- Wen, H., Zhao, D., Wang, W., Hua, X., Yu, W., 2025. Exploring the spatial distribution structure of intercity human mobility networks under multimodal transportation systems in China. *J. Transp. Geogr.* 123, 104144.
- Wong, S.D., Walker, J.L., Shaheen, S.A., 2021. Bridging the gap between evacuations and the sharing economy. *Transportation* 48 (3), 1409–1458.
- Woo, S., Yoon, S., Kim, J., Hwang, S.W., Kweon, S.J., 2021. Optimal cooling shelter assignment during heat waves using real-time mobile-based floating population data. *Urban Clim.* 38, 100874.
- Xia, J., Curtin, K.M., Huang, J., Wu, D., Xiu, W., Huang, Z., 2019. A carpool matching model with both social and route networks. *Comput. Environ. Urban. Syst.* 75, 90–102.
- Xu, Y., Tong, Y., Shi, Y., Tao, Q., Xu, K., Li, W., 2020. An efficient insertion operator in dynamic ridesharing services. *IEEE Trans. Knowl. Data Eng.* 34 (8), 3583–3596.
- Yoon, S., Woo, S., Kim, J., Hwang, S.W., Kweon, S.J., 2022. The location routing problem for cooling shelters during heat waves. *Urban Clim.* 44, 101138.
- Yu, W., Wang, W., Hua, X., Zhao, D., Ngoduy, D., 2025. Dynamic patterns of intercity mobility and influencing factors: insights from similarities in spatial time-series. *J. Transp. Geogr.* 124, 104154.