Transportation Research Part B 150 (2021) 227–259

Contents lists available at ScienceDirect

Transportation Research Part B

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

Proactive shuttle dispatching in large-scale dynamic dial-a-ride systems

Amirmahdi Tafreshian [a], Mojtaba Abdolmaleki [a], Neda Masoud [a,*], Huizhu Wang [b]

[a] Civil and Environmental Engineering, University of Michigan Ann Arbor, United States of America
[b] Ford Motor Company, Greater Detroit Area, United States of America

ARTICLE INFO

Keywords:
Proactive optimization
Shuttle dispatching
On-demand dial-a-ride problem
Large-scale optimization
Offline and online optimization framework

ABSTRACT

The problem of dispatching shuttles to serve trip requests can be mathematically formulated as a dial-a-ride problem (DARP). With on-demand mobility services gaining more popularity due to the recent developments in the gig economy, communication technologies, and urbanization, the real-time application of DARP is attracting ever more interest. However, the fact that the size of DARP grows exponentially with number of requests and number of available seats renders the current solution methodologies inadequate for online applications. In order to tackle this issue, we propose a general framework that shifts much of the computational burden of the optimization problems that need to be solved into an offline phase, thereby addressing on-demand requests with fast and high-quality solutions in real time. Using numerical experiments, we demonstrate the benefits of the proposed method. Furthermore, we conduct sensitivity analyses to show the performance of our methodology under different parameter settings.

1. Introduction

The problem of dispatching a fleet of shuttles to serve trip requests can be mathematically formulated as a dial-a-ride problem. Although well-studied, the dial-a-ride problem has been traditionally used for dispatching shuttles in para-transit systems, where trip requests are known well ahead of time. With on-demand shared-use mobility services gaining more popularity as a result of high cost of shuttle ownership and the prospect of upcoming driver-less fleet, the real-time application of the dial-a-ride problem is attracting more interest. The successful implementation of on-demand mobility services for large-scale systems requires the dial-a-ride problem to accommodate high-occupancy shuttles that enable pooling requests with similar itineraries. However, the fact that the computational complexity of this problem grows exponentially with number of requests and available seats, and the possibility of changing the itinerary of shuttles en-route in real-time render the current solution methodologies inadequate for online applications.

Typically, mobility services operate under one of two models: (1) having riders register their trip requests ahead of time. This would enable solving the routing and scheduling problems offline, and providing high quality solutions. However, such systems are not convenient for customers since they require in advance planning on their part; (2) serving on-demand trips by heuristically inserting online requests into existing shuttle routes. This approach is much more flexible and more appealing to the customers. However, the insertion heuristics are typically very simple (e.g., assigning a request to the closest shuttle), in the interest of reducing the wait time for the newly joined customer. As such, this approach fails to take into account system-wide quality-of-service measures. This research introduces a general framework to bridge this gap by providing near-optimal insertion methodologies using system-level information, in real time. As such, our framework shifts much of the computational burden of the optimization

* Corresponding author.
E-mail address: nmasoud@umich.edu (N. Masoud).

https://doi.org/10.1016/j.trb.2021.06.002
Received 19 March 2020; Received in revised form 29 March 2021; Accepted 4 June 2021
Available online 29 June 2021

problems that need to be solved into an offline phase, thereby addressing the on-demand requests with fast and high quality solutions. Furthermore, in order to improve the utilization rate of shuttles, we seek to dispatch our shuttles proactively, and not wait for the demand to be realized first. The framework, therefore, introduces a data-driven approach that starts with a high quality forecast of future trips based on historical data. Using numerical examples, we demonstrate the effectiveness of our methodology applied to a fairly large-scale problem.

The rest of this paper can be summarized as the following: In Section 2 we review the related work in static and dynamic dial-a-ride problem. The general structure and assumptions of the problem are introduced in Section 3. In Section 4, we first propose a data-driven approach for finding a pool of useful shuttle routes in an offline phase. Further, in this section, an online framework for dispatching shuttles in real-time is proposed. The result of a numerical experiment and some interesting aspects of our proposed model are discussed in Section 5. We conclude our findings and provide some suggestions for future research in Section 6.

2. Literature review

The Dial-a-ride problem (DARP) is a generalization of the pick-up and delivery vehicle routing problem (PDVRP), and vehicle routing problem with time windows (VRPTW). Compared to other types of routing problems, DARP is usually considered for transporting humans with tight time windows, and the capacity of vehicles are mostly binding (Cordeau and Laporte, 2007). The static version of this problem is a well-studied problem in combinatorial optimization (Healy and Moll, 1995). In static DARP, it is assumed that all requests have been realized prior to the routing and scheduling of the fleet. This problem was initially introduced for scheduling and transporting elderly and disable people.

In recent decades, with the advent of advanced communication technologies and computational tools, the dynamic form of DARP as well as its counterparts, such as ridesharing and ridesourcing, have received more attention (Masoud and Jayakrishnan, 2017b; Lloret-Batlle et al., 2017; Tafreshian et al., 2020; Masoud and Jayakrishnan, 2017c). The dynamic DARP has especially attracted immense attention in recent years due to the rise in popularity of multi-modal transportation, where vehicles can be used to complement public transportation in real-time (Yan et al., 2019; Regue et al., 2016; Masoud et al., 2017; Nam et al., 2018). Interest in efficient operational schemes for autonomous vehicle fleet that can provide door-to-door transportation has been another motivator for the recent revival of interest in the dynamic DARP (Hyland and Mahmassani, 2018; Zhang et al., 2020; Masoud and Jayakrishnan, 2016, 2017a).

Jaw et al. (1986) were among the first to consider a dynamic DARP with time constraints. They proposed a simple and efficient insertion heuristic that influenced the work of many others afterwards. Interestingly, they showed that considering a batch of customers to simultaneously insert in a route does not result in a large improvement, but it certainly comes at a much higher computational cost. Madsen et al. (1995) considered a DARP with multiple capacities and multiple objective functions for a project by the Copenhagen Fire-Fighting Service (CFFS). They proposed another insertion heuristic that proved to provide a solution in real time. Mitrović-Minić et al. (2004) devised double horizon-based insertion and improvement heuristics that account for both short-term and long-term decisions. Based on their experiments, considering the slack time in the distant future may result in decreasing the routing costs. Coslovich et al. (2006) developed a two-stage insertion heuristic based on route perturbations. In the offline phase this method creates a feasible neighborhood for the current route of a vehicle that is moving between 2 stops. Then, in real-time it applies a simple insertion rule to decide whether to insert the new request or not. Using numerical examples, they showed that the result is close to that of static problem. However, they relaxed the capacity constraint due to its low possibility of being violated. Xiang et al. (2008) proposed a flexible scheduling scheme that can handle different sources of uncertainty in a dynamic DARP. They devised a heuristic approach based on local search and introduced a secondary objective function to diversify their search. Häme (2011) proposed an adaptive insertion algorithm that can find the optimal solution for static and dynamic single-vehicle DARP with time windows in small instances. For larger instances, however, this approach uses a local search heuristic. They further analyze the complexity of their approach and test it by various numerical experiments. Ma et al. (2013), using the lazy shortest path calculation strategy, devised a schedule allocation algorithm to insert a new user's trip request into the schedule of the taxi that satisfies the query with minimum incurred travel distance for the ridesharing system.

Maalouf et al. (2014) developed an efficient dynamic fuzzy logic algorithm to insert a request in vehicles. They further considered two objectives in their algorithm to optimize the trade-off between level of service (customers' detour times) and company's performance (miles traveled by vehicles). Ritzinger et al. (2016) proposed a hybrid meta-heuristic that combines an exact dynamic programming with a meta-heuristic that is based on dynamic programming into a large neighborhood search. One advantage of their model is that it does not require a commercial MIP solver. Jafari et al. (2016) proposed a dial-a-ride system where each passenger places a bid for their request and specifies their pick-up and drop-off time windows. The operator selects the requests considering vehicle capacities, placed bids, and the operating cost to maximize the net profit.

In the earlier studies reviewed in this paper, researchers devised reactive solutions to respond to the dynamic nature of requests in a DARP. In a more advanced work, Bent and Van Hentenryck (2004) proposed a multiple scenario approach that continuously generates routing plans based on realized and anticipated customers to maximize the number of served requests in a stochastic dynamic DARP. They further introduce a consensus function to rank these plans in real time. Using numerical experiments, they show the prominent advantages of using stochastic information in vehicle routing. In another seminal work, Ichoua et al. (2006) introduced a real-time setting that accounts for the future request arrivals by strategically inserting some dummy customers in vehicle routes. They also use a parallel Tabu Search algorithm to investigate the neighborhood of current routes in real time. Schilde et al. (2011) considered a dynamic and stochastic DARP for a partially dynamic para-transit system. In this system, the inbound trips are scheduled ahead of time, but the outbound trips are unknown and will be realized in real time. In this work, they suggest using

historical data to predict the demand for return trips. Applying different meta-heuristic techniques in the literature, they showed that incorporating trip information based on historical data can lead to large improvements.

Another scenario-based heuristic was introduced by Ghilas et al. (2016) to solve a dynamic and stochastic DARP in the existence of some fixed scheduled lines. In this study, they developed an adaptive large neighborhood search algorithm inside a sample average approximation algorithm. Using numerical experiments, they showed the impact of integrating their pick-up and delivery vehicles with scheduled lines. Ulmer et al. (2017) introduced a route-based Markov decision process (MDP) similar to the conventional MDP for routing problems, except that the definition of a state also includes route plans that will be updated in different steps. This change in the definition of state also gives rise to a change in the definition of the reward function. Wei et al. (2017) introduced a look-ahead insertion policy for a shared-taxi system that inserts requests based on a trade-off between operating cost, wait time, and detour time. They further devised a reinforcement learning approach that exploits historical data to learn the parameter governing this trade-off. Alonso-Mora et al. (2017) presented a novel framework that assigns customers to vehicles using a reactive anytime optimal algorithm, and re-positions the idle vehicles using an LP formulation. They further claim that their approach can be easily adjusted to accommodate historical data. They conducted a comprehensive study on the New York City taxi dataset where they found that almost all taxi trips can be satisfied by much fewer vehicles when sharing rides is allowed. Lowalekar et al. (2018) proposed a multi-stage stochastic optimization formulation that incorporates future demand from historical data. They also presented a worst-case theoretical bound on the performance of their algorithm. Using numerical experiments on the New York Taxi dataset, they show the advantages of their proposed method over other algorithms in the literature. More recently, Li et al. (2019) suggested deploying vans proactively by using multiple scenarios of future requests and traffic conditions to generate and evaluate potential decisions for the position of vans. For a more comprehensive study of the related works on static and dynamic DARP variants and their solution methodologies, we refer interested readers to Berbeglia et al. (2007), Madsen et al. (1995), Santos and Xavier (2015), Paquette et al. (2013), Psaraftis (1980), Cordeau (2006), Paquette et al. (2012), Cordeau and Laporte (2003, 2007) and Ho et al. (2018).

Our paper contributes to the literature in the following ways: (*i*) We propose a new local search algorithm for generating a pool of useful routes in the offline phase; (*ii*) we introduce an efficient min-cost flow problem to find the optimal subset of customers served by a fixed route; and (*iii*) we propose an online framework that exploits the information from historical data to deploy shuttles proactively and change their routes if necessary in real time.

3. Problem statement

In this paper, we study a dynamic dial-a-ride/ridesourcing problem for an on-demand shuttle service that provides ride-share services to a set of passengers on a (directed) road network. We assume that the study region can be represented by a connected, directed graph $G = (S, \mathcal{E})$, where $S = \{1, 2, \dots, S\}$ denotes the set of stations, and $\mathcal{E} \subseteq S \times S$ denotes the set of transportation links. The stations are strategically positioned in the road network to ensure that there is at least one station in the walking distance of a typical trip origin/destination. We further discretize the study time horizon into an ordered set of indexed time intervals of small duration, say 1 min. We represent this set by $\mathcal{T} = \{0, 1, 2, \dots, T\}$. We assume that the shortest-path (SP) travel time between every two stations can be extracted from time-dependent travel-time matrices, denoted by $\tau(t, i, j)$. Entry (t, i, j) in this matrix holds the SP travel time between station i and station j if one departs from station i at time t.

Let $\mathcal{H} = \{1, 2, \dots, H\}$ denote the set of passengers (users) that dynamically enter the system during the study time horizon. The trip of a passenger h can be characterized by its origin and destination stations, denoted by $o(h) \in S$ and $d(h) \in S$, respectively, and the earliest departure time (EDT) from the trip origin, denoted by $e(h) \in \mathcal{T}$. In this study, we introduce a constant parameter ω that accounts for the maximum pick-up waiting time of every passenger at their origin station. As a result, the pick-up time of passenger h, represented by $p(h)$, should fall within the interval $[e(h), e(h) + \omega]$. Furthermore, in order to maintain a certain level of service, we introduce another constant parameter Ω that limits the detour of every passenger from their shortest path. We denote the latest arrival time (LAT) of passenger h at their trip's destination station by $l(h)$, where $l(h) = p(h) + \tau(p(h), o, d) + \Omega$. We assume passengers announce their trips at short notice prior to the start of their pick-up time windows and expect a response from the operator no later than their earliest departure time. We also assume that from historical data and/or survey responses, the distribution of the number of requests for an origin–destination pair, (o, d), that arrive at the system at time interval e can be estimated with mean $\phi(e, o, d)$ (e.g., the time-dependent travel demand may follow a Poisson distribution with rate $\phi(e, o, d)$).

The shuttle service company is in charge of a fleet of maximum K homogeneous high-capacity shuttles (also referred to as carpooling vans) denoted by $\mathcal{K} = \{1, 2, \dots, K\}$, with C available seats for each shuttle, that operate during the time horizon $\mathcal{T}$. Every shuttle starts/ends empty in the beginning/end of the horizon from/at any station. It can be easily shown that our methodology can be simply adapted to special cases where there exist a set of heterogeneous origins and destinations as well as capacities for shuttles. As a result of having discrete sets of time and space, we can represent the time-expanded network in our study with a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{N}, \mathcal{L})$. Every node $n = (t, i) \in \mathcal{N}$ in this graph is a tuple whose first entry is the time and the second entry is the station in the road network. Moreover, the edge set $\mathcal{L}$ consists of all edges $\ell = ((t, i), (q, j))$ such that $q = t + \tau(t, i, j)$. The route of every shuttle in the study time horizon can be represented by a path (sequence of nodes) in this graph that connects a node with $t = 0$ to a node with $q = T$. Finally we assume that shuttles can wait at any station for any number of time steps as long as they can serve their on-board and scheduled customers within their specified time windows. This assumption implies that all edges of type $\ell = \{((t, i), (t + 1, i)),\ \forall\, i \in S, t \in \mathcal{T}\}$ also exist in graph $\mathcal{G}$.

Finally, without loss of generality, we assume that the system aims to maximize the total origin–destination shortest-path driving distances of customers, i.e., the summation of the origin–destination shortest-path distances of the trips served by the system. Since

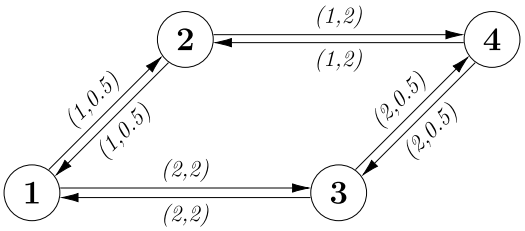


Fig. 1. The road network of the toy example. The tuples on directed links denote the shortest-path travel time (in minutes) and driving distance (in miles).

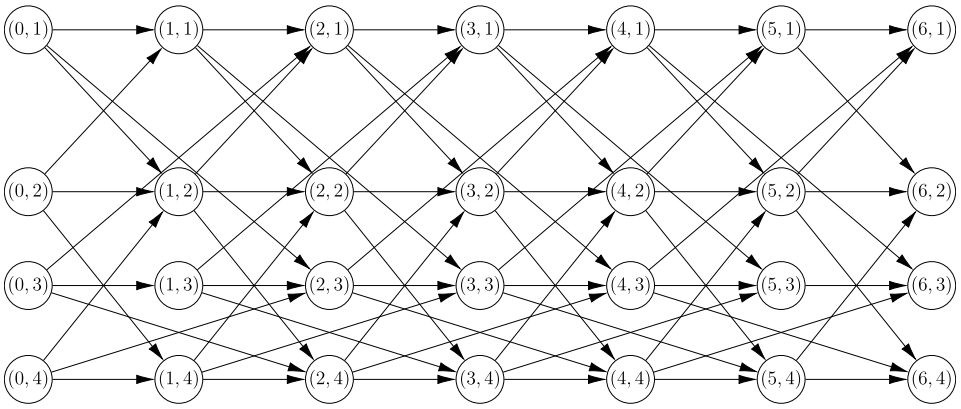


Fig. 2. The time-expanded network of the toy example. A node n in this time-expanded network is annotated by the tuple (t, s), where t is a time step and s is a physical location on the road network.

fare of customers are usually proportionate to the distance of their trips, this objective function is aligned with the purpose of shuttle service companies in maximizing profit. Thus hereafter, we refer to the objective function as "Revenue" with unit of mile. However, note that the proposed methodology is independent of choice of objective function. It is worth mentioning that the introduction of the pool of useful trajectories makes our methodology particularly suitable for a central fleet operator with full control of all shuttles' paths during the specified time horizon, e.g., an autonomous fleet operator.

We end this section by introducing a toy example that enables us to clearly illustrate the proposed routing algorithms in our method.

3.1. A toy example

Let us consider a road network that consists of 4 stations, as shown in Fig. 1. In this figure, the shortest-path travel time (in minutes) and driving distance (in miles) of every link is presented by a tuple for that link.

We further assume that the time horizon is 6 min with time steps of 1 min. As a result, the time-expanded network of this example is demonstrated in Fig. 2. Every node (t, i) in this network represents the spatio-temporal position of a shuttle, e.g., node (3,1) indicates that the shuttle is located in station 1 at time 3. In addition, there exists an edge between node (t, i) and (q, j) if the shuttle can depart from station i at time t and arrive at station j at time q. Hence, any path from the leftmost nodes to the rightmost ones of this network can be considered as a complete route of a shuttle on the road network during the specified time horizon.

The demand forecast for this toy example is provided in Table 1. Every entry (e, i, j) in ϕ represents the average number of requests from origin station i to destination station j that entered the system at time e, e.g. $\phi(2, 1, 4) = 8.0$ tells us that 8.0 requests for travel from station 1 to station 4 are expected to arrive at time 2. We further assume that customers are available for pick-up upon their arrival into the system for 2 time steps, i.e., $\omega = 2$. Thus, as an instance, the total expected number of requests that can be picked up by a shuttle from station 1 at time 2 and be dropped off at station 4 can be calculated as $\phi(1, 1, 4) + \phi(2, 1, 4) = 7.9 + 8.0 = 17.9$. Finally, we assume that only 1 shuttle with 10 seats is available for serving requests, and the maximum allowed detour time, Ω, is set to 3.

4. Solution methodology

In this section, we present a data-driven methodology that exploits the useful patterns hidden in historical data to respond to the dynamic and stochastic nature of the shuttle dispatching problem. This methodology can be clearly divided into two distinct, yet related, parts, which are implemented in offline and online phases. The offline portion of the solution methodology aims to find a

Table 1
The demand rate of the toy example.

$$\phi(0,.,.)=\begin{bmatrix}0.0 & 3.3 & 5.2 & 8.0\\ 1.2 & 0.0 & 3.5 & 6.9\\ 1.6 & 3.6 & 0.0 & 2.1\\ 5.3 & 6.3 & 9.4 & 0.0\end{bmatrix},\ \phi(1,.,.)=\begin{bmatrix}0.0 & 2.5 & 5.1 & 7.9\\ 1.4 & 0.0 & 3.8 & 6.6\\ 3.2 & 2.8 & 0.0 & 1.2\\ 6.3 & 5.6 & 8.5 & 0.0\end{bmatrix},\ \phi(2,.,.)=\begin{bmatrix}0.0 & 2.5 & 5.7 & 8.0\\ 1.4 & 0.0 & 4.6 & 6.7\\ 2.2 & 3.6 & 0.0 & 1.0\\ 5.2 & 5.8 & 7.4 & 0.0\end{bmatrix}$$

$$\phi(3,.,.)=\begin{bmatrix}0.0 & 2.7 & 5.0 & 8.0\\ 0.9 & 0.0 & 5.1 & 6.7\\ 2.7 & 3.3 & 0.0 & 1.0\\ 5.9 & 5.9 & 7.4 & 0.0\end{bmatrix},\ \phi(4,.,.)=\begin{bmatrix}0.0 & 1.8 & 5.1 & 8.3\\ 1.6 & 0.0 & 0.0 & 7.5\\ 1.6 & 0.0 & 0.0 & 1.3\\ 6.3 & 7.1 & 8.2 & 0.0\end{bmatrix},\ \phi(5,.,.)=\begin{bmatrix}0.0 & 2.3 & 0.0 & 0.0\\ 1.0 & 0.0 & 0.0 & 6.7\\ 0.0 & 0.0 & 0.0 & 0.0\\ 0.0 & 5.7 & 0.0 & 0.0\end{bmatrix}$$

pool of near-optimal shuttle routes under different realizations of demand requests. In practice, many different trip patterns can be realized, each giving rise to a distinct set of optimal shuttle routes. These trip patterns are governed by a probability density function that can be learned from the historical data. In this research, for instance, we assume that the arrival of passengers in the system follows a Poisson distribution with a mean estimated from past requests. In the online phase, the shuttles will be proactively routed through one of the previously-devised routes. In case of having accurate demand forecasts, we expect that the shuttles readily serve a good portion of passengers en-route. However, the alternative routes in our pool can be used at any time to change the route of a shuttle toward more promising directions in real-time. In what follows, we first present a method to find this pool of routes in an offline phase. Next, we introduce a procedure to use this pool of routes in an online phase, where the realized demand could be stochastically different from the predicted one.

4.1. Offline planning

As mentioned earlier, the offline phase of our method aims to construct a pool of useful (near-optimal) routes based on historical data. Toward this end, we propose a multi-step framework that takes the time-expanded network under study and the average rate of demand as its inputs, and provides a set of shuttle routes as its output. This general framework, presented in Algorithm 1, is built on Algorithms 2, 3, and 4, which will be described in further details later.

Suppose that we are interested in obtaining a set of M alternative routes per shuttle. Let us denote alternative route $m = 1, \dots, M$ for shuttle k by $\mathcal{P}_m^k$. Also, we refer to the first route ($m = 1$) as the base route of each shuttle. In this framework, we first find the base routes using the average demand forecast ϕ in network graph $\mathcal{G}$. These routes help us determine a tentative direction for each shuttle in time-expanded network $\mathcal{G}$. We assume that each shuttle will be deployed on this primary route in the online phase of our problem.

In order to find the base routes, we first create K initial routes by partitioning the average demand, ϕ, into K approximately uniform clusters, ϕ^k, and then constructing a full route for each cluster ϕ^k using a routing algorithm (which will be described later) in network $\mathcal{G}$. For partitioning ϕ into clusters, we use the methodology described in Tafreshian and Masoud (2020). After finding the initial base routes, we set the total revenue, denoted by z, to zero, and repeat a clustering and routing procedure until the total revenue converges to a fixed value. At each iteration of this procedure, we first calculate the distance between every trip in ϕ and current base routes, and assign the demand to the closest shuttle by solving the problem in 1. For measuring the distance between trip (e, o, d) in ϕ and base route $\mathcal{P}_1^k = [(t_1, s_1), (t_2, s_2), \dots, (t_x, s_x)]$, we use the equation:

$$\eta^k(e,o,d) = \min_{\substack{e\le p\le e+\omega \\ p+\tau(p,o,d)\le d\le p+\tau(p,o,d)+\Omega \\ (t_i,s_i),(t_j,s_j)\in\mathcal{P}_1^k}} \left\{|t_i - p| + \tau(p,o,s_i) + |t_j - d| + \tau(l,d,s_j)\right\} \tag{3}$$

which finds the spatiotemporal distance between the origin and destination of the trip with their closet nodes in the route. After assigning the demand to the closest shuttle, we use the routing algorithm in 4 to adjust the base routes for each shuttle. We also find the optimal revenue, γ_1^k, and served requests, f_1^k, for each shuttle using a min-cost flow algorithm (which will be described later in Algorithm 3). Since all requests for a given origin–destination pair may not be served by its closest shuttle, we repeat this process (for R times) by assigning the unserved demand to its next closest shuttle.

Note that using the average demand forecast enables us to take into account the average travel pattern. However, relying merely on average demand has two drawbacks. First, it does not necessarily represent a feasible demand profile due to continuity of demand rate for every (t, i, j). Second, it does not capture the variability in demand. In order to mitigate these issues, we realize $M-1$ different demand profiles from the Poisson distribution with average rate of ϕ, and construct an alternative route per each demand profile. Let us denote these demand profiles with Φ_m, $\forall\ m = \{2, 3, \dots, M\}$. It is worth mentioning that there is no specific priority among different shuttles, and indices 1 through K do not represent any order among them. Also, there is no specific priority among the alternative routes of a single shuttle, and indices 2 through M for each shuttle do not represent any order among these routes.

Before finding an alternative route $m \in \{2, 3, \dots, M\}$ for shuttle k, we introduce a 2-step procedure (see Algorithm 2) to reduce the time-expanded network to a graph in the neighborhood of the base route. In the first step, we find a sub-graph of $\mathcal{G}$, denoted by $\mathcal{G}'$, in which every node has a spatio-temporal distance of maximum ϵ from its closest node in the base route. In the second step, we first find the optimal set of passengers (from the demand table Φ_m^k) that can be served by the base route $\mathcal{P}_1^k$ (using Algorithm 4 which will be discussed later). Let us denote the set of spatio-temporal nodes at which the served passengers are picked up by $\mathcal{N}_f$. Then, we obtain a sub-graph of $\mathcal{G}'$ in which every node is accessible by nodes in $\mathcal{N}_f$, i.e., there exists a path between every node

Algorithm 1: Offline framework

Input: Time-expanded network: $\mathcal{G}$
Average demand forecast: ϕ
Threshold for reducing the graph around base: ϵ
Number of alternative routes per shuttle: M
Output: Pool of shuttle routes: $\mathcal{P}$
Partition the average demand in ϕ into K uniform clusters, $\phi^k \quad \forall\, k \in \mathcal{K}$;
for $k \in \mathcal{K}$ **do**
  $\mathcal{P}_1^k \leftarrow$ Find a base route for shuttle k using demand ϕ^k in network $\mathcal{G}$ (see Algorithm 4) ;
$z \leftarrow 0$;
repeat
  $\eta^k(e,o,d) \leftarrow$ Find the distance between every trip (e,o,d) in ϕ and base route $\mathcal{P}_1^k, \quad \forall\, k \in \mathcal{K}, \forall\, e \in \mathcal{T}, \forall\, o \in \mathcal{S}, \forall\, d \in \mathcal{S}$;
  Create the clustering problem:

$$\min \quad \sum_{k=1}^{K} \sum_{(e,o,d)} \eta^k(e,o,d)\, \phi^k(e,o,d) \tag{1a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \phi^k(e,o,d) = \phi(e,o,d) \qquad \forall\, k \in \mathcal{K}, \forall\, e \in \mathcal{T}, \forall\, o \in \mathcal{S}, \forall\, d \in \mathcal{S} \tag{1b}$$

  for $r = 1, 2, \ldots, R$ **do**
    $\phi^k \leftarrow$ Distribute demand between shuttles by solving the clustering problem in 1;
    for $k \in \mathcal{K}$ **do**
      $\mathcal{P}_1^k \leftarrow$ Find a base route for shuttle k using demand ϕ^k in network $\mathcal{G}$ (see Algorithm 4) ;
      $\gamma_1^k, f_1^k \leftarrow$ Find optimal revenue and served requests given $\mathcal{P}_1^k$ and ϕ^k (see Algorithm 3);
      Add $\phi^k(e,o,d) \le f_1^k(e,o,d)$ to the clustering problem in 1, $\quad \forall\, e \in \mathcal{T}, \forall\, o \in \mathcal{S}$, $\forall\, d \in \mathcal{S}$;
    $z \leftarrow \sum_{k=1}^{K} \gamma_1^k$;
until z converges;
for $m = 1, 2, \ldots, M$ **do**
  $\boldsymbol{\Phi}_m(e,o,d) \leftarrow$ Sample from $Poi(\phi(e,o,d)) \quad \forall\, e \in \mathcal{T}, \forall\, o \in \mathcal{S}, \forall\, d \in \mathcal{S}$;
  $\eta^k(e,o,d) \leftarrow$ Find the distance between every trip (e,o,d) in $\boldsymbol{\Phi}_m$ and base route $\mathcal{P}_1^k, \quad \forall\, k \in \mathcal{K}, \forall\, e \in \mathcal{T}, \forall\, o \in \mathcal{S}, \forall\, d \in \mathcal{S}$;
  Create the clustering problem:

$$\min \quad \sum_{k=1}^{K} \sum_{(e,o,d)} \eta^k(e,o,d)\, \boldsymbol{\Phi}_m^k(e,o,d) \tag{2a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \boldsymbol{\Phi}_m^k(e,o,d) = \boldsymbol{\Phi}_m(e,o,d) \qquad \forall\, k \in \mathcal{K}, \forall\, e \in \mathcal{T}, \forall\, o \in \mathcal{S}, \forall\, d \in \mathcal{S} \tag{2b}$$

  for $r = 1, \ldots, R$ **do**
    $\boldsymbol{\Phi}_m^k \leftarrow$ Distribute demand between shuttles by solving the clustering problem in 2;
    for $k \in \mathcal{K}$ **do**
      $\mathcal{G}_m^k \leftarrow$ Reduce $\mathcal{G}$ to a network around $\mathcal{P}_1^k$ using ϵ and $\boldsymbol{\Phi}_m^k$ (see Algorithm 2);
      $\mathcal{P}_m^k \leftarrow$ Find an alternative route for shuttle k using demand $\boldsymbol{\Phi}_m^k$ in network $\mathcal{G}_m^k$ (see Algorithm 4) ;
      $f_m^k \leftarrow$ Find optimal served requests given $\mathcal{P}_m^k$ and $\boldsymbol{\Phi}_m^k$ (see Algorithm 3);
      Add $\boldsymbol{\Phi}_m^k(e,o,d) \le f_m^k(e,o,d)$ to the clustering problem in 2, $\quad \forall\, e \in \mathcal{T}, \forall\, o \in \mathcal{S}$, $\forall\, d \in \mathcal{S}$;

$n' \in \mathcal{N}_f$ and every node n in $\mathcal{G}'$. Note that the accessibility is a two sided relationship, and the pair order just depends on the time dimension of the two nodes (see West et al., 1996).

Using this procedure, one can ensure that (1) there are some common spatio-temporal nodes between different alternative routes in case the shuttle is required to change its itinerary en-route in real-time, and (2) the set of passengers that can be served by different routes are highly overlapping, and thus, changing the route is not likely to violate the destination of on-board passengers.

Suppose, as an instance in our toy example, the base route is $\mathcal{X} = \{(0,1),(1,2),(2,1),(3,2),(4,1),(5,2),(6,1)\}$ and $\epsilon = 1$. By taking the first step, the time-expanded network can be reduced as shown in Fig. 3. In this network, every node is either on the path (black solid lines), or 1 min away from it (black dashed lines). Now, suppose we take a Poisson sample from the average demand in Table 1 as shown in Table 2. Based on the realized demand table and number of available seats, a shuttle can optimally pick up 1, 4, and 2 customers at nodes $(2,1)$, $(3,2)$, and $(4,1)$, respectively, on the specified path. Hence, we set $\mathcal{N}_f = \{(2,1),(3,2),(4,1)\}$. Fig. 4 shows the result of applying the second step using this information. This figure clearly shows that every complete route from the leftmost nodes to the rightmost ones must pass through all nodes in $\mathcal{N}_f$.

In order to find alternative route m for each shuttle, we follow the same process of clustering and routing as in finding the base routes, with random demand forecast $\boldsymbol{\Phi}_m^k$ and sub-graph $\mathcal{G}_m^k$. This procedure will be repeated $M-1$ times to yield a pool of $K \times M$ routes. In the next two parts, we elaborate on two of the sub-procedures required to implement Algorithm 1 – we first propose an algorithm to efficiently find the optimal subset of passengers served by a fixed route. Next, we introduce a local search algorithm to find the route of a single shuttle based on a demand table.

Algorithm 2: Reduce time-expanded network around a route

Input: Time-expanded network: $\mathcal{G}$
A realization of demand: Φ
Base route: $\mathcal{X}$
Threshold for reducing the graph around base: ϵ
Output: Reduced time-expanded network: $\mathcal{G}''$
Step 1:
$\mathcal{N}' \leftarrow \varnothing$;
for $n \in \mathcal{N}$ **do**
  for $n' \in \mathcal{X}$ **do**
    if length of Shortest Path (SP) between n and n' in $\mathcal{G} \leq \epsilon$ **then**
      $\mathcal{N}' \leftarrow \mathcal{N}' \cup \{n\}$
$\mathcal{G}' \leftarrow$ sub-graph of $\mathcal{G}$ induced by $\mathcal{N}'$ (West et al., 1996) ;
Step 2:
$f \leftarrow$ Find optimal served requests given $\mathcal{X}$ and Φ;
$\mathcal{N}_f \leftarrow$ Find stations in $\mathcal{X}$ at which served requests in f are picked up ;
$\mathcal{N}'' \leftarrow \varnothing$;
for $n \in \mathcal{N}'$ **do**
  for $n' \in \mathcal{N}_f$ **do**
    if node n is accessible by n' in $\mathcal{G}'$ **then**
      $\mathcal{N}'' \leftarrow \mathcal{N}'' \cup \{n\}$;
$\mathcal{G}'' \leftarrow$ sub-graph of $\mathcal{G}'$ induced by $\mathcal{N}''$;

Table 2
A demand profile for the toy example.

$$\Phi = \left[\begin{bmatrix} 0 & 0 & 7 & 4 \\ 2 & 0 & 5 & 7 \\ 3 & 3 & 0 & 0 \\ 5 & 5 & 9 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 4 & 9 \\ 2 & 0 & 5 & 6 \\ 4 & 5 & 0 & 1 \\ 7 & 7 & 8 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 2 & 8 \\ 3 & 0 & 3 & 4 \\ 5 & 4 & 0 & 3 \\ 8 & 6 & 5 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 4 & 8 & 3 \\ 1 & 0 & 2 & 3 \\ 1 & 2 & 0 & 1 \\ 1 & 6 & 9 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 2 & 1 & 6 \\ 1 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 \\ 3 & 4 & 9 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 \end{bmatrix}\right]$$

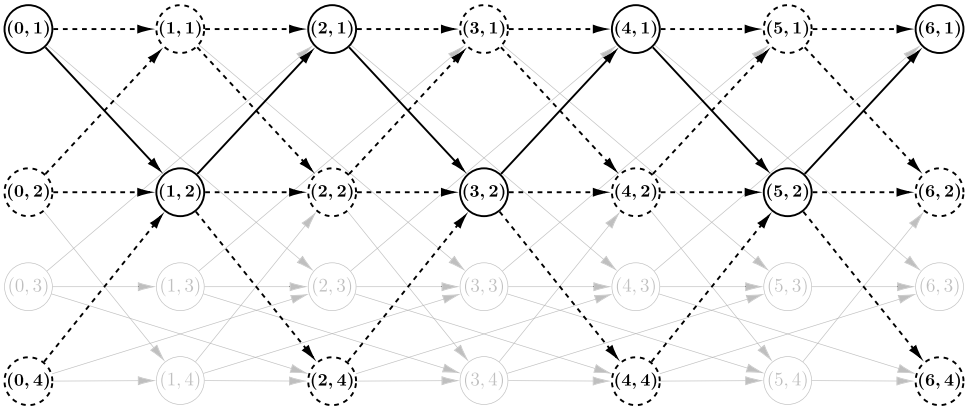


Fig. 3. The reduced time-expanded network of the toy example in step 1.

4.1.1. Finding the optimal set of passengers served by a fixed route

Suppose that we have a capacitated shuttle with a fixed route $\mathcal{X} = [n_1, n_2, \dots, n_x]$, and that there is a set of passengers whose pick-up and drop-off nodes belong to the sequence of nodes in $\mathcal{X}$. We are looking for the best subset of passengers that can be served by this shuttle. In this section, we propose an optimal method to solve this problem efficiently. This method is motivated by the concept of k-coloring in interval graphs. Next, we briefly describe the maximum weighted k-coloring problem and show how it is related to the problem at hand.

In graph theory, maximum weighted k-coloring of intervals is a well-known problem. In this problem, one is required to color a set of n' open intervals (from the real line), denoted by $I_1, I_2, \dots, I_{n'}$, with a set of K colors, such that two intersecting intervals receive distinct colors. Every interval is also associated with a positive weight of $w_i \ \forall \ i \in \{1, 2, \dots, n'\}$. Further, suppose that $v_1 < v_2 < \cdots < v_r$ represent the unique set of ordered endpoints of these intervals. It is shown in Carlisle and Lloyd (1995) that this problem can be formulated as a min-cost flow problem in a weighted DAG G' with nodes $s = v_0 < v_1 < \cdots < v_{r+1} = t$ and two sets of edges, called clique edges and interval edges. The clique edges $(v_{i-1}, v_i) \ \forall \ 1 \leq i \leq (r+1)$ have capacity K and cost 0. The interval

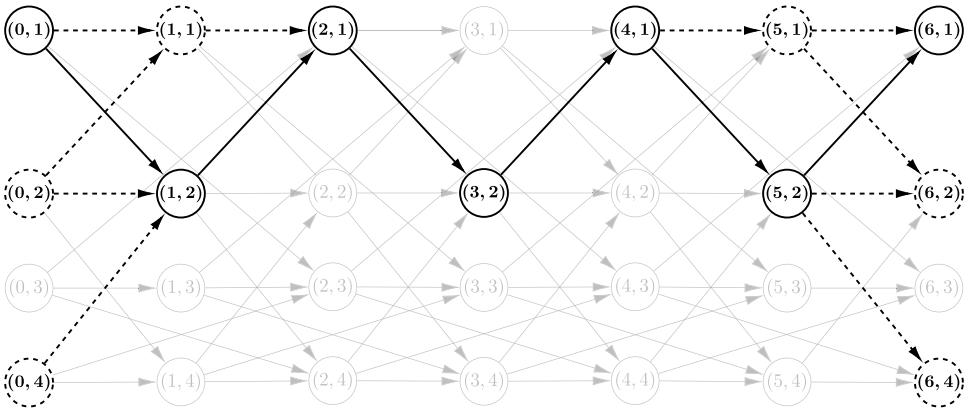


Fig. 4. The reduced time-expanded network of the toy example in step 2.

edges (v_j, v_h), with v_j and v_h being defined as the left and right endpoint of interval I_i $\forall\ 1 \le i \le n'$, have capacity 1 and cost $-w_i$. Also let the demand and supply of source node s and target node t be K. In what follows, we present a result from Carlisle and Lloyd (1995) regarding this graph theoretic problem.

Lemma 1 (*Carlisle and Lloyd, 1995*)**.** *Given a min-cost flow of size K in G′, the intervals corresponding to the interval-edges of flow 1 are exactly the intervals in some maximum weight k-coloring of the n′ intervals.*

Now, we show that the problem of finding the best subset of passengers to serve by a fixed route is a variant of the k-coloring problem. The trip of every passenger can be considered as an interval from its pick-up node to its drop-off node. We set the weight of an interval equal to the revenue obtained by serving a passenger from its associated pick-up node to its drop-off node. We can also think of the C seats in a shuttle as the distinct colors in the k-coloring problem. Hence, our objective is to assign these seats to passengers such that every seat can be occupied by at most one passenger at any point throughout the fixed route, while maximizing the total revenue obtained by serving these passengers. In Algorithm 3, we provide a pseudo-code to formulate the problem at hand as a min-cost flow problem in a weighted DAG, denoted by $\mathscr{G} = (\mathscr{N}, \mathscr{L}, \mathscr{W}, \mathscr{U})$, where $\mathscr{N}, \mathscr{L}, \mathscr{W}$, and $\mathscr{U}$ represent the set of nodes, edges, weights and capacities, respectively.

Note that there are two subtle difference between our problem and the k-coloring problem. First, the passenger intervals are closed from the left (i.e., at their pick-up nodes). In order to handle this issue, we define a drop-off and a pick-up node per each node $n_i \in \mathscr{X}$, denoted by n_i^{d} and n_i^{p}, respectively. As a result, we have two types of clique edges, namely the ones that connect the consecutive drop-off nodes and the ones that connect each drop-off node to its consecutive pick-up node. Second, the intervals in our problem are not unique. In other words, we can have more than one passenger request from node $n_i = (t, i)$ to node $n_j = (q, j)$. As a result, the capacity of interval edges is equal to the number of passengers that are available to be picked up at node $n_i = (t, i)$ and dropped off at $n_j = (q, j)$.

Now, let us describe the above procedure using an instance from our toy example. Suppose that $\mathscr{X} = \{(0,3),(2,4),(3,2),(4,1),(6,3)\}$ and we have realized the demand profile $\boldsymbol{\Phi}$ in Table 2. Table 3 shows the number of available customers, obtained from $\boldsymbol{\Phi}$, for each combination of origin–destination stations in route $\mathscr{X}$, e.g., the shuttle can pick up $\boldsymbol{\Phi}(1,4,2) + \boldsymbol{\Phi}(2,4,2) = 6 + 7 = 13$ customers from station 4 at time 2 and drop them off at station 2 at time 3. The trip intervals of these customers are depicted in Fig. 5. The first column, (t, i, q, j), shows the origin–destination combination, the second column, $\rho(t, i, j)$, shows the profit of each interval (i.e., shortest-path driving distance between stations i and j departing from i at time t), and the third column shows the trip intervals on a timeline. The value on top of each interval presents the frequency of that trip interval obtained from Table 3. This figure clearly indicates that we are not able to serve all of the customers. For instance, at time 2, 3+3+13+15+13=47 intervals intersect while we have only 10 seats in our shuttle. Therefore, we aim to find the most profitable subset of intervals that does not violate the capacity of the shuttle at any time.

As stated above, this can be formulated as a min-cost flow problem depicted in Fig. 6(a). For each station, we define two nodes of drop-off (dashed line) and pick-up (solid line) except the last one for which we only have a drop-off node. We further show the clique edges with dashed lines and the interval edges with solid lines. The tuple on top of each edge shows the negative of the cost and the capacity of that edge. Finally, we assume that the first drop-off node is a source node with supply of 10 and the last drop-off node is a sink node with demand of 10. The solution to this capacitated min-cost flow problem, depicted in Fig. 6(b), provides the most profitable subset of customers in $\boldsymbol{\Phi}$ that can be served by route $\mathscr{X}$. Note that the reader must distinguish between the capacity of interval edges in min-cost flow problem and the capacity of shuttles, C, as the former represents the number of available customers for a certain origin–destination combination while the latter is a given input that are reflected in the capacity of clique edges. In our toy example, for instance, the capacity of arc $(2,4,6,3)$ is 13 that is higher than capacity of shuttle, 10. That being said, for computational purposes, we can always replace the capacities above C with C since we know that there is no way to have a flow of greater than C in this min-cost flow problem.

We end this section by presenting a remark that shows the efficiency of the proposed min-cost flow algorithm.

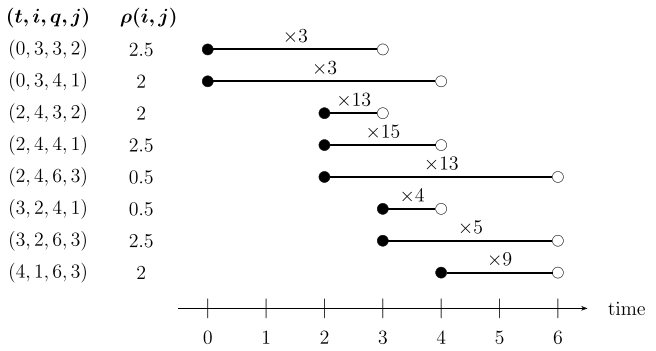


Fig. 5. Trip intervals of the demand in Table 3.

Remark 1. The optimal subset of passengers served by a single shuttle that visits a fixed route of x stations can be found by a min-cost flow problem with worst-case running time of $\mathcal{O}(Cx^2)$.

Proof. Following Lemma 1, one can easily show that the min-cost flow is a valid formulation for the problem at hand. In order to find the complexity of the algorithm, we divide the computation into two parts of constructing graph $\mathcal{G}$, and solving the min-cost flow problem on $\mathcal{G}$. For constructing the graph, the bottleneck is to find the interval edges. Since there are x stops in route $\mathcal{X}$, there are $\frac{x(x-1)}{2}$ possible combinations for pick-up and drop-off nodes for constructing the interval edges. Thus, the complexity of constructing graph $\mathcal{G}$ is $\mathcal{O}(x^2)$. Also, note that graph $\mathcal{G}$ is an acyclic graph with integer capacities. According to Tarjan (1983), the min-cost flow in this graph has the time complexity of $\mathcal{O}(|\mathcal{L}|+CS(x,|\mathcal{L}|))$, where $S(x,|\mathcal{L}|)$ is the complexity of finding the shortest path in a graph with x nodes and $|\mathcal{L}|$ edges. Using the Dijkstra's algorithm with Fibonacci heaps, it can be shown that the overall complexity of the min-cost flow in Algorithm 3 is $\mathcal{O}(|\mathcal{L}| + C(|\mathcal{L} + x\log x|))$. In the most computationally-intensive scenario that arises when $|\mathcal{L}| = x^2$, the running time complexity of the algorithm is $\mathcal{O}(Cx^2)$, which is linear in C, and quadratic in x. □

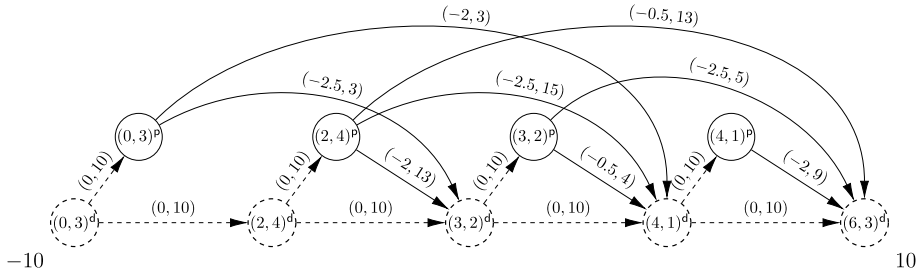
Algorithm 3: Find the sequence of pick-up and drop-offs that provide the optimal revenue, given a fixed route and demand forecast

Input: Fixed Route of shuttle: $\mathcal{X} = [n_1, n_2, \dots, n_x]$
Demand forecast: Φ
Output: Optimal revenue of shuttle: γ
Optimal number of served trips: f

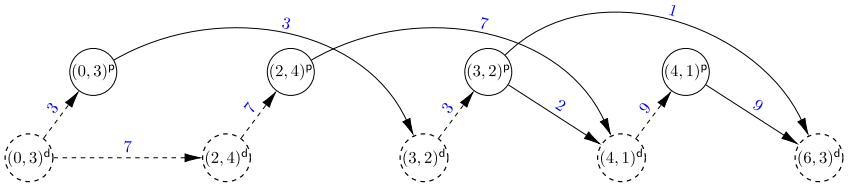
$\mathcal{N} \leftarrow \{n_1^{\mathsf{d}}, n_1^{\mathsf{p}}, n_2^{\mathsf{d}}, n_2^{\mathsf{p}}, \dots, n_x^{\mathsf{d}}, n_x^{\mathsf{p}}\}$;
$\mathcal{D}(n_1^{\mathsf{d}}) \leftarrow -C$; $\mathcal{D}(n_x^{\mathsf{p}}) \leftarrow C$; $\mathcal{D}(n_i^{\mathsf{d}}) \leftarrow 0 \ \forall\, i \in \{2, \dots, x\}$; $\mathcal{D}(n_i^{\mathsf{p}}) \leftarrow 0 \ \forall\, i \in \{1, \dots, x-1\}$;
$\mathcal{L} \leftarrow \{\}$;
for $i = 1, 2, \dots, x-1$ **do**
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{n_i^{\mathsf{d}}, n_i^{\mathsf{p}}\}; \mathcal{W}(n_i^{\mathsf{d}}, n_i^{\mathsf{p}}) \leftarrow 0; \mathcal{U}(n_i^{\mathsf{d}}, n_i^{\mathsf{p}}) \leftarrow C$;
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{n_i^{\mathsf{d}}, n_{i+1}^{\mathsf{d}}\}; \mathcal{W}(n_i^{\mathsf{d}}, n_{i+1}^{\mathsf{d}}) \leftarrow 0; \mathcal{U}(n_i^{\mathsf{d}}, n_{i+1}^{\mathsf{d}}) \leftarrow C$;
for $i = 1, 2, \dots, x-1$ **do**
  for $j = i+1, i+2, \dots, x$ **do**
    $(p, o) \leftarrow n_i^{\mathsf{p}}$;
    $(l, d) \leftarrow n_j^{\mathsf{d}}$;
    if d not visited again between n_i^{p} and n_j^{d} **then**
      if $(l - p) \le \tau(p, o, d) + \Omega$ **then**
        $\mathcal{L} \leftarrow \mathcal{L} \cup \{n_i^{\mathsf{p}}, n_j^{\mathsf{d}}\}$;
        if o visited again between n_i^{p} and n_j^{d} **then**
          $\mathcal{U}(n_i^{\mathsf{p}}, n_j^{\mathsf{d}}) \leftarrow \Phi(p, o, d)$;
        else
          $t_0 \leftarrow$ Last time o visited in time window $[p - \omega, p]$;
          $\mathcal{U}(n_i^{\mathsf{p}}, n_j^{\mathsf{d}}) \leftarrow \sum_{t=t_0+1}^{e} \Phi(t, o, d)$;
        $\mathcal{W}(n_i^{\mathsf{p}}, n_j^{\mathsf{d}}) \leftarrow -\rho(p, o, d)$;
Define graph $\mathcal{G} = (\mathcal{N}, \mathcal{L}, \mathcal{D}, \mathcal{W}, \mathcal{U})$;
Solve min-cost flow on $\mathcal{G}$;
$f \leftarrow$ Optimal flow on edges of type $(n_i^{\mathsf{d}}, n_j^{\mathsf{p}})$;
$\gamma \leftarrow -1\times$ Optimal flow cost ;

Table 3
The potential customers for route $\mathcal{X} = \{(0,3),(2,4),(3,2),(4,1),(6,3)\}$ in the toy example.

	(2,4)	(3,2)	(4,1)	(6,3)
(0,3)	0	3	3	0
(2,4)		13	15	13
(3,2)			4	5
(4,1)				9



(a) min-cost flow problem



(b) optimal flows with the optimal revenue of 46.5

Fig. 6. The min-cost flow problem to find the most profitable set of passengers to be served through path $\mathcal{X} = \{(0,3),(2,4),(3,2),(4,1),(6,3)\}$ and its optimal solution.

It is worth mentioning that our min cost flow formulation, aiming to find the optimal set of passengers served by a fixed route, carries similarities with the min cost flow formulation in Jafari et al. (2016). However, as we have demonstrated in Remark 1, our formulation differs from the min-cost flow formulation of Jafari et al. (2016) from two stand points. First, Our min-cost flow formulation does not depend on the number of passengers; rather, it depends on the number of origin–destination pairs. This results in significant computational savings in crowded metropolitan networks such as the Manhattan network. Second, our min-cost flow formulation has fewer number of edges, which again results in less computational effort.

4.1.2. Shuttle routing

Consider the problem of routing a single shuttle on road network G during time horizon $\mathcal{T}$. The route of a shuttle, denoted by $\mathcal{X}$, can be defined as a path in a time-expanded network G that connects a node with time 0 to a node with time T. Given a set of demand requests in Φ, the goal of this problem is to find a route that maximizes the revenue of served passengers. This problem is a well-known combinatorial problem that is NP-hard. In order to provide a high-quality solution to this problem, we propose a local search method in Algorithm 4. This local search algorithm is motivated by the concept of dynamic programming for finding the longest path in a weighted DAG. In this approach, the nodes will be observed in a topological order and the value of a node, which is defined as the longest path until that node, will be updated based on the value of its predecessor nodes and the weights of their edges connecting them to that node. After computing the value of all nodes, the longest path can be found by backtracking from the node with the highest value.

Similar to this approach, we consider a value at each node of $\mathcal{G}$ in our local search, denoted by $\varphi(n)$. This value can be defined as the highest revenue of routes that pass by node n. Since there are an exponential number of routes passing by each node, we propose an iterative framework that reduces the search space by fixing the nodes visited either prior to or after node n. We choose the route passing by the node with the highest value as the best route, denoted by $\mathcal{X}^*$. Let us define $\mathcal{F}_\alpha(n)$ as the partial route from a node in time 0 to node n at iteration α, hereafter referred to as the "predecessor route". Similarly, define $\mathcal{B}_\alpha(n)$ as the partial route from node n to a node in time T at iteration α, hereafter referred to as the "successor route". Note that connecting these two partial routes at node n yields a complete route, $\mathcal{X}$. Let us represent the connecting operator by $\oplus$.

The local search starts by constructing the predecessor route for all n in $\mathcal{N}$ at iteration 0 using Algorithm 5. In this algorithm we traverse the list of nodes in order of their time components, and for each node, pick one of its immediate predecessor nodes at random. This enables us to create a path from any node at time 0 to every node very quickly. In the next iteration, the predecessor routes will be fixed from the previous iteration and Algorithm 6 finds the successor routes through a backward local search. This will be followed by a forward local search using Algorithm 7 that fixes the successor routes from the backward local search and

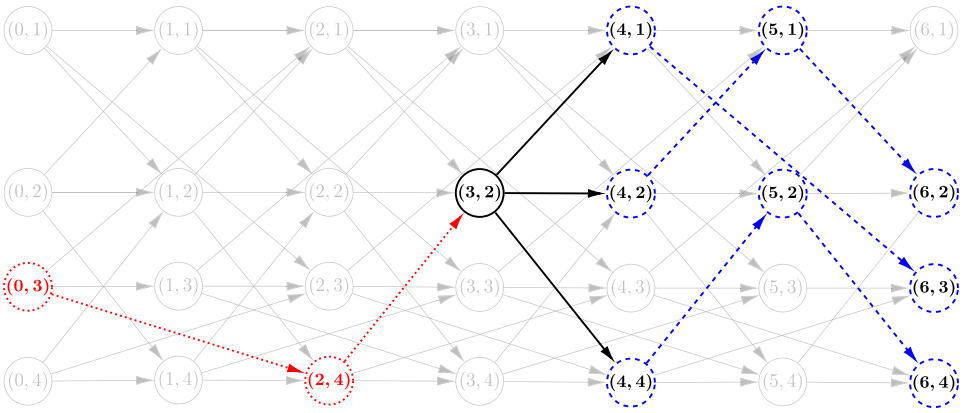


Fig. 7. Backward local search for (3,2).

updates the predecessor routes as well as the node with the highest value denoted by n^*. This process will continue until the relative gap between the highest node value of two consecutive forward local search is less than an improvement factor of $\zeta \in (0,1)$. At convergence, one can retrieve the best route as the one that passes by the node n^*. Since the initial solution for routing is constructed randomly, we repeat the entire process Π times, and choose the solution that yields the highest node value at convergence. In what follows, we describe the backward and forward local search procedures in more detail.

In the backward local search, we assume that the predecessor route of each node is fixed from the previous iteration. We visit nodes in the reversed topological order of $\mathcal{G}$. For all nodes n with no outbound edges, we initialize the successor route to include only node n. At every node $n = (t, i)$ that has outbound edges, we consider all node v's that can be visited immediately from node n in graph $\mathcal{G}$. We create a complete route $\mathcal{X}$ by connecting the predecessor route of node n and the successor route of node v, and apply the min-cost algorithm to find the optimal revenue for fixed route $\mathcal{X}$ given demand forecast $\boldsymbol{\Phi}$. Upon completing this task for all v's, we choose the best v and denote it by v^*. As a result, the successor route of node n can be created by appending node n to the successor route of v^*. The whole process will be repeated until we reach the nodes with time 0 in $\mathcal{G}$. The highest node value will be stored in $\varphi^{*\mathcal{B}}$. The forward local search will be implemented next, with the similar general structure except the fact that the successor routes are fixed and the predecessor routes will be updated in a topological order of nodes in $\mathcal{G}$.

Now, let us demonstrate the backward and forward routing algorithms in the context of our toy example. For this example, we assume that all edges in the time-expanded network are available. Further suppose that we are currently at iteration $\alpha = 1$, and we have implemented the backward routing algorithm until node $(3,2)$. Fig. 7 shows the step in which we want to decide about the node that the shuttle visits immediately after (3,2). There are three choices since $\delta^+((3,2)) = \{(4,1),(4,2),(4,4)\}$. The edges corresponding to these 3 options are shown in solid black lines. Also, assume that we have found the successor routes of (4,1), (4,2), and (4,4) in earlier steps as: $\mathcal{B}_1((4,1)) = \{(4,1),(6,3)\}$, $\mathcal{B}_1((4,2)) = \{(4,2),(5,1),(6,2)\}$ and $\mathcal{B}_1((4,4)) = \{(4,4),(5,2),(6,4)\}$. These three partial routes are shown in blue dashed lines. Finally assume that from the initial solution the predecessor route of (3,2) has been chosen as $\mathcal{F}_0(3,2) = \{(0,3),(2,4),(3,2)\}$. This partial route is shown with red dotted arrows. Using the connecting operator $\oplus$,

Algorithm 4: A local search for single shuttle routing

Input: Time-expanded network: $\mathcal{G}$
Demand forecast: $\boldsymbol{\Phi}$
Output: Local optimal shuttle route: $\mathcal{X}^*$
for $\pi = 1, 2, \cdots, \Pi$ **do**
  $\alpha \leftarrow 0$;
  $\Delta \leftarrow +\infty$;
  $\varphi_0^{*\mathcal{F}} \leftarrow 0$;
  $\mathcal{F}_0(n) \leftarrow$ Find an initial solution with $\mathcal{G}$ and $\boldsymbol{\Phi}$ (see Algorithm 5);
  repeat
    $\alpha \leftarrow \alpha + 1$;
    $\varphi_\alpha^{*\mathcal{B}}, \mathcal{B}_\alpha(n) \leftarrow$ Solve backward local search with $\mathcal{G}$ and $\boldsymbol{\Phi}$ and $\mathcal{F}_{\alpha-1}(n)$ (see Algorithm 6);
    $\varphi_\alpha^{*\mathcal{F}}, \mathcal{F}_\alpha(n), n^* \leftarrow$ Solve forward local search $\mathcal{G}$ and $\boldsymbol{\Phi}$ and $\mathcal{B}_\alpha(n)$ (see Algorithm 7);
    $\Delta \leftarrow \frac{\varphi_\alpha^{*\mathcal{F}} - \varphi_{\alpha-1}^{*\mathcal{F}}}{\varphi_\alpha^{*\mathcal{F}}}$;
  until $\Delta < \zeta$;
  $\mathcal{X}^* \leftarrow \mathcal{F}_\alpha(n^*) \oplus \mathcal{B}_\alpha(n^*)$;
Retrieve the best solution of Π replications ;

Algorithm 5: An Initial solution for routing

Input: Time-expanded Network: $\mathcal{G}$
Demand forecast: Φ
Output: Predecessor route of node $n \in \mathcal{N}$: $\mathcal{F}_0(n)$
for $q = 0, 1, \ldots, T$ **do**
  for $j \in S$ **do**
    $n \leftarrow (q, j)$;
    if $q = 0$ **then**
      $\mathcal{F}_0(n) \leftarrow [n]$;
      $\varphi_0(n) \leftarrow 0$;
    else
      $u = (t, i) \leftarrow$ draw randomly from $\delta^-(n)$;
      $\mathcal{F}_0(n) \leftarrow \mathcal{F}_0(u) \oplus [n]$;
      $\varphi_0(n) \leftarrow 0$;

Algorithm 6: Backward local search for routing

Input: Time-expanded Network: $\mathcal{G}$
Route of shuttle until node n: $\mathcal{F}_{\alpha-1}(n)$
Demand forecast: Φ
Output: Maximum revenue of shuttle at iteration α: $\varphi_\alpha^{*\mathcal{B}}$
Successor route of node $n \in \mathcal{N}$: $\mathcal{B}_\alpha(n)$
$\varphi_\alpha^{*\mathcal{B}} \leftarrow 0$;
for $t = T, T-1, \ldots, 0$ **do**
  for $i \in S$ **do**
    $n \leftarrow (t, i)$;
    if $\delta^+(n) = \varnothing$ **then**
      $\mathcal{B}_\alpha(n) \leftarrow [n]$;
      $\varphi_\alpha(n) \leftarrow 0$;
    else
      $\gamma^* \leftarrow -\infty$;
      for $v \in \delta^+(n)$ **do**
        $\mathcal{X} \leftarrow \mathcal{F}_{\alpha-1}(n) \oplus \mathcal{B}_\alpha(v)$;
        $\gamma \leftarrow$ Find the optimal revenue given route $\mathcal{X}$ and demand Φ (Algorithm 3);
        if $\gamma > \gamma^*$ **then**
          $\gamma^* \leftarrow \gamma$;
          $v^* \leftarrow v$;
      $\mathcal{B}_\alpha(n) \leftarrow [n] \oplus \mathcal{B}_\alpha(v^*)$;
      $\varphi_\alpha(n) \leftarrow \gamma^*$;
      if $\varphi_\alpha(n) > \varphi_\alpha^{*\mathcal{B}}$ **then**
        $\varphi_\alpha^{*\mathcal{B}} \leftarrow \varphi_\alpha(n)$;

we can construct three complete routes of $\mathcal{X}_1 = \{(0,3),(2,4),(3,2),(4,1),(6,3)\}$, $\mathcal{X}_2 = \{(0,3),(2,4),(3,2),(4,2),(5,1),(6,2)\}$, and $\mathcal{X}_3 = \{(0,3),(2,4),(3,2),(4,4),(5,2),(6,4)\}$. For every complete route, we solve the min-cost flow problem in Algorithm 3. Suppose we obtain three values of 25.5, 27, and 31.5. Hence, we let $\varphi((3,2)) = 31.5$ and set $\mathcal{B}_1((3,2)) = \{(4,4),(5,2),(6,4)\}$. We follow the same procedure for the rest of the nodes in the reverse topological order. Next, we start from node (0,1) and traverse nodes in the forward direction by applying the forward routing algorithm. Let us consider again node (3,2), but this time in the forward routing algorithm as shown in Fig. 8. Since $\delta^-((3,2)) = \{(2,1),(2,2),(2,4)\}$, we have three options for the node that the shuttle visits right before node (3,2). From earlier steps, assume that we have found $\mathcal{F}_1((2,1)) = \{(0,1),(1,2),(2,1)\}$, $\mathcal{F}_1((2,2)) = \{(0,2),(1,4),(2,2)\}$ and $\mathcal{F}_1((2,4)) = \{(0,3),(2,4)\}$. Also, we showed that from implementing the backward routing algorithm for (3,2), we got $\mathcal{B}_1((3,2)) = \{(3,2),(4,4),(5,2),(6,4)\}$. Using the connecting operator $\oplus$, we can construct three complete routes of $\mathcal{X}_1 = \{(0,1),(1,2),(2,1),(3,2),(4,4),(5,2),(6,4)\}$, $\mathcal{X}_2 = \{(0,2),(1,4),(2,2),(3,2),(4,4),(5,2),(6,4)\}$, and $\mathcal{X}_3 = \{(0,3),(2,4),(3,2),(4,4),(5,2),(6,4)\}$. For every complete route, we solve the min-cost flow problem in Algorithm 3. Suppose we obtain three values of 35, 37.5, and 31.5. Hence, we let $\varphi((3,2)) = 37.5$ and set $\mathcal{F}_1((3,2)) = \{(0,2),(1,4),(2,2),(3,2)\}$. We continue the whole process in Algorithm 4 until convergence.

Next, we present a lemma followed by a proposition that together prove the convergence of the local search algorithm.

Lemma 2. *At each step of the local search algorithm in 4, the objective function either improves or stays the same.*

Algorithm 7: Forward local search for routing

Input: Time-expanded Network: $\mathcal{G}$
Route of shuttle from node n: $\mathcal{B}_\alpha(n)$
Demand forecast: Φ
Output: Maximum revenue of shuttle at iteration α: $\varphi_\alpha^{*\mathcal{F}}$
Predecessor route of node $n \in \mathcal{N}$: $\mathcal{F}_\alpha(n)$
Node with the highest revenue: n^*

$\varphi_\alpha^{*\mathcal{F}} \leftarrow 0$;
for $j \in S$ **do**
  $n \leftarrow (q, j)$;
  if $\delta^-(n) = \varnothing$ **then**
    $\mathcal{F}_\alpha(n) \leftarrow [n]$;
    $\varphi_\alpha(n) \leftarrow 0$;
  else
    $\gamma^* \leftarrow -\infty$;
    for $u \in \delta^-(n)$ **do**
      $\mathcal{X} \leftarrow \mathcal{F}_\alpha(u) \oplus \mathcal{B}_\alpha(n)$;
      $\gamma \leftarrow$ Find optimal revenue given route $\mathcal{X}$ and demand requests in Φ (see Algorithm 3) ;
      if $\gamma > \gamma^*$ **then**
        $\gamma^* \leftarrow \gamma$;
        $u^* \leftarrow u$;
    $\mathcal{F}_\alpha(n) \leftarrow \mathcal{F}_\alpha(u^*) \oplus [n]$;
    $\varphi_\alpha(n) \leftarrow \gamma^*$;
    if $\varphi_\alpha(n) > \varphi_\alpha^{*\mathcal{F}}$ **then**
      $\varphi_\alpha^{*\mathcal{F}} \leftarrow \varphi_\alpha(n)$;
      $n^* \leftarrow n$;

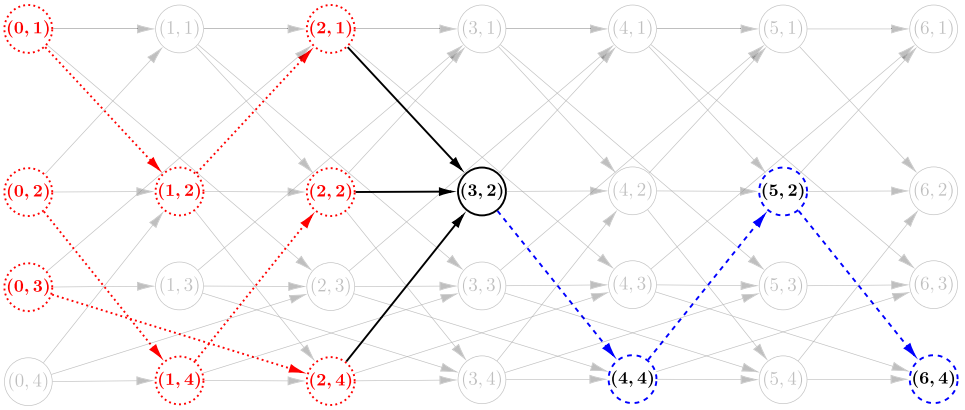


Fig. 8. Forward local search for (3,2).

Proof. At every move of the backward/Forward local search, we have the option to visit the same route or choose a node that results in a route with higher revenue. As a result the overall revenue either increases or stays the same at every move of the algorithm. □

Proposition 1. *Given an integer table of demand forecast, the objective, revenue, of the local search algorithm in 4 converges to a local optimum in polynomial time.*

Proof. (1) There are possibly many, but finite number of ways to serve requests in table Φ. Therefore, the solution set is finite; (2) based on Lemma 2, we infer that the solution always improves or stays the same, at which point we stop. Since the objective function is bounded from above, the algorithm is guaranteed to converge. From (1) and (2) the algorithm converges in finite number of iterations. In order to find a bound on the number of iterations before convergence, we assume that we iterate between the forward and backward local search only when $\varphi_\alpha^{*\mathcal{F}} - \varphi_{\alpha-1}^{*\mathcal{F}} \geq \zeta\ \varphi_\alpha^{*\mathcal{F}}$ where $\zeta \in (0, 1)$. Thus, for every two consecutive forward local searches, the inequality $\varphi_{\alpha-1}^{*\mathcal{F}} \leq (1-\zeta)\ \varphi_\alpha^{*\mathcal{F}}$ holds. If we start with the initial solution $\varphi_0^{*\mathcal{F}}$ and terminate with solution $\varphi_n^{*\mathcal{F}}$ after n iterations, then $\varphi_0^{*\mathcal{F}} \leq (1-\zeta)^n\ \varphi_n^{*\mathcal{F}} \leq (1-\zeta)^n\ \varphi^*$ where φ^* denotes the optimal node value. Since $1 + x \leq e^x$ for every $x \in \mathbb{R}$, we have

$\varphi_0^{*F} \leq e^{-\zeta n}\ \varphi^*$. This bounds the number of iterations n by $\mathcal{O}\big(\frac{1}{\zeta}\log(\frac{\varphi^*}{\varphi_0^{*F}})\big)$, which is polynomial. Moreover, at every forward/backward local search, we solve a min-cost flow problem per each edge of the time-expanded network. Thus, based on Remark 1, the worst-case running time of a forward/backward local search is $\mathcal{O}(|\mathcal{L}|\ Cx^2)$. Therefore, the entire local search algorithm in 4 has a polynomial time complexity of $\mathcal{O}\big(\frac{1}{\zeta}\log(\frac{\varphi^*}{\varphi_0^{*F}})\Pi\ |\mathcal{L}|\ Cx^2\big)$. Note that since φ^* is unknown, we can replace it with a valid upper bound $\bar{\varphi}$, which denotes the highest node value and can be calculated as $Cx\bar{\rho}$ where $\bar{\rho} = \max_{\forall\,(t,o,d)} \rho$. □

In Algorithm 1, we introduce a procedure to find the base routes. This procedure iterates over some clustering steps until the total revenue, z, converges. We finalize this section by presenting a Proposition regarding the convergence of z.

Proposition 2. *If we initiate local search algorithms in 4 from previous solutions, the procedure in Algorithm 1 converges to a fixed value in a finite number of steps.*

Proof. We know that the total revenue is bounded from above. Therefore, it suffices to show that in every step of our procedure, the value of z improves or stays the same. Let us assume any initial partitioning of passengers among shuttles. Based on Lemma 1, we know that the revenue of each shuttle improves or stays the same. Therefore, after applying the local search to each partition, the total revenue either improves or stays the same. Also, the distance function (3) ensures that any trip that can be served by a shuttle has the distance of zero. Therefore, the clustering problem in 1 assigns those requests that were served by every shuttle in the previous clustering step to the same shuttle. Now, if we incorporate the previous route of every shuttle in the initial solution of the local search algorithm in 4, we can ensure that revenue for each shuttle does not decrease. Finally, in every re-clustering step, we only assign the unserved requests to the next closest shuttle, which guarantees again that the shuttle can achieve the same revenue by only serving the requests from the previous step in the worst case. As a result, the revenue of all shuttles continuously improves and the result follows. □

4.2. Online routing

In this section, we introduce an online routing framework based on the results of the offline planning discussed in the previous section. Algorithm 8 provides a high-level description of the proposed method. Let us denote the simulation time by θ. Also let us introduce four tables of ψ_k, ψ^{ca}, ψ_k^{sc}, and ψ_k^{on}. Table ψ_k holds the expected future demand that can be served by shuttle k, and will be set to the average demand assigned to shuttle k in the offline phase, i.e., $\psi_k = \sum_{m=1}^{M} \Phi_m^k / M$. Note that ψ_k can be set to a high-quality demand forecast if one is available. Table ψ^{ca} holds the number of realized passengers that have arrived before time θ. The last two tables, ψ_k^{sc} and ψ_k^{on}, hold the number of scheduled and on-board passengers on shuttle k at time θ, respectively. The scheduled passengers are those whose demand has been realized and assigned to a shuttle in the past, but have not been picked up by the shuttle yet. The algorithm starts by initializing the route of each shuttle, denoted by $\mathcal{R}^k\ \forall\ k \in \mathcal{K}$, to the base route from the offline phase, i.e., $\mathcal{P}_1^k$. Also, we initialize the realized number of requests to 0. We set the simulation clock to 0 and increment it by 1 unit of time step until the end of the time horizon. At each simulation time, we repeat the following procedure.

We realize a set of passengers to arrive at the system during the interval $[\theta - 1, \theta]$ and update ψ^{ca}, accordingly. Similar to the offline phase, we adopt a multi-step clustering approach to distribute requests between shuttles. To this end, we define a proximity measure, denoted by $\chi^k(e,o,d)$, between trip (e,o,d) and shuttle k as follows:

$$\chi^k(e,o,d) = \frac{1}{M}\sum_{m=1}^{M} I\big(\text{route } \mathcal{P}_m^k \text{ can serve trip } (e,o,d)\big), \tag{4}$$

where $I(.)$ is an indicator function that takes the value 1 if the statement is true and 0 otherwise. This equation simply states that the proximity of a trip to a shuttle is equal to the probability of shuttle k serving this request using different alternative routes. After finding the proximity between all realized trips and shuttles, we construct the clustering problem in 5. Next, we repeat the following procedure for R times.

We first obtain the candidate requests for each shuttle, denoted by ψ_k^{ca}, by solving the clustering problem in 5. For every shuttle k, we find the next stop on its route, denoted by n_y. Also, we update on-board and scheduled passengers from the previous simulation time step. Next, a cost–benefit analysis will be conducted (Algorithm 9) to find the set of realized passengers that can be served by this shuttle, taking into account the possibility of changing the shuttle's route at node n_y. This can be accomplished by finding all possible alternative routes for shuttle k that intersect at node n_k and satisfy the time windows and origin/destination stations of on-board and scheduled passengers, hereafter refereed to as the set of feasible alternatives. After finding the best feasible alternative, if this is the last clustering step ($r = R$), we change the route of the shuttle and update the time windows of on-board and scheduled passengers; Otherwise, we add a number of constraints to the problem in stmxrefeq:cluster(5) to make sure the unserved requests will be assigned to the next closest shuttle.

We implement Algorithm 9 on the set of feasible alternatives to carry out a cost–benefit analysis. The result of this cost–benefit analysis determines the alternative route the shuttle follows. In this algorithm we first find the pick-up/drop-off nodes of on-board and scheduled passengers. Next, we construct a graph similar to the one in the offline phase, with a few changes. In the online graph, we create three copies of each node in route $\mathcal{X} = \{n_y, \dots, n_i, \dots, n_x\}$, denoted by $n_i^{\mathrm{d}}, n_i^{\mathrm{p1}}$, and n_i^{p2}. The first copy accounts for the drop-off at node n_i, $\forall\ i :\ \ y \leq i \leq x$. The next two copies model the pick-up of the realized and predicted passengers at node n_i, respectively. In addition, we multiply the cost on edges of type $(n_i^{\mathrm{p2}}, n_i^{\mathrm{d}})$ by a factor $\beta \in (0,1)$ to take into account the fact that

Algorithm 8: Online framework

Input: Pool of Shuttle routes: $\mathcal{P}$
Average demand forecast for Shuttle k: $\psi_k \quad \forall k \in \mathcal{K}$
Output: Optimal revenue of Shuttle: $\mathcal{Z}$
$\mathcal{R}^k \leftarrow \mathcal{P}_1^k, \quad \forall k \in \mathcal{K}$;
$\psi^{ca}(e,o,d) \leftarrow 0 \quad \forall e \in \mathcal{T}, \forall o \in \mathcal{S}, \forall d \in \mathcal{S}$;
$\psi_k^{sc}(e,o,d) \leftarrow 0 \quad \forall k \in \mathcal{K}, \forall e \in \mathcal{T}, \forall o \in \mathcal{S}, \forall d \in \mathcal{S}$;
$\psi_k^{on}(e,o,d) \leftarrow 0 \quad \forall k \in \mathcal{K}, \forall e \in \mathcal{T}, \forall o \in \mathcal{S}, \forall d \in \mathcal{S}$;
for $\theta = 0, 1, \dots, T$ **do**
  $\psi_k(e,o,d) \leftarrow 0 \quad \forall k \in \mathcal{K}, \forall e \in \{0, \dots, \theta\}, \forall o \in \mathcal{S}, \forall d \in \mathcal{S}$;
  $\psi^{ca} \leftarrow$ Update realized requests based on passenger arrivals ;
  $\chi^k(e,o,d) \leftarrow$ Find the proximity between every trip (e,o,d) in ψ^{ca} and base route $\mathcal{P}_1^k, \quad \forall k \in \mathcal{K}, \forall e \in \mathcal{T}, \forall o \in \mathcal{S}, \forall d \in \mathcal{S}$;
  Create the clustering problem:

$$\max \quad \sum_{k=1}^{K} \sum_{(e,o,d)} \chi^k(e,o,d)\, \psi_k^{ca}(e,o,d) \tag{5a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \psi_k^{ca}(e,o,d) = \psi^{ca}(e,o,d) \qquad \forall k \in \mathcal{K}, \forall e \in \mathcal{T}, \forall o \in \mathcal{S}, \forall d \in \mathcal{S} \tag{5b}$$

  for $r = 1, 2, \dots, R$ **do**
    $\psi_k^{ca} \leftarrow$ Distribute demand between Shuttles by solving the clustering problem in 5;
    for $k \in \mathcal{K}$ **do**
      $n_y \leftarrow$ Find the next node to be visited on $\mathcal{R}^k$;
      $\psi_k^{on}, \psi_k^{sc} \leftarrow$ Update on-board and scheduled passengers on Shuttle k;
      $\gamma^* \leftarrow 0$;
      for $m = 1, 2, \dots, M$ **do**
        $n_{y'} \leftarrow$ Find the next node to be visited on $\mathcal{P}_m^k$;
        if $n_y = n_{y'}$ **then**
          if Route $\mathcal{P}_m^k$ satisfies passengers in ψ_k^{on} and ψ_k^{sc} **then**
            $\mathcal{X} \leftarrow$ Route $\mathcal{P}_m^k$ from $n_{y'}$;
            $\gamma, f_k^{ca} \leftarrow$ Find optimal revenue given $\mathcal{X}$, ψ_k, ψ_k^{ca}, ψ_k^{on}, ψ_k^{sc} (see Algorithm 9);
            if $\gamma > \gamma^*$ **then**
              $\gamma^* \leftarrow \gamma$; $m^* \leftarrow m$; $f_k^{ca^*} \leftarrow f_k^{ca}$;
      if $r = R$ **then**
        $\mathcal{R}^k \leftarrow$ Update route of Shuttle k based on route $\mathcal{P}_{m^*}^k$ from n_y;
        Update pick-up and drop-off times of on-board and scheduled ;
      else
        Add $\psi_k^{ca}(e,o,d) \le f_k^{ca}(e,o,d)$ to the clustering problem in 5, $\quad \forall e \in \mathcal{T}$, $\forall o \in \mathcal{S}, \forall d \in \mathcal{S}$;
        $\mathcal{Z} \leftarrow \mathcal{Z} + \gamma^*$;

traversing these edges corresponds to serving demand that is only expected, and has not yet been realized. Finally, we alter the demand of origin and destination nodes of on-board and scheduled customers to ensure that these customers will be picked up by the min-cost flow algorithm. More precisely, we add the number of on-board/scheduled customers to the demand of their origin nodes and subtract it from the demand of their destination nodes.

Let us demonstrate a small instance of this cost–benefit analysis using our toy example. Suppose that we have found two alternative routes of $\mathcal{P}_1^1 = \{(0,2),(1,1),(2,2),(3,1),(4,2),(5,4),(6,2)\}$ and $\mathcal{P}_2^1 = \{(0,2),(1,4),(2,2),(3,4),(4,2),(5,1),(6,2)\}$ for our shuttle in the offline phase. These routes are shown in Fig. 9 with dashed and solid lines, respectively. Suppose that the current simulation time is $\theta = 2$, and the current route of the shuttle is $\mathcal{R} = \mathcal{P}_1^1 = \{(0,2),(1,1),(2,2),(3,1),(4,2),(5,1),(6,2)\}$. Suppose that we want to evaluate the revenue of the shuttle by changing the route to $\mathcal{P}_2^1$ at node $(2,2)$. Assume that the shuttle has already picked up 2 customers at time 1 from station 1, whose destination station is 4. Also, 7 customers are scheduled to be picked up at time 2 from station 2, where 5 of them want to go to station 4 and the rest want to go to station 1. Also, at this simulation time, $\theta = 2$, the system has received 10 trip requests at station 2, of which 6 go to station 4 and the rest go to station 1, and 5 trip requests at station 4 that go station 2. This information is summarized in Table 4. In Table 4, the on-board and scheduled customers are specified in red (the first number in each cell) and candidate customers are specified in black (the second number in each cell). Blue numbers (the third number in each cell) indicate the expected number of requests based on historical data.

We can model this online problem as a min-cost flow problem as described in Algorithm 9 and shown in Fig. 9. As such, we define 3 distinct nodes in (2,2) for drop-off (black dashed circle), candidates' pick-up (black solid circle), and expected customers' pick-up (blue solid circle). Similarly, we define 3 distinct nodes in (3,4), (4,2), and (5,1). Finally, we define a single drop-off circle for node (6,2). Similar to the offline phase, the clique edges are shown with dashed lines and the interval edges with solid lines. However, there is a difference in the online phase when specifying the cost of interval edges. For expected customers, we multiply

Algorithm 9: Find optimal revenue in real-time

Input: Fixed Route of shuttle: $\mathcal{X} = [n_y, n_{y+1}, n_{y+2}, \dots, n_x]$
Demand forecast: ψ_k
Realized demand: ψ_k^{ca}
On-board passengers count: ψ_k^{on}
Scheduled passengers count: ψ_k^{sc}
Output: Optimal revenue of shuttle: γ
Optimal number of served passengers from realized demand: f^{ca}

pick-up times of on-board passengers $\leftarrow n_y^{\mathsf{P}_1}$;
Find drop-off times of on-board passengers based on route $\mathcal{X}$;
Find pick-up and drop-off times of scheduled passengers based on route $\mathcal{X}$;
$\mathcal{N} \leftarrow \{n_y^{\mathsf{d}}, n_y^{\mathsf{P}_1}, n_y^{\mathsf{P}_2}, n_{y+1}^{\mathsf{d}}, n_{y+1}^{\mathsf{P}_1}, n_{y+1}^{\mathsf{P}_2}, \dots, n_x^{\mathsf{d}}, n_x^{\mathsf{P}_1}, n_x^{\mathsf{P}_2}\}$;
$\mathscr{D}(n_y^{\mathsf{d}}) \leftarrow -C$; $\mathscr{D}(n_x^{\mathsf{P}}) \leftarrow C$; $\mathscr{D}(n_i^{\mathsf{d}}) \leftarrow 0 \,\forall\, i \in \{y+1, \dots, x\}$; $\mathscr{D}(n_i^{\mathsf{P}}) \leftarrow 0 \,\forall\, i \in \{y, \dots, x-1\}$
$\mathscr{L} \leftarrow \{\}$;
for $i = y, y+1, \dots, x$ **do**
  $\mathscr{L} \leftarrow \mathscr{L} \cup \{n_i^{\mathsf{d}}, n_i^{\mathsf{P}_1}\}$; $\mathscr{W}(n_i^{\mathsf{d}}, n_i^{\mathsf{P}_1}) \leftarrow 0$; $\mathscr{U}(n_i^{\mathsf{d}}, n_i^{\mathsf{P}_1}) \leftarrow C$;
  $\mathscr{L} \leftarrow \mathscr{L} \cup \{n_i^{\mathsf{P}_1}, n_i^{\mathsf{P}_2}\}$; $\mathscr{W}(n_i^{\mathsf{P}_1}, n_i^{\mathsf{P}_2}) \leftarrow 0$; $\mathscr{U}(n_i^{\mathsf{P}_1}, n_i^{\mathsf{P}_2}) \leftarrow C$;
  $\mathscr{L} \leftarrow \mathscr{L} \cup \{n_i^{\mathsf{d}}, n_{i+1}^{\mathsf{d}}\}$; $\mathscr{W}(n_i^{\mathsf{d}}, n_{i+1}^{\mathsf{d}}) \leftarrow 0$; $\mathscr{U}(n_i^{\mathsf{d}}, n_{i+1}^{\mathsf{d}}) \leftarrow C$;
for $i = y, y+1, \dots, x-1$ **do**
  for $j = i+1, i+2, \dots, x$ **do**
    $(p, o) \leftarrow n_i^{\mathsf{P}_1}$; $(l, d) \leftarrow n_j^{\mathsf{d}}$;
    if d not visited again between n_i^{P} and n_j^{d} **then**
      if $(l - p) \le \tau(p, o, d) + \Omega$ **then**
        $\mathscr{L} \leftarrow \mathscr{L} \cup \left\{n_i^{\mathsf{P}_1}, n_j^{\mathsf{d}}) \cup (n_i^{\mathsf{P}_2}, n_j^{\mathsf{d}}\right\}$;
        $\mathscr{D}(n_i^{\mathsf{d}}) \leftarrow \mathscr{D}(n_i^{\mathsf{d}}) + \psi_k^{sc}(p, o, d) + \psi_k^{on}(p, o, d)$;
        $\mathscr{D}(n_j^{\mathsf{d}}) \leftarrow \mathscr{D}(n_j^{\mathsf{d}}) - \psi_k^{sc}(p, o, d) + \psi_k^{on}(p, o, d)$;
        if o visited again between n_i^{P} and n_j^{d} **then**
          $\mathscr{U}(n_i^{\mathsf{P}_1}, n_j^{\mathsf{d}}) \leftarrow \psi_k^{ca}(p, o, d)$;
          $\mathscr{U}(n_i^{\mathsf{P}_2}, n_j^{\mathsf{d}}) \leftarrow \psi_k(p, o, d)$;
        else
          $t_0 \leftarrow$ Last time o visited in time window $[p - \omega, p]$;
          $\mathscr{U}(n_i^{\mathsf{P}_1}, n_j^{\mathsf{d}}) \leftarrow \sum_{t=t_0+1}^{\mathsf{p}} \psi_k^{ca}(t, o, d)$;
          $\mathscr{U}(n_i^{\mathsf{P}_2}, n_j^{\mathsf{d}}) \leftarrow \sum_{t=t_0+1}^{\mathsf{p}} \psi_k(t, o, d)$;
        $\mathscr{W}(n_i^{\mathsf{P}_1}, n_j^{\mathsf{d}}) \leftarrow -\rho(p, o, d)$; $\mathscr{W}(n_i^{\mathsf{P}_2}, n_j^{\mathsf{d}}) \leftarrow -\rho(p, o, d) \times \beta$;
Define graph $\mathscr{G} = (\mathcal{N}, \mathscr{L}, \mathscr{D}, \mathscr{W}, \mathscr{U})$;
Solve min-cost flow on $\mathscr{G}$;
$f^{ca} \leftarrow$ Optimal flow on edges of type $(n_i^{\mathsf{P}_1}, n_j^{\mathsf{d}})$;
$\gamma \leftarrow -1\times$ Optimal flow cost ;

the original cost by a discount factor, β, to account for the fact that these requests are only expected and have not been realized yet. In addition, we initially set the demand of the drop-off nodes at (2,2), (3,4), (4,2), (5,1), and (6,2) to −10, 0, 0, 0, and 10, respectively. Next, we alter these values to make sure that the optimal flow satisfies the on-board and scheduled customers. As such, we add 9 units to the demand of the drop-off node at (2,2), because all the on-board and scheduled customers start from this node. Next, we subtract 7 units from the demand of the drop-off node at (3,4) and , since 7 customers will be dropped off at this node. Finally, we subtract 2 units from the demand of the drop-off node at (5,1) where the rest of customers will be dropped off. As a result, the demand of the drop-off nodes at (2,2), (3,4), (4,2), (5,1), and (6,2) are set to −1, −7, 0, −2, and +10, respectively. As shown in 10(b), solving the min-cost flow problem in this graph yields an optimal revenue of 17.4 given $\beta = 0.5$. If we follow the same procedure for the original route, $\mathcal{P}_1^1$, we obtain an optimal revenue of 14.5 for the same value of β. Therefore, our analysis suggests that the shuttle is better off by changing its route from $\mathcal{P}_1^1$ to $\mathcal{P}_2^1$ for the rest of its journey.

5. Numerical experiment

In this section, we showcase the performance of our framework under two different case studies. In the first case study, we consider a simulated ridesharing dataset over a well-known small transportation network, namely the Nguyen–Dupuis network. We further conduct an extensive sensitivity analysis on different parameters in both the offline and the online phases of the framework. Next, in order to showcase the scalability of our framework, we evaluate the performance of the proposed method in serving the taxi trips in the Manhattan area using the 2016 New York Taxi dataset.

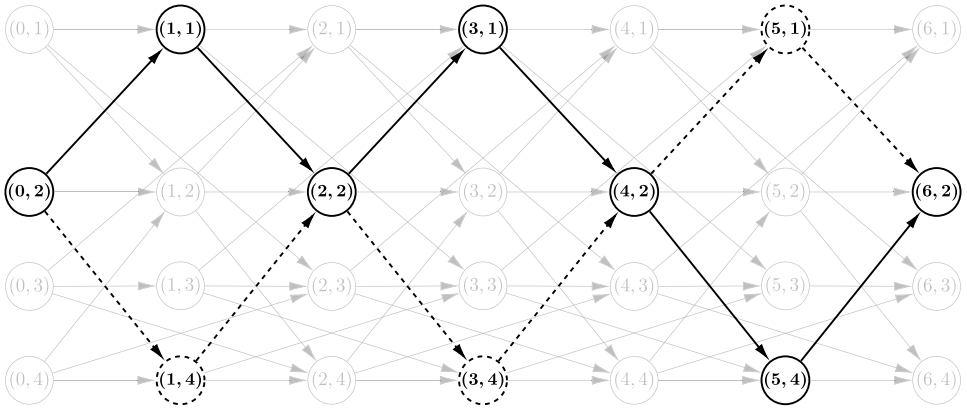
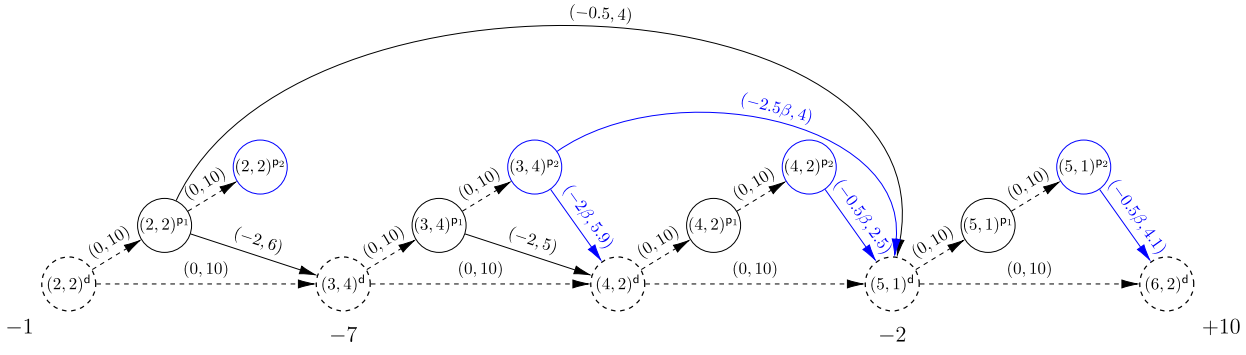
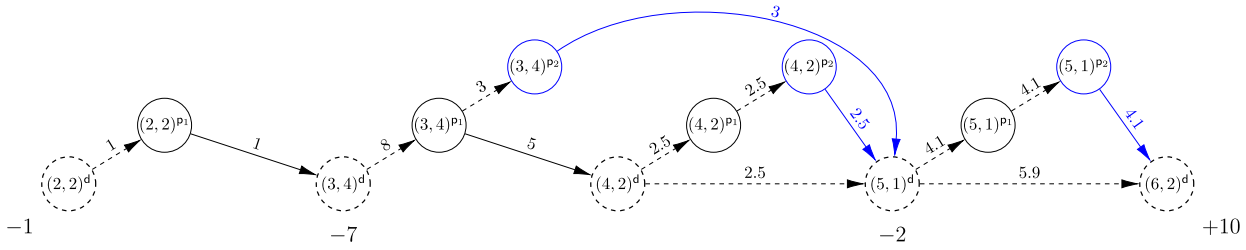


Fig. 9. The current and alternative route for the shuttle.



(a) min-cost flow problem



(b) optimal flows with the optimal revenue of 17.4 given $\beta = 0.5$

Fig. 10. Min-cost flow for path $\mathcal{P}_2^1$ at $\theta = 2$ and its optimal solution. The black dashed circles model drop-offs, the black solid circles model candidate customers' pick-ups, and the blue solid circles model the expected pick-ups. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
The number of on-board/scheduled customers (in red), realized customers (in black), and expected number of requests (in blue) for path $\mathcal{P}_2^1$ at $\theta = 2$.

	(2,2)	(3,4)	(4,2)	(5,1)	(6,2)
(0,2)	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
(1,1)	0, 0, 0	2, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
(2,2)		5, 6, 0	0, 0, 0	2, 4, 0	0, 0, 0
(3,4)			0, 5, 5.9	0, 0, 5.9	0, 0, 0
(4,2)				0, 0, 2.5	0, 0, 0
(5,1)					0, 0, 4.1

In all case studies, we compare the performance of our algorithm under three different configurations with two well-known state-of-the-art models proposed by Ma et al. (2013) and Alonso-Mora et al. (2017), which are modified based on the assumptions

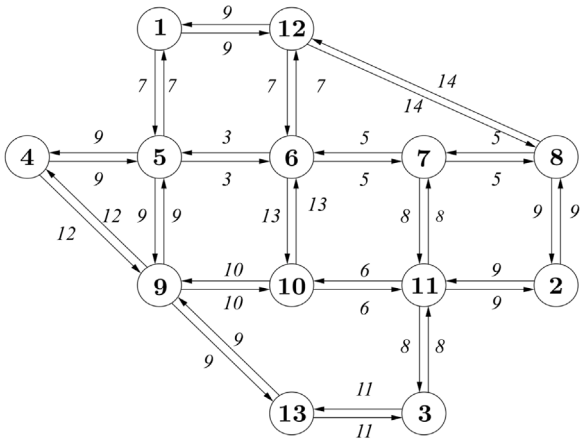


Fig. 11. Nguyen–Dupuis network.

stated in Section 3. In the first configuration, we consider a full version of our method with each shuttle having 50 alternative routes. We use the average rate of demand ϕ, obtained from historical data, as our demand prediction, with the influence factor of β. In the second configuration, we still use alternative routes for shuttles, but we assume that there is no future demand forecast, i.e., $\beta = 0$. Finally, in the third configuration, we assume that neither the alternative routes nor the demand forecast is available. This configuration leads to a system that resembles fixed-route transit. Hereafter, we refer to these three configurations as the "Full", "No Pred.", and "No Alter." methods, respectively. The T-share insertion method proposed by Ma et al. (2013) and the dynamic trip-shuttle assignment model proposed by Alonso-Mora et al. (2017), which are hereinafter referred to as "Insertion" and "Assignment", respectively, serve as benchmarks for performance evaluation. In order to evaluate the performance of these methods, we introduce the following measures:

- $\mathcal{Z}$: Total revenue of all shuttles (in miles)
- $\mathcal{M}$: Matching rate of customers served by all shuttles
- $\mathcal{W}$: Waiting time of served customers averaged over all shuttle and customers (in minutes)
- $\mathcal{D}$: Detour time of served customers averaged over all shuttle and customers (in minutes)
- $\mathcal{U}$: Utilization rate of shuttles' seats averaged over all shuttle and time steps
- $\mathcal{A}$: Number of times shuttles switch their routes averaged over all shuttle (only for Full and No Pred. methods)
- Θ: Average computation time (in milliseconds) of online framework over all time steps

In the following subsections, we denote the online performance measures corresponding to Full, No Pred., No Alter., Insertion and Assignment methods by indices 1, 2, 3, 4, and 5, respectively. We also show the value of the parameter β in the full configuration by a superscript for these measures. For instance, $\mathcal{Z}_1^{0.5}$ denotes the total revenue of shuttles using the full configuration of our proposed method with $\beta = 0.5$.

All experiments are conducted on a 56-core 3.50 GHz Intel Xeon machine with a 64-bit version of the Windows 10 operating system with 256.0 GB of RAM. All methods are coded in `Python 3.7` and min-cost flow problems are solved using the `OR-Tools` package. Next, we present the experiment setups and the results of various experiments for the two case studies.

5.1. Nguyen-Dupuis case study

The Nguyen–Dupuis network consists of 13 nodes, 38 links, and 13 OD pairs as shown in Fig. 11. The italic number on every edge shows the constant link travel time in minutes. For this network, we assume a constant speed of 15 miles per hour on every link. Hence, the distance in miles between every two link can be found as $\rho(t, o, d) = 0.25 \times \tau(t, o, d),\ \forall\, t \in \mathcal{T}, \forall\, o \in \mathcal{S}, \forall\, d \in \mathcal{S}$.

We further assume that a fleet of 50 homogeneous shuttles with capacity 10 is available. For large-scale implementation, we assume a study time horizon of 60 min, with time steps of 1 min. We further assume that from historical data, the arrival of demand requests for each time step and OD pair follows a Poisson distribution with average rate being stored in ϕ presented in Table B.1. Note that we assume that ϕ only contains the average count of passengers that arrived at the system and could be served during the given time horizon. The expected total number of trips is assumed to be 1000 and the arrival time of a request and its origin and destination are determined based on a uniform random distribution. We also set the waiting time ω and the detour budget time Ω to 5 and 10 min, respectively.

5.1.1. Results

For generating base and alternative routes of each shuttle, we implemented the offline framework in Algorithm 1 with $M = 50$, $R = 5$, and $\epsilon = 10$ minutes. In other words, after finding the base routes, the time-expanded network for each shuttle was reduced to a sub-graph whose nodes are at most 10 min away from the closest node on its base route. Next, we implemented demand clustering

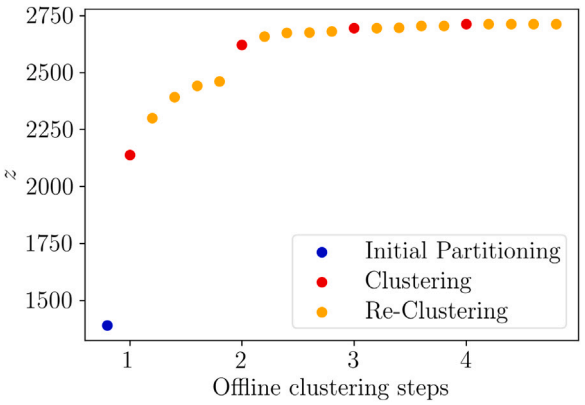


Fig. 12. Convergence of the total revenue for the base routes in the offline phase of the Nguyen Dupuis case study.

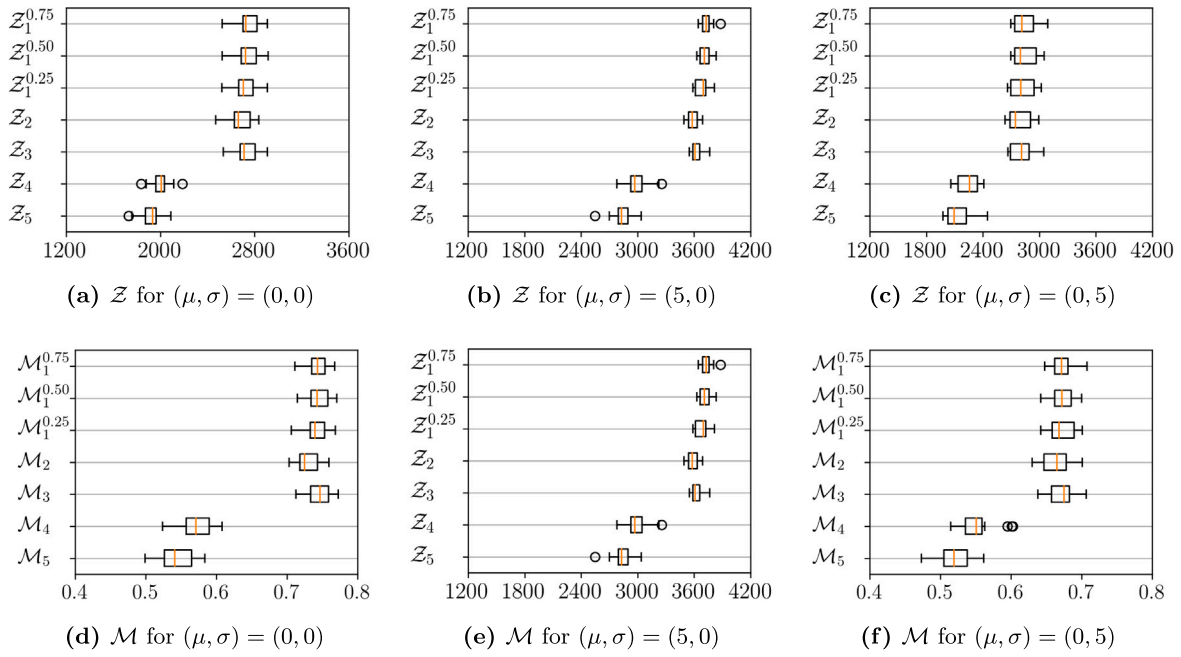


Fig. 13. The result of different methods on the performance measures in the Nguyen Dupuis case study.

and applied Algorithm 4 (for five iterations) in the reduced graphs to obtain 49 alternative routes per base route using different realizations of a Poisson distribution with mean ϕ. We set the number of replications, Π, and improvement factor, ζ, in the local search algorithm to 10 and 0.01, respectively. Thus, for finding each route, we continue the local search until the relative gap is less than 0.01, and we repeat the entire process with 10 initial solutions and pick the one that yields the highest revenue. Fig. 12 shows the convergence of the total revenue for the base routes, z, in the offline phase as stated in Proposition 2. This figure clearly shows that the total revenue increase at every step of the clustering and re-clustering approach outlined in Algorithm 1 until it converges to a fixed value.

In the online phase, we consider 3 configurations of our proposed method and compare their results with the two benchmark methods. In these experiments, we define two additional parameters, μ and σ, to generate scenarios where the actual demand is a shifted (by μ) or scaled (by σ) version of values in ϕ. For instance, a scenario with $\mu = 0$, $\sigma = 5$, means that we simulate customer arrivals based on a Poisson distribution with the rate of $\phi' \sim N(\phi + 0, 5)$.

The box plots in Fig. 13 demonstrate the revenues and matching rates of experiments for different methods under the three demand scenarios. Also, Table 5 summarizes the average revenue and matching rate over 20 simulation runs. In order to evaluate the significance of differences in the values of this table, we conducted paired t-test between all pairwise combinations of methods under the three scenarios, the p-value of which are summarized in Table E.1. Table 5 shows that our proposed method significantly outperforms both benchmark methods. This table also suggests that in cases where the expected demand is close to the realized one,

Table 5
The results of different methods on the performance measures averaged over simulation runs in the Nguyen–Dupuis case study.

μ	σ	β	Full		No Pred.		No Alter.		Insertion		Assignment	
			$\overline{\mathcal{Z}}_1$	$\overline{\mathcal{M}}_1$	$\overline{\mathcal{Z}}_2$	$\overline{\mathcal{M}}_2$	$\overline{\mathcal{Z}}_3$	$\overline{\mathcal{M}}_3$	$\overline{\mathcal{Z}}_4$	$\overline{\mathcal{M}}_4$	$\overline{\mathcal{Z}}_5$	$\overline{\mathcal{M}}_5$
0	0	0.95	2748.90	0.74	2677.08	0.73	2731.89	0.75	2003.29	0.57	1914.25	0.54
		0.50	2741.19	0.74								
		0.25	2725.04	0.74								
5	0	0.95	3729.16	0.56	3583.72	0.55	3629.71	0.56	2997.8	0.47	2835.92	0.44
		0.50	3713.06	0.56								
		0.25	3677.31	0.57								
0	5	0.95	2847.92	0.67	2785.86	0.66	2804.75	0.67	2242.68	0.55	2137.66	0.52
		0.50	2842.04	0.67								
		0.25	2823.32	0.67								

Table 6
Impact of the spatio-temporal threshold, ϵ, on the average revenue, $\mathcal{Z}$, in the Nguyen–Dupuis case study.

μ	σ	β	Full			No Pred.			No Alter.	Insertion	Assignment
			$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$			
0	0	0.95	2745.39	2748.90	2740.58	2687.34	2677.08	2690.22	2732.71	2003.29	1905.72
		0.50	2736.75	2741.19	2733.26						
		0.25	2712.49	2725.04	2708.52						
5	0	0.95	3701.12	3729.16	3720.26	3592.11	3583.72	3554.56	3628.69	2997.80	2851.28
		0.50	3676.54	3713.06	3694.45						
		0.25	3626.26	3677.31	3638.58						
0	5	0.95	2838.06	2847.92	2840.58	2765.89	2785.86	2760.79	2807.85	2242.68	2138.80
		0.50	2836.90	2842.04	2828.25						
		0.25	2810.98	2823.32	2801.10						

it is more appropriate to stay on the base route, as $\mathcal{Z}_3$ yields high revenue in the first scenario of μ and σ. However, as the mean or variance of the realized demand deviates from the expected demand, we may benefit from switching to alternative routes. In this table, the best value of β is embolden for each scenario. Although the highest value of β in all scenarios has been shown to provide the highest average revenue, the p-values in Table E.1 indicate that the right choice of its value depends on the performance of our demand forecast. More specifically, high values of β provide higher revenues when the forecast is accurate (scenario 1). However, as the mean/variance of demand forecast deviates further from the mean/variance of the realized demand, lower values of β become comparable or even preferable.

This table also suggests that the result of the Insertion method surprisingly outperforms that of the Assignment method, even through the latter considers all combinations of trip sharing and assigns the shuttles to the best combinations optimally at any time step. The reason behind this observation resides in the fact that assigning customers to shuttles optimally without considering the future unobserved demand may increase the level of nearsightedness of a method to the point that a FCFS insertion method can outperform it. Overall, these experiments suggests that the Full configuration of our method is the most robust strategy in a stochastic and dynamic environment although the margin of its outperformance start to shrink by deviating from a perfect demand forecast.

5.1.2. Sensitivity analysis

In this section, we study the impact of ϵ by repeating the experiments with three values of 5, 10, and 15 min for this parameter. Tables 6, 7, C.1, C.2, C.3, C.4, and C.5 summarize the impact of changing ϵ on the performance measures averaged over 20 simulation runs. Also, Table E.3 contains the p-value of the pairwise comparisons between all combinations of ϵ for all measures, methods, and demand scenarios.

Table 6 suggests that for a given number of alternative routes, as ϵ increases, the total revenue of the Full method first increases and then it starts to decrease. This is due to the fact that although increasing ϵ allows the alternatives of a shuttle to cover a larger region, this happens at the price of having smaller number of similar spatio-temporal nodes between alternative routes. Thus, the chance of serving on-board or scheduled customers decreases as we increase the value of ϵ. Also, decreasing ϵ to a very small value prevents the shuttle to explore different regions, and hence, results in alternatives serving similar customers. It is worth mentioning that Table E.3 does not show a statistically significant difference between different values of epsilon for scenarios 1 and 3.

Interestingly, Table E.3 shows a different trend for the No Pred. method. More specifically, it indicates that the lower the value of ϵ, the higher the revenue. This is mainly due to the fact that using alternative routes without predicted future demand causes the online cost–benefit analysis fail to find the best choice for the remaining part of a shuttle's trajectory. Overall, smaller choice of ϵ is preferred when we have access to a high-quality demand forecast. However, note that we may need to consider larger values of ϵ when the mean or variance of realized demand differs from what we predicted. Table 7 indicates that ϵ does not seem to have a major impact on matching rates of the Full and No Pred. methods.

Table 7
Impact of the spatio-temporal threshold, ϵ, on the average matching rate, $\mathcal{M}$, in the Nguyen–Dupuis case study.

μ	σ	β	Full			No Pred.			No Alter.	Insertion	Assignment
			$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$			
0	0	0.95	0.74	0.74	0.74	0.73	0.73	0.73	0.75	0.57	0.54
		0.50	0.74	0.74	0.74						
		0.25	0.74	0.74	0.74						
5	0	0.95	0.56	0.56	0.56	0.55	0.55	0.55	0.56	0.47	0.44
		0.50	0.56	0.56	0.56						
		0.25	0.56	0.57	0.56						
0	5	0.95	0.67	0.67	0.67	0.66	0.66	0.66	0.67	0.55	0.52
		0.50	0.67	0.67	0.67						
		0.25	0.67	0.67	0.67						

Table 8
The result of different methods on the performance measures averaged over simulation runs in the Manhattan case study.

Method	$^*\beta$	$\overline{z}$	$\overline{\mathcal{M}}$	$\overline{\mathcal{A}}$	$\overline{w}$	$\overline{D}$	$\overline{U}$	$\overline{\pi}$
Full	0.75	15295.24	0.19	3.31	2.73	2.07	0.69	916.99
	0.50	15380.86	0.20	3.62	2.84	2.09	0.70	896.05
	0.25	15300.37	0.20	4.31	2.93	2.07	0.70	827.80
No Pred.	–	15093.66	0.21	2.91	2.96	2.05	0.69	544.10
No Alter.	–	15181.47	0.21	–	2.94	1.98	0.69	76.08
Insertion	–	10948.23	0.13	–	2.75	3.25	0.51	714.04
Assignment	–	11331.41	0.13	–	2.97	3.26	0.52	107.82

The impact of ϵ on the rest of measures has been presented in Appendix C. In the online experiments above, we assumed that customer arrivals followed a Poisson distribution with rate ϕ. In order to see the impact of arrival distribution, we repeat the experiments by assuming a Uniform distribution for the arrival of customers over time, and compare its impact on different measures with those of Poisson distribution. The results of this analysis can be found in Appendix D.

5.2. Manhattan case study

In order to showcase the scalability of the framework, we conduct an experiment in the Manhattan area in New York City (NYC) using the NYC Taxi dataset. The dataset contains the longitude and latitude of trip origins and destinations and the pick-up times for all trips. We define 184 stations that are strategically positioned in the Manhattan area to ensure there is at least one station within the walking distance of a typical trip origin/destination (see Fig. 14). The shortest-path driving distances and travel times between all stations are obtained from the Google API.

In this case study, we assume that the shuttle service company possesses a fleet of 100 shuttles with capacity of 10 that operate during the evening pick hours in the Manhattan area. All shuttles start their trips from a station in the center of the region (the black station in Fig. 14) at 18:00, and end their trips in the same station at 21:00. As such, we obtain a historical demand table ϕ by taking the average of trip counts for every minute and origin/destination stations from July 2015 to May 2016. We further reserve the demand of June 2016 for running simulations in the online framework. In the offline phase, we construct 50 alternative routes for each shuttle by assuming a Poisson distribution for the arrival of customers with rate of ϕ, and setting the parameters ϵ, R, Π, and ζ to 15 min, 5, 100, and 0.01, respectively.

5.2.1. Results

The result of different methods on performance measures in the online experiments are shown in Fig. 15. The average of these measures are summarized in Table 8. Table E.2 provides the p-values of the corresponding paired t-tests for all pairwise comparisons. The presented performance measures in Table 8 are obtained by averaging the results of 30 days of simulation. Moreover, we considered three different values for the influence factor β in the full configuration of our proposed method. In terms of total revenue, this table clearly shows that our proposed methodology significantly outperforms the two myopic benchmark methods, as the revenue of our method (independent of the choice of configuration) is at least 34% larger than those of the benchmark methods. The main reason behind this substantial difference is that the benchmark methods tend to aggressively respond to the realized demand without considering the unobserved future demand. Therefore, a high percentage of future demand is likely to be rejected due to the limited capacity of shuttles, especially in circumstances with high demand rate. We can also infer that both allowing for changing routes and incorporating expected demand can lead to a slight (but statistically significant) increase in revenue.

Table E.2 also shows that 100 shuttles are able to serve 20 percent of all requests (around 50 000 on average) under our proposed method, which is at least 7% larger than those of the benchmark methods. This table also indicates that we are more likely to switch a shuttle's route when using the expected demand. The highest value of β that yields the highest revenue for the full method indicates that our demand forecast is not near perfect. That is why shuttles can improve their revenues by diverging from the base routes to follow alternative routes. However, the comparison between the revenues of different configurations suggests that allowing for

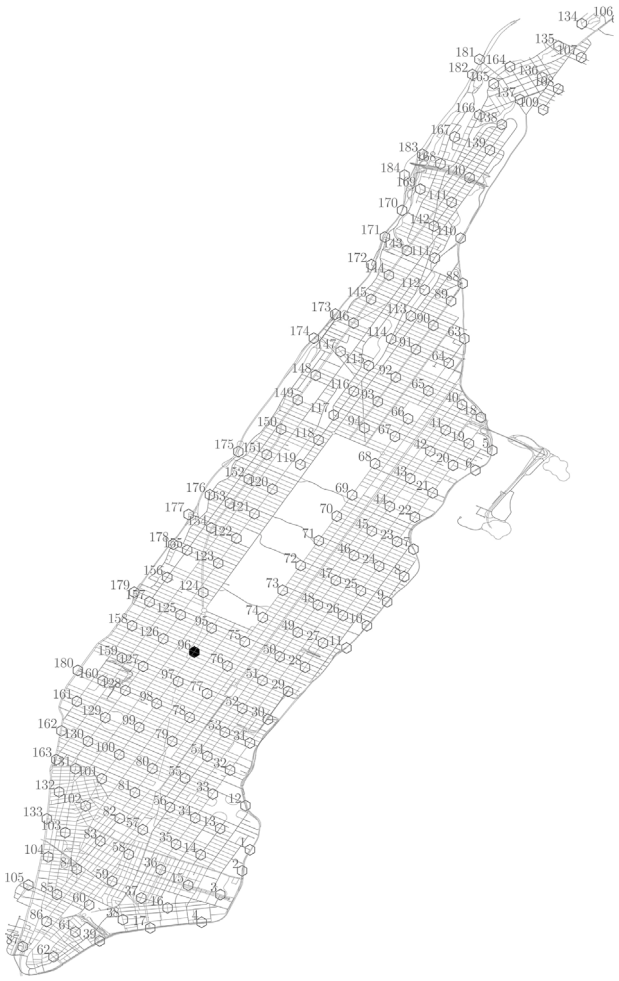


Fig. 14. Road network of Manhattan, NY.

alternating between routes does not result in higher revenue unless we consider the effect of unobserved future demand in our cost-benefit analyses.

In terms of waiting time, we experience somewhat longer waits under our proposed method than the Insertion method. However, the detour time is significantly lower under our proposed method. The first observation originates from the fact that the Insertion method assigns customers to their nearby shuttles, and the second one can be justified by the fact that the myopic method inserts customers on a shuttle's route one at a time. Comparing the utilization rates of shuttles reveals that they are about 20% larger under our method than the benchmark methods. This is due to the fact that the number of customers served by the proposed method are much higher than the Insertion method. Hence, shuttle seats are occupied for a larger proportion of the study horizon.

Finally, this table indicates that all methods take less than 1 s to make a decision at each simulation time step, confirming that the proposed method can be implemented in close to real-time in practice. Note that all of the online computations for all the methods except the Insertion method have been parallelized over the shuttles. It is noteworthy that the No Alter. method is even faster than the Assignment method, while the other two configurations have higher computation times. There are 2 reasons for this difference: (1) the Full and No Pred. methods consider alternative routes per each shuttle, (2) the No Alter. method does not need to reschedule the on-board and scheduled customers. Also, incorporating the expected demand results in larger min-cost flow network problems that have longer computation times under the Full method. It should be emphasized that all computations are parallelized over shuttles on a 50 core environment. However, the cost–benefit analysis of alternative routes in the online phase can be also implemented on multiple cores and completed in parallel, resulting in significantly lower computation times of both Full and No Pred. methods.

Overall, the results of our case study once again confirm that the effectiveness of a shuttle dispatching system can be improved immensely by using the historical trip data. It is worth mentioning that although the No Pred. and No Alter. configurations do not use the expected demand in the cost–benefit analysis of online experiments, they exploit this information when generating the routes in the offline phase.

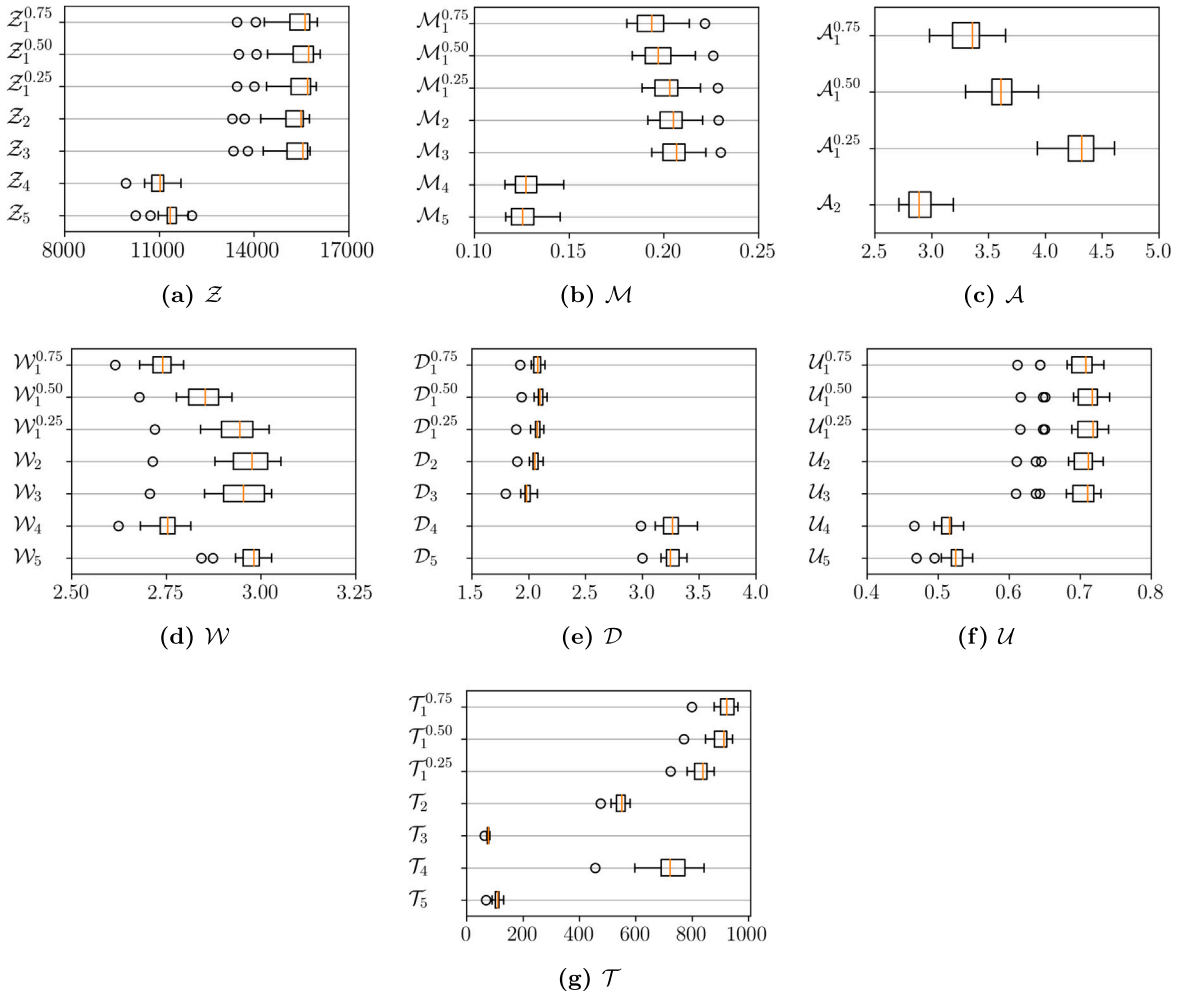


Fig. 15. The result of different methods on the performance measures in the Manhattan case study.

Video 1 visualizes the deployment of all shuttles to serve on-demand requests in the Manhattan area on June 4, 2017 from 7 a.m to 10 a.m. In this video, every shuttle is represented by a circle with a distinct color, and its size corresponds to the number of on-board customers. Also, video 2 demonstrates the deployment of a single shuttle on March 3, 2017 from 7 a.m to 10 a.m. In this video, different alternative routes are represented by different colors. Therefore, the shuttle's color changes when it switches between its alternative routes. The red and blue pins represent the pick-up and drop-off locations of customers, respectively.

6. Conclusion

In this paper we propose a general framework to solve the shuttle dispatching problem efficiently and with high level of accuracy. Our method includes an offline phase, where the computationally-heavy shuttle routing and scheduling problems are solved. In this phase, we generate a large pool of routes that could be valuable under different realizations of demand. These routes are deliberately generated with high spatio-temporal overlap between them to facilitate the transfer of a shuttle from one to another. The efficiency of the offline phase stems from a number of polynomial-time, non-myopic local search algorithms that despite posing little computational burden, enable a near-thorough exploration of the feasible region. In the online portion of the framework, shuttles are routed proactively based on the results from the offline phase. Moreover, the online phase enables shuttles to switch between the pool of routes, allowing them to react to stochastic changes in travel patterns.

We applied our methodology to two networks with different network sizes and demand volumes and patterns. For the online experiments, we introduced three configurations of our methodology by posing assumptions about the existence of a demand forecast and availability of high-quality alternative shuttle routes. We compared the performance of these methods with two state-of-the art algorithms for large-scale dynamic dial-a-ride problems. Results indicated that regardless of the configuration, our method led to a substantial increase in the matching rate and the revenue of both case studies. These experiments also suggested that customers

Table A.1
List of notations.

Notation	Description
Parameters	
S	Number of stations
T	Number of time intervals
K	Number of Shuttles
C	Capacity of a Shuttle
ω	Maximum waiting time of passengers in their origin stations
Ω	Maximum allowed detour from the SP travel time of passengers
M	Number of alternative routes for Shuttle k from historical data
ϵ	Spatio-temporal threshold for building a sub-graph around base route
β	Influence factor $\in (0,1)$ for the cost of expected demand in the cost–benefit analysis
Π	Number of replications for the local search
R	Number of re-clustering steps for the clustering problems
ζ	Improvement factor $\in (0,1)$ for stopping the local search
Sets	
$\mathcal{S}$	Set of stations in the road network
$\mathcal{E}$	Set of transportation links in the road network
$\mathcal{T}$	Ordered set of time intervals
$\mathcal{N}$	Set of nodes in the time-expanded network
$\mathcal{H}$	Set of passengers
$\mathcal{K}$	Set of Shuttles
$\delta^+(n)$	Set of successor nodes of node n in time-expanded network
$\delta^-(n)$	Set of predecessor nodes of node n in time-expanded network
$\mathcal{L}$	Set of edges in the time-expanded network
$\mathcal{N}'$	Set of nodes in the min-cost flow network
$\mathcal{L}'$	Set of edges in the min-cost flow network
Functions	
G	Connected, directed graph of the road network
$\mathcal{G}$	DAG of the time-expanded network
$\mathcal{D}(n)$	Node demand in the min-cost network
$\mathcal{W}(n,n')$	Weight of edges in the min-cost network
$\mathcal{U}(n,n')$	Capacity of edges in the min-cost network
$\mathcal{G}'$	Weighted DAG of the min-cost network
Tables	
$\tau(t,i,j)$	SP travel time (in min) between stations i and j departing from i at time t
$\rho(t,i,j)$	SP travel distance (in miles) between stations i and j departing from i at time t
$\phi(e,o,d)$	Average demand from origin o to destination d entering system at time e
Variables	
$\mathcal{F}_\alpha(n)$	Partial route of Shuttle until node n at iteration α
$\mathcal{B}_\alpha(n)$	Partial route of Shuttle from node n at iteration α
$\mathcal{X}$	A fixed route of Shuttle
$\mathcal{X}^*$	A local optimal route of Shuttle
$\varphi_\alpha(n)$	revenue of Shuttle by visiting node n at iteration α
φ^*_α	Maximum revenue of Shuttle at iteration α
Δ	Relative gap between maximum revenue of two consecutive forward local search
$f^k_m(e,o,d)$	Optimal number of served trips from origin o to destination d with EDT e by alternative m of Shuttle k
$\mathcal{G}^k_m$	The reduced time-expanded network around the base route of Shuttle k based on demand profile m
γ	Optimal revenue of Shuttle in the min-cost network
θ	Simulation time step
z	Total revenue of all base routes in the offline phase
$\phi^k(e,o,d)$	Average demand from origin o to destination d with EDT at time e assigned to Shuttle k
$\eta^k(e,o,d)$	Distance between base route of Shuttle k and requests with origin o to destination d with EDT at time e in the offline phase
$\boldsymbol{\Phi}_m(e,o,d)$	Request counts from origin o to destination d with EDT at time e in the mth demand profile sampled from the Poisson distribution with mean $\phi(e,o,d)$
$\boldsymbol{\Phi}^k_m(e,o,d)$	Request counts from origin o to destination d with EDT at time e in the mth demand profile assigned to Shuttle k
$\psi(e,o,d)$	Estimated request counts from origin o to destination d with EDT at time e
$\psi^{ca}(e,o,d)$	Realized request counts from origin o to destination d with EDT at time e
$\psi^{ca}_k(e,o,d)$	Number of requests from origin o to destination d with EDT at time e assigned to Shuttle k
$\psi^{on}_k(p,o,d)$	Number of on-board passengers on Shuttle k
$\psi^{sc}_k(p,o,d)$	Number of scheduled passengers on Shuttle k
$\xi^k(e,o,d)$	Proximity between base route of Shuttle k and requests with origin o to destination d with EDT at time e in the online phase

(*continued on next page*)

Table A.1 (*continued*).

Notation	Description
$\mathcal{P}_m^k$	Alternative route m for Shuttle k computed in the offline framework
$\mathcal{R}_k$	Route of Shuttle k in real time
Measures	
$\mathcal{Z}$	Total revenue of all Shuttles (in miles) averaged over all runs
$\mathcal{M}$	Matching rate of customers served by all Shuttles averaged over all runs
$\mathcal{W}$	Waiting time of served customers averaged over all Shuttle, runs, and customers
$\mathcal{D}$	Detour time of served customers averaged over all Shuttle, runs, and customers
$\mathcal{U}$	Utilization rate of Shuttles' seats averaged over all Shuttle, runs, and time steps
$\mathcal{A}$	Number of times Shuttles switch their routes averaged over all Shuttle and runs
Θ	Average computation time (in seconds) of online framework over all runs and all time steps

Table B.1
Hourly rate of demand between all pairs of origin destination stations in the Nguyen–Dupuis case study.

O	D												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	5	22	2	5	0	0	0	0	0	0	12	0
2	12	0	13	1	2	0	0	15	0	0	25	0	0
3	22	4	0	15	4	0	0	0	0	0	18	0	5
4	14	7	11	0	34	10	0	0	22	0	0	0	0
5	1	10	10	2	0	0	0	0	8	0	0	0	0
6	5	3	13	8	12	0	6	31	7	15	30	8	0
7	17	1	12	22	9	18	0	16	2	4	10	9	2
8	0	10	4	7	0	0	9	0	12	4	9	13	4
9	11	1	4	12	0	0	15	0	0	27	9	0	4
10	0	0	0	2	0	14	3	0	30	0	9	0	17
11	0	5	6	2	0	12	13	0	20	0	0	0	28
12	5	0	0	10	5	5	6	9	0	31	0	0	9
13	0	0	15	0	0	0	0	0	10	8	0	0	0

generally experience lower detour times under the proposed method than the benchmark methods. Among the three configurations of our method, the Full method always yielded the highest revenue, but it came at the cost of increased computational time. However, this increase in the computational burden can be partially alleviated by evaluating the alternative routes on multiple cores in parallel.

Through various case studies, we inferred that only following the base routes provides sufficiently large revenues. However, we showed that we can improve the revenues even further with the help of alternating routes and incorporating demand forecast. Finally, the result of the Manhattan case study suggested that considering historical demand, even using a very simple predictive model such as a Poisson distribution, could significantly improve the performance of a shuttle dispatching system.

CRediT authorship contribution statement

Amirmahdi Tafreshian: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Roles/Writing - original draft, Writing - review & editing. **Mojtaba Abdolmaleki:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Roles/Writing - original draft, Writing - review & editing. **Neda Masoud:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Roles/Writing - original draft, Writing - review & editing. **Huizhu Wang:** Conceptualization, Investigation, Resources, Roles/Writing - original draft.

Acknowledgments

The work described in this paper was supported by research grants from the Ford Motor Company, the Michigan Institute for Data Science (MIDAS), and National Science Foundation awards 1831347 and 2046372.

Appendix A. Notations for the proposed framework

See Table A.1.

Appendix B. Average demand for Nguyen–Dupuis network

See Table B.1.

Table C.1
Impact of the spatio-temporal threshold, ϵ, on the average waiting time, $\mathcal{W}$, in the Nguyen–Dupuis case study.

μ	σ	β	Full			No Pred.			No Alter.	Insertion	Assignment
			$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$			
0	0	0.95	1.41	1.41	1.40	1.41	1.39	1.37	1.39	1.3	1.81
		0.50	1.41	1.4	1.39						
		0.25	1.41	1.39	1.39						
5	0	0.95	1.46	1.48	1.48	1.53	1.51	1.51	1.48	1.56	1.92
		0.50	1.48	1.48	1.49						
		0.25	1.50	1.50	1.50						
0	5	0.95	1.42	1.42	1.43	1.43	1.42	1.42	1.41	1.36	1.82
		0.50	1.42	1.41	1.42						
		0.25	1.43	1.41	1.42						

Table C.2
Impact of the spatio-temporal threshold, ϵ, on the average detour time, $\mathcal{D}$, in the Nguyen–Dupuis case study.

μ	σ	β	Full			No Pred.			No Alter.	Insertion	Assignment
			$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$			
0	0	0.95	3.86	3.82	3.83	3.78	3.67	3.74	3.81	1.91	1.95
		0.50	3.83	3.78	3.78						
		0.25	3.80	3.69	3.79						
5	0	0.95	3.98	3.97	3.96	3.86	3.82	3.85	4.05	2.28	2.24
		0.50	3.94	3.96	3.89						
		0.25	3.87	3.87	3.85						
0	5	0.95	3.87	3.86	3.85	3.82	3.77	3.77	3.87	2.02	2.01
		0.50	3.85	3.81	3.85						
		0.25	3.83	3.78	3.82						

Table C.3
Impact of the spatio-temporal threshold, ϵ, on the average seat utilization rate of Shuttles, $\mathcal{U}$, in the Nguyen–Dupuis case study.

μ	σ	β	Full			No Pred.			No Alter.	Insertion	Assignment
			$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$			
0	0	0.95	45.84	45.85	45.75	44.89	44.49	44.89	45.7	30.5	28.95
		0.50	45.66	45.69	45.52						
		0.25	45.22	45.29	45.17						
5	0	0.95	61.91	62.32	62.17	60.13	59.8	59.47	61.27	46.29	43.85
		0.50	61.54	62.12	61.64						
		0.25	60.68	61.45	60.74						
0	5	0.95	47.43	47.59	47.45	46.28	46.47	46.13	47.09	34.21	32.58
		0.50	47.39	47.46	47.25						
		0.25	46.95	47.11	46.8						

Table C.4
Impact of the spatio-temporal threshold, ϵ, on the average number of switches between alternative routes, $\mathcal{A}$, in the Nguyen–Dupuis case study.

μ	σ	β	Full			No Pred.			No Alter.	Insertion	Assignment
			$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$			
0	0	0.95	2.78	3.01	2.71	0.49	0.49	0.50	–	–	–
		0.50	2.8	3.04	2.77						
		0.25	2.88	3.1	2.92						
5	0	0.95	3.9	3.97	3.7	0.84	0.9	1.06	–	–	–
		0.50	3.95	4.06	3.87						
		0.25	3.99	4.09	3.94						
0	5	0.95	2.87	3.07	2.85	0.55	0.59	0.66	–	–	–
		0.50	2.99	3.16	3.01						
		0.25	3.09	3.2	3.13						

Appendix C. Sensitivity analysis on ϵ in the Nguyen–Dupuis case study (cont'd)

From the results of Tables C.1, C.2, and C.3, we infer that ϵ does not significantly affect the out-of-shuttle waiting time, in-shuttle detour time, and the occupancy of seats in a shuttle under our proposed method.

Table C.5
Impact of ϵ on the average computation time of every time step, Θ, in the Nguyen–Dupuis case study.

μ	σ	β	Full			No Pred.			No Alter.	Insertion	Assignment
			$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 15$			
		0.95	21.00	23.61	21.34						
0	0	0.50	20.32	23.07	21.05	15.00	17.49	15.30	1.28	21.30	1.70
		0.25	20.00	22.71	20.57						
		0.95	27.77	31.11	28.39						
5	0	0.50	26.25	29.86	27.05	19.29	22.28	19.40	1.94	46.34	3.73
		0.25	25.08	28.84	26.07						
		0.95	21.27	23.65	21.54						
0	5	0.50	20.42	23.32	20.88	14.96	17.45	15.18	1.38	23.32	1.94
		0.25	20.16	22.70	20.34						

Table D.1
Impact of arrival distribution on the average revenue, $\mathcal{Z}$, in the Nguyen–Dupuis case study.

μ	σ	β	Full		No Pred.		No Alter.		Insertion		Assignment	
			Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform
		0.95	2748.9	2817.14								
0	0	0.50	2741.19	2811.7	2677.08	2735.1	2731.89	2785.81	2003.29	2067.86	1914.25	1962.24
		0.25	2725.04	2785.94								
		0.95	3729.16	3770.66								
5	0	0.50	3713.06	3749.98	3583.72	3640.84	3629.71	3675.12	2997.8	3127.34	2835.92	2930.15
		0.25	3677.31	3717.95								
		0.95	2847.92	2937.48								
0	5	0.50	2842.04	2936.15	2785.86	2852.84	2804.75	2884.05	2242.68	2274.21	2137.66	2196.38
		0.25	2823.32	2914.35								

Table D.2
Impact of arrival distribution on the average matching rate, $\mathcal{M}$, in the Nguyen–Dupuis case study.

μ	σ	β	Full		No Pred.		No Alter.		Insertion		Assignment	
			Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform
		0.95	0.74	0.74								
0	0	0.50	0.74	0.75	0.73	0.73	0.75	0.75	0.57	0.58	0.54	0.54
		0.25	0.74	0.74								
		0.95	0.56	0.55								
5	0	0.50	0.56	0.55	0.55	0.54	0.56	0.55	0.47	0.47	0.44	0.44
		0.25	0.57	0.55								
		0.95	0.67	0.67								
0	5	0.50	0.67	0.67	0.66	0.66	0.67	0.67	0.55	0.54	0.52	0.51
		0.25	0.67	0.67								

Table D.3
Impact of arrival distribution on the average waiting time, $\mathcal{W}$, in the Nguyen–Dupuis case study.

μ	σ	β	Full		No Pred.		No Alter.		Insertion		Assignment	
			Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform
		0.95	1.41	1.39								
0	0	0.5	1.4	1.37	1.39	1.37	1.38	1.36	1.3	1.33	1.82	1.83
		0.25	1.39	1.37								
		0.95	1.48	1.5								
5	0	0.5	1.48	1.5	1.51	1.55	1.47	1.52	1.56	1.59	1.94	1.93
		0.25	1.5	1.52								
		0.95	1.42	1.41								
0	5	0.5	1.41	1.4	1.42	1.39	1.4	1.37	1.36	1.39	1.82	1.87
		0.25	1.41	1.4								

Table C.4 clearly suggests that the capability of shuttles to switch their routes and the shuttle revenues demonstrate the same trend as we increase the value of ϵ. This table further reveals that the shuttles change their routes more frequently in the second and third scenarios. These scenarios mimic the situation where the realized online demand significantly differs from the historical demand. As a result, it is more likely for shuttles to obtain requests that are not on their current routes. This table also indicates that using demand prediction helps us use our alternative routes more effectively, as the values of $\mathcal{A}_1$ are much higher than the

Table D.4
Impact of arrival distribution on the average detour time, D, in the Nguyen–Dupuis case study.

μ	σ	β	Full		No Pred.		No Alter.		Insertion		Assignment	
			Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform
0	0	0.95	3.82	3.92								
		0.50	3.78	3.9	3.67	3.79	3.77	3.88	1.91	1.87	1.9	1.97
		0.25	3.69	3.84								
5	0	0.95	3.97	4.15								
		0.50	3.96	4.12	3.82	4	4.02	4.19	2.28	2.22	2.22	2.27
		0.25	3.87	4.04								
0	5	0.95	3.86	3.99								
		0.50	3.81	3.95	3.77	3.85	3.85	3.94	2.02	2.07	2.05	2.08
		0.25	3.78	3.88								

Table D.5
Impact of arrival distribution on the average seat utilization rate of shuttles, $\mathcal{U}$, in the Nguyen–Dupuis case study.

μ	σ	β	Full		No Pred.		No Alter.		Insertion		Assignment	
			Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform
0	0	0.95	45.85	47.16								
		0.50	45.69	47.04	44.49	45.68	45.61	46.71	30.5	31.32	29.04	29.84
		0.25	45.29	46.55								
5	0	0.95	62.32	63.39								
		0.50	62.12	63.09	59.8	61.15	61.23	62.35	46.29	47.92	43.57	45.04
		0.25	61.45	62.49								
0	5	0.95	47.59	49.29								
		0.50	47.46	49.22	46.47	47.76	47.02	48.49	34.21	34.76	32.59	33.52
		0.25	47.11	48.78								

Table D.6
Impact of arrival distribution on the average number of switches between alternative routes, $\mathcal{A}$, in the Nguyen–Dupuis case study.

μ	σ	β	Full		No Pred.		No Alter.		Insertion		Assignment	
			Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform
0	0	0.95	3.01	2.99								
		0.5	3.04	3.05	0.49	0.53	–	–	–	–	–	–
		0.25	3.1	3.06								
5	0	0.95	3.97	3.78								
		0.5	4.06	3.93	0.9	0.89	–	–	–	–	–	–
		0.25	4.09	3.96								
0	5	0.95	3.07	3.1								
		0.5	3.16	3.16	0.59	0.62	–	–	–	–	–	–
		0.25	3.2	3.16								

Table D.7
Impact of arrival distribution on the average computation time of every time step, Θ, in the Nguyen–Dupuis case study.

μ	σ	β	Full		No Pred.		No Alter.		Insertion		Assignment	
			Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform	Poisson	Uniform
0	0	0.95	23.61	22.78								
		0.50	23.07	22.43	17.49	15.48	1.71	1.41	21.30	19.49	2.20	1.79
		0.25	22.71	21.05								
5	0	0.95	31.11	30.75								
		0.50	29.86	29.33	22.28	21.84	2.62	2.33	46.34	42.02	4.72	4.35
		0.25	28.84	28.45								
0	5	0.95	23.65	22.76								
		0.50	23.32	23.00	17.45	16.78	1.51	1.46	23.32	23.01	2.14	2.12
		0.25	22.70	22.36								

values of $\mathcal{A}_2$. Finally, Table C.5 presents the computation time of every time step in our simulation. This table suggests that our proposed method can be easily applied in real time as the computation time of every time step takes less than a few milliseconds. This table also indicates that the computation time of the Full model is more than other configurations. This is not surprising as we take alternative routes as well as future demand into account when conducting a cost–benefit analysis in the Full method.

Table E.1
Paired t-test *p*-value for all pairwise comparison of the results in the Nguyen–Dupuis case study.

(a) $\mathcal{Z}$ for $(\mu,\sigma)=(0,0)$

	$\overline{\mathcal{Z}}_1^{0.50}$	$\overline{\mathcal{Z}}_1^{0.25}$	$\overline{\mathcal{Z}}_2$	$\overline{\mathcal{Z}}_3$	$\overline{\mathcal{Z}}_4$	$\overline{\mathcal{Z}}_5$
$\overline{\mathcal{Z}}_1^{0.75}$	0.00	0.00	0.00	0.01	0.00	0.00
$\overline{\mathcal{Z}}_1^{0.50}$		0.00	0.00	0.10	0.00	0.00
$\overline{\mathcal{Z}}_1^{0.25}$			0.00	0.25	0.00	0.00
$\overline{\mathcal{Z}}_2$				0.00	0.00	0.00
$\overline{\mathcal{Z}}_3$					0.00	0.00
$\overline{\mathcal{Z}}_4$						0.00

(b) $\mathcal{Z}$ for $(\mu,\sigma)=(5,0)$

	$\overline{\mathcal{Z}}_1^{0.50}$	$\overline{\mathcal{Z}}_1^{0.25}$	$\overline{\mathcal{Z}}_2$	$\overline{\mathcal{Z}}_3$	$\overline{\mathcal{Z}}_4$	$\overline{\mathcal{Z}}_5$
$\overline{\mathcal{Z}}_1^{0.75}$	0.02	0.00	0.00	0.00	0.00	0.00
$\overline{\mathcal{Z}}_1^{0.50}$		0.00	0.00	0.00	0.00	0.00
$\overline{\mathcal{Z}}_1^{0.25}$			0.00	0.00	0.00	0.00
$\overline{\mathcal{Z}}_2$				0.00	0.00	0.00
$\overline{\mathcal{Z}}_3$					0.00	0.00
$\overline{\mathcal{Z}}_4$						0.00

(c) $\mathcal{Z}$ for $(\mu,\sigma)=(0,5)$

	$\overline{\mathcal{Z}}_1^{0.50}$	$\overline{\mathcal{Z}}_1^{0.25}$	$\overline{\mathcal{Z}}_2$	$\overline{\mathcal{Z}}_3$	$\overline{\mathcal{Z}}_4$	$\overline{\mathcal{Z}}_5$
$\overline{\mathcal{Z}}_1^{0.75}$	0.18	0.00	0.00	0.00	0.00	0.00
$\overline{\mathcal{Z}}_1^{0.50}$		0.00	0.00	0.00	0.00	0.00
$\overline{\mathcal{Z}}_1^{0.25}$			0.00	0.03	0.00	0.00
$\overline{\mathcal{Z}}_2$				0.05	0.00	0.00
$\overline{\mathcal{Z}}_3$					0.00	0.00
$\overline{\mathcal{Z}}_4$						0.00

(d) $\mathcal{M}$ for $(\mu,\sigma)=(0,0)$

	$\overline{\mathcal{M}}_1^{0.50}$	$\overline{\mathcal{M}}_1^{0.25}$	$\overline{\mathcal{M}}_2$	$\overline{\mathcal{M}}_3$	$\overline{\mathcal{M}}_4$	$\overline{\mathcal{M}}_5$
$\overline{\mathcal{M}}_1^{0.75}$	0.49	0.19	0.00	0.16	0.00	0.00
$\overline{\mathcal{M}}_1^{0.50}$		0.03	0.00	0.25	0.00	0.00
$\overline{\mathcal{M}}_1^{0.25}$			0.00	0.04	0.00	0.00
$\overline{\mathcal{M}}_2$				0.00	0.00	0.00
$\overline{\mathcal{M}}_3$					0.00	0.00
$\overline{\mathcal{M}}_4$						0.00

(e) $\mathcal{M}$ for $(\mu,\sigma)=(5,0)$

	$\overline{\mathcal{M}}_1^{0.50}$	$\overline{\mathcal{M}}_1^{0.25}$	$\overline{\mathcal{M}}_2$	$\overline{\mathcal{M}}_3$	$\overline{\mathcal{M}}_4$	$\overline{\mathcal{M}}_5$
$\overline{\mathcal{M}}_1^{0.75}$	0.09	0.01	0.00	0.77	0.00	0.00
$\overline{\mathcal{M}}_1^{0.50}$		0.03	0.00	0.12	0.00	0.00
$\overline{\mathcal{M}}_1^{0.25}$			0.00	0.00	0.00	0.00
$\overline{\mathcal{M}}_2$				0.00	0.00	0.00
$\overline{\mathcal{M}}_3$					0.00	0.00
$\overline{\mathcal{M}}_4$						0.00

(f) $\mathcal{M}$ for $(\mu,\sigma)=(0,5)$

	$\overline{\mathcal{M}}_1^{0.50}$	$\overline{\mathcal{M}}_1^{0.25}$	$\overline{\mathcal{M}}_2$	$\overline{\mathcal{M}}_3$	$\overline{\mathcal{M}}_4$	$\overline{\mathcal{M}}_5$
$\overline{\mathcal{M}}_1^{0.75}$	0.79	0.39	0.00	0.17	0.00	0.00
$\overline{\mathcal{M}}_1^{0.50}$		0.23	0.00	0.13	0.00	0.00
$\overline{\mathcal{M}}_1^{0.25}$			0.00	0.75	0.00	0.00
$\overline{\mathcal{M}}_2$				0.00	0.00	0.00
$\overline{\mathcal{M}}_3$					0.00	0.00
$\overline{\mathcal{M}}_4$						0.00

Table E.2
Paired t-test *p*-value for all pairwise comparison of the results in the Manhattan case study.

(a) $\mathcal{Z}$

	$\overline{\mathcal{Z}}_1^{0.50}$	$\overline{\mathcal{Z}}_1^{0.25}$	$\overline{\mathcal{Z}}_2$	$\overline{\mathcal{Z}}_3$	$\overline{\mathcal{Z}}_4$	$\overline{\mathcal{Z}}_5$
$\overline{\mathcal{Z}}_1^{0.75}$	0.00	0.00	0.68	0.00	0.00	0.00
$\overline{\mathcal{Z}}_1^{0.50}$		0.00	0.00	0.00	0.00	0.00
$\overline{\mathcal{Z}}_1^{0.25}$			0.00	0.00	0.00	0.00
$\overline{\mathcal{Z}}_2$				0.00	0.00	0.00
$\overline{\mathcal{Z}}_3$					0.00	0.00
$\overline{\mathcal{Z}}_4$						0.00

(b) $\mathcal{M}$

	$\overline{\mathcal{M}}_1^{0.50}$	$\overline{\mathcal{M}}_1^{0.25}$	$\overline{\mathcal{M}}_2$	$\overline{\mathcal{M}}_3$	$\overline{\mathcal{M}}_4$	$\overline{\mathcal{M}}_5$
$\overline{\mathcal{M}}_1^{0.75}$	0.02	0.00	0.00	0.00	0.00	0.00
$\overline{\mathcal{M}}_1^{0.50}$		0.00	0.00	0.00	0.00	0.00
$\overline{\mathcal{M}}_1^{0.25}$			0.00	0.00	0.00	0.00
$\overline{\mathcal{M}}_2$				0.00	0.00	0.00
$\overline{\mathcal{M}}_3$					0.00	0.00
$\overline{\mathcal{M}}_4$						0.24

(c) $\mathcal{A}$

	$\overline{\mathcal{A}}_1^{0.50}$	$\overline{\mathcal{A}}_1^{0.25}$	$\overline{\mathcal{A}}_2$
$\overline{\mathcal{A}}_1^{0.75}$	0.00	0.00	0.00
$\overline{\mathcal{A}}_1^{0.50}$		0.00	0.00
$\overline{\mathcal{A}}_1^{0.25}$			0.00

(d) $\mathcal{W}$

	$\overline{\mathcal{W}}_1^{0.50}$	$\overline{\mathcal{W}}_1^{0.25}$	$\overline{\mathcal{W}}_2$	$\overline{\mathcal{W}}_3$	$\overline{\mathcal{W}}_4$	$\overline{\mathcal{W}}_5$
$\overline{\mathcal{W}}_1^{0.75}$	0.00	0.00	0.00	0.00	0.01	0.00
$\overline{\mathcal{W}}_1^{0.50}$		0.00	0.00	0.00	0.00	0.00
$\overline{\mathcal{W}}_1^{0.25}$			0.00	0.00	0.00	0.00
$\overline{\mathcal{W}}_2$				0.00	0.00	0.62
$\overline{\mathcal{W}}_3$					0.00	0.06
$\overline{\mathcal{W}}_4$						0.00

(e) $\mathcal{D}$

	$\overline{\mathcal{D}}_1^{0.50}$	$\overline{\mathcal{D}}_1^{0.25}$	$\overline{\mathcal{D}}_2$	$\overline{\mathcal{D}}_3$	$\overline{\mathcal{D}}_4$	$\overline{\mathcal{D}}_5$
$\overline{\mathcal{D}}_1^{0.75}$	0.00	0.08	0.03	0.00	0.00	0.00
$\overline{\mathcal{D}}_1^{0.50}$		0.00	0.00	0.00	0.00	0.00
$\overline{\mathcal{D}}_1^{0.25}$			0.02	0.00	0.00	0.00
$\overline{\mathcal{D}}_2$				0.00	0.00	0.00
$\overline{\mathcal{D}}_3$					0.00	0.00
$\overline{\mathcal{D}}_4$						0.72

(f) $\mathcal{U}$

	$\overline{\mathcal{U}}_1^{0.50}$	$\overline{\mathcal{U}}_1^{0.25}$	$\overline{\mathcal{U}}_2$	$\overline{\mathcal{U}}_3$	$\overline{\mathcal{U}}_4$	$\overline{\mathcal{U}}_5$
$\overline{\mathcal{U}}_1^{0.75}$	0.00	0.00	0.54	0.80	0.00	0.00
$\overline{\mathcal{U}}_1^{0.50}$		0.08	0.00	0.00	0.00	0.00
$\overline{\mathcal{U}}_1^{0.25}$			0.00	0.00	0.00	0.00
$\overline{\mathcal{U}}_2$				0.55	0.00	0.00
$\overline{\mathcal{U}}_3$					0.00	0.00
$\overline{\mathcal{U}}_4$						0.01

(g) $\mathcal{T}$

	$\overline{\mathcal{T}}_1^{0.50}$	$\overline{\mathcal{T}}_1^{0.25}$	$\overline{\mathcal{T}}_2$	$\overline{\mathcal{T}}_3$	$\overline{\mathcal{T}}_4$	$\overline{\mathcal{T}}_5$
$\overline{\mathcal{T}}_1^{0.75}$	0.00	0.00	0.00	0.00	0.00	0.00
$\overline{\mathcal{T}}_1^{0.50}$		0.00	0.00	0.00	0.00	0.00
$\overline{\mathcal{T}}_1^{0.25}$			0.00	0.00	0.00	0.00
$\overline{\mathcal{T}}_2$				0.00	0.00	0.00
$\overline{\mathcal{T}}_3$					0.00	0.00
$\overline{\mathcal{T}}_4$						0.00

Appendix D. Sensitivity analysis on the distribution of arrivals in the Nguyen–Dupuis case study

The results of Poisson and Uniform arrivals are summarized in Tables D.1, D.2, D.3, D.4, D.5, D.6, and D.7. From Table D.1, we infer that changing the arrival distribution from Poisson to Uniform results in a slight increase in the revenue of all methods, albeit the differences more or less stay the same. This might seem contradictory given our assumption of Poisson arrivals for generating the routes in the offline phase. However, it should be noted that a Poisson arrival indicates exponential inter-arrival rates. Given the memory-less property of the exponential inter-arrival times, the Poisson arrivals simulate a highly stochastic system. In contrast, arriving uniformly over time increases the chance of a customer to become available for at least one shuttle. This is why the revenue increases not only for the three configurations of our method, but also for the two benchmark methods. Table D.2, however, does not show a meaningful change in the matching rate of methods under the Uniform distribution.

Table D.3 does not show a significant change in the waiting times. However, one can infer from Table D.4 a slight increase in the detour time of the three configurations of our method by using Uniform distributions. This again originates from the fact that

Table E.3
Paired t-test p-values for sensitivity analysis on ϵ.

Measure	μ	σ	β	Full			No Pred.		
				(5,10)	(5,15)	(10,15)	(5,10)	(5,15)	(10,15)
			0.95	0.58	0.53	0.13			
	0	0	0.50	0.46	0.62	0.10	0.22	0.67	0.06
			0.25	0.14	0.67	0.01			
			0.95	0.00	0.00	0.17			
$\mathcal{Z}$	5	0	0.50	0.00	0.01	0.03	0.50	0.01	0.04
			0.25	0.00	0.29	0.00			
			0.95	0.09	0.72	0.26			
	0	5	0.50	0.42	0.31	0.02	0.04	0.64	0.02
			0.25	0.10	0.28	0.01			
			0.95	0.64	0.33	0.07			
	0	0	0.50	0.41	0.26	0.02	0.34	0.41	0.07
			0.25	0.12	0.65	0.01			
			0.95	0.01	0.41	0.04			
$\mathcal{M}$	5	0	0.50	0.08	0.84	0.06	0.77	0.06	0.11
			0.25	0.00	0.57	0.00			
			0.95	0.21	0.48	0.10			
	0	5	0.50	0.70	0.09	0.02	0.44	0.45	0.11
			0.25	0.34	0.07	0.01			
			0.95	0.99	0.66	0.71			
	0	0	0.50	0.16	0.08	0.58	0.02	0.00	0.07
			0.25	0.05	0.00	0.43			
			0.95	0.17	0.03	0.39			
$\mathcal{W}$	5	0	0.50	0.80	0.43	0.23	0.20	0.16	0.81
			0.25	0.97	0.69	0.76			
			0.95	0.69	0.41	0.15			
	0	5	0.50	0.39	0.95	0.25	0.43	0.21	0.62
			0.25	0.28	0.84	0.30			
			0.95	0.02	0.19	0.40			
	0	0	0.50	0.02	0.03	0.79	0.00	0.20	0.01
			0.25	0.00	0.65	0.00			
			0.95	0.42	0.38	0.87			
$\mathcal{D}$	5	0	0.50	0.53	0.01	0.00	0.18	0.69	0.04
			0.25	0.87	0.29	0.39			
			0.95	0.31	0.25	0.91			
	0	5	0.50	0.08	0.87	0.03	0.02	0.09	0.80
			0.25	0.04	0.81	0.15			
			0.95	0.00	0.19	0.00			
	0	0	0.50	0.00	0.68	0.00	1.00	0.72	0.73
			0.25	0.01	0.45	0.02			
			0.95	0.37	0.01	0.00			
$\mathcal{A}$	5	0	0.50	0.20	0.27	0.00	0.13	0.00	0.00
			0.25	0.17	0.58	0.00			
			0.95	0.01	0.73	0.00			
	0	5	0.50	0.02	0.85	0.01	0.35	0.00	0.04
			0.25	0.12	0.49	0.15			
			0.95	0.88	0.52	0.19			
	0	0	0.50	0.81	0.31	0.06	0.00	1.00	0.00
			0.25	0.64	0.73	0.12			
			0.95	0.00	0.00	0.08			
$\mathcal{U}$	5	0	0.50	0.00	0.41	0.00	0.14	0.01	0.15
			0.25	0.00	0.76	0.00			
			0.95	0.07	0.81	0.22			
	0	5	0.50	0.49	0.29	0.04	0.28	0.43	0.07
			0.25	0.12	0.32	0.04			
			0.95	0.00	0.00	0.00			
	0	0	0.50	0.00	0.00	0.00	0.00	0.00	0.00
			0.25	0.00	0.00	0.00			
			0.95	0.00	0.00	0.00			
$\mathcal{T}$	5	0	0.50	0.00	0.00	0.00	0.00	0.00	0.00
			0.25	0.00	0.00	0.00			
			0.95	0.00	0.00	0.00			
	0	5	0.50	0.00	0.00	0.00	0.00	0.00	0.00
			0.25	0.00	0.00	0.00			

Table E.4
Paired t-test p-values for sensitivity analysis on the arrival distribution.

Measure	μ	σ	β	Full	No Pred.	No Alter.	Insertion	Assignment
			0.95	0.02				
	0	0	0.50	0.03	0.08	0.08	0.05	0.14
			0.25	0.06				
			0.95	0.02				
$\mathcal{Z}$	5	0	0.50	0.07	0.00	0.02	0.00	0.02
			0.25	0.04				
			0.95	0.02				
	0	5	0.50	0.01	0.04	0.02	0.44	0.12
			0.25	0.02				
			0.95	0.75				
	0	0	0.50	0.79	0.92	0.91	0.20	0.98
			0.25	0.95				
			0.95	0.00				
$\mathcal{M}$	5	0	0.50	0.00	0.00	0.00	0.92	0.84
			0.25	0.00				
			0.95	0.41				
	0	5	0.50	0.57	0.34	0.54	0.01	0.20
			0.25	0.56				
			0.95	0.23				
	0	0	0.50	0.17	0.46	0.54	0.40	0.47
			0.25	0.27				
			0.95	0.22				
$\mathcal{W}$	5	0	0.50	0.17	0.06	0.05	0.29	0.57
			0.25	0.28				
			0.95	0.78				
	0	5	0.50	0.58	0.17	0.08	0.33	0.01
			0.25	0.48				
			0.95	0.01				
	0	0	0.50	0.00	0.01	0.01	0.51	0.31
			0.25	0.00				
			0.95	0.00				
$\mathcal{D}$	5	0	0.50	0.00	0.00	0.00	0.36	0.28
			0.25	0.00				
			0.95	0.06				
	0	5	0.50	0.07	0.21	0.19	0.42	0.56
			0.25	0.20				
			0.95	0.74				
	0	0	0.50	0.88	0.33	–	–	–
			0.25	0.52				
			0.95	0.03				
$\mathcal{A}$	5	0	0.50	0.08	0.78	–	–	–
			0.25	0.07				
			0.95	0.81				
	0	5	0.50	1.00	0.38	–	–	–
			0.25	0.66				
			0.95	0.75				
	0	0	0.50	0.79	0.92	0.91	0.20	0.98
			0.25	0.95				
			0.95	0.00				
$\mathcal{U}$	5	0	0.50	0.00	0.00	0.00	0.92	0.84
			0.25	0.00				
			0.95	0.41				
	0	5	0.50	0.57	0.34	0.54	0.01	0.20
			0.25	0.56				
			0.95	0.00				
	0	0	0.50	0.00	0.00	0.00	0.00	0.00
			0.25	0.00				
			0.95	0.00				
$\mathcal{T}$	5	0	0.50	0.00	0.00	0.00	0.00	0.00
			0.25	0.00				
			0.95	0.00				
	0	5	0.50	0.00	0.00	0.00	0.00	0.00
			0.25	0.00				

the shuttle routes are built based on Poisson distribution. Hence, although the shuttles can still visit stations, but the order in which stations are visited may result in longer detours. The results in Table D.5 are also consistent with our previous observations. As the matching rate and detour times go up under the Uniform distribution, we expect higher utilization rates.

In addition Table D.6 may suggest that under the Uniform distribution, shuttles alter their routes less frequently. However, this change in the value of $\mathcal{A}$ does not seem to be statistically significant. Finally, Table D.7 shows a slight decrease in the computation times of our proposed method under the Uniform arrivals case. This is due to the fact that Poisson arrivals may give rise to larger pools of customers in some time steps. Hence the capacity of arcs in the min-cost flow network increases, which causes the optimization to take longer to solve. However, it is worth mentioning that the magnitude of decrease in computation times seem to be negligible. Overall, we can conclude that the performance of the proposed framework is robust to the choice of customers' arrival distribution.

Appendix E. Paired T-test P-values

See Tables E.1–E.4.

References

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., Rus, D., 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proc. Natl. Acad. Sci. 114 (3), 462–467.

Bent, R.W., Van Hentenryck, P., 2004. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Oper. Res. 52 (6), 977–987.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems: a classification scheme and survey. TOP 15 (1), 1–31.

Carlisle, M.C., Lloyd, E.L., 1995. On the k-coloring of intervals. Discrete Appl. Math. 59 (3), 225–235.

Cordeau, J.-F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. Oper. Res. 54 (3), 573–586.

Cordeau, J.-F., Laporte, G., 2003. The dial-a-ride problem (DARP): Variants, modeling issues and algorithms. Q. J. Belg. Fr. Ital. Oper. Res. Soc. 1 (2), 89–101.

Cordeau, J.-F., Laporte, G., 2007. The dial-a-ride problem: models and algorithms. Ann. Oper. Res. 153 (1), 29–46.

Coslovich, L., Pesenti, R., Ukovich, W., 2006. A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. European J. Oper. Res. 175 (3), 1605–1615.

Ghilas, V., Demir, E., Van Woensel, T., 2016. A scenario-based planning for the pickup and delivery problem with time windows, scheduled lines and stochastic demands. Transp. Res. B 91, 34–51.

Häme, L., 2011. An adaptive insertion algorithm for the single-vehicle dial-a-ride problem with narrow time windows. European J. Oper. Res. 209 (1), 11–22.

Healy, P., Moll, R., 1995. A new extension of local search applied to the dial-a-ride problem. European J. Oper. Res. 83 (1), 83–104.

Ho, S.C., Szeto, W., Kuo, Y.-H., Leung, J.M., Petering, M., Tou, T.W., 2018. A survey of dial-a-ride problems: Literature review and recent developments. Transp. Res. B 111, 395–421.

Hyland, M., Mahmassani, H.S., 2018. Dynamic autonomous vehicle fleet operations: optimization-based strategies to assign avs to immediate traveler demand requests. Transportation Research Part C: Emerging Technologies 92, 278–297.

Ichoua, S., Gendreau, M., Potvin, J.-Y., 2006. Exploiting knowledge about future demands for real-time vehicle dispatching. Transp. Sci. 40 (2), 211–225.

Jafari, E., Rambha, T., Khani, A., Boyles, S.D., 2016. The for-profit dial-a-ride problem on dynamic networks. In: The 95th Annual Meeting of the Transportation Research Board. Washington, DC (No. 16-4686).

Jaw, J.-J., Odoni, A.R., Psaraftis, H.N., Wilson, N.H., 1986. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transp. Res. B 20 (3), 243–257.

Li, D., Antoniou, C., Jiang, H., Xie, Q., Shen, W., Han, W., 2019. The value of prepositioning in smartphone-based vanpool services under stochastic requests and time-dependent travel times. Transp. Res. Rec. 2673 (2), 26–37.

Lloret-Batlle, R., Masoud, N., Nam, D., 2017. Peer-to-peer ridesharing with ride-back on high-occupancy-vehicle lanes: Toward a practical alternative mode for daily commuting. Transp. Res. Rec. 2668 (1), 21–28.

Lowalekar, M., Varakantham, P., Jaillet, P., 2018. Online spatio-temporal matching in stochastic and dynamic domains. Artificial Intelligence 261, 71–112.

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, IEEE, pp. 410–421.

Maalouf, M., MacKenzie, C.A., Radakrishnan, S., Court, M., 2014. A new fuzzy logic approach to capacitated dynamic dial-a-ride problem. Fuzzy Sets and Systems 255, 30–40.

Madsen, O.B., Ravn, H.F., Rygaard, J.M., 1995. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. Ann. Oper. Res. 60 (1), 193–208.

Masoud, N., Jayakrishnan, R., 2016. Formulations for optimal shared ownership and use of autonomous or driverless vehicles. In: Proceedings of the Transportation Research Board 95th Annual Meeting. pp. 1–17.

Masoud, N., Jayakrishnan, R., 2017a. Autonomous or driver-less vehicles: Implementation strategies and operational concerns. Transp. Res. E 108, 179–194.

Masoud, N., Jayakrishnan, R., 2017b. A decomposition algorithm to solve the multi-hop Peer-to-Peer ride-matching problem. Transp. Res. B 99, 1–29.

Masoud, N., Jayakrishnan, R., 2017c. A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system. Transp. Res. B 106, 218–236.

Masoud, N., Nam, D., Yu, J., Jayakrishnan, R., 2017. Promoting peer-to-peer ridesharing services as transit system feeders. Transp. Res. Rec.: J. Transp. Res. Board 2650, 74–83.

Mitrović-Minić, S., Krishnamurti, R., Laporte, G., 2004. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. Transp. Res. B 38 (8), 669–685.

Nam, D., Yang, D., An, S., Yu, J.G., Jayakrishnan, R., Masoud, N., 2018. Designing a transit-feeder system using multiple sustainable modes: Peer-to-peer (p2p) ridesharing, bike sharing, and walking. Transp. Res. Rec. 2672 (8), 754–763.

Paquette, J., Bellavance, F., Cordeau, J.-F., Laporte, G., 2012. Measuring quality of service in dial-a-ride operations: the case of a Canadian city. Transportation 39 (3), 539–564.

Paquette, J., Cordeau, J.-F., Laporte, G., Pascoal, M.M., 2013. Combining multicriteria analysis and tabu search for dial-a-ride problems. Transp. Res. B 52, 1–16.

Psaraftis, H.N., 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transp. Sci. 14 (2), 130–154.

Regue, R., Masoud, N., Recker, W., 2016. Car2work: A shared mobility concept to connect commuters with workplaces. Transp. Res. Rec.: J. Transp. Res. Board 2542, 102–110.

Ritzinger, U., Puchinger, J., Hartl, R.F., 2016. Dynamic programming based metaheuristics for the dial-a-ride problem. Ann. Oper. Res. 236 (2), 341–358.

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 (19), 6728–6737.

Schilde, M., Doerner, K.F., Hartl, R.F., 2011. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Comput. Oper. Res. 38 (12), 1719–1730.
Tafreshian, A., Masoud, N., 2020. Trip-based graph partitioning in dynamic ridesharing. Transp. Res. C 114, 532–553.
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. 12 (2–3), 44–60.
Tarjan, R.E., 1983. Data Structures and Network Algorithms, Vol. 44. SIAM.
Ulmer, M.W., Goodson, J.C., Mattfeld, D.C., Thomas, B.W., 2017. Route-Based Markov Decision Processes for Dynamic Vehicle Routing Problems. Technical Report. Braunschweig.
Wei, C., Wang, Y., Yan, X., Shao, C., 2017. Look-ahead insertion policy for a shared-taxi system based on reinforcement learning. IEEE Access 6, 5716–5726.
West, D.B., et al., 1996. Introduction to Graph Theory, Vol. 2. Prentice Hall Upper Saddle River, NJ.
Xiang, Z., Chu, C., Chen, H., 2008. The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. European J. Oper. Res. 185 (2), 534–551.
Yan, X., Levine, J., Zhao, X., 2019. Integrating ridesourcing services with public transit: an evaluation of traveler responses combining revealed and stated preference data. Transportation Research Part C: Emerging Technologies 105, 683–696.
Zhang, Z., Tafreshian, A., Masoud, N., 2020. Modular transit: Using autonomy and modularity to improve performance in public transportation. Transp. Res. E 141, 102033.