Case Studies on Transport Policy 20 (2025) 101462

Contents lists available at ScienceDirect

Case Studies on Transport Policy

journal homepage: www.elsevier.com/locate/cstp

A Multi-Objective model for shared-ride automated services: Reducing the price of anarchy through centralized matching

Min-Ci Sun, Luca Quadrifoglio *

Zachry Department of Civil and Environmental Engineering, Texas A&M University, College Station, TX 77843-3136, United States

ARTICLE INFO

Keywords:
Automated Mobility-On-Demand
Shared-Ride Services
Dynamic Rider Matching
Robotaxi

ABSTRACT

The rise of Automated Mobility-on-Demand (AMoD) services, such as Waymo and Zoox, has transformed urban mobility. However, the growing demand and expanding robotaxi fleets have the potential to exacerbate congestion. This study introduces a novel centralized ride-matching procedure to enhance the scheduling efficiency of Shared-Ride Automated Mobility-on-Demand Services (SRAMODS). The proposed multi-objective adaptive model addresses the perspectives of on-site, in-vehicle riders, and robotaxi operators by minimizing on-site waiting time, in-vehicle travel duration, and detour distance at each pairing epoch. The proposed model dynamically matches riders within each epoch, with each robotaxi accommodating up to 4 riders simultaneously. A case study using Chicago demand data demonstrates that different weighting distributions lead to distinct matching outcomes, and a balanced weighting across all objectives minimizes the total time spent. Compared to traditional decentralized ride-matching, SRAMODS reduces the price of anarchy—measured as distance traveled per rider—by up to 15%, highlighting the benefits of centralized control. These findings provide policy insights to encourage shared robotaxi adoption through centralized coordination, improving urban mobility while reducing operational inefficiencies and congestion.

1. Introduction

Transportation Network Companies (TNCs), such as Uber and Lyft, have reformed urban mobility over the past decade by providing services through online platforms. Unlike traditional public transportation, TNCs offer greater flexibility and convenience to users. In San Francisco, approximately 170,000 TNC vehicle trips occurred during a typical weekday, taking 15 % of all vehicle trips, while conventional taxi only accounts for 1 % (Castiglione et al., 2017). In 2019, there were about 91.1 million TNC rides in Massachusetts, indicating a 12.0 % increase from 2018 and a substantial 40.6 % rise from 2017 (Mass.gov, 2019). However, the growth of such Mobility-on-Demand (MoD) services has raised concerns about the external impacts, particularly congestion challenges (Diao et al., 2021; Erhardt et al., 2019).

The advance of autonomous driving technology in MoD has created a new pattern of transit systems—automated mobility on demand (AMoD). In the AMoD system, passengers request journeys using a mobile app, and the system provides the service 24/7 by a centrally managed fleet of AVs. We refer to such self-driving taxis as robotaxis, which have gained more popularity due to their decent fares and more reliable service. Compared with human-driven vehicles, robotaxi systems cut the labor costs of drivers and reflect lower fares. Also, with the development of autonomous driving technology, people have more confidence in its efficiency and safety. Hence, an increase in AMoD demand can be anticipated (Nahmias-Biran et al., 2019). For instance, Waymo, established in 2016, has a service area covering Phoenix, San Francisco, and Los Angeles, and has served over 700,000 ride-hailing trips without human drivers in 2023. Now it is planning to expand the service area to Austin (Waymo, 2023). Zoox, founded in 2014, conducted the first on-road test in Foster City, California, in 2023, and is scheduled to operate in San Francisco and Las Vegas (Zoox, 2023a, 2023b).

To mitigate potential congestion resulting from increasing demand, shared-ride services have emerged as a key approach in AMoD systems, offering travelers with similar spatial and temporal trips the option to share rides at a lower cost. The service, known as "Shared-Ride Automated Mobility-on-Demand Services (SRAMODS)" (Hyland and Mahmassani, 2020), utilized automated, non-human drivers (robotaxis) and a centralized system for ride-matching optimization and fleet management.

* Corresponding author.
E-mail addresses: mincisun5296@tamu.edu (M.-C. Sun), quadrifo@tamu.edu (L. Quadrifoglio).

https://doi.org/10.1016/j.cstp.2025.101462
Received 7 December 2024; Received in revised form 23 April 2025; Accepted 23 April 2025
Available online 26 April 2025

However, current analysis indicates that shared-ride trips only represent a small percentage of total MoD services in cities. The lack of growth can be attributed to various factors, such as an increase in users' in-vehicle times and operational difficulties for TNCs (Hou et al., 2020). As a result, the probability of successfully matching rides remains low (Alonso-González et al., 2021; Si et al., 2023). For example, in Toronto, with 15 % of sharing requests among all, only 52 % of these requests were matched (Young et al., 2020). In a case study of Denver, only 14.8 % of sharing requests received a matching ride (Henao and Marshall, 2019).

In addition, TNCs like UberX Share rely on decentralized matching mechanisms, where riders are paired based primarily on immediate distance savings, and human drivers retain the choice to accept or reject matches (Rinkus, 2014). This decentralized matching generates a cost of "anarchy" as individual agents—both riders and drivers—act in their self-interest, leading to a sort of "user equilibrium" matching. Such individualistic matching may result in suboptimal system outcomes, including longer waiting and travel duration for riders and increased operational costs for providers. For instance, while assigning the second-best rider match to a driver could benefit the TNCs overall, drivers naturally prioritize their most convenient match. This leads to situations where other drivers must cover longer distances to pick up the less optimal riders, adding extra operational costs and overall waiting time.

Moreover, TNCs often prioritize distance savings when matching riders. For instance, in Fig. 1, the shared-ride trip aims to minimize total travel distance. However, the vehicle may arrive at Passenger 2's location before the scheduled pick-up time, resulting in a stop that increases the total travel duration for Passenger 1. This highlights the importance of considering not only distance-based efficiency but also user satisfaction to achieve a more holistic optimization.

Driven by advancements in robotaxi technology, growing demand for MoD services, and the limitations of a narrow focus on a single objective, this study proposes a novel rider-matching model for SRAMODS, seeking to optimize robotaxi operations and overcome inefficiencies inherent in human-driven ride-sharing. Unlike conventional decentralized TNC matching, which prioritizes immediate distance savings and relies on driver discretion, our model introduces a centralized, real-time decision-making framework that holistically optimizes ride-matching. The key innovation lies in integrating adaptive multi-objective optimization, which balances riders' on-site waiting time, in-vehicle travel duration, and detour distance while ensuring system-wide efficiency. Additionally, our dynamic trip-based rider insertion mechanism enables full utilization of vehicle capacity—unlike static approaches that pre-group riders before assigning vehicles. This structured, real-time coordination enhances the viability of SRAMODS, addressing the low match rates in urban TNC services and paving the way for scalable deployment.

The remainder of the paper is structured as follows. Section 2 provides a literature review regarding shared-ride services. Section 3 introduces the operation framework for the SRAMODS model. Section 4 verifies the proposed model using the Chicago taxi data. Section 5 concludes the findings and discusses limitations for future research.

2. Literature review

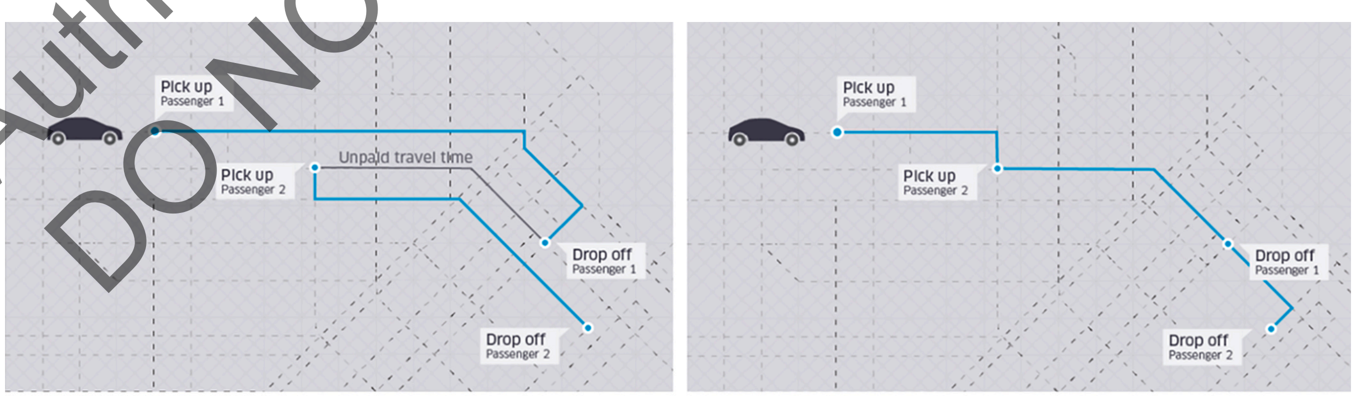
2.1. Overview of shared mobility and Ride-Matching

The proposed model for SRAMODS belongs to the field of shared economy in transportation. Before diving into the rider-matching model, we have to clarify some relevant terms, including:

(1) Shared-ride service: Shared-ride service is a broader term involving various forms of ride-sharing, taxi-pooling, ride-sourcing, and other services where multiple passengers share a vehicle (Castellanos et al., 2022; Shaheen and Cohen, 2019). This term is used in this study to describe sharing rides in robotaxis.
(2) Ride-sharing: Drivers in ride-sharing have specific origins and destinations, which distinguishes it from services with professional drivers. Based on the size of participants, ride-sharing can be further categorized into carpooling and vanpooling. Carpooling refers to a smaller group of people who share a private vehicle to commute together regularly, typically for work or school, while vanpooling is similar to carpooling but involves a larger vehicle (Chan and Shaheen, 2012). The matching in ride-sharing can be sorted into one-to-one, one-to-many, and multi-hop ride-matching (Tafreshian et al., 2020).
(3) Taxi-pooling: The service is applied to taxis when passengers share a ride with others who have similar routes (Yan et al., 2012).
(4) Ride-hailing and ride-sourcing: Both ride-hailing and ride-sourcing involve booking on-demand rides through digital platforms, but passengers do not share rides with other riders in this service (Wang and Yang, 2019). Ride-hailing involves professional drivers and licensed vehicles, but ride-sourcing allows non-professional drivers to offer rides using their personal cars.

2.2. Research Status

Previous research has tackled various challenges and optimizations associated with shared-ride services. Agatz et al. (2012) systematically outlined optimization challenges for ride-sharing, particularly in operations research, while Furuhata et al. (2013) classified existing ride-sharing systems and identified ride-sharing matching patterns, highlighting coordination difficulties in synchronizing itineraries, schedules, and costs among participants. Wang and Yang (2019) undertook a comprehensive literature review covering demand, pricing, supply,



(a) Two Separate Trips without Ridesharing (b) A Shared-ride Trip with Minimized Total Travel Distance

Fig. 1. Illustration of rider matching in UberPOOL.

incentives, and platform operations for ride-sourcing, and different objectives of previous works are compared. Despite these efforts, the SRAMODS problem, as a relatively new transportation pattern, remains underexplored. General ride-sharing models are not fully applicable to SRAMODS as robotaxis do not have fixed origins and destinations. Additionally, a centralized controller is capable of assigning robotaxis based on system-wide objectives.

Current decentralized matching in traditional taxis and TNCs allows drivers the flexibility to accept or reject matches. This decentralized, self-interested matching, or “user equilibrium”, generates a cost of “anarchy” in that both drivers and riders pursue immediate personal gains, sometimes at the expense of optimal system efficiency (Foti et al., 2021, 2017; Wolfson and Lin, 2017). Such individualistic matching can lead to suboptimal outcomes, such as increased rider wait times, longer travel durations, and higher operating costs for providers (Shen et al., 2019). By contrast, a centralized control system can ensure compliance among vehicles. Through centralized oversight, matching algorithms can prioritize both system efficiency and user satisfaction, creating a more balanced and mutually beneficial solution.

The SRAMODS problem, which lacks fixed origins and destinations, can be regarded as an extension of the dial-a-ride problem, where each request consists of disparate pick-up and drop-off locations (Berbeglia et al., 2010). Generally, these problems contain constraints to control user inconveniences, such as time windows or maximum travel time. As one of the studies pioneering in dial-a-ride problems, Psaraftis (1980) considered a single-vehicle case, seeking to minimize a weighted combination of waiting time and riding time. Nevertheless, SRAMODS requires more dynamic, real-time adjustments due to the autonomous nature of the vehicles and the need to respond to changing conditions. Besides, the increased number of vehicles within the system adds complexity to rider-vehicle matching as well as pick-up/drop-off sequence arrangement compared to dial-a-ride problems (Berbeglia et al., 2010).

To address the challenges of real-time rider-vehicle matching, a bipartite graph is often generated to evaluate the feasibility of various vehicle-rider matching conditions (Tafreshian and Masoud, 2020). Alonso-Mora et al. (2017) improved such request-vehicle matches and proposed a request-trip-vehicle graph (RTV graph), exploring the possibility of matching some riders and treating them as a new trip to match drivers. The authors started from a greedy assignment and improved it through a constrained optimization. Inheriting the RTV graph, Shah et al. (2020) proposed a multi-stage stochastic program that incorporated approximate dynamic programming within each time interval, and neural networks were leveraged for the approximation process. The RTV graph can be viewed as a set packing formulation where grouped riders are vertices in the bipartite graph. Sun et al. (2020) further developed an efficient graph-based approach to generate all necessary vehicle routes and a column generation-based heuristic algorithm to solve the set packing formulation.

However, in such graph-based matching, once a vehicle is dispatched to a rider or a rider group, the vehicle route is final, and no new riders can be assigned to this vehicle (Sun et al., 2020). Although Santos and Xavier (2015) introduced a swap operator to exchange riders among matched vehicles, the model is more suitable for drivers having fixed itineraries and time windows. In the problem context of SRAMODS, where robotaxis operate 24/7, the matching process needs to be more dynamic, allowing for the insertion of new riders at any time without waiting for a complete trip cycle.

Dynamic matching decomposes time into several intervals (epochs), seeking to optimize the matching within each stage. Hyland and Mahmassani (2020) propose a more dynamic model where vehicles’ locations are tracked every second. The matching process is optimized in every epoch, around 15 s, assigning the vehicles to riders and updating the status of vehicles. The authors further incorporated the unpredictability of potential delays from travelers. Similarly, Yu and Shen (2020) adopted approximate dynamic programming (ADP), for which they showed properties of the approximated value function at each stage.

Although the shared-ride service and related ride-sharing issues have been researched, solutions that simultaneously address the needs of in-vehicle riders, on-site riders, and operators require more exploration. Current TNCs often prioritize distance savings when matching riders, aiming to attract riders and drivers with cost savings from longer trip distances (Rinkus, 2014). However, this approach neglects riders’ time windows, potentially leading to longer waiting times and travel durations. Dandl et al. (2021) proposed a tri-level model considering travelers, MoD providers, and authorities, whereas the model focused on travelers’ willingness to choose transportation modes and simulated the shared-ride solutions instead of optimizing shared-ride matches.

While existing research has addressed various facets of shared-ride systems, significant gaps remain in addressing the dynamic nature of robotaxi services and holistic objective exploration. The following section outlines how this study contributes to bridging these gaps through a novel multi-objective model.

2.3. Study contribution and innovation

To address the gap in considering multiple aspects of robotaxi services, this study proposes a multi-objective model that incorporates the interests of in-vehicle riders, on-site riders, and SRAMODS operators. Building on the framework of Hyland and Mahmassani (2020), where matching is solved in a rolling manner over each epoch, this project considers multiple objectives to explore various solutions and expands the matching capacity from 2 riders to 4 riders.

This study distinguishes itself in several key areas:

(1) **Multi-objective Optimization Framework with Adaptive Weight Tuning:** Unlike previous models focused on single metrics such as travel time or distance (Horn, 2002; Lu et al., 2022), this study addresses the integration of multiple objectives, combining waiting time, travel duration, and detour distance. The model enables flexible weight adjustments to align with operational goals and accommodate the diverse needs of in-vehicle riders, on-site riders, and fleet operators. We evaluate the impact of different weight settings on system performance, and the results highlight the importance of adaptive tuning in achieving flexible and optimized outcomes.

(2) **Trip-Based Rider Adding:** The proposed model implements a real-time, dynamic rider-inserting process that evaluates vehicle capacity at the beginning of matching intervals. Unlike traditional methods that assign riders to vehicles until reaching maximum capacity (Alonso-Mora et al., 2017; Sun et al., 2020), this approach enables incremental rider assignments based on evolving trip conditions. As illustrated in Fig. 2, our trip-based approach adds new riders to existing trips so that the vehicles can follow a continuous sequence of pick-ups and drop-offs. This approach can reduce idle time and enhance capacity utilization compared to a fixed-group assignment strategy.

(3) **Performance Validation:** The model is validated using real-world data to assess performance improvements resulting from shared rides, and operational benefits are quantified in comparison to human-driven systems, underscoring the efficiencies of a centrally managed robotaxi fleet.

3. Methodology

3.1. Problem Statement

The SRAMODS model introduces a structured, centrally-controlled approach to robotaxi ride-matching in urban environments. Unlike traditional, “anarchic” ride-matching models such as those used by conventional taxi systems—where drivers often operate independently and may select rides based on personal convenience—this model utilizes

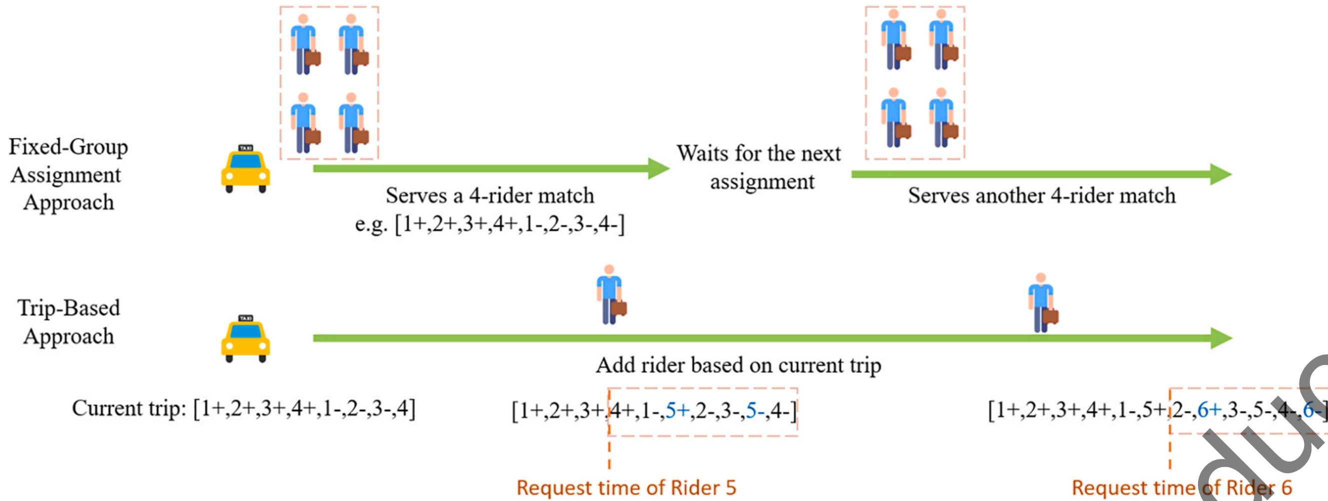


Fig. 2. Illustration of the trip-based rider-adding approach.

a demand-responsive, interval-based matching system. This centralized control minimizes the inefficiencies that arise when individual drivers prioritize short-term personal gains over optimal service distribution.

The objective of this research is to optimize rider matching considering on-site waiting time, in-vehicle duration, and detour distance, developing a system that caters to the requirements of both robotaxi operators and users. Following the current ride-hailing operation, where users will be more likely paired with neighboring vehicles, we develop a vehicle searching algorithm to capture those vehicles around the request. Thereafter, for each rider-robotaxi match, a list of candidate routes will be built considering capacity constraints, and an optimization model will be applied to identify the best route for each match. Eventually, a bipartite graph standing for all potential matches within an interval can be generated to optimize the robotaxi assignment. Following this, the route and arrival schedule of the robotaxi are updated, and the process is repeated for the subsequent interval. The overall procedure can be summarized in Fig. 3.

The SRAMODS model processes requests at defined intervals (e.g., every minute), enabling the controller to assess all pending requests and allocate robotaxis based on real-time conditions. As presented in Fig. 4, rider matching for pick-ups scheduled within interval t is conducted at the start of that interval, allowing seamless implementation of matching decisions within interval t. Likewise, matching for riders with pick-up times in interval $t+1$ is executed at the beginning of that interval. Riders who do not request one interval ahead of their pick-up time will have their matching deferred to the next interval.

3.2. Assumptions

The assumptions adopted for the problem formulation are listed as follows:

(1) Centralized Control over Fleet: The robotaxis fleet is of a fixed size, with a centralized assignment center having full operational control over all robotaxis. This control helps mitigate the "anarchy" associated with decentralized systems, where individual vehicles might act in self-interest, potentially prioritizing convenience over system efficiency. By centralizing the decision-making, the model enhances collective efficiency by regulating assignments to better serve overall system goals.
(2) Speed Assumption: The speed of the robotaxis is assumed to be constant along the route for all of the robotaxis.

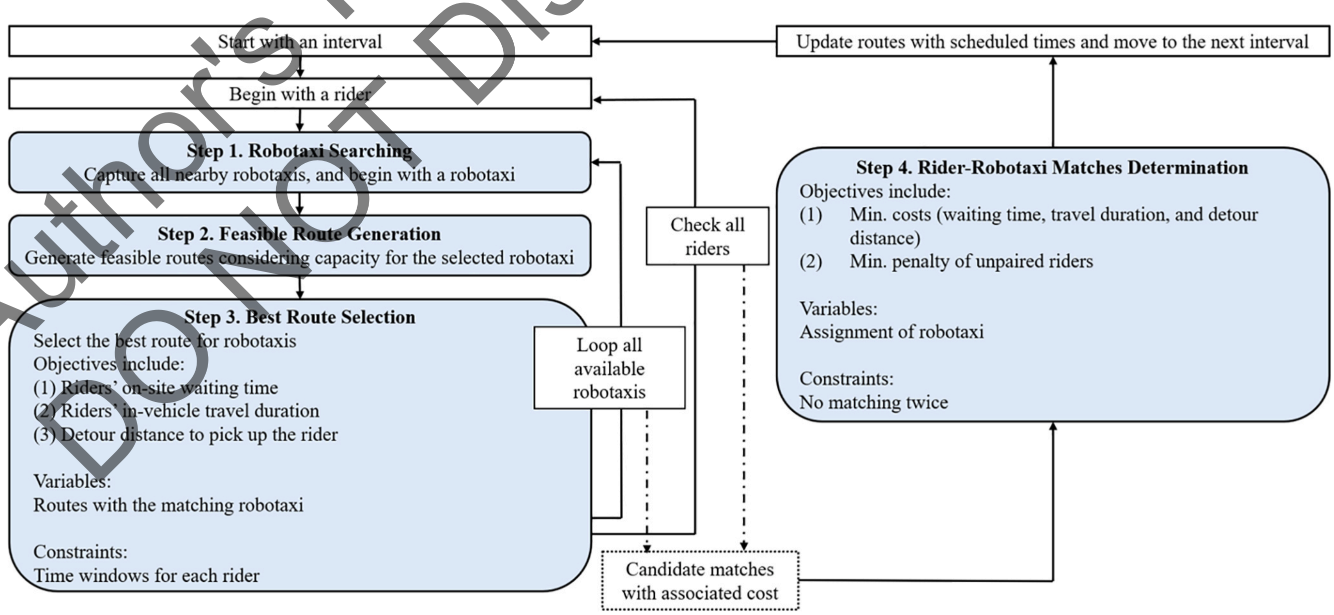


Fig. 3. Summary of the proposed SRAMODS model.

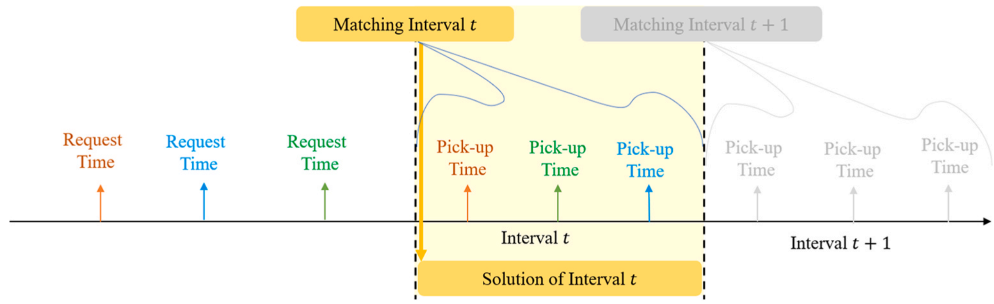


Fig. 4. The timeline of the matching process.

(3) Known Request Parameters: The information of riders' expected pickup time, drop-off time, locations, and detour tolerance are contained in a request, allowing for more accurate scheduling. In practice, if the detour tolerance is missing, it can be inferred from historical data to ensure a reasonable shared trip distance.
(4) Capacity Maximization: Robotaxis can accommodate up to four riders simultaneously, and all riders are assumed to be open to sharing rides. This capacity limit is designed to optimize vehicle usage, enabling a greater number of shared rides and thus improving profitability for stakeholders.
(5) Alternative Transit Assumption: If riders cannot be paired with a robotaxi, they are assumed to seek alternative transit options. This reflects real-world dynamics where unmet riders would look for other modes of transportation.
(6) Distance Calculation Assumption: The Manhattan distance is used to calculate the travel distance between points. In practice, local network configurations could be used to fine-tune distance accuracy.

3.3. Matching procedure and problem Formulations

The proposed SRAMODS model aims to optimize rider matching by considering on-site waiting time, in-vehicle travel duration, and detour distance, creating a system that meets the needs of both robotaxi operators and users. The model consists of four key steps: (1) vehicle searching, (2) feasible route generation, (3) best route selection, and (4) rider-robotaxi matching optimization.

Step 1: Vehicle Searching.

The vehicle searching algorithm is designed to identify and capture available nearby vehicles at a specific request time, as matching a new request with every vehicle can be highly inefficient. The concept centers on evaluating the route of a robotaxi spatially and temporally. If there are segments aligning with the time window of a new request and located within a certain proximity of the request's origin, then the robotaxi is considered available. To illustrate, in Fig. 5, three robotaxis are in the vicinity of the new request. Robotaxi 1, which is on call within the search area of the new request, will be matched with the request. Robotaxi 2 enters the search area, but the new request's time window has elapsed; thus, it will not be taken into account. Robotaxi 3 passes by the search area of the new request within its time window; therefore, it will be considered. The vehicle searching concept is outlined in **Algorithm 1**. In the algorithm, we term "point" to indicate the sequential placement in the existing route where the new rider can be inserted.

The design of the search area is similar to the existing ride-hailing service but with the addition of projecting future routes. A decent search area not only reduces the complexity of the problem but also eliminates potential matches that involve long distances.

Algorithm 1. Vehicle Searching

Input: The route of a robotaxi, the location and time windows of the new request
Output: Status of the robotaxi
points_within_timewindows = *GetPoints*(robotaxi route, new request's time windows)
  // Identify the segments of the robotaxi's route that falls within the new request's time windows
Status = False
for point ∈ point_within_timewindows **do** *// Check if any of the robotaxi's visiting points spatially fall within the search area.*
  if InSearchingArea(point, new request's location) == True **then**
    Status = True
    break
  end if
return Status

Step 2: Route Generation

Given the selected robotaxis from Step 1, feasible routes considering capacity will be generated in this step. By comparing the new requests' time windows, the inserting positions can be specified. However, as the taxi has limited capacity, the routes exceeding the capacity will be excluded.

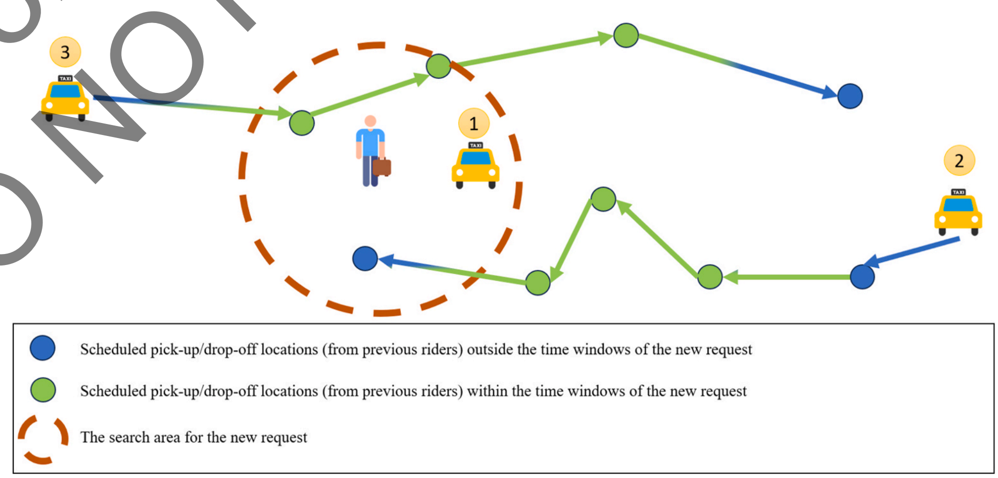


Fig. 5. The search area of a new request.

This study develops a route generation algorithm to enumerate all potential routes, as outlined in **Algorithm 2**. For a matched robotaxi, the algorithm identifies trip segments within the new rider's time window. The first available position is set as the pick-up point, and the algorithm iterates through subsequent positions as potential drop-off points. At each drop-off position, it checks for capacity violations—if the capacity is exceeded, all later drop-off points are disregarded. The process then repeats with the next pick-up position.

As illustrated in Fig. 6, ideally, the robotaxi can drop off the new rider at any position within the time window. However, as there are riders already assigned to this robotaxi, we have to exclude the sequence that may not suffice to accommodate the scheduled rider. After enumerating all potential routes, considering the efficiency, only the first top k routes with minimum detour distance will be chosen for further matching investigation in Step 3.

Algorithm 2. Route Generation

Input: The route of a robotaxi, the location and time windows of the new request
Output: A new route
pos_{start}, pos_{end} = *GetPosition*(original route, new request' time windows)
$pos_{pickup} = pos_{start}$
new_route_list = empty list
while $pos_{pickup} < pos_{end}$ **do** *// Pick-up point begins from the start until the end positions*
  $pos_{dropoff} = pos_{pickup} + 1$
  while $pos_{dropoff} < pos_{end}$ **do**
    newroute = Insert(original route, pos_{pickup}, $pos_{dropoff}$)
    if ViolateMaxCapacity(new_route)==False **then** *// check capacity*
      add new_route to new_route_list
    else
      break *//After this point, the capacity will be violated*
    $pos_{dropoff}$ += 1*// Move to the next drop-off position*
    end if
  pos_{pickup} += 1*// Move to the next pick-up position*
new_route_list = TopKRoutes(new_route_list) *//Get the first few routes with the shortest distance*
return new_route_list

Step 3: Best Route Selection and Cost Function

Given candidate routes for each rider-robotaxi pair, this section aims to select the best route with the minimum cost, where the cost function considers on-site waiting time, in-vehicle duration, and detour distance. The index r represents the candidate route for the request with a robotaxi, $r \in R$. The set I_r signifies all visiting points along route r, which includes riders' pick-up point subset I_r^+ and drop-off point subset I_r^-. The set definition is illustrated in Fig. 7. Let X_r be a binary decision variable whose value equals 1 if a route is selected, and 0 otherwise. Time-related continuous variables (at_i, wt_i, tt_i) denote arrival time, waiting time, and total travel duration for each rider. The relevant notations and descriptions are organized in Table 1, and the route selection model is formulated as follows.

Objective function:

$$minZ_{overallcost} = \sum_{r\in R}\left(\alpha\sum_{i\in I_r^+} wt_i + \beta\sum_{i\in I_r^+} tt_i + \gamma D_r\right)X_r \quad (1)$$

Subject to:

$$at_i \le et_i + M(1 - X_r) \quad i \in I_r^-,\ r \in R \quad (2)$$

$$wt_i \ge at_i - st_i \quad i \in I_r^+,\ r \in R \quad (3)$$

$$at_{i+1} \ge at_i + d_{i,i+1}/V \quad i \in I_r \backslash \{lastpoint\},\ r \in R \quad (4\text{-}1)$$

$$at_{i+1} \ge st_i + d_{i,i+1}/V \quad i \in I_r^+,\ r \in R \quad (4\text{-}2)$$

$$tt_i \ge at_j - at_i - M(1 - Z_i) \quad i = j,\ i \in I_r^+,\ j \in I_r^-,\ r \in R \quad (5\text{-}1)$$

$$tt_i \ge at_j - st_i - MZ_i \quad i = j,\ i \in I_r^+,\ j \in I_r^-,\ r \in R \quad (5\text{-}2)$$

$$\sum_{r\in R} X_r = 1 \quad (6)$$

$$at_i \ge 0 \quad i \in I_r,\ r \in R \quad (7)$$

$$wt_i \ge 0 \quad i \in I_r^+,\ r \in R \quad (8)$$

$$tt_i \ge 0 \quad i \in I_r^+,\ r \in R \quad (9)$$

$$X_r \in \{0, 1\} \quad r \in R \quad (10)$$

$$Z_i \in \{0, 1\} \quad i \in I_r^+,\ r \in R \quad (11)$$

The objective function, Eq. (1), seeks to select the route for a rider-robotaxi pair that minimizes the riders' travel duration, on-site waiting time, and detour distance of robotaxis to pick up the new rider. The practitioners can adjust the weights of each objective.

Constraint (2) is to ensure that the scheduled drop-off time is prior to the end time of the rider's time windows. M is introduced to invalidate the constraint for those unselected routes. Constraint (3) calculates the on-site waiting time of rider i, $i \in I_r^+$, and Constraint (7) ensures the non-negativity of the waiting time. Constraints (4-1) and (4-2) update the arrival time at each point around the route. The arrival time of the current time will be the arrival time at the previous node plus travel time, as displayed in Constraint (4-1). However, if the previous node is a pick-up point, belonging to set I_r^+, we have to further consider the actual departure time at the previous node, as the robotaxi may arrive earlier and wait until picking him/her up, which is handled by Constraint (4-2).

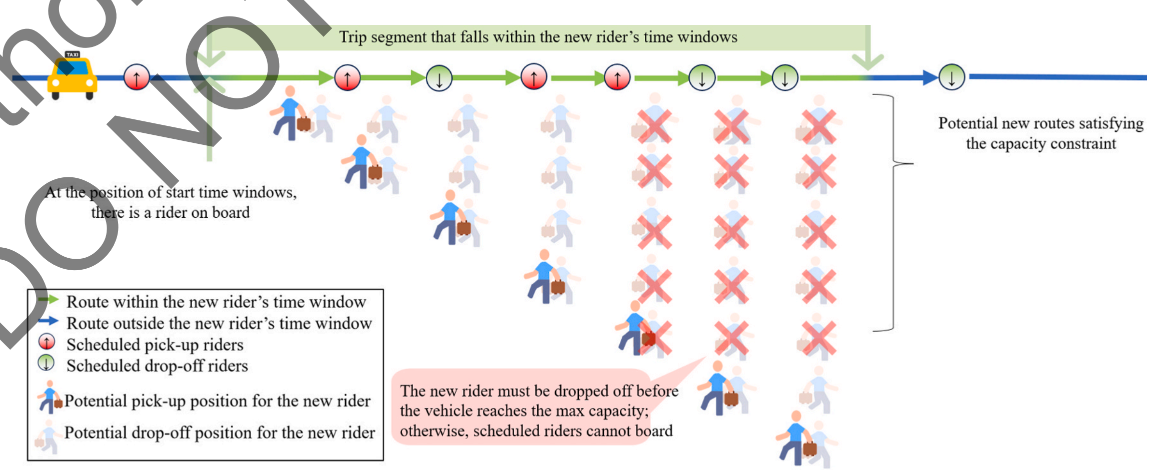


Fig. 6. Illustration for feasible route generation process.

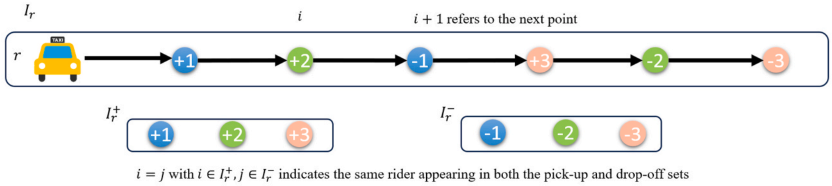


Fig. 7. Illustration of visiting point sets.

Table 1
Terminology for route selection and cost function.

Sets	
R	The set of routes for a rider-robotaxi pair generated in Step 2, indexed by r.
I_r	The set of riders' pick-up, drop-off, and the robotaxi's initial positions for a route r, indexed by i.
I_r^+	The set of riders' pick-up points for route r, indexed by i.
I_r^-	The set of riders' drop-off points for route r, indexed by i.
Variables	
X_r	A binary variable, which is 1 if the route r is selected; otherwise, 0.$r \in R$
Z_i	An indicator on pick-up nodes, representing whether the rider waits or the robotaxi waits, $i \in I_r^+$ and $r \in R$.
at_i	Arrival time at the rider's pick-up/ drop-off point, $i \in I_r^+$ refers to the pick-up point and $i \in I_r^-$ refers to the drop-off point.
wt_i	Waiting time of rider i, $i \in I_r^+$ and $r \in R$.
tt_i	In-vehicle travel duration of rider i of route r, $i \in I_r^+$ and $r \in R$.
Parameters	
α, β, γ	Weights of on-site waiting time, in-vehicle travel duration, and detour distance.
et_i	End time of rider i's time windows, $i \in I_r^-$ and $r \in R$.
st_i	Start time of rider i's time windows, $i \in I_r^+$ and $r \in R$.
$d_{i,i+1}$	The distance from point i towards $i+1$, $i \in I_r \backslash \{lastpoint\}$ and $r \in R$.
D_r	The marginal increase in distance of route r while inserting a new rider, $r \in R$.
V	The constant speed of robotaxis along the route.
M	A relatively big number to invalidate the constraints.

Constraints (5-1) and (5-2) are used to calculate the total riding time for each rider. The binary variable, Z_i, indicates whether the rider waits or not. Eq. (6) ensures a route of the examined robotaxi will be selected for the new request (unless the optimization is infeasible due to time window violations). Constraints (7)–(9) enforce the non-negativity of time-related variables, and Constraints (10) and (11) ensure the binary nature of variables X_r and Z_i.

The optimization model will suggest the best route with the total cost, which will serve as the link cost in the rider-robotaxi matching graph in Step 4. While evaluating arrival times along the route, the formulation accounts for all visiting nodes I_r in route r, including the robotaxi's initial position as well as riders' pick-up and drop-off points. Since the trip sequence continuously extends, we can treat the arrival times at nodes visited before the current time (epoch time) as constants, based on the previously determined route, to reduce the computational burden.

Step 4: Rider-Robotaxi Matching Optimization

The optimization model introduced in Step 3 returns the best route for a certain rider-robotaxi match with the associated cost. By considering all the riders within an interval, a bipartite graph representing rider-robotaxi matching can be generated, where the weights on links stand for the cost, as presented in Fig. 8. When the links are determined, each robotaxi's route and arrival schedule will be updated based on the route scheme from Step 3.

Let i be the index of riders in a matching interval, belonging to the rider set N, and v be the index of captured robotaxis, belonging to the vehicle set V. The matching of (i, v), denoted by l, belongs to the link set L.

A binary variable, Y_l, indicates whether the link is selected, $l \in L$. Parameter, c_l, represents the overall cost associated with time and distance for the link l, $l \in L$, and a constant p is employed to penalize unpaired riders.

Keep in mind that not all riders and robotaxis are linked in the bipartite graph, as Step 3 may indicate infeasibility for certain matches, or the robotaxis may be outside the search area for certain requests.

Objective function:

$$minZ_{cost} = \sum_{l \in L} c_l Y_l + \sum_{i \in N} p \left(1 - \sum_{\substack{l \in L^i \\ L^i: \text{links connecting to rider } i}} Y_l \right) \tag{12}$$

Subject to:

$$\sum_{\substack{l \in L^i \\ L^i: \text{links connecting to rider } i}} Y_l \leq 1 \quad i \in N \tag{13}$$

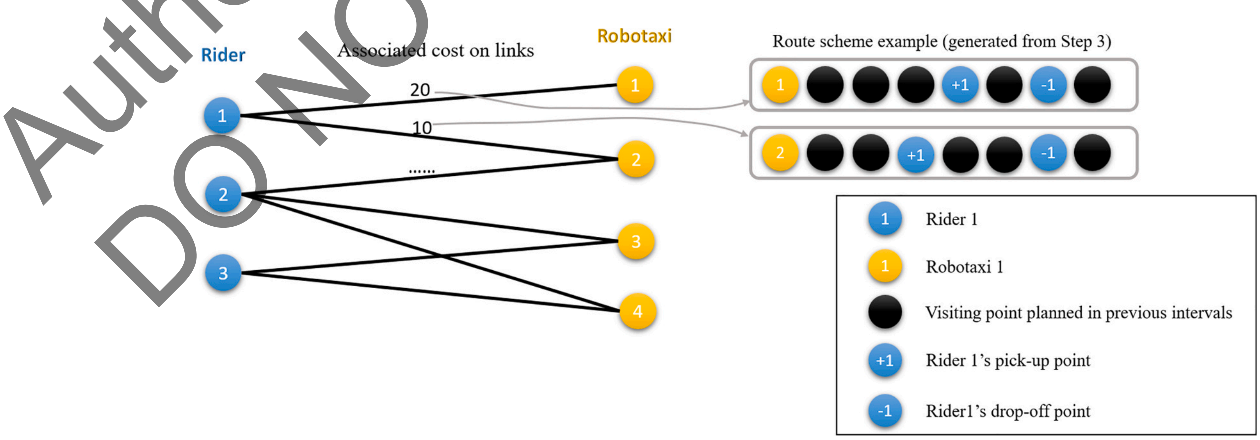


Fig. 8. A bipartite graph of rider-robotaxi matching within an interval.

$$\sum_{\substack{l \in L^v \\ L^v:\text{links connecting to robotaxi } v}} Y_l \leq 1 \quad v \in V \tag{14}$$

$$Y_l \in \{0,1\} \quad l \in L \tag{15}$$

The objective function (12) is to choose routes with minimal costs. The penalty cost term must be included; otherwise, the model will generate solutions where no links are selected to achieve the minimum cost. By introducing the penalty cost, the model can produce a solution that prioritizes pairing as many riders as possible while minimizing associated costs.

Eqs. (13) and (14) guarantee that each rider and vehicle are paired no more than once. Eq. (15) establishes the binary nature of the variable. Leveraging the total unimodularity of the constraint matrix, the assignment problem formulation allows for the relaxation of the integer program into a linear program.

3.4. Summary

This study presents a centralized ride-matching framework for SRAMODS, optimizing shared mobility by dynamically inserting new requests into ongoing trips of robotaxis. The model enhances efficiency through adaptive multi-objective optimization, balancing the benefits of both riders and operators. This study does not consider the need for charging. If a robotaxi's battery level drops too low, the centralized controller can simulate a "fake" ride request, directing the vehicle to the nearest charging station.

4. Case study

4.1. Data Description

The proposed SRAMODS model is tested using weekday taxi data from Chicago, Illinois (City of Chicago, 2024). Chicago's high urban density, diverse transportation choices, and varied travel behaviors make it an ideal city to evaluate SRAMODS, reflecting the real-world complexities that are essential for assessing the model's applicability (Zhou et al., 2019). For this case study, we examine daily trip volumes from November 5 to December 30, 2023, seeing higher trip volumes on weekdays compared to weekends, with a peak on Tuesday, as shown in Fig. 9 (a). For testing, we select Thursday, November 9, and focus on the peak period from 2:00 to 3:00 PM, as illustrated in Fig. 9 (b).

Key input parameters are displayed in Table 2. A 5-km searching radius balances the reach for available robotaxis without excessive detours, while a 1-minute examining interval captures real-time demand fluctuations. The chosen maximum capacity of 4 seats reflects common shared-ride limits, optimizing vehicle use without compromising rider comfort. These choices align with practical ride-sharing considerations and enhance model applicability.

Table 2
Parameters for the case study.

Parameter	Step	Value	Units
Searching radius	Step 1	5	km
Max capacity	Step 2	4	riders/ robotaxi
Top ***k*** route	Step 2	3	–
Weights (α, β, γ)	Step 3	(0.4, 0.3, 0.3)	–
Examining time interval	Step 4	1	minute
Robotaxi speed, $\boldsymbol{V}$	Step 4	25	km/hr
Fleet size	Step 4	100 (we also test 50,150, 200)	vehicles
Requests	Step 4	1,416	travelers

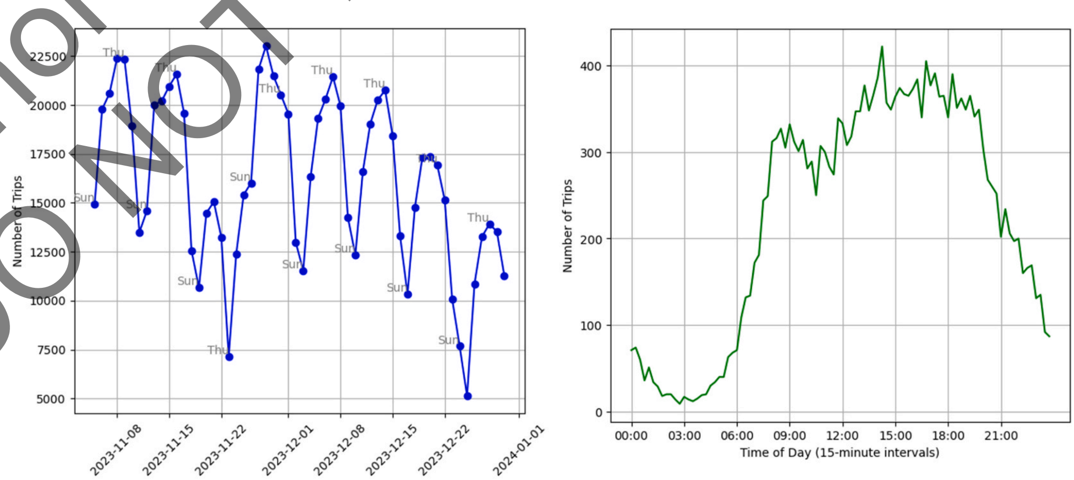
The pairing procedure is computed via the Python interface, and the commercial solver (Gurobi 10.0.1) is adopted to solve the binary integer programming using a laptop with an Intel Core i7-1255U and 16 GB DDR4-3200. The proposed SRAMODS model with one-hour taxi data can be solved within 10 min.

4.2. Performance given different weights of Multi-Objectives

To examine the impact of weight distribution outlined in Eq. (1), Step 3, this section explores all potential scenarios from weights (1,0,0), (0.9,0.1,0), (0.9,0,0.1), (0.8,0.2,0), and so forth to (0,0,1), where the sum of weights equals 1.0. Fig. 10 presents the average on-site waiting time, in-vehicle travel duration, and distance for each weight scenario given 100 robotaxis. In the chart, orange, green, and pink colors represent the scenarios where one objective is set as 0 to observe the relationship of the other two.

Fig. 11 examines the trade-offs between pairs of objectives from different perspectives: Subfigure (a) shows that prioritizing distance minimization increases travel duration, as the model may suggest shorter routes that require onboard riders to wait longer for new passengers to join the trip. Subfigure (b) reveals that reducing waiting time has little effect on travel distance, but neglecting it may lead to significant delays. Subfigure (c) indicates an inverse relationship between waiting time and travel duration—minimizing wait times often results in longer onboard travel duration due to suboptimal routing.

To better observe the domination between scenarios, Fig. 12 presents the Pareto front from multiple perspectives. Subfigure (a) shows an



(a) Daily taxi trip volume (Nov 5 – Dec 30, 2023) (b) 15-minute-interval taxi trip volume on Nov 9, 2023

Fig. 9. Daily taxi trip volume (Nov 5 – Dec 30, 2023) and 15-minute interval volume (Nov 9).

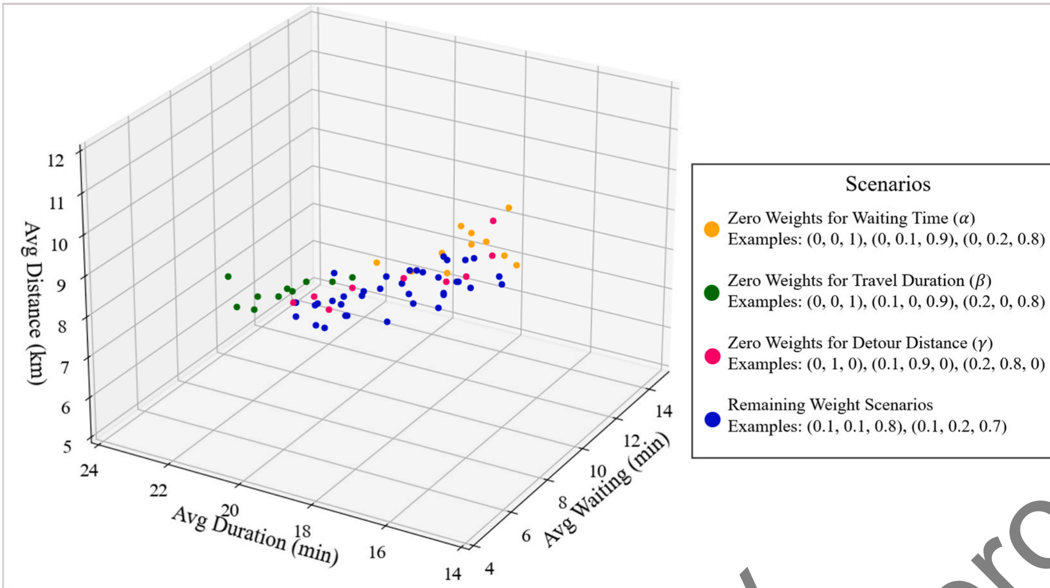


Fig. 10. Impact of different weightings on waiting time, travel duration, and detour distance.

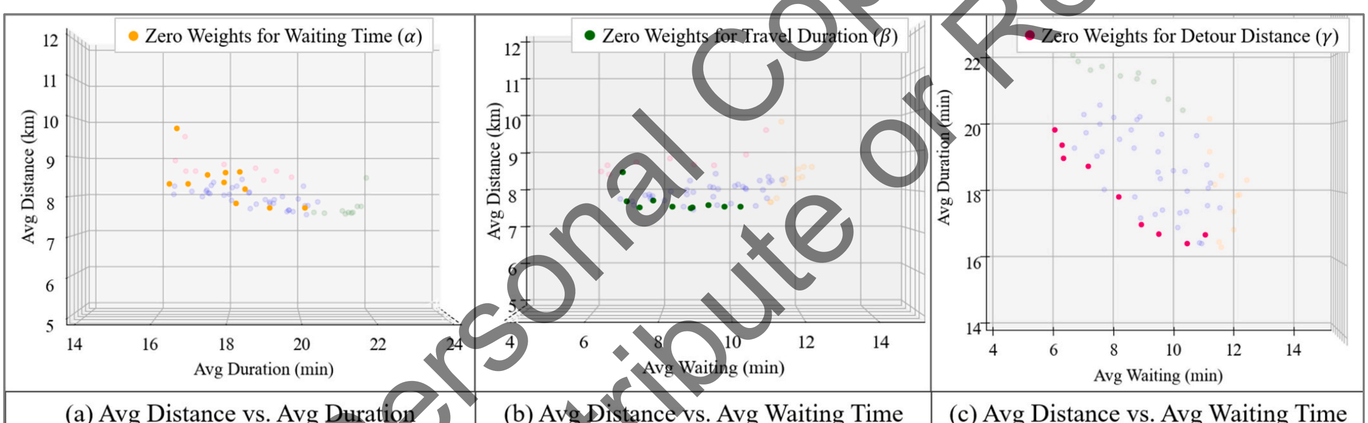


Fig. 11. Trade-offs between objectives from different perspectives.

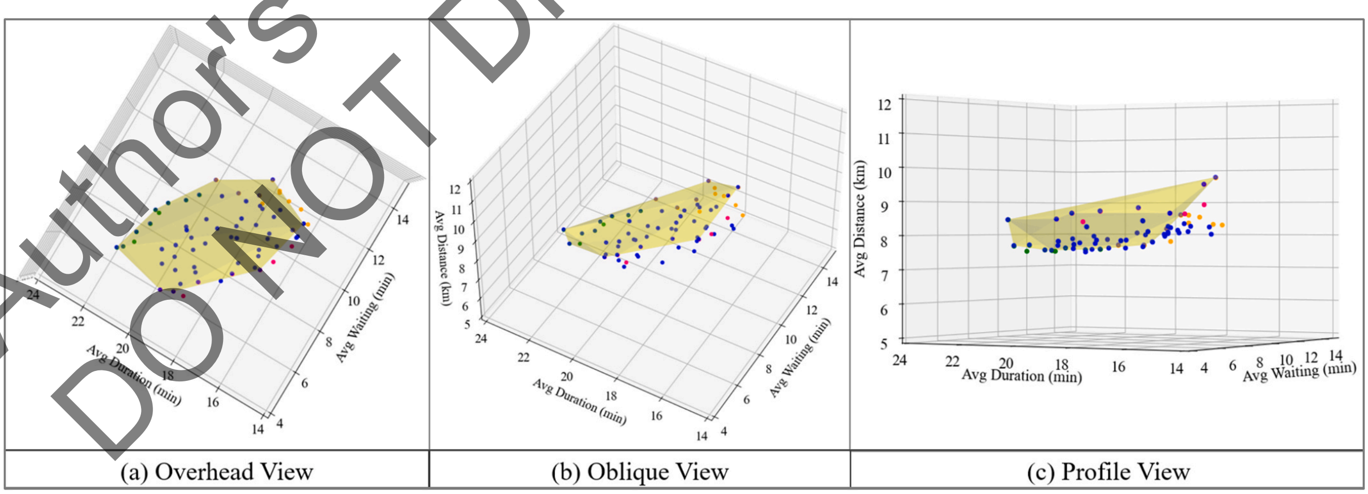


Fig. 12. Pareto front visualization in scenario testing.

overhead view, highlighting a circular Pareto boundary that encloses the majority of scenarios. Subfigures (b) and (c) reveal a bowl-shaped Pareto front in the side view. Overall, no solution dominates in terms of the on-site waiting time, in-vehicle duration, and travel distance. Instead, the Pareto front underscores the trade-offs between objectives, allowing practitioners to tailor configurations to specific operational priorities.

To further investigate the impact of various weight configurations on riders' on-site waiting time and in-vehicle duration, Fig. 13 examines two scenarios where the detour distance weight (γ) is set to 0 and 0.2. In both subfigures, the x-axis shows varying allocations of waiting time weight (α). Subfigure (b) extends only to $\alpha = 0.8$ because 0.2 of the total weight is reserved for detour distance (γ). When $\alpha = 0$, the average waiting time is around 11 min in both cases. As α increases, waiting times decrease, but travel duration rises. The total time spent (sum of waiting and travel durations) is minimized when waiting time and duration weights (α and β) are evenly distributed. Additionally, as γ increases from 0 to 0.2, the total time spent increases because more priority is given to reducing detour distance, leaving less focus on minimizing on-site waiting time and in-vehicle duration. These observations stress that there isn't an ideal weight configuration that optimizes both waiting and travel times simultaneously.

4.3. Significance of 2 + Riders

This study proposes a dynamic trip-based rider insertion to add riders to an ongoing trip, efficiently utilizing the vehicle capacity. Building on the proposed dynamic trip-based rider insertion approach, this section evaluates the impact of increasing vehicle capacity, using a weight distribution of (0.4, 0.3, 0.3). Four vehicle fleet size scenarios are considered, as shown in Fig. 14. A maximum capacity of 1 reflects the concept of no shared ride.

Fig. 14 (a) demonstrates that shared-ride services reduce the riders' average waiting time by increasing opportunities for riders to join ongoing trips. While shared rides with more passengers may extend in-vehicle travel time due to route deviations, this increase can be offset by reduced waiting time. Fig. 14 (b) shows that the average distance decreases with higher capacity because it provides greater flexibility to reduce unnecessary detours and improve overall efficiency. Overall, increasing both vehicle capacity and fleet size can reduce total time spent and travel distance.

Fig. 14 (c) illustrates how increasing capacity accommodates more travelers across all four fleet size scenarios. However, as fleet size grows, the marginal reduction in unpaired riders decreases. For example, the decline from 150 to 200 vehicles is smaller compared to the reduction from 50 to 100 vehicles. This suggests that while expanding fleet size improves ride-matching efficiency, the benefits diminish beyond a certain point, highlighting the importance of optimizing both capacity and fleet size for maximum system efficiency.

4.4. Price of anarchy reduction

This section aims to compare robotaxis and human drivers in shared-ride services. UberX Share and Lyft Shared had both disclosed that their rider matching principle primarily focuses on minimum total matched distance (Rinkus, 2014; Timothy Brown, 2016). This study models human drivers' decisions by using maximum distance savings as the matching criterion, as illustrated in Fig. 15. The "anarchic" nature of human drivers is implemented by preventing blocking pairs (Unmatched riders and drivers who would prefer each other over their current assignments) (Wang et al., 2018).

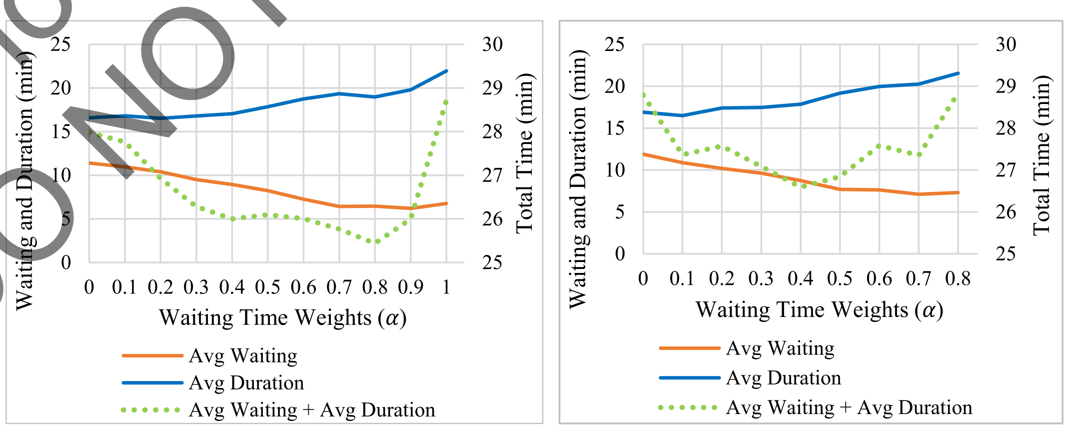
We consider the following three cases:

(a) Human Drivers (Benchmark): Self-interested human drivers emphasize individual distance savings.
(b) Robotaxi Prioritizing Distance Minimization (Weights: 0, 0, 1): The proposed robotaxi model focuses solely on minimizing travel distance.
(c) Robotaxi with Balanced Priorities (Weights: 0.4, 0.3, 0.3): The proposed robotaxi model balances on-site waiting time, in-vehicle duration, and travel distance.

Fig. 16 compares the performance of these scenarios using Case (a) as the benchmark. Case (b), which prioritizes distance minimization, achieves a 15 % reduction in average distance but at the expense of increased on-site waiting time and in-vehicle duration. This result indicates that systematic matching can achieve better distance savings compared to self-interested drivers.

Case (c) achieves a 20 % reduction in waiting time and a 13 % reduction in distance. There is no significant difference in travel duration. This result is reasonable since cases (a) and (b) do not explicitly optimize travel duration. In case (c), the reduction in on-site waiting time may cause some robotaxis to arrive earlier than the new riders' start time windows, resulting in slightly longer in-vehicle durations for onboard riders. However, these routes are still preferred because the benefits from reduced distance and waiting time outweigh the slight increase in in-vehicle duration. Consequently, the average in-vehicle duration remains comparable across all three cases.

The 15 % distance reduction achieved by centralized matching focusing distance (Case b) can be interpreted as a reduction in the "price of anarchy," where individual drivers' choices lead to suboptimal system-wide efficiency. Case (c) outperforms human-driven cases on all three objectives and performs only slightly worse than case (b) in terms



(a) $\gamma = 0$ (b) $\gamma = 0.2$

Fig. 13. Waiting time and in-vehicle duration under a fixed weight on detour distance.

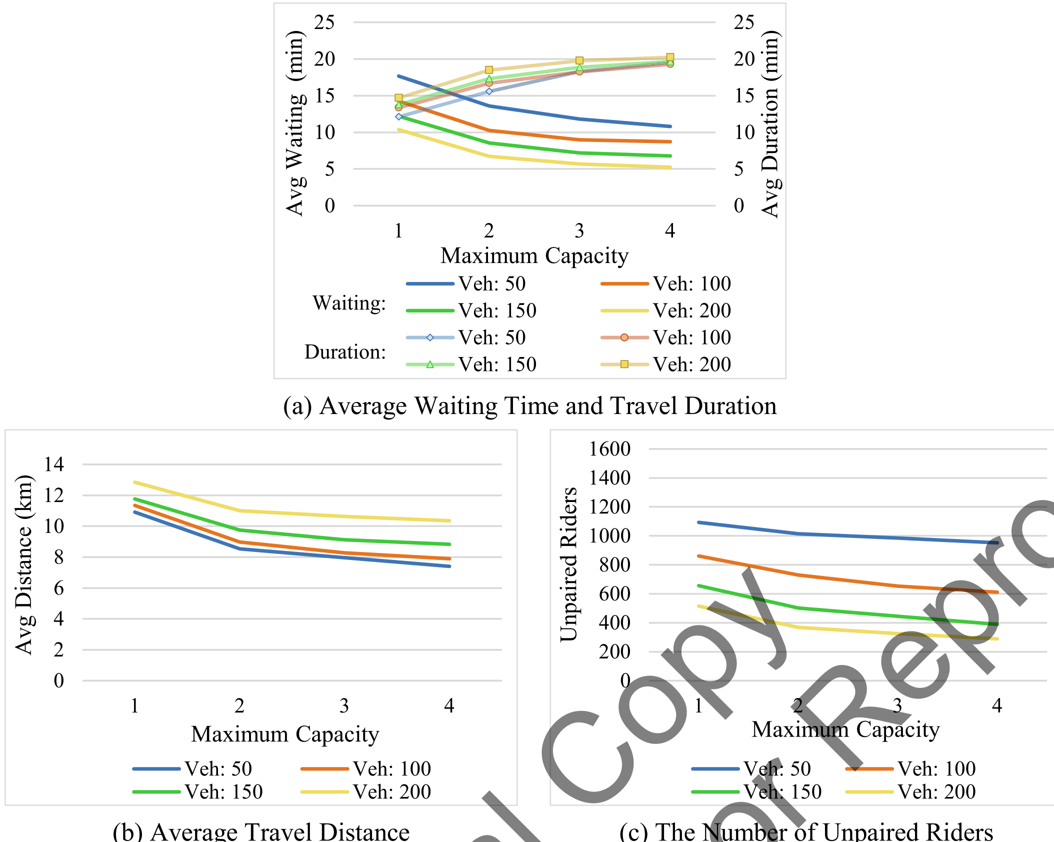


Fig. 14. Objectives under different maximum capacities and vehicle numbers.

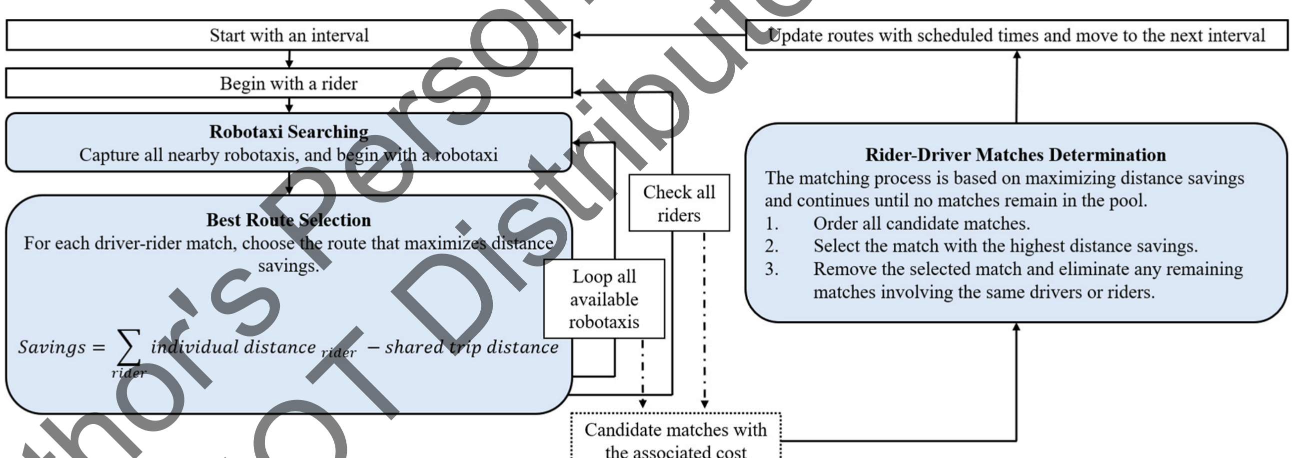


Fig. 15. The matching procedure designed for TNC taxi assignments.

of distance alone, highlighting the importance of a centralized system and multifaceted objective consideration.

4.5. *Policy recommendations*

Despite the potential to provide better service and reduce total distance, the application of shared-ride service in Robotaxi is still in its early stages. To encourage the growth and sustainability of shared-ride services, we propose a mix of quantitative and qualitative policy measures, aligned with our model outcomes and insights.

A. Quantitative measures:

(1) Regulate Riders' Minimum Satisfaction

Our findings suggest that placing all weight on distance minimization can undermine performance in terms of on-site waiting time and in-vehicle travel duration. Hence, while operators may aim to minimize total travel distance, regulations should ensure that waiting time or travel duration does not become excessive, safeguarding user satisfaction.

(2) Provide Financial Incentives for Operators

Sensitivity results indicate that increasing the share of pooled rides

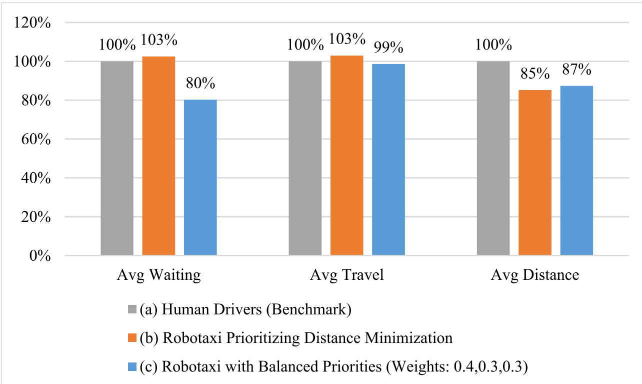


Fig. 16. Performance using human-driven vehicles as the benchmark.

significantly reduces total travel distance. Financial incentives—such as subsidies, tax credits, or HOV access—should target operators who utilize vehicle capacity efficiently, encouraging wider adoption of shared services.

(3) Ensure Profit Returns for Riders

As higher occupancy leads to more efficient distance savings, governments can subsidize the travel costs of riders who are willing to share rides with multiple passengers. This not only promotes system efficiency but also maintains affordability for users.

B. Qualitative includes:
(1) Ensure Fair and Transparent Rider Compensation

While shared rides may involve longer trip times, cost savings remain a key benefit. The fare structure and cost-saving policy should be monitored to ensure fairness and transparency.

(2) Educate the Public on Environmental and Economic Benefits

The government can propagate model-based evidence to raise awareness of how shared rides reduce congestion and emissions. Highlighting individual savings may also encourage travelers to shift from private cars.

(3) Develop Safety Standards

As riders have to share the vehicle with others, it is important to create a safe and reliable environment. The company can install surveillance and alert systems in response to any emergencies.

By implementing these policies, governments can support a transportation ecosystem that maximizes the benefits of robotaxi services while addressing challenges such as urban congestion and environmental impact, thereby achieving the Sustainable Development Goals (SDGs) in the city.

5. Conclusion

With the increasing popularity of robotaxis in many cities, proactive planning for shared-ride services is essential. This study develops a multi-objective model for scheduling SRAMODS, integrating the perspectives of on-site riders, in-vehicle riders, and robotaxi operators. By minimizing waiting time, travel duration, and detour distance, this study provides more efficient and holistic solutions. Our model leverages the centralization of robotaxi systems for system-wide optimization, which has the potential to mitigate the inefficiencies (or 'price of anarchy') caused by self-interested driver decisions. Moreover, unlike existing ride-sharing models that predefine match groups and assign vehicles, our approach uses dynamic trip-based insertion, ensuring continuous fleet adjustments to maximize service efficiency.

The process begins by identifying robotaxis near a new rider and generating feasible routes for these vehicles. Among these routes, the one with the minimum cost, considering waiting time, travel duration, and detour distance, is selected using a Mixed-Integer Programming (MIP) formulation. Finally, the matching between robotaxis and riders is determined based on the generated candidates using bipartite matching. Various scenarios, including weight distribution, maximum seat capacity, and the number of robotaxis, are analyzed. Key findings are summarized as follows:

- Emphasizing one objective improves its performance but compromises others.

The result indicates that no solution dominates in terms of waiting time, travel duration, and distance optimization. Instead, emphasizing a specific objective can improve its performance but compromise the others. However, evenly distributing weights on waiting time and travel duration leads to the minimum total time spent.

- Allowing more riders per vehicle enhances overall efficiency.

Increasing the maximum seat capacity enables better pairing, reducing the average travel distance per request and operational costs. Although individual travel time may slightly increase, shorter waiting times typically offset this effect, resulting in a total trip time comparable to non-sharing settings.

- Adding more robotaxis increases coverage but may raise costs.

A larger fleet accommodates more requests and improves access to service, but excessive fleet sizes can lead to diminishing returns and unnecessary operational costs. Efficient fleet sizing is essential to balance demand coverage with cost-effectiveness.

- Using centralized robotaxi control improves system performance.

The robotaxi service significantly improves system efficiency by reducing distances by up to 15 %. When a balanced weight is applied across objectives, improvements can be achieved in waiting time, travel time, and travel distance simultaneously. The centralized robotaxi control and systematic matching processes overcome the inefficiencies (the price of anarchy) that arise from the independent and self-interested decisions of human drivers.

Future investigations should quantify the economic benefits of reduced travel distances to encourage the adoption of shared mobility services. Further analysis is also needed to evaluate how different objective weight distributions perform in high-demand versus low-demand scenarios. Additionally, future studies can incorporate rider preferences, such as gender, safety concerns, and other demographic factors, which could significantly impact rider matching and travel behavior.

This study aims to draw policymakers' attention to the promotion of shared-ride services. Local governments play a critical role in integrating shared-ride services with autonomous vehicles. To support this, several recommendations are provided, including offering tax incentives to encourage participation and establishing clear guidelines for sharing networks. These measures can help encourage user participation and support the integration of shared-ride services into urban transport systems.

Author Contributions

The authors confirm their contribution to the paper as follows: study conception and design: M. Sun; data collection and processing: M. Sun;

analysis and interpretation of results: M. Sun; draft manuscript preparation: M. Sun, L. Quadrifoglio. The authors reviewed the results and approved the final version of the manuscript.

Funding sources

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

CRediT authorship contribution statement

Min-Ci Sun: Writing – original draft, Visualization, Validation, Software, Methodology, Data curation, Conceptualization. **Luca Quadrifoglio:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agatz, N., Erera, A., Savelsbergh, M., Wang, X., 2012. Optimization for dynamic ride-sharing: a review. Eur. J. Oper. Res. 223, 295–303. https://doi.org/10.1016/j.ejor.2012.05.028.

Alonso-González, M.J., Cats, O., van Oort, N., Hoogendoorn-Lanser, S., Hoogendoorn, S., 2021. What are the determinants of the willingness to share rides in pooled on-demand services? Transp. (Amst) 48, 1733–1765. https://doi.org/10.1007/s11116-020-10110-2.

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., Rus, D., 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. PNAS 114, 462–467. https://doi.org/10.1073/pnas.1611675114.

Berbeglia, G., Cordeau, J.F., Laporte, G., 2010. Dynamic pickup and delivery problems. Eur. J. Oper. Res. 202, 8–15. https://doi.org/10.1016/j.ejor.2009.04.024.

Castellanos, S., Grant-Muller, S., Wright, K., 2022. Technology, transport, and the sharing economy: towards a working taxonomy for shared mobility. Transp. Rev. 42, 318–336. https://doi.org/10.1080/01441647.2021.1968976.

Castiglione, J., Chang, T., Cooper, D., Hobson, J., Logan, W., Young, E., Charlton, B., Wilson, C., Mislove, A., Chen, L., Jiang, S., 2017. TNCs Today A Profile of San Francisco Transportation Network Company Activity.

Chan, N.D., Shaheen, S.A., 2012. Ridesharing in North America: past, present, and future. Transp. Rev. 32, 93–112. https://doi.org/10.1080/01441647.2011.621557.

City of Chicago, 2024. Taxi Trips (2013-2023)-Chicago Data Portal [WWW Document]. URL https://data.cityofchicago.org/Transportation/Taxi-Trips-2013-2023-/wrvz-psew/about_data (accessed 3.20.25).

Dandl, F., Engelhardt, R., Hyland, M., Tilg, G., Bogenberger, K., Mahmassani, H.S., 2021. Regulating mobility-on-demand services: tri-level model and Bayesian optimization solution approach. Transp. Res. C Emerg Technol. 125. https://doi.org/10.1016/j.trc.2021.103075.

Diao, M., Kong, H., Zhao, J., 2021. Impacts of transportation network companies on urban mobility. Nat Sustain 4, 494–500. https://doi.org/10.1038/s41893-020-00678-z.

Erhardt, G.D., Roy, S., Cooper, D., Sana, B., Chen, M., Castiglione, J., 2019. Do transportation network companies decrease or increase congestion?.

Foti, L., Lin, J., Wolfson, O., 2021. Optimum versus Nash-equilibrium in taxi ridesharing. GeoInformatica 25, 423–451. https://doi.org/10.1007/s10707-019-00379-6.

Foti, L., Lin, J., Wolfson, O., Rishe, N.D., 2017. The Nash Equilibrium Among Taxi Ridesharing Partners, in: GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems. Association for Computing Machinery. https://doi.org/10.1145/3139958.3140028.

Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.E., Wang, X., Koenig, S., 2013. Ridesharing: the state-of-the-art and future directions. Transp. Res. B Methodol. 57, 28–46. https://doi.org/10.1016/j.trb.2013.08.012.

Henao, A., Marshall, W.E., 2019. The impact of ride-hailing on vehicle miles traveled. Transp. (Amst) 46, 2173–2194. https://doi.org/10.1007/s11116-018-9923-2.

Horn, M.E.T., 2002. Fleet scheduling and dispatching for demand-responsive passenger services. Transp. Res. C Emerg Technol. 10, 35–63.

Hou, Y., Garikapati, V., Weigl, D., Henao, A., Moniot, M., Sperling, J., 2020. Factors influencing willingness to pool in ride-hailing trips. Transp. Res. Rec. 2674, 419–429. https://doi.org/10.1177/0361198120915886.

Hyland, M., Mahmassani, H.S., 2020. Operational benefits and challenges of shared-ride automated mobility-on-demand services. Transp Res Part A Policy Pract 134, 251–270. https://doi.org/10.1016/j.tra.2020.02.017.

Lu, W., Quadrifoglio, L., Lee, D., Zeng, X., 2022. The ridesharing problem without predetermined drivers and riders: formulation and heuristic. Transp. Lett. 00, 1–11. https://doi.org/10.1080/19427867.2022.2116674.

Mass.gov, 2019. Rideshare in Massachusetts.

Nahmias-Biran, B.H., Oke, J.B., Kumar, N., Basak, K., Araldo, A., Seshadri, R., Akkinepally, A., Azevedo, C.L., Ben-Akiva, M., 2019. From traditional to automated mobility on demand: a comprehensive framework for modeling on-demand services in simmobility. Transp. Res. Rec. 2673, 15–29. https://doi.org/10.1177/0361198119853553.

Psaraftis, H.N., 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transp. Sci. 14, 130–154.

Rinkus, S., 2014. How UberPool works [WWW Document]. Uber Blog. URL https://www.uber.com/blog/driving-with-uberpool/ (accessed 7.25.24).

Santos, D.O., Xavier, E.C., 2015. Taxi and ride sharing: a dynamic dial-a-ride problem with money as an incentive. Expert Syst. Appl. 42, 6728–6737. https://doi.org/10.1016/j.eswa.2015.04.060.

Shah, S., Lowalekar, M., Varakantham, P., 2020. Neural Approximate Dynamic Programming for On-Demand Ride-Pooling. The Thirty-Fourth AAAI Conference on Artificial Intelligence.

Shaheen, S., Cohen, A., 2019. Shared ride services in North America: definitions, impacts, and the future of pooling. Transp. Rev. https://doi.org/10.1080/01441647.2018.1497728.

Shen, W., Achar, R., Lopes, C.V., 2019. Toward understanding the impact of user participation in autonomous ridesharing systems. IEEE Trans. Antennas and Propagation.

Si, H., Shi, J., Hua, W., Cheng, L., De Vos, J., Li, W., 2023. What influences people to choose ridesharing? An overview of the literature. Transp. Rev. 43, 1211–1236. https://doi.org/10.1080/01441647.2023.2208290.

Sun, Y., Chen, Z.L., Zhang, L., 2020. Nonprofit peer-to-peer ridesharing optimization. Transp. Res. E Logist Transp. Rev. 142. https://doi.org/10.1016/j.tre.2020.102053.

Tafreshian, A., Masoud, N., 2020. Trip-based graph partitioning in dynamic ridesharing. Transp. Res. C Emerg Technol. 114, 532–553. https://doi.org/10.1016/j.trc.2020.02.008.

Tafreshian, A., Masoud, N., Yin, Y., 2020. Frontiers in service science: Ride matching for peer-to-peer ride sharing: a review and future directions. Serv. Sci. https://doi.org/10.1287/serv.2020.0258.

Timothy Brown, 2016. Matchmaking in Lyft Line [WWW Document]. Medium . URL https://eng.lyft.com/matchmaking-in-lyft-line-9c2635fe62c4 (accessed 7.24.24).

Wang, H., Yang, H., 2019. Ridesourcing systems: a framework and review. Transp. Res. B Methodol. 129, 122–155. https://doi.org/10.1016/j.trb.2019.07.009.

Wang, X., Agatz, N., Erera, A., 2018. Stable matching for dynamic ride-sharing systems. Transp. Sci. 52, 850–867. https://doi.org/10.1287/trsc.2017.0768.

Waymo, 2023. Waymo significantly outperforms comparable human benchmarks over 7 + million miles of rider-only driving [WWW Document]. URL https://waymo.com/blog/2023/12/waymo-significantly-outperforms-comparable-human-benchmarks-over-7-million/ (accessed 3.29.24).

Wolfson, O., Lin, J., 2017. Fairness versus optimality in ridesharing, in: Proceedings - 18th IEEE International Conference on Mobile Data Management, MDM 2017. Institute of Electrical and Electronics Engineers Inc., pp. 118–123. https://doi.org/10.1109/MDM.2017.25.

Yan, S., Chen, C.Y., Wu, C.C., 2012. Solution methods for the taxi pooling problem. Transp. (Amst) 39, 723–748. https://doi.org/10.1007/s11116-011-9354-9.

Young, M., Farber, S., Palm, M., 2020. The true cost of sharing: a detour penalty analysis between UberPool and UberX trips in Toronto. Transp. Res. D Transp. Environ. 87. https://doi.org/10.1016/j.trd.2020.102540.

Yu, X., Shen, S., 2020. An integrated decomposition and approximate dynamic programming approach for on-demand ride pooling. IEEE Trans. Intell. Transp. Syst. 21, 3811–3820. https://doi.org/10.1109/TITS.2019.2934423.

Zhou, X., Wang, M., Li, D., 2019. Bike-sharing or taxi? Modeling the choices of travel mode in Chicago using machine learning. J. Transp. Geogr. 79. https://doi.org/10.1016/j.jtrangeo.2019.102479.

Zoox, 2023a. Zoox on Public Roads Fact Sheet [WWW Document]. URL chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.datocms-assets.com/106048/1696542543-zoox-worlds-first-robotaxi-on-public-roads-fact-sheet.pdf (accessed 3.30.24).

Zoox, 2023b. Building a new kind of transportation [WWW Document]. URL chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.datocms-assets.com/106048/1696542919-zoox-backgrounder.pdf (accessed 3.30.24).