DEPARTMENT OF ENGINEERING MANAGEMENT

On-Demand Bus Routing Problem with Dynamic Stochastic Requests and Prepositioning

Ying Lian, Flavien Lucas & Kenneth Sörensen

UNIVERSITY OF ANTWERP
Faculty of Business and Economics
City Campus
Prinsstraat 13, B.226
B-2000 Antwerp
Tel. +32 (0)3 265 40 32
www.uantwerpen.be

FACULTY OF BUSINESS AND ECONOMICS

DEPARTMENT OF ENGINEERING MANAGEMENT

On-Demand Bus Routing Problem with Dynamic Stochastic Requests and Prepositioning

Ying Lian, Flavien Lucas & Kenneth Sörensen

RESEARCH PAPER 2022-004
MAY 2022

University of Antwerp, City Campus, Prinsstraat 13, B-2000 Antwerp, Belgium
Research Administration – room B.226
phone: (32) 3 265 40 32
e-mail: joeri.nys@uantwerpen.be

D/2022/1169/004

On-Demand Bus Routing Problem with Dynamic Stochastic Requests and Prepositioning

Ying Lian[a,*], Flavien Lucas[b], Kenneth Sörensen[a]

[a] *ANT/OR - Operations Research Group, Department of Engineering Management, University of Antwerp*
[b] *IMT Nord Europe, Institut Mines-Télécom, Univ. Lille, Centre for Digital Systems, F-59000 Lille, France*

Abstract

The On-Demand Bus Routing Problem (ODBRP) is defined as a large-scale dial-a-ride problem with bus station assignment. Specifically, each passenger can have alternative stations to board and alight; then, station pairs with the smallest total User Ride Time (URT) are chosen for overall efficiency. In the dynamic ODBRP (DODBRP), buses are only dispatched to the stations with known requests. However, this paper considers prepositioning: buses are sent to stations where new requests are likely to appear if the expected number of served requests has increased consequently. A heuristic algorithm with variable neighborhood search (VNS) is proposed to solve this dynamic and stochastic ODBRP, with multiple scenarios representing different realizations of stochastic requests. Experimental data show the superiority of prepositioning compared to DODBRP. On average, 24.27% - 38.80% more passengers can be served with the use of prepositioning with a simultaneous reduction from 2.06% to 5.93% of the average URT. In addition, different parameters are investigated to test robustness, such as instance sizes, station distributions, ratios of dynamic requests, probabilities of stochastic requests, time windows, and levels of estimation accuracy of stochastic requests.

Keywords: routing, stochastic requests, dynamic requests, multiple scenarios, prepositioning

1. Introduction

There are three types of public transit systems: scheduled public transports, taxi services, and on-demand transit services. Among them, scheduled public buses are cost-efficient and environmentally friendly; however, passengers must adjust their travel plans according to the buses' fixed routes and schedules. Contrarily, taxis are flexible in terms of routes and schedules, while they are expensive for passengers and the environment. Thus, on-demand public transit service has emerged in response to cost-efficiency and customization. It has been in use particularly in applications involving the elderly and people with disabilities. However, recently, there is an increased interest in using it in public transportation also because of the emergence of advanced technology and concern for the environment. In academic field, these problems of vehicle routing and scheduling to transit passengers are typically modeled as "Dial-A-Ride Problems" (DARP), which manages

*Corresponding author.
Email addresses: `ying.lian@uantwerpen.be` (Ying Lian), `flavien.lucas@imt-nord-europe.fr` (Flavien Lucas), `kenneth.sorensen@uantwerpen.be` (Kenneth Sörensen)

either door-to-door transportation or simply assigns passengers to their closest station to board and alight. However, the ODBRP includes bus station assignment, where each passenger has alternative stations to board and alight and the best stations are pairs with the best objective function value. Therefore, the bus station assignment improves the objective function value such as total URT substantially.

Each request is a ride from an origin to a destination with desired time. The existing literature has limitations wherein it only considers sending vehicles to locations with received requests as against those locations from where future requests have a high probability of originating (defined as "potential requests" in this paper). This can cause either missed requests owing to violation of time windows or higher routing costs to serve them. Hence, in the present study, a heuristic algorithm is developed to benefit from the information of stochastic potential requests for a dynamic and stochastic ODBRP, inspired by the emergence of accurate demand forecast such as Tang et al., 2017; Kong et al., 2016; Jia et al., 2016. More specifically, certain requests in this model are known in advance; in other words, they are static and deterministic, while other requests are dynamic and stochastic, with a certain probability of appearing in real time. Under the current problem setting, instead of waiting for the stochastic and dynamic requests to appear and then routing the buses to serve them, we use the probability of the stochastic requests and prepare the bus routes and schedule proactively.

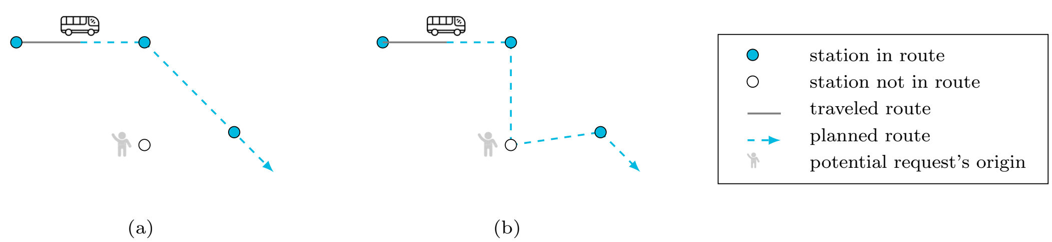


Figure 1: conceptual illustration of prepositioning

In particular, we investigate the benefit of sending vehicles to the stations with potential requests, that is, the meaning of "prepositioning" in this paper. A conceptual illustration of prepositioning is shown in Figure 1: (a) first, a potential request is detected; (b) then, the proposed algorithm tries to find a routing and schedule solution to serve it, while in the DODBRP, a potential request in (a) is ignored. Regarding stochastic requests, multiple scenarios are used to represent possible realizations. In addition, our algorithms of prepositioning explicitly consider both the removal and non-removal of empty stations. More specifically, after the realization of each time bucket, certain stations may become redundant once the aimed stochastic requests are not materialized. Thus, removal of empty stations is adopted as the recourse action.

During experiments, artificial instances are generated to test the algorithm's performance and conduct sensitivity analysis. Therefore, superiority of prepositioning over pure dynamic ODBRP and robustness hold among different parameters including instance sizes, station distributions, dynamic request percentage, probabilities of stochastic requests, and various lengths of time windows. Finally, the performance of our algorithms was validated under inaccurate estimation of the probability of stochastic requests and the result was comparable with the pure dynamic case.

The contributions of this study can be listed as follows:

It uses prepositioning with regard to stochastic requests when solving the bus routing and scheduling problem, which can improve service quality significantly;

It explicitly investigates the impact with/without removal of empty stations after each time bucket;

It investigates a wide range of parameter settings such as multiple instance sizes, station distributions, ratios of dynamic requests, probabilities of stochastic requests, alternative lengths of time windows, and finally diverse levels of estimation accuracy of stochastic requests. The robustness is proved and thus can provide guidance for practical usage of this algorithm.

2. Literature review

Our study is closely related to DARP, a variant of the vehicle routing problem (VRP), especially its dynamic and stochastic variation. Generally, DARP is used to design vehicle routes and schedules to transport passengers from their origins to destinations, with vehicles starting and ending at a depot.

In mathematical formulation, the most common objective function here is to minimize the operation costs such as total travel distance or duration subject to certain constraints related to passengers' time windows, maximum duration or detour, vehicle capacity, coupling and precedence. The latter two respectively require each passenger to board and alight from the same bus and alight after boarding.

Depending on whether information is known with certainty when computing a solution, DARP can be categorized as static or dynamic as well as deterministic or stochastic. With regard to requests, all requests are known in advance in the static case, while a part is revealed as execution proceeds in the dynamic case and real-time adjustment of solutions is required. However, if request information is known with certainty, it is deterministic; otherwise, it is stochastic. Compared with static or deterministic, dynamic and/or stochastic has received less consideration. For a recent comprehensive review of all variants of DARP, see Ho et al., 2018.

2.1. Dynamic DARP

Dynamic DARP is a frequently encountered problem when requests arrive in real time while execution proceeds and updated solution is required to respond to new requests. In this regard, periodic re-optimization is one of the standard methods to transfer dynamic DARP to static (Pillac et al., 2013), as a result, dynamic DARP can be decomposed into a series of static DARP. Subsequently, depending on how the update is triggered, dynamic DARP can be categorized as either by event (a new request comes in or a vehicle arrives at a station) (Wong et al., 2014; Hanne et al., 2009; Berbeglia et al., 2012; Marković et al., 2015; Santos and Xavier, 2015; Melis and Sörensen, 2021), or by pre-determined duration, and one of the most common methods of the latter is rolling horizon (Yang et al., 1999; Luo and Schonfeld, 2011). Dynamic DARP requires the development of techniques to manage real-time requests with quick response as well as provide high-quality solutions. Thus, research in dynamic DARP has focused on heuristics to insert new requests and/or post-optimization.

2.2. Stochastic DARP

Some studies use stochastic information to prepare for uncertain events and generate solutions with higher performance. This variation is called stochastic DARP. Different types of stochastic information have been investigated in literature, such as failures of vehicles, accidents, traffic jams, stochastic delays at pickup locations, drop-off times, stochastic travel speed, cancellation of requests (Issaoui et al., 2013; Heilporn et al., 2011; Maalouf et al., 2014; Schilde et al., 2014; Hyytiä et al., 2010; Xiang et al., 2008). To solve them, different methods have been addressed, such as two-stage stochastic programming, fuzzy logic control, and local search heuristics.

Specific to stochastic requests, variable models and solution methods have been applied. In their study, Ichoua et al., 2006 exploited knowledge about future demands and showed that a vehicle waiting at the position of its last request can be beneficial if new nearby requests have a high probability of appearing. A parallel tabu search heuristic was developed to solve this problem. Later, Ho and Haugland, 2011 brought up a probabilistic DARP where each user requires service only with a given probability and their objective was to minimize the expected travel cost, a priori set of routes was created under uncertainty in the first stage, while in the second stage, unrequested vertices were skipped. They proposed a time-efficient local search and a tabu search procedure to solve it. Schilde et al., 2011 investigated whether using stochastic information about return transports is beneficial when planning routes. They compared different heuristics categorized based on whether considering stochastic future request and maintaining a pool of solutions together with pure dynamic DARP, and they concluded that integrating stochastic information about return transports can have a positive effect on the solution quality if certain conditions are met. Per the research in recent years, Lowalekar et al., 2018 tackled a large-scale online spatio-temporal matching problem of taxi and customers. They proposed a multi-stage stochastic optimization formulation and showed superiority of the incorporation future demand. In addition, an algorithm to send vehicles to potential requests under stochastic traffic conditions has been proposed by Li et al., 2019. They used scenarios to represent different samples of future requests and traffic conditions. Experiments with real-world data show prepositioning can improve average profit and reduce waiting time. As an alternative method to deal with the stochastic requests, Tafreshian et al., 2021 generated a large pool of routes that could be valuable under different realizations of demand, in a large-scale DARP system. In the online phase, shuttles are routed proactively based on the results from the offline phase. Moreover, the online phase enables shuttles to switch between different routes in the pool allowing them to react to stochastic changes in travel patterns.

Our study is similar to the dynamic stochastic DARP in Schilde et al., 2011 and the DARP with prepositioning (Li et al., 2019), however, stochastic traffic information is not included in our study, compared with Li et al., 2019. Besides, prepositioning is not considered in Schilde et al., 2011. Instead, they only used stochastic requests to evaluate the solution quality without inserting the corresponding stations. Moreover, bus station assignment is highlighted in ODBRP (Melis and Sörensen, 2022), which significantly improves the overall performance of the solution, compared with standard door-to-door transportation in DARP. Moreover, whether to adopt the recourse action (with/without removal of empty stations) is explicitly investigated. Last but not least, we quantitively perform sensitivity analysis among various instance sizes, station distributions, ratios of dynamic requests, probabilities of stochastic requests, lengths of time windows, and finally levels of estimation accuracy of stochastic requests.

3. Problem Description

A two-stage passenger transportation problem with dynamic and stochastic requests is discussed in this study. In the first stage, static ODBRP is solved, which consists of bus routing and scheduling, passenger-bus assignment, as well as bus station assignment. Therefore, passengers are planned to be transported with the minimum total URT. In the second stage, solutions are modified in real time to serve dynamic and stochastic requests. Particularly, stations to serve stochastic requests can be inserted in bus routes and the corresponding schedules can be changed. Subsequently, the candidate solution with the largest expected number of served passengers among scenarios is implemented. Thus, in the reminder of this section, we respectively explain the ODBRP and the prepositioning approach with scenarios.

3.1. First stage: the static ODBRP

The procedure of the ODBRP to transport passengers is as follows. First, passengers send in travel requests, possibly via a mobile application or website. The requests can be sent in advance, for example, one day before their trip, or in real time to have a service as soon as possible. In this regard, the former type of passengers are categorized as static, while the latter dynamic. The static passengers are guaranteed to be served, while the dynamic passengers could be rejected if and only if any of the time windows or the bus capacity constraints are violated. Besides, passengers also specify their locations of origin and destination, while the buses are only allowed to dwell at predefined stations, so the feasible station(s) within walking distance of each passenger's origin and destination are calculated.

A formal description of the problem is as follows.

Let G = (V, A) be a directed graph, where V is the set of nodes and A is the set of arcs. The nodes in set V represent each predefined station. Unlike the most general form of transportation problems, we do not explicitly have a depot node. Instead, each bus starts from its first passenger's pickup station, and ends at its last passenger's drop-off station, while the route from or to a depot is neglected. Each node $\in V$ can be visited more than once (even by a same bus) or never. For each arc $(i, j) \in A$, the travel time t_{ij} is constant over time.

A homogeneous fleet of buses is disposed for this operation, each with a finite capacity q.

Passengers are geographically dispersed within a service area, each with a location of origin and destination, allowing us to assign to each request all stations within the walking distance. Note, the locations of origins and destinations do not belong to V, since the ODBRP does not deal with door-to-door transportation. In our problem setting, at least one station is guaranteed for each passenger to respectively board and alight. Let R_s be the set of static requests known in advance, $R_d(t)$ the set of dynamic and stochastic requests will be revealed over time. Specifically, a dynamic request $\in R_d(t)$ is revealed at time e_r. As we assume each dynamic request simply requires a ride as soon as possible, e_r is thus the earliest allowed pickup time, while the requests do not have to be picked up exactly at e_r. Then the latest arrival time l_r is calculated accordingly as explained in the next paragraph. However, in practice, dynamic requests can also require a ride later than e_r, and the impact of the gap between the received time and the earliest allowed pickup time is worthy of future investigation. For simplicity, each request corresponds to one passenger. Let the binary variables a^u_{ps} and a^o_{ps} respectively denote if passenger p can be assigned to station s for pickup and drop-off.

Passengers also indicate either their desired departure time or arrival time, then we implicitly calculate a hard time window for each passenger. In literature such as Cordeau and Laporte, 2003, a general formulation of time windows distinguishes inbound or outbound requests, as well as has separate time windows for pickup and drop-off, together with a constraint for the maximal riding time. However, in the proposed model, each request has one single hard time window of earliest departure and latest arrival $[e_r, l_r]$. The duration of each time window is set to $f \times t_{dir}$, where f is a constant, and t_{dir} is the direct travel time from the get-on to the get-off station. The value of e_r is randomly generated and then l_r is calculated as $l_r = e_r + f \times t_{dir}$. If there are more than one alternates as mentioned before, the station pair to calculate t_{dir} and thus l_r will be randomly chosen among all the combinations of the alternative get-on and -off stations. Since any alternative stations is within the walking distance of the passengers' origins and destinations, the travel time difference of different combinations are considered as negligible. The parameter f controls how strict the time windows are. A small value leads to a strict time window and vice versa.

The station sequence that each bus b has to visit is denoted SS_b, which is the routing solution of the ODBRP. Each station in SS_b is visited only when needed, i.e. at least one passenger needs to get on or off.

Each bus b's arrival and departure times at station $s \in SS_b$ are respectively denoted as t^a_{bs} and t^d_{bs}.

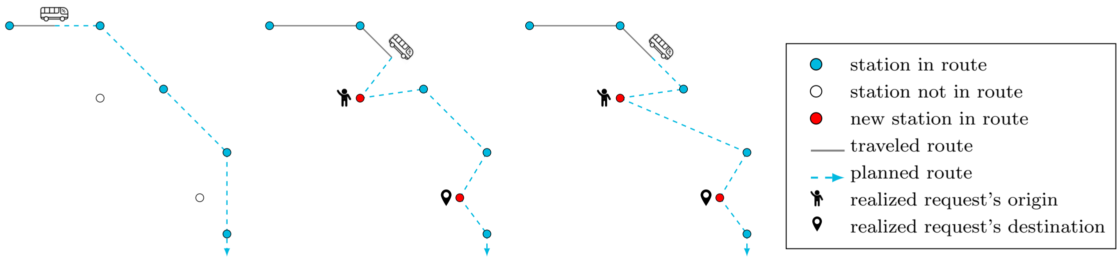
Buses are allowed to wait at station s, even if there are passengers on board, under the condition that at least one passenger r is boarding at s and the arrival t^a_{bs} is prior to the earliest departure e_r. Therefore, the departure time t^d_{bs} may be later than the arrival time t^a_{bs}, and the difference is called *waiting time* t^w_{bs}, where $t^w_{bs} = t^d_{bs} - t^a_{bs}$. The reason to allow waiting is that it is a way to decrease the total URT and thus indirectly increase the possibility to serve more passengers. Although it increases the URT of passengers aboard, it can reduce others'. However, if the holding policy does not apply, we assume passengers board and alight immediately when the bus arrives, and this service duration is omitted, namely t^d_{bs} is simply equal to t^a_{bs}.

Passenger-bus assignment, i.e. which bus to serve each passenger, is also a decision to be made by the ODBRP. Therefore, each bus is assigned a list of passengers $P_b(t)$ who will be served, including their pickup and drop-off times and locations. The list may vary when new requests come in as time t lapses.

The exact mathematical formulation of static ODBRP can be found in Appendix A, where the objective function is to minimize the total URT, subject to time windows, bus capacity, precedence and coupling.

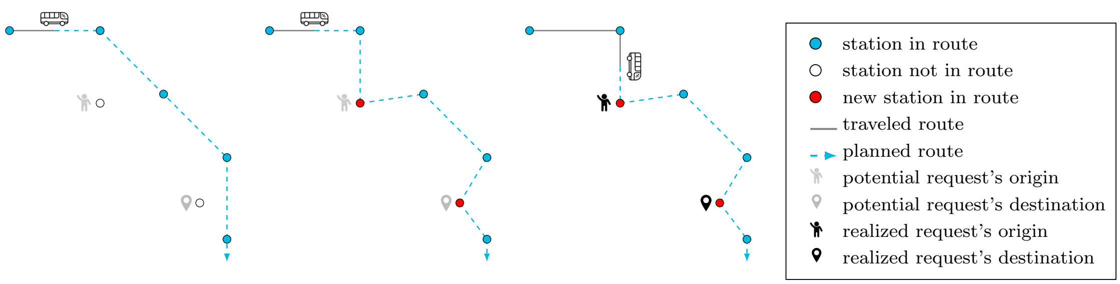
3.2. Second stage: prepositioning with dynamic and stochastic requests

Then in the real-time stage, the ODBRP with dynamic and stochastic requests as well as prepositioning is solved. In order to differentiate prepositioning from DODBRP, let us start from the graphic illustrations of both cases. In DODBRP (Figure 2), the bus originally runs along the planned route (Figure 2a); the moment a dynamic request appears, the algorithm is triggered to find a feasible routing and scheduling solution to serve it. Given consideration to human drivers, altering the bus route immediately from its current position is not allowed such as in Figure 2c, but only after visiting its next station (Figure 2b). In the prepositioning case (Figure 3), the moment this potential request is detected (Figure 3a), the algorithm is triggered as in (Figure 3b), and then when this request actually appears, a bus could be already on the way to picking up the corresponding passenger (Figure 3c).



(a) (b) (c)

Figure 2: procedure of DODBRP



(a) (b) (c)

Figure 3: procedure of prepositioning

To implement prepositioning, we first discretize the time horizon into an ordered set $\tau = \{0, 1, 2, ..., T\}$ of time buckets, each with a small constant duration H such as 5 minutes. We use scenarios to represent the samples of stochastic requests. $S(t) = \{S_1(t), S_2(t), ..., S_N(t)\}$ is the set of scenarios used at time t_τ, each contains the dynamic requests that appear in the time bucket $[t_\tau, t_{\tau+1})$. In addition, N denotes the number of scenarios, which is fixed for all time buckets. Each dynamic and stochastic request has a probability to appear in each scenario.

Scenarios can be generated according to historical data or a request forecasting model in practice. While in this study, each stochastic request is simply assigned a probability $prob \in (0, 1)$, and whether it appears in the set of scenarios $S(t)$ is i.i.d. In other words, the expected number of scenarios containing this request is thus $prob \times N$.

Let us make two simplifications of our model: first, for each stochastic request, only whether it appears or not is unknown, but the time to receive it is deterministic if it appears. In other words, if a request is not sent at a specific time point, it is impossible to be sent later on either. Thus, for all scenarios containing it, its information are exactly the same, i.e. its time window, a^u_{ps} and a^o_{ps}. Given this assumption, after the realization of each time bucket, it could be a reasonable recourse action to remove the stations prepared for the no-show requests and adjust the schedule at each station. A conceptual illustration of the recourse action is in Figure 4: (4a) first, get-on and get-off stations are inserted for a potential request; (4b) once the request does not materialize, the corresponding stations are removed if and only if they have not been visited yet, or going to be visited as the next station. The reason for keeping the next station unchanged is human drivers are assumed in this project; however, autonomous buses could change the routes immediately. In contrast, if the recourse action is not applied, the route and the schedule remain the same as in (figure 4c). With or without this recourse action and the impact are also investigated in this paper.

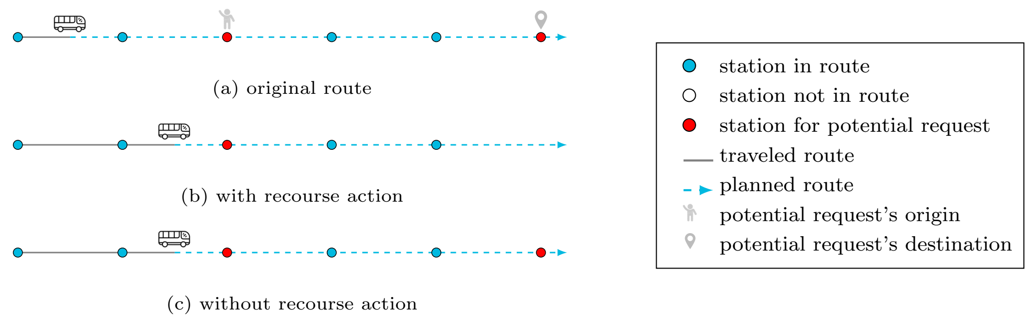


Figure 4: conceptual illustration of the recourse action in prepositioning

Our second simplification assumes the scenario set $S(t)$ contains all the combinations of the stochastic requests, despite it can hardly hold in reality. Since each stochastic request is independent from others, then if there are m stochastic requests within a time bucket, and each appears or not, the possible combinations of request realization is thus 2^m, namely it grows exponentially with m. Therefore if m is large while the number of scenarios N is small, none of the scenarios may represent the true realization. However, our second simplification in this work neglects the impact of N. We leave for future study the analysis of N's size and its impact on the accuracy of representing stochastic requests, as well as on the effectiveness of prepositioning.

The objective function varies between two stages. In the first stage, all the requests have to be served and the objective function is to minimize the total URT, while in the second stage, it is unrealistic to impose the buses to serve all the dynamic requests owing to the hard time windows and the assumption that there are no more buses to be dispatched, therefore, the objective function is to maximize the expected number of served passengers. Thus, the aim of this research is to design an online algorithm to proactively route buses to achieve the objective in the second stage. Meanwhile, given dynamic requests are revealed over time, it is likely to achieve local optima if we only aim to maximize the expected number of served passengers, so we also minimize the total URT of all accepted requests in each time bucket to indirectly increase the possibility to serve more passengers later on.

4. Solution method

This section explains the solution method in two stages sequentially. In the first stage containing only static and deterministic requests, a greedy insertion heuristic in section 4.1.1 is applied to construct the initial solution, followed by a VNS to improve the solution explained in section 4.1.2. Then, in the second stage the requests are dynamic and stochastic, while in each scenario, the problem to be solved is a dynamic and deterministic ODBRP where the greedy insertion and VNS are applied again, and their similarities as well as differences compared to the first stage are explained in 4.2.1; additionally, the procedure of prepositioning is illustrated in section 4.2.2.

4.1. The first stage: static and deterministic ODBRP

In the first stage, greedy insertion is implemented to construct the initial solution, this is explained in section 4.1.1; then the VNS with three LS operators is implemented to improve the solution quality, with details in section 4.1.2.

4.1.1. Greedy insertion

Greedy insertion is the constructive heuristic used each time inserting a new request. To start with, the requests are ranked according to their earliest allowed pickup time e_r, i.e. the request with the smallest e_r is the first. Then the procedure of greedy insertion is, for each request r, we first try to loop over all the buses and try to insert it in every position of each bus. The order of the buses is simply the id of each bus; each position is checked in sequence to insert the get-on and get-off stations. If there are more than one feasible solutions, the one with the minimal total URT is chosen; if no feasible solution can be found, the r is rejected. After this insertion procedure with every request, VNS is triggered to further possibly reduce the total URT.

4.1.2. VNS

Following the constructive heuristic, VNS is applied to improve the solution quality. The neighborhood structure consists of three local search operators: alternative get-off stations, swap, and reinsert. The concept of each LS operator is as follows.

Alternative get-off stations means to replace a passenger's get-off station with its substitute, if the passenger r has at least two options. If this station is solely in use by this passenger, it is simply replaced; otherwise, the original station remains, while the substitute is inserted right in front.

Swap means to exchange a passenger's get-off station with one in an earlier position. If the passenger's get-off station is also occupied by other passengers, swap is allowed only if the precedence constraint still applies to them.

Finally, the operator *reinsert* tries to remove a passenger from a route and reinsert it into an another route if the total URT is reduced.

The three LS operators are applied in the same sequence as introduced. Each operator terminates if the solution is not further improved after a round with respect to each passenger. The procedure of VNS is outlined in algorithm 1.

Algorithm 1: VNS, post-optimization after greedy insertion

Result: improve initial solution with VNS

for *Each passenger P* **do**
    Alternative();
    Implement the solution with minimal total URT;

for *Each passenger P* **do**
    Swap();
    Implement the solution with minimal total URT;

for *Each passenger P* **do**
    Reinsert();
    Implement the solution with minimal total URT;

4.2. The second stage: dynamic and stochastic ODBRP

The second stage deals with dynamic and stochastic requests. Similar to the first stage, the greedy insertion and VNS are also applied upon and after inserting new requests. However, there are differences from the first stage, both with and without prepositioning. The details are explained in section 4.2.1. The procedure of applying prepositioning is then explained in section 4.2.2.

4.2.1. Greedy insertion and VNS: similarities and differences

Greedy insertion and VNS are applied in both stages. Besides, in the second stage, there are two cases depending on whether prepositioning is applied. The without prepostioning case is simply the DODBRP, that is, inserting each dynamic request in the current solution after it is actually received; the with prepositioning case can insert each stochastic request in the current solution proactively, as described in section 3 and will be explained in section 4.2.2.

The greedy insertion procedure described in section 4.1.1 is identical with the three cases: static requests, dynamic requests with or without prepositioning. Namely, all the requests are sorted in ascending order of their earliest allowed departure and every combination of the positions in each bus is tested to insert r, then the solution with the minimum total URT is selected or r is rejected if no feasible solution is found.

However, there are differences in terms of the request set and allowed positions. First, the set of r is different. For the initial solution in the first stage, the set is R_s, while for the DODBRP, $r \in R_d(t)$; finally, for ODBRP with prepositioning, the request set is $S_n(\tau)$, that is, the scenario in use at time t_τ.

The second difference concerns the allowed range of modification of the bus routes and schedule, both for greedy insertion and VNS. In the second stage when the current time is t ($t \geq 0$), the range can be modified is limited, i.e. the routes have been traveled or the stations will be visited the next when the time is t cannot change as explained in section 3.2, while in the static stage there is no limitation. As for t, it is variant as the simulation executes. Particularly, for the DODBRP, $t := e_r$ each time a dynamic request r appears at e_r (since we assume for each dynamic request the moment it is received the corresponding passenger is ready to depart in section 3.2), while for the case with prepositioning, t is equal to t_τ, as τ is set to $0, 1, 2, ..., T$ sequentially in section 3.2.

4.2.2. Method of prepositioning

In this subsection, the procedure to implement prepositioning with scenarios is explained in detail.

As explained in section 3, the requests are deterministic in each scenario and the problem to be solved is thus the deterministic ODBRP. Consequently, in each scenario, the buses' routing and scheduling after t_τ are changed accordingly to maximize the number of accepted new requests in the time bucket with the smallest total URT. We call the corresponding solution of buses' routing and scheduling a candidate solution $C_n(\tau)$. Among all candidate solutions generated by each scenario, the one fits all scenarios the best (will be explained in the next sentences) is selected. To be specific, each candidate solution is tested on other scenarios $S_1(\tau), S_2(\tau), ..., S_N(\tau)$, to calculate how many passengers can be served and the total URT, i.e. $obj_n(\tau)$ (both the number of served passengers and the total URT are stored in $obj_1(\tau), obj_2(\tau), ..., obj_N(\tau)$). In other words, given each candidate solution's routes and schedules at each station, the requests of each other scenario are inserted in sequence: each request can be served if and only if no violation occurs; each accepted request is inserted at the position with the least increase in the total URT. Then for each candidate solution $C_n(\tau)$, the average value of served requests among all scenarios is calculated, together with the

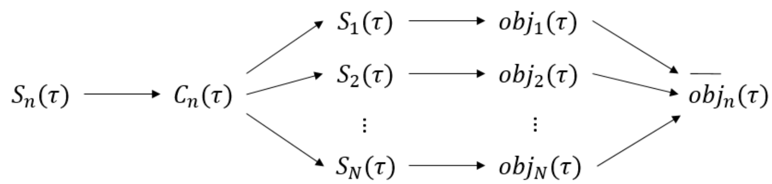


Figure 5: Illustration of the scenario-based method

average total URT, the two average values form $\overline{obj_n(\tau)}$. The motivation of the aforementioned procedure is, since at time t_τ, it is unknown which stochastic dynamic request will actually realize, it is preferable to maximize the expected value of served passengers. Afterwards, among all candidate solutions, the one with the largest expected number of served passengers will be implemented in the next time bucket; if two or more candidate solutions result in the largest expected number of served passengers, the one with the smallest total URT will be implemented: this criterion corresponds to the objective function described in section 3. The procedure illustrated above is also shown in Figure 5, which is to generate $C_n(\tau)$ given $S_n(\tau)$, and subsequently evaluate $C_n(\tau)$ among all scenarios $S(\tau)$ by the average objective value.

When at time t_τ, one specific scenario $S_n(\tau-1)$ is actually realized, and for each stochastic request between $[t_{\tau-1},\ t_\tau)$, it becomes deterministic, either has appeared or not. Namely, the stochastic requests contained in $S_n(\tau-1)$ have appeared, while those that have not appeared are false information. Apparently, the realized scenario does not necessarily correspond to the one generating the best candidate solution, so there can be empty stations in routes, which have been intended for the no-show requests. According to our assumption, this can cause detours and increase the total URT, which potentially prevents the buses serving upcoming requests, unless any of them happen to use an empty station timely. However, our model aims for on-demand transportation and tries to avoid an instance when a bus passes by a station with no one getting on or off. Hence in terms of the recourse action at t_τ, we then have two strategies, one is to remove the empty stations, while the other is without the recourse action. However, for the former strategy, the empty stations already visited before t_τ or upon being visited as the next station cannot be modified, instead, only the remaining route from the next station to the end. Algorithm 2 outlines the procedure of ODBRP with prepositioning, where the *if*-block (line 3 - 5) is thus the recourse action for the former strategy ($\tau > 0$ such that there is one realized scenario), while it does not exist for the latter one. For easier comparison, the procedure to insert dynamic requests in DODBRP is also outlined in Algorithm 3.

Algorithm 2: Procedure of ODBRP with prepositioning

for *realized_scenario* = *1, 2, ..., N* **do**
  for τ = *0, 1, ..., T* **do**
    if $\tau > 0$ **then**
      for *b in all buses* **do**
        for *empty stations* **do**
          Remove empty station i iff $Ta_{bi} > t_\tau$ and i is not the station b is running towards;
    for *n* = *1, 2, ..., N* **do**
      for *each* $r \in S_n(\tau)$ **do**
        insert r at the position with min. increase of total URT;
        VNS, Algorithm 1;
      Evaluate $C_n(t)$;
    Implement best $C_n(t)$;
Calculate the average objective value of N scenarios.

Algorithm 3: Procedure to insert dynamic requests, without prepositioning

for *realized_scenario* = *1, 2, ..., N* **do**
  for *each* $r \in R_{dn}$ **do**
    insert r at the position with min. increase of total URT;
    VNS, Algorithm 1;
Calculate the average objective value of N scenarios.

5. Computational Experiments

All tests were performed in C++ on a Windows 10 computer system, an Intel® Core^{TM} i7-8850H, a 2.60Ghz processor and 16GB RAM. In this section, we generated instances to test our algorithm's performance. First, how test instances were generated is explained in section 5.1. Then different experiments were conducted to test the effectiveness and robustness in addition to provide guidance for practical usage. Subsequently, the results of different parameters and variant instances are presented in section 5.2.

5.1. Data generation

A set of instances are randomly created for experiment. To mimic the travel demands in an urban area in real world, we design a dense neighborhood with the following parameter settings of stations and requests.

There are 12 stations with two distributions, i.e. along a single line (denoted Ln), and in a grid with 3 rows × 4 columns (denoted Gr), as shown in Figure 6. Then the travel time between two neighboring stations is equal and set to 3 minutes.

(a) L_n (b) G_r

Figure 6: Shapes of the instances

As for the requests, all of them travel from left to right in Ln, while they can travel in any direction in Gr. The motivation to set the travel direction is, we would like to use Ln to simulate a bus line where even if stations can be skipped, not much travel distance can be saved. This case can happen commonly in practice, such as when the buses running almost along a straight line and arterial roads. Similarly, we would like to use Gr to represent a popular area within which passengers travel in any direction. We assume Ln and Gr are two simple constitutional units in real road network, however, the lack of more realistic networks can be considered as a limitation of this work. Requests' origins and destinations are randomly generated near the stations. The maximum number of feasible get-on or get-off choices is limited to 2. With regard to the number of requests, we have 3 different levels: small, medium, and large. To start with, we assume all static requests are deterministic, while all dynamic requests are stochastic, and the probability of each dynamic and stochastic request's materializing has three levels: 30%, 50%, and 80%.

We then set the ratio between the static and expected dynamic requests equal to 2:3 when the probability is 50%. Given these parameters, the small instances contain 16 static requests and 48 dynamic requests, then in total 64 requests to consider, thus 24 expected dynamic requests if the probability is 50%. However, if we consider that only 30% of the expected dynamic requests will effectively appear, thus the instances contain only 14.4 dynamic requests on average; the instances still have 16 static request, thus in total 30.4 requests to schedule on average. Similarly, when the probability becomes 80%, the instances contain 38.4 dynamic requests on average, and with addition of 16 static requests, there are in total 54.4 requests. Subsequently, the medium and large instances are assigned in the same manner. Namely, the medium instances have 28 static requests and 84 dynamic requests, then in total 112 requests, and thus respectively 25.2, 42, 67.2 expected dynamic requests for the probability 30%, 50% and 80%. Furthermore, the large instances have 80 static requests, 240 dynamic requests, then in total 320 requests, and thus respectively 25.2, 42, 67.2 expected dynamic requests for the probability 30%, 50% and 80%. The exact numbers are listed in table 1 for readers' convenience. In the table, the following notations are in use: the sizes S —small, M —medium, L —large; % —probability of stochastic requests to materialize; No. static —number of static requests; D —number of dynamic requests, R—number of total requests, E(D) —expected number of dynamic requests, E(R) —expected number of total requests. These instances serve as our basic instances, but in order to further perform sensitivity analysis, we also vary the ratio between static and expected dynamic requests, which will be explained below.

Table 1: Number of requests per instance class

Distribution	Size	%	No. static	D	R	E(D)	E(R)
Ln	S	30	16	48	64	14,4	30,4
	S	50	16	48	64	24	40
	S	80	16	48	64	38,4	54,4
	M	30	28	84	112	25,2	53,2
	M	50	28	84	112	42	70
	M	80	28	84	112	67,2	95,2
	L	30	80	240	320	72	152
	L	50	80	240	320	120	200
	L	80	80	240	320	192	272
Gr	S	30	16	48	64	14,4	30,4
	S	50	16	48	64	24	40
	S	80	16	48	64	38,4	54,4
	M	30	28	84	112	25,2	53,2
	M	50	28	84	112	42	70
	M	80	28	84	112	67,2	95,2
	L	30	80	240	320	72	152
	L	50	80	240	320	120	200
	L	80	80	240	320	192	272

The earliest allowed time to start operating is 0. Each request's earliest departure time is randomly chosen from the time period [0, 20]. The longest travel time between stations is 11 × 3 = 33 minutes for Ln, and $\sqrt{(3 \times 3)^2 + (3 \times 2)^2} \approx 10.8$ for Gr. We choose our time window length, i.e. l_r - e_r = 1.6 $\times t_{dir}$ and 1.1 $\times t_{dir}$ respectively for broad and strict TW, where e_r is randomly generated and t_{dir} is the direct ride time as introduced in section 3. Given the above calculation, for Ln, our operation time is [0, 20 + 33 × 1.6] = [0, 72.8] for broad TW, and [0, 20 + 33 × 1.1] = [0, 56.3] for strict, while for Gr, [0, 20 + 10.8 × 1.6] ≈ [0, 37.3] for broad TW, and [0, 20 + 10.8 × 1.1] ≈ [0, 31.9] for strict.

To serve the requests, we have $min(No.static, 20)$ homogeneous buses, each with 6 seats. More precisely, if the number of static requests is smaller than 20, then the number of buses is equal to the number of static requests. Otherwise, the number of buses are fixed to 20 despite the size of requests. The motivation for this setting is, in the ODBRP the route from and to the depot(s) are neglected as aforementioned in section 3, as the travel from and to the depot(s) do not contribute to the total URT. Hence, the case when an empty bus is dispatched from the depot to serve a dynamic request is simply neglected. As for the case when the number of buses is equal to the number of static requests, each bus is initially dispatched to serve a static request, while when dynamic requests come later, their routes and schedule can be adjusted for them. The entire setting can be seen realistic since in practice buses are still scheduled even they are mostly empty and new passengers come dynamically and stochastically.

5.2. Results

To begin with, we evaluate the impact of factor H which means the length of a time bucket to look ahead, i.e. the stochastic requests within[t, t+H) is considered. Small H means we make a relatively short-term plan and vice versa. We set H respectively equal to 2, 5, 10, 20 minutes and tested among all instances the solution quality. The average relative results measured by the number of served passengers compared with pure DODBRP are presented in table 2 with the following notations: P denotes number of served passengers, while U denotes average URT. For P, positive values indicate that prepositioning approach serves more passengers, while for U, negative (positive) values indicate that prepositioning approach reduces (increases) average URT. As shown in the table, the effectiveness of prepositioning decreases monotonously as H increases. Average URT is excluded from the objective function. However, this value is also listed as a reference. From this, we conclude only considering the stochastic requests in near future is most beneficial. The reasons can be, first, the problem is more complicated owing to the size of requests for larger H, thus it is harder to find a decent solution. Besides, shorter H results in more rounds of algorithm execution and local search, thus the solution is likely to be improved. Compared with literature, despite different problem settings, Schilde et al., 2011 also obtained the minimized tardiness as their primary objective when H was the smallest, while Li et al., 2019 obtained larger profit, lower waiting time yet larger detour as H increased.

After H is determined, we fix H = 5 and compare our solution methods, that is, prepositioning with/without removal of empty stations, with pure DODBRP. The average results among all 18 instances are listed in table 3. In the tables 3, 4 and 5, the notations are as following: P —number of served passengers, U —average URT, Pr + (-) Re —prepositioning with (without) removal of empty stations, br (st) —broad (strict) TW. For P, positive values indicate that prepositioning approach serves more passengers, while for U, negative (positive) values indicate that prepositioning approach reduces (increases) average URT.

Table 2: Average gap in solution quality relative to pure dynamic ODBRP with variant lengths of H

H in min	2	5	10	20
gap(P) in %	43,05	37,01	26,80	1,77
gap(U) in %	-1,31	-0,72	1,05	0,01

The results are shown in percentage relative to its DODBRP counterpart. The results show the superiority of prepositioning over DODBRP, in terms of more expected served passengers and less average URT. This holds under different situations of demand intensity, station distribution, time window constraints, and probability of stochastic requests. More precisely, compared with DODBRP, using prepositioning can serve on average 24.27% - 38.80% more requests, and simultaneously reduce by 2.06% - 5.93% the average URT. The effectiveness of prepositioning holds for both station distributions. We could infer from this that prepositioning can benefit both transportation along single bus lines and within popular areas. The benefit of using prepositioning becomes more obvious as demand level changes from low, medium to high. In addition, if the stochastic requests have higher probability of materializing, the benefit is also more significant. This conclusion contradicts paper Schilde et al., 2011, where their similar algorithm has a lager advantage over the pure myopic approach, when the stochastic requests are less likely to happen. We assume two possible reasons. The first one is the different approach, i.e. they used stochastic requests only to evaluate the solution quality without inserting the corresponding stations. The other possible reason could due to their formulation of stochastic requests, the time when each stochastic request occurs is also stochastic within a time interval, this adds considerably more uncertainty in the system. While in our model, the time of which is deterministic. Another finding is the case with removal of empty stations is in general slightly worse than the version of without. One would expect the opposite: If the expected stochastic requests have not materialized and thus the corresponding stations are likely to be redundant, it could reduce detours by skipping them for Gr, and/or possible waiting time at them for Ln and Gr. However, the statistical analysis of the solutions show that this is mainly because the stations prepared for stochastic requests are usually at the end of a bus route or due to a zero (or rather small) detour when an empty station is in line with (or close to) its preceding and succeeding ones. A larger network with more distant stations could be interesting for future work.

Another set of experiments was conducted to see the influence of the ratio between static and dynamic requests. In specific, apart from the ratio of static and the expected number of dynamic requests 2:3 given in the last section 5.1, two more levels 1:4 and 3:2 were tested, while the expected number of total requests remain the same as 2:3. The detailed dimensions for each instance and the corresponding results are in table 4. As the results show, the more dynamic (and stochastic) requests, the more benefit from prepositioning.

Finally, as explained in section 3, each stochastic request has a probability *prob* of appearing. In this regard, the impact of forecast accuracy was also tested by varying *prob*. More specifically, solutions were generated with one probability while requests with another probability were actually realized. The results of our algorithm under inaccurate estimation are presented in table 5, with the following notations: % Forecast —the data used to generate solutions, % True —the data actually realize.

The overall performance under different cases is comparable with DODBRP. A few observations can be drawn from the results: larger instances suffer more from inaccurate estimation, so are broader time windows; besides, overestimation is slightly better than underestimation. Nevertheless, the gaps are small.

To summarize, the above observations are encouraging. Despite the limitations of the modeling, the satisfactory solution quality of the prepositioning shows a possible advantage to being applied in practice.

Table 3: Average gap in solution quality relative to pure dynamic ODBRP

Distribution	Size	%	No. static	D	R	E(D)	E(R)	Pr + Re + br (%)		Pr - Re +br (%)		Pr + Re + st (%)		Pr - Re + st (%)	
								P	U	P	U	P	U	P	U
	S	30	16	48	64	14,4	30,4	20,10	1,46	20,10	1,46	12,30	-1,24	12,30	-1,24
	S	50	16	48	64	24	40	31,23	-0,74	32,02	-1,35	34,68	-3,32	35,14	-3,68
	S	80	16	48	64	38,4	54,4	46,46	-2,78	47,14	-2,38	35,83	-6,92	35,83	-6,92
	M	30	28	84	112	25,2	53,2	26,04	-0,88	27,60	0,76	19,61	-2,83	20,26	-2,32
Ln	M	50	28	84	112	42	70	39,36	-2,29	38,90	-1,23	34,95	-6,39	34,95	-6,17
	M	80	28	84	112	67,2	95,2	47,63	0,51	47,83	0,99	50,38	-6,00	49,87	-6,23
	L	30	80	240	320	72	152	36,28	-4,81	36,66	-5,56	29,96	-8,68	30,87	-9,20
	L	50	80	240	320	120	200	40,17	-3,95	40,75	-4,34	37,46	-9,20	37,57	-8,11
	L	80	80	240	320	192	272	54,83	-8,44	54,90	-6,94	57,38	-8,76	59,33	-8,45
	S	30	16	48	64	14,4	30,4	18,41	-2,70	25,37	-2,97	20,00	-4,04	21,14	-4,35
	S	50	16	48	64	24	40	25,78	-5,44	37,78	-5,22	15,87	-5,16	25,40	-6,23
	S	80	16	48	64	38,4	54,4	41,24	-3,36	49,27	-1,44	17,56	-5,49	24,39	-7,05
	M	30	28	84	112	25,2	53,2	25,44	-3,25	25,15	-3,22	14,84	-1,43	17,97	-1,42
Gr	M	50	28	84	112	42	70	41,52	-2,69	44,56	-2,65	22,74	-2,74	23,47	-2,88
	M	80	28	84	112	67,2	95,2	49,22	-0,43	50,55	-0,71	35,54	-4,63	38,55	-3,91
	L	30	80	240	320	72	152	25,13	-3,11	24,88	-1,57	21,10	-3,87	21,50	-3,87
	L	50	80	240	320	120	200	37,19	-1,93	38,09	-2,13	33,03	-5,41	34,47	-5,60
	L	80	80	240	320	192	272	54,04	-2,90	53,58	-3,37	37,72	-4,19	41,47	-4,53
average-all								36,67	-2,65	38,62	-2,33	29,50	-5,01	31,36	-5,12
average Ln								38,01	-2,44	38,43	-2,06	34,73	-5,93	35,13	-5,81
average Gr								35,33	-2,87	38,80	-2,59	24,27	-4,10	27,60	-4,43
S								30,54	-2,26	35,28	-1,98	22,71	-4,36	25,70	-4,91
M								38,20	-1,51	39,10	-1,01	29,68	-4,00	30,85	-3,82
L								41,27	-4,19	41,48	-3,98	36,11	-6,68	37,54	-6,63
30								25,23	-2,22	26,63	-1,85	19,64	-3,68	20,67	-3,73
50								35,87	-2,84	38,68	-2,82	29,79	-5,37	31,83	-5,45
80								48,90	-2,90	50,54	-2,31	39,07	-6,00	41,58	-6,18

Table 4: Average gap in solution quality relative to pure dynamic ODBRP, variant ratios

Distribution	Size	Ratio	No. static	D	R	E(D)	E(R)	Pr + Re + br (%)		Pr - Re + br (%)		Pr + Re + st (%)		Pr - Re + st (%)	
								P	U	P	U	P	U	P	U
	S	1:4	8	64	72	32	40	41,71	-6,14	46,52	-2,78	26,11	-2,96	30,57	-4,35
	S	2:3	16	48	64	24	40	31,23	-0,74	32,02	-1,35	34,68	-3,32	35,14	-3,68
	S	3:2	24	32	56	16	40	19,28	-2,75	21,90	-3,07	15,52	-3,51	20,94	-5,80
	M	1:4	14	112	126	56	70	47,75	0,70	45,35	0,76	53,41	-3,06	52,65	-3,53
Ln	M	2:3	28	84	112	42	70	39,36	-2,29	38,90	-1,23	34,95	-6,39	34,95	-6,17
	M	3:2	42	56	98	28	70	13,04	-2,53	13,23	-2,88	9,27	-3,32	9,76	-3,54
	L	1:4	40	320	360	160	200	62,42	1,32	61,52	1,28	58,82	-4,68	59,25	21,47
	L	2:3	80	240	320	120	200	40,17	-3,95	40,75	-4,34	37,46	-9,20	37,57	-8,11
	L	3:2	120	160	280	80	200	17,71	-2,47	17,94	-2,64	19,38	-25,37	22,25	-5,15
	S	1:4	8	64	72	32	40	61,82	-3,52	63,03	-3,70	41,84	-9,22	43,88	-8,53
	S	2:3	16	48	64	24	40	25,78	-5,44	37,78	-5,22	15,87	-5,16	25,40	-6,23
	S	3:2	24	32	56	16	40	15,69	-1,46	19,93	-2,55	7,11	-2,83	10,28	-3,02
	M	1:4	14	112	126	56	70	47,57	-3,62	52,81	-3,98	31,55	-6,67	43,32	-9,12
Gr	M	2:3	28	84	112	42	70	41,52	-2,69	44,56	-2,65	22,74	-2,74	23,47	-2,88
	M	3:2	42	56	98	28	70	15,42	-3,06	19,27	-3,23	8,09	-2,30	8,09	-2,30
	L	1:4	40	320	360	160	200	50,19	-0,19	52,32	-0,94	32,54	-5,10	36,29	-5,48
	L	2:3	80	240	320	120	200	37,19	-1,93	38,09	-2,13	33,03	-5,41	34,47	-5,60
	L	3:2	120	160	280	80	200	16,63	-1,51	16,33	-0,67	11,89	-2,99	12,21	-2,80
average-all								34,69	-2,35	36,79	-2,30	27,46	-5,79	30,03	-3,60
average Ln								34,74	-2,09	35,35	-1,80	32,18	-6,87	33,68	-2,10
average Gr								34,64	-2,60	38,24	-2,79	22,74	-4,71	26,38	-5,11
1:4								50,63	-1,37	51,13	-0,25	40,71	-5,28	44,33	-1,59
2:3								35,87	-2,84	38,68	-2,82	29,79	-5,37	31,83	-5,45
3:2								16,68	-2,58	17,69	-2,86	11,88	-6,72	13,92	-3,77
S								30,74	-3,21	33,48	-2,40	23,52	-4,50	27,70	-5,27
M								34,11	-2,25	35,69	-2,20	26,67	-4,08	28,70	-4,59
L								40,10	-1,70	40,07	-1,90	27,55	-4,20	30,93	-5,09

Table 5: Average gap in solution quality relative to pure dynamic ODBRP, inaccurate probabilities

Distribution	Size	% Forecast	% True	Pr + Re + br (%)		Pr - Re + br (%)		Pr + Re + st (%)		Pr - Re + st (%)	
				P	U	P	U	P	U	P	U
Ln	S	30	80	-1,68	-1,10	-2,02	-0,06	0,83	-0,24	0,00	0,00
	S	80	30	0,00	0,18	0,00	0,60	0,00	0,00	1,07	1,03
	M	30	80	-0,59	-0,42	0,00	-0,46	-1,01	0,26	-1,01	0,19
	M	80	30	-1,04	-0,01	-1,82	-0,09	-0,65	0,03	-0,98	-0,12
	L	30	80	-2,34	-0,89	-4,07	-0,84	-2,23	0,12	-1,76	-0,22
	L	80	30	-1,36	-0,54	-1,75	0,29	-1,82	0,12	-2,59	0,10
Gr	S	30	80	0,36	0,05	3,65	-0,16	0,98	0,83	3,41	1,03
	S	80	30	1,00	-0,11	4,98	-0,13	0,00	0,00	1,71	-0,29
	M	30	80	0,44	0,21	1,33	0,24	1,20	0,23	2,41	0,47
	M	80	30	0,58	0,13	1,17	0,26	2,34	-0,18	5,08	-0,37
	L	30	80	1,65	-0,47	0,83	-0,52	0,31	-0,56	1,10	-0,15
	L	80	30	-0,50	0,03	-0,38	2,08	0,79	-0,18	1,38	-0,32
average				-0,29	-0,25	0,16	0,10	0,06	0,03	0,82	0,11
Ln-ave				-1,17	-0,46	-1,61	-0,09	-0,81	0,05	-0,88	0,16
Gr-ave				0,59	-0,03	1,93	0,30	0,94	0,02	2,52	0,06
S				-0,08	-0,25	1,65	0,06	0,45	0,15	1,55	0,44
M				-0,15	-0,02	0,17	-0,01	0,47	0,08	1,37	0,04
L				-0,64	-0,47	-1,34	0,25	-0,74	-0,13	-0,47	-0,15
underestimate				-0,36	-0,44	-0,05	-0,30	0,02	0,11	0,69	0,22
overestimate				-0,22	-0,05	0,37	0,50	0,11	-0,04	0,94	0,01

6. Conclusion

In this paper, we explicitly consider sending vehicles to stations with potential requests and propose heuristic algorithms with or without the recourse action for the dynamic stochastic on-demand bus routing problem. The main objective of this work is to investigate whether taking future requests into account while routing and scheduling is beneficial to solution quality. As a result, this is the case among a wide range of variants. More specifically, the prepositioning strategy outperforms the pure dynamic solution, especially when the size of requests is larger, the stochastic requests have higher probability to materialize, or the percentage of dynamic requests is larger.

With respect to the prepositioning strategy, we also found the time bucket H is an important factor of the performance, which determines the period in which potential requests are considered. To be specific, a rather short H is preferable to maintain a significant improvement of solution quality, as we test H ranging from 2 to 20 minutes, the solution quality monotonously decreases: for the case of 2 minutes, the expected number of served passengers is 43.05% more than the pure dynamic case, while 20 minutes, the result is only 1.77%.

In addition, the impact of inaccurate forecast of stochastic requests is investigated, in terms of the probability of which to materialize: the performance of the proposed method is slightly worse yet comparable to the dynamic ODBRP when there is a lack of knowledge concerning the uncertain demand; however, the more accurate this knowledge is, the better the proposed method performs. In general, prepositioning strategy can be powerful and robust among a wide range of parameter settings. Besides, tiny differences exist among variant sizes of requests and whether the probability is overestimated or underestimated.

Another interesting finding is the recourse action where empty stations are removed after each realized time bucket does not improve the solution quality. Instead, without recourse action slightly outperforms in nearly all cases, even when the probability of stochastic requests are inaccurate. This is usually because the stations prepared for stochastic requests are at the end of a bus route, or due to a zero (or rather small) detour when an empty station is in line with (or close to) its preceding and succeeding ones.

In future work we plan to investigate more realistic formulation of stochastic requests, and other heuristic algorithms to solve this type of problems.

Acknowledgements

The authors thank University of Antwerp for full funding.

CRediT authorship contribution statement

Ying Lian: Conceptualization, Methodology, Software, Formal analysis, Investigation, Drafting, Writing - review and editing, Visualization. **Flavien Lucas**: Validation, Writing - review and editing, Formal analysis, Visualization, Supervision. **Kenneth Sörensen**: Conceptualization, Validation, Formal analysis, Resources, Writing - review and editing, Project administration, Funding acquisition, Supervision.

References

Berbeglia, G., Cordeau, J.-F., and Laporte, G. (2012). A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. *INFORMS Journal on Computing*, 24(3):343–355.

Cordeau, J.-F. and Laporte, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37(6):579–594.

Hanne, T., Melo, T., and Nickel, S. (2009). Bringing robustness to patient flow management through optimized patient transports in hospitals. *INFORMS Journal on Applied Analytics*, 39(3):241–255.

Heilporn, G., Cordeau, J.-F., and Laporte, G. (2011). An integer l-shaped algorithm for the dial-a-ride problem with stochastic customer delays. *Discrete Applied Mathematics*, 159(9):883–895.

Ho, S. C. and Haugland, D. (2011). Local search heuristics for the probabilistic dial-a-ride problem. *OR Spectrum*, 33(4):961–988.

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

Hyytiä, E., Aalto, S., Penttinen, A., and Sulonen, R. (2010). A stochastic model for a vehicle in a dial-a-ride system. *Operations Research Letters*, 38(5):432–435.

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

Issaoui, B., Zidi, I., Zidi, K., and Ghedira, K. (2013). A distributed approach for the resolution of a stochastic dial a ride problem. In *2013 International Conference on Advanced Logistics and Transport*, pages 205–210.

Jia, Y., Wu, J., and Du, Y. (2016). Traffic speed prediction using deep learning method. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1217–1222.

Kong, X., Xu, Z., Shen, G., Wang, J., Yang, Q., and Zhang, B. (2016). Urban traffic congestion estimation and prediction based on floating car trajectory data. *Future Generation Computer Systems*, 61:97–107.

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

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

Luo, Y. and Schonfeld, P. (2011). Online rejected-reinsertion heuristics for dynamic multivehicle dial-a-ride problem. *Transportation Research Record*, 2218(1):59–67.

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

Marković, N., Nair, R., Schonfeld, P., Miller-Hooks, E., and Mohebbi, M. (2015). Optimizing dial-a-ride services in maryland: Benefits of computerized routing and scheduling. *Transportation Research Part C: Emerging Technologies*, 55:156–165.

Melis, L. and Sörensen, K. (2021). The real-time on-demand bus routing problem: what is the cost of dynamic requests? Working paper, University of Antwerp.

Melis, L. and Sörensen, K. (2022). The static on-demand bus routing problem: large neighborhood search for a dial-a-ride problem with bus station assignment. *International Transactions in Operational Research*, 29(3):1417–1453.

Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11.

Santos, D. O. and Xavier, E. C. (2015). Taxi and ride sharing: A dynamic dial-a-ride problem with money as an incentive. *Expert Systems with Applications*, 42(19):6728–6737.

Schilde, M., Doerner, K., and Hartl, R. (2011). Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers and Operations Research*, 38(12):1719–1730.

Schilde, M., Doerner, K., and Hartl, R. (2014). Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem. *European Journal of Operational Research*, 238(1):18–30.

Tafreshian, A., Abdolmaleki, M., Masoud, N., and Wang, H. (2021). Proactive shuttle dispatching in large-scale dynamic dial-a-ride systems. *Transportation Research Part B: Methodological*, 150:227–259.

Tang, J., Liu, F., Zou, Y., Zhang, W., and Wang, Y. (2017). An improved fuzzy neural network for traffic speed prediction considering periodic characteristic. *IEEE Transactions on Intelligent Transportation Systems*, 18(9):2340–2350.

Wong, K., Han, A., and Yuen, C. (2014). On dynamic demand responsive transport services with degree of dynamism. *Transportmetrica A: Transport Science*, 10(1):55–73.

Xiang, Z., Chu, C., and Chen, H. (2008). The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. *European Journal of Operational Research*, 185(2):534–551.

Yang, J., Jaillet, P., and Mahmassani, H. S. (1999). On-line algorithms for truck fleet assignment and scheduling under real-time information. *Transportation Research Record*, 1667(1):107–113.

Appendix A. Mathematical model

Table A.6: Variables and parameters of the ODBRP

x_{snb}	1 if the n-th station of bus b is bus station s and 0 otherwise
y^u_{pnb}	1 if passenger p is picked up at the n-th station of bus b and 0 otherwise
y^o_{pnb}	1 if passenger p is dropped off at the n-th station of bus b and 0 otherwise
q_{nb}	net number of passengers picked up (or dropped off) at the n-th station of bus b
t^a_{nb}	arrival time of bus b at its n-th station
t^d_{nb}	departure time of bus b at its n-th station
T_p	user ride time of passenger p
B	the fleet of buses
P	the set of transportation requests, $\|P\|$ denotes the number of requests
S	the set of bus stations
Q	capacity of bus
a^u_{ps}	1 if passenger p can be assigned to station s for pick-up
a^o_{ps}	1 if passenger p can be assigned to station s for drop-off
e_p	earliest pick-up time for passenger p
l_p	latest drop-off time for passenger p
$TT_{ss'}$	travel time between station s and station s'

The objective function is to minimize total URT. Constraints (A.2) enforce the fact that a bus can only stop at one station at the same time. Constraints (A.3) make sure the positions used in the bus route are used consecutively and start at the first position. Constraints (A.4) and (A.5) respectively enforce a bus to stop at one station if and only if at least one passenger uses it either to board or alight. Constraints (A.6) and (A.7) respectively impose that a station is designated to a passenger to board/alight only if the station belongs to the passenger, i.e. it is within the predefined walking distance. Constraints (A.8) impose for any two consecutive stations, the arrival time at the later station is (larger than or) equal to the departure time at the previous one plus the travel time, where the travel time depends on the departure time. Constraints (A.9) guarantee the departure time at a passenger's pickup station is greater than or equal to the earliest allowed value. Correspondingly, constraints (A.10) guarantee the arrival time at a passenger's drop-off station is smaller or equal to the latest allowed value. Constraints (A.11) impose that the pickup station precedes the corresponding drop-off one for any passengers. Constraints (A.12) enforce each passenger gets on and gets off the same bus. Constraints (A.13) make sure each passenger is served at most once. Together with constraints (A.18), every request is served once and only once. Constraints (A.14) forbid two consecutive stations be the same. Constraints (A.15) calculate the net capacity at each station, which is equal to the number of passengers getting on minus the one of getting off. Consequently, constraints (A.16) forbid the violation of bus capacity. Constraints (A.17) calculate the URT of each passenger, i.e. the arrival time at the get-off station minus the departure time at the get-on station. Constraints (A.19 - A.22) define the range for each variable.

$$\min \text{URT} = \sum\nolimits_p T_p \tag{A.1}$$

s.t.

$$\sum\nolimits_s x_{snb} \leq 1 \qquad \forall n \in N, b \in B \tag{A.2}$$

$$\sum\nolimits_s (x_{snb} - x_{s(n+1)b}) \geq 0 \qquad \forall n \in N, b \in B \tag{A.3}$$

$$M \sum\nolimits_s x_{snb} - \sum\nolimits_p (y^u_{pnb} + y^o_{pnb}) \geq 0 \qquad \forall n \in N, b \in B \tag{A.4}$$

$$\sum\nolimits_s x_{snb} - \sum\nolimits_p (y^u_{pnb} + y^o_{pnb}) \leq 0 \qquad \forall n \in N, b \in B \tag{A.5}$$

$$x_{snb} + y^u_{pnb} - a^u_{ps} \leq 1 \qquad \forall s \in S, n \in N, p \in P, b \in B \tag{A.6}$$

$$x_{snb} + y^o_{pnb} - a^o_{ps} \leq 1 \qquad \forall s \in S, n \in N, p \in P, b \in B \tag{A.7}$$

$$t^a_{(n+1)b} - t^d_{nb} - TT_{ss'} + (x_{snb} + x_{s'(n+1)b} - 2)(-M) \geq 0 \qquad \forall s, s' \in S \mid s \neq s', n \in N, b \in B \tag{A.8}$$

$$t^d_{nb} - e_p + (y^u_{pnb} - 1)(-M) \geq 0 \qquad \forall p \in P, n \in N, b \in B \tag{A.9}$$

$$t^a_{nb} - l_p + (y^o_{pnb} - 1)M \leq 0 \qquad \forall p \in P, n \in N, b \in B \tag{A.10}$$

$$\sum\nolimits_n (n y^u_{pnb} - n y^o_{pnb}) \leq 0 \qquad \forall p \in P, b \in B \tag{A.11}$$

$$\sum\nolimits_n (y^u_{pnb} - y^o_{pnb}) = 0 \qquad \forall p \in P, b \in B \tag{A.12}$$

$$\sum\nolimits_b \sum\nolimits_n y^u_{pnb} \leq 1 \qquad \forall p \in P \tag{A.13}$$

$$x_{snb} + x_{s(n+1)b} \leq 1 \qquad \forall s, n, b \tag{A.14}$$

$$\sum\nolimits_p (y^u_{pnb} - y^o_{pnb}) - q_{nb} = 0 \qquad \forall n \in N, b \in B \tag{A.15}$$

$$\sum\nolimits_{n' \leq n} q_{n'b} \leq Q \qquad \forall n, n' \in N \mid n \geq n', b \in B \tag{A.16}$$

$$T_p + (2 - y^o_{pn'b} - y^u_{pnb})M - t^a_{n'b} + t^d_{nb} \geq 0 \qquad \forall n, n' \in N \mid n' > n, p \in P, b \in B \tag{A.17}$$

$$\sum\nolimits_p \sum\nolimits_b \sum\nolimits_n y^u_{pnb} = |P| \tag{A.18}$$

$$x_{snb} \in \{0, 1\} \qquad \forall s \in S, n \in N, b \in B \tag{A.19}$$

$$y^u_{pnb} \in \{0, 1\} \qquad \forall p \in P, n \in N, b \in B \tag{A.20}$$

$$y^o_{pnb} \in \{0, 1\} \qquad \forall p \in P, n \in N, b \in B \tag{A.21}$$

$$q_{nb} \in \mathbb{Z} \qquad \forall n \in N, b \in B \tag{A.22}$$