Front. Eng. Manag. 2025, 12(4): 983–1004
https://doi.org/10.1007/s42524-025-4174-3

RESEARCH ARTICLE

Mengsi ZHOU, Yadong WANG

Optimal routing and request selection for multiple service routes in a demand-adaptive transit system

Abstract The demand-adaptive system (DAS) has been recognized as a promising transit mode for demand with high fluctuations. In this paper, we optimize the routes and request selection for a DAS with multiple service routes. Currently, most studies on DAS focus on optimizing single-route systems, where each area is exclusively served by one route and heuristic pre-assignations of requests are made. In contrast, our study addresses a more generalized routing and request selection problem for a DAS with multiple service routes. This problem jointly assigns requests to the service routes and determines the resulting routes while considering the pickup and delivery locations and the reserved boarding time for each request. A mixed-integer linear programming (MILP) model is developed to minimize the sum of bus travel time cost, passenger in-vehicle and waiting time costs, and request rejection penalties. A tailored adaptive large neighborhood search algorithm (ALNS) solves this optimization model efficiently. The numerical experiments show that, under the same optimality conditions, the proposed algorithm outperforms the exact algorithm implemented by GORUBI in terms of solution quality and computation time. The ALNS algorithm also reports cost reductions of up to 50% in comparison with prevailing benchmark metaheuristics. Moreover, the multi-route DAS in this paper has a lower rejection rate and objective value than the single-route systems examined in previous studies.

Keywords demand-adaptive systems, multi-route design, request selection, adaptive large neighborhood search heuristic

Received Jun. 8, 2024; revised Sep. 16, 2024; accepted Oct. 16, 2024

Mengsi ZHOU, Yadong WANG (✉)
School of Economics and Management, Nanjing University of Science and Technology, Nanjing 210094, China
E-mail: wang_yd@foxmail.com

This research was supported by the National Natural Science Foundation of China (Grant Nos. 72371130 and 72001108).

1 Introduction

In areas with high transportation demand, traditional bus systems with fixed routes and schedules usually operate efficiently and provide a high level of service due to their high degree of resource sharing (Crainic et al., 2010; Errico et al., 2013). However, in environments with low demand or large variability in demand, traditional bus systems become inefficient. In such situations, the buses either run empty or are extremely crowded, both are equally unpleasant. Empty buses indicate ineffective operations by the transit agency while overcrowded buses render passengers uncomfortable. Demand responsive transit (DRT) systems have been implemented to address this situation by offering high degree of flexibility in transport means to customers (Palmer et al., 2004). A well-known example of DRT is the Dial-a-Ride (DAR) system (Ho et al., 2018). DRT provides a highly customized service by designing routes and schedules based on request information such as pickup and delivery locations, and reserved boarding times. However, this high degree of flexibility comes with inherent drawbacks. For instance, the service relies entirely on incoming requests, which can vary significantly across different time periods, making it difficult for both the transit agency and passengers to forecast itineraries, schedules, and stop locations. Therefore, it is difficult to integrate DRT with traditional transit systems.

To address these issues, the demand-adaptive system (DAS) has been proposed, which integrates the advantages of fixed-route transit (FRT) and DRT (Malucelli et al., 1999). In low demand areas, the DAS is more flexible than FRT (Becker et al., 2013) and more cost-efficient than DRT (Fittante and Lubin, 2016). As illustrated in Fig. 1, DAS provides service for a set of compulsory stops with predefined time windows, also known as the master schedule. The time window brings some degree of both flexibility and regularity for DAS. Due to the compulsory stops, passengers can use DAS without reservation, just like in traditional bus systems. DAS is also

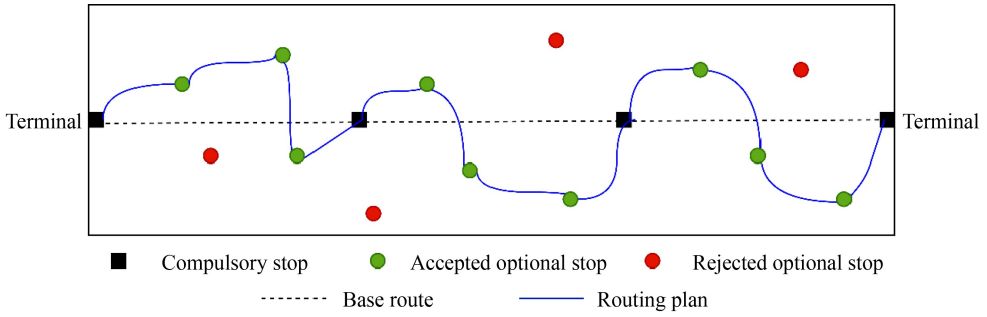

Fig. 1 A DAS route deviating base route to serve optional stops.

allowed to deviate from the base route between two consecutive compulsory stops to serve personalized requests, which embodies its flexibility. In addition, the time windows allow passengers to plan their journeys and transfer to other modes of transit at the compulsory stops. Given these characteristics, DAS is suited to low demand areas (e.g., city suburbs, rural areas) or periods (e.g., non-rush hours, evenings, weekends) where public transit services are required (Crainic et al., 2005). Table 1 compares DAS with FRT and DRT across five aspects, indicating that DAS could strike an ideal balance between affordability and flexibility that may be factored into transit system use. For these advantages, DAS is in operation in numerous cities worldwide, such as the route deviation service in Mason County, Washington, and the City of St. Joseph, Missouri (Koffman, 2004; Potts et al., 2010).

With the wide adoption of the DAS, its design problem has become increasingly important. To the best of our knowledge, most studies focus on optimizing single-route systems, with requests preassigned to each service route as the input for the problem. In contrast, this paper addresses a more general case in which multiple routes exist in the DAS. We explore how to optimally assign requests to each service route and determine the resultant service routes based on the pickup and delivery locations, as well as the reserved boarding time of each request. Model based on mixed-integer linear programming is designed to tackle this problem. However, optimizing this problem is challenging due to its large scale and computational complexity. Therefore, we propose an adaptive large neighborhood search heuristic to solve it in a reasonable time frame. This study aims to equip transit agencies in planning their transit services more efficiently in low demand areas.

2 Literature review

The planning of DAS has attracted significant attention within the academic literature. Errico et al. (2013) classifies decision-making into three distinct levels: (1) the strategic level, which involves selecting operating policies, designing service zones, and optimizing fleet size; (2) the tactical level, which focuses on drafting the master schedule, establishing base routes, and determining service headways; (3) the operational level, which includes routing, scheduling, and request assignments.

At the strategic level, key tasks include the selection of operating policies, design of service zones, and optimization of fleet size, among others. Previous studies have explored the transition point of demand density between DAS and FRT policies (Quadrifoglio and Li, 2009; Qiu et al., 2014a), as well as comparisons between different DAS policies (Zheng et al., 2018a). Many researchers have assumed that the overall service area can be segmented into multiple smaller zones, each independently serviced by a DAS route. Building upon this premise, they have designed service parameters pertinent to these zones. For instance, Li and Quadrifoglio (2009) determined the optimal number of zones necessary for feeder services, while Nourbakhsh and Ouyang (2012) proposed hub-and-spoke and grid network models into DAS, seeking optimal layouts for networks, service zones for each bus, and bus headways. Furthermore, Kim et al. (2019) optimized both headway and service zone size in a many-to-one demand pattern. Sipetas and Gonzales (2021) investigated optimal spacing for checkpoints and determined appropriate service zone widths.

The tactical level primarily addresses the design of base routes, master schedules, and service headways. For

Table 1 The comparison among DAS, FRT and DRT

Transit system type	Stop	Route	Schedule	Flexibility	Cost
FRT	Compulsory stops	Fixed	Fixed	Low	Low
DRT	Optional stops	Flexible	Flexible	High	High
DAS	Compulsory stops and optional stops	Fixed part: base route Flexible part: route deviating the base route to serve optional stops	Fixed part: time windows at compulsory stops Flexible part: departure time at compulsory stops within the time windows	Medium	Medium

base route design, Errico et al. (2021) focused on selecting compulsory stops, assigning optional stops per segment, and optimizing stop sequences for a single-route DAS in the stationary-demand case. In contrast, Yang et al. (2016) explored route planning for multiple routes, leveraging existing networks and heterogeneous demand patterns. Recently, Li et al. (2023a) and Li and Tang (2023) examined the DAS route design problem with meeting points. In another study, Li et al. (2023b) optimized flexible service lengths across various segments and the service range of meeting stops. Regarding master schedule design, some studies optimized the slack time allocated to each flexible-route segment (Fu, 2002; Crainic et al., 2010), whereas others analyzed the relationship between service cycle time and the length and width of the service area (Zhao and Dessouky, 2008). In relation to service headway design, Chen and Nie (2017) developed a hybrid transit system that integrates multiple DAS routes with multiple FRT routes, optimizing headways for both types of service.

The operational level primarily concentrates on vehicle routing, scheduling, request assignments and related activities. The main objectives are generally to minimize operational costs and enhance service quality. Crainic et al. (2005) explored the request selection and vehicle routing problem within the context of a single route. In contrast, Quadrifoglio et al. (2007) developed a strategy to route multiple trips along a single route using an insertion heuristic in real-time scenarios. Quadrifoglio et al. (2008), on the other hand, examined a static situation in which all requests were known in advance. Pei et al. (2019) investigated the selection of optional stops that would only be visited if passengers expressed a strong willingness to pay. Further studies by Galarza Montenegro et al. (2021, 2022) focused on optimizing the timetable, route, and request assignment within a multi-bus, single-trip feeder service. Their subsequent research (Galarza Montenegro et al., 2023) expanded into more complex scenarios involving multiple buses and trips, incorporating headway decisions into the optimization model. Jin et al. (2023) considered departure conditions, differentiated fare structures, and passenger service evaluations at compulsory stops for DAS. Additionally, some studies aimed to reduce rejection rates by introducing meeting stops or utilizing accepted optional stops (Qiu et al., 2014b; Zheng et al., 2019). Other research explored the integration of Modular Autonomous Vehicles (MAVs) with DAS (Liu et al., 2021; Tang et al., 2023). However, the aforementioned operational-level studies primarily focus on single-route optimization. In the context of multiple routes, Pang et al. (2017) and Lu et al. (2020) investigated the coordinated scheduling of multi-route DAS, concentrating on request-to-route assignment and transfer issues within predetermined vehicle routes. Unlike our research, these studies did not involve the design of individual DAS routes. Specifically, they positioned only one or two optional stops between two consecutive compulsory stops, with vehicles prohibited from turning back, thereby predetermining the visiting sequence.

According to the review above, most previous studies have focused on optimizing DAS in a single-route context (Crainic et al., 2005; Quadrifoglio et al., 2007; Quadrifoglio et al., 2008; Pei et al., 2019; Galarza Montenegro et al., 2021, 2022, 2023), neglecting the design of multi-route DAS. Although some research has addressed multi-route scenarios, these studies concentrate on either designing service area parameters at the strategic and tactical levels (Nourbakhsh and Ouyang, 2012; Yang et al., 2016; Chen and Nie, 2017; Kim et al., 2019) or request-to-route assignment and transfer issues within predetermined vehicle routes (Pang et al., 2017; Lu et al., 2020). They all fail to account for the multi-route design and request selection from a holistic optimization perspective. There are two significant limitations associated with the single-route approach. First, in practice, a service area typically includes multiple routes, rendering the single-route approach inadequate for accurately representing real-world conditions. Secondly, most single-route studies pre-assign requests to specific routes as input for the optimization problem, which restricts the flexibility to allocate requests across multiple routes.

In light of these shortcomings, this paper is the first to address the multi-route case at the operational level and thus proposes the **multi-route DAS design and request selection problem (MDRP)**. This paper presents three key contributions:

(1) An innovative problem is proposed. First, it extends the focus of previous single-route studies by designing multiple routes simultaneously, increasing its generalizability and real-world applicability. Second, it optimally allocates requests to service routes while accounting for the pickup and delivery locations as well as the reserved boarding times for each request.

(2) A new optimization model is developed to identify the optimal routing and request selection for multiple service routes within a DAS.

(3) An efficient heuristic algorithm has been specifically designed to solve the optimization model within a reasonable timeframe. Numerical experiments validate the effectiveness of the heuristic algorithm and reveal that the multi-route case demonstrates greater system efficiency compared to the single-route case.

The remainder of this paper is organized as follows: Section 3 presents the MDRP, which is further formulated as a mixed-integer linear programming (MILP) model. Section 4 proposes a customized adaptive large neighborhood search (ALNS) heuristic algorithm to solve the model. In Section 5, numerical experiments are conducted to assess the model's applicability and the efficiency of the solution algorithm. Finally, conclusions are provided in Section 6.

3 Problem statement and model formulation

3.1 Demand-adaptive systems

DAS typically function in areas with low transportation demand or during off-peak periods. Generally, a DAS comprises multiple routes, each served by several buses. The system includes optional stops, which are visited when a service request is received and considered profitable. Passengers who request service at these optional stops are classified as active users, whereas those traveling exclusively between compulsory stops are termed passive users.

To accommodate passengers at optional stops, the vehicle must deviate from the most direct route between two consecutive compulsory stops. This route segment between consecutive compulsory stops is referred to as a segment. Prior studies (e.g., Crainic et al., 2005) have assigned requests to specific segments on designated routes before planning; however, this study relaxes this constraint, allowing for broader allocation of requests across the entire DAS coverage area.

At each compulsory stop, designated time windows define the earliest and latest departure times (EDT and LDT). A bus may arrive at any time before the LDT. Nonetheless, if it arrives prior to the EDT, it must wait until the EDT to proceed with the journey. To ensure consistent service quality, all time windows are of uniform width. Passengers boarding at compulsory stops are required to arrive before the EDT and may experience a waiting period if the bus arrives after that time.

3.2 Problem settings

This paper targets the operational-level MDRP, assuming given tactical and strategic decisions such as the service area of the DAS, the compulsory stop locations, their sequence, and the associated time windows. Additionally, we operate under the assumption that all requests are known prior to the planning phase.

The focus of this study is on optimizing routing and request selection for multiple routes, given a specific set of requests within a defined time frame and low-demand area. DAS caters to four categories of requests: O2O (pick-up and drop-off at both optional stops), C2O (pick-up at a compulsory stop and drop-off at an optional stop), O2C (pick-up at an optional stop and drop-off at a compulsory stop), and C2C (pick-up and drop-off at both compulsory stops). C2C passengers utilize DAS as a FRT and their itineraries do not impact bus routing. Consequently, for simplicity, C2C passengers are excluded from this model. Furthermore, consistent with the methodologies of Crainic et al. (2005) and Quadrifoglio et al. (2007), this study assumes that capacity constraints are negligible, given that DAS operates in low-demand regions where capacity is generally sufficient.

Under these assumptions, the transit agency must make three crucial decisions simultaneously to minimize total costs, which include bus travel time cost, passenger in-vehicle and waiting time costs, and penalties for rejected requests.

(1) (Request selection) Which requests should be selected from the complete set of issued requests? Not all requests can be fulfilled due to the time windows at compulsory stops. The model aims to identify a set of requests that minimizes the total cost.

(2) (Request assignment and multi-route design) Which route and segment should a request be assigned to? Specifically, between which two consecutive compulsory stops should a pick-up or drop-off stop be positioned? What sequence should be established for visiting compulsory and optional stops? An effective assignment and routing strategy can reduce detours and accommodate more passengers.

(3) (Timetable design) What timetables will simultaneously satisfy both the time windows at compulsory stops and the reserved boarding times of the selected requests? The timetables directly impact the passenger in-vehicle and waiting time costs.

The MDRP aims to optimally select the request set and design multiple routes within the DAS.

3.3 Model formulation

This section presents the MILP formulation of the MDRP. We assume that the DAS operates in a region with $|K|$ routes (denoted by $k = 1, 2, ..., |K|$). For convenience of modeling, the number of compulsory stops on each route is set to be the same, denoted by P. However, the model can easily accommodate varying numbers of compulsory stops across different routes by incorporating dummy stops. We consider a single trip for each route, assuming that all trips maintain the same direction (either entirely outbound or inbound). Each request is characterized by a pick-up stop, a drop-off stop, and a reserved boarding time. If a request is accepted by the system, the vehicle is required to pick up the passenger after the reserved boarding time at the pick-up stop and subsequently transport them to the designated drop-off stop. Optional stops are assumed to be situated within the service area of the DAS. The formulation specifically addresses three types of requests: an O2O request is modeled with both a pick-up and drop-off stop; a C2O request is represented by its drop-off stop; and an O2C request is represented by its pick-up stop. As a C2O or O2C request involves a compulsory stop on a DAS route, we refer to the "corresponding compulsory stop" and "corresponding route" for this request. Moreover, we assume that a C2O or O2C request can exclusively be assigned to its "corresponding route" without accounting

for transfers. For modeling simplicity, we omit dwell time (i.e., service time) at each optional or compulsory stop, as it does not affect our conclusions.

3.3.1 Notation

All notions used in the optimization model are summarized in Table 2.

3.3.2 Mixed-integer linear programming model

Given the above definitions, the MDRP is formulated as a mixed-integer programming model.

$$\begin{aligned}\min\ & \lambda_1 \sum_{k\in K}\sum_{i\in V}\sum_{j\in V} T_{i,j}x_{i,j,k} + \lambda_2 \sum_{r\in R} N_r(t_r^{\text{drop}} - t_r^{\text{pick}}) \\ & +\lambda_3 \sum_{r\in R_1\cup R_3} N_r \max\{0,\ t_r^{\text{pick}} - Q_r\} \\ & +\lambda_4[\sum_{r\in R_1\cup R_3} N_r(1-y_{s(r)}) + \sum_{r\in R_2} N_r(1-y_{d(r)})]\end{aligned} \tag{1}$$

subject to

$$y_{i,k} = 1, \quad \forall i \in F_k, \quad \forall k \in K \tag{2}$$

$$\sum_{k\in K} y_{i,k} \leqslant 1, \quad \forall i \in V \tag{3}$$

$$y_{s(r),k} = y_{d(r),k}, \quad \forall r \in R_1, \quad \forall k \in K \tag{4}$$

$$y_{d(r),k} = 0, \quad \forall r \in R_2, \quad \forall k \in K, \quad k \neq l(r) \tag{5}$$

$$y_{s(r),k} = 0, \quad \forall r \in R_3, \quad \forall k \in \mathrm{K}, \quad k \neq l(r) \tag{6}$$

$$y_i = \sum_{k\in K} y_{i,k}, \quad \forall i \in V \tag{7}$$

$$y_{i,k} = \sum_{j\in V, j\neq i} x_{j,i,k} = \sum_{j\in V, j\neq i} x_{i,j,k}, \quad \forall i \in V/F^0, \quad \forall k \in K \tag{8}$$

$$\sum_{j\in V, j\neq i} x_{i,j,k} = 1, \quad i = f(k,1), \quad \forall k \in K \tag{9}$$

$$\sum_{i\in V, i\neq j} x_{i,j,k} = 1, \quad j = f(k,P) \quad \forall k \in K \tag{10}$$

$$t_{j,k}^{\text{arr}} \geqslant t_{i,k}^{\text{dep}} + T_{i,j} - M(1-x_{i,j,k}), \quad \forall i,j \in V, \quad \forall k \in K \tag{11}$$

$$t_{j,k}^{\text{arr}} \leqslant t_{i,k}^{\text{dep}} + T_{i,j} + M(1-x_{i,j,k}), \quad \forall i,j \in V, \quad \forall k \in K \tag{12}$$

$$t_i^{\text{arr}} = \sum_{k\in K} t_{i,k}^{\text{arr}}, \quad \forall i \in V \tag{13}$$

$$t_i^{\text{dep}} = \sum_{k\in K} t_{i,k}^{\text{dep}}, \quad \forall i \in V \tag{14}$$

$$t_{i,k}^{\text{dep}} \geqslant t_{i,k}^{\text{arr}}, \quad \forall i \in F_k, \quad \forall k \in K \tag{15}$$

$$t_{i,k}^{\text{dep}} \geqslant A_{i,k}, \quad \forall i \in F_k, \quad \forall k \in K \tag{16}$$

$$t_{i,k}^{\text{dep}} \leqslant B_{i,k}, \quad \forall i \in F_k, \quad \forall k \in K \tag{17}$$

$$t_{i,k}^{\text{dep}} \leqslant t_{i,k}^{\text{arr}} + M(1-g_{i,k}), \quad \forall i \in F_k, \quad \forall k \in K \tag{18}$$

$$t_{i,k}^{\text{dep}} \leqslant A_{i,k} + Mg_{i,k}, \quad \forall i \in F_k, \quad \forall k \in K \tag{19}$$

$$t_{s(r)}^{\text{dep}} \geqslant t_{s(r)}^{\text{arr}}, \quad \forall r \in R_1 \cup R_3 \tag{20}$$

$$t_{s(r)}^{\text{dep}} \geqslant Q_r y_{s(r)}, \quad \forall r \in R_1 \cup R_3 \tag{21}$$

$$t_{s(r)}^{\text{dep}} \leqslant t_{s(r)}^{\text{arr}} + M(1-h_{s(r)}), \quad \forall r \in R_1 \cup R_3 \tag{22}$$

$$t_{s(r)}^{\text{dep}} \leqslant Q_r y_{s(r)} + Mh_{s(r)}, \quad \forall r \in R_1 \cup R_3 \tag{23}$$

$$t_{d(r)}^{\text{arr}} = t_{d(r)}^{\text{dep}}, \quad \forall r \in R_1 \cup R_2 \tag{24}$$

$$t_r^{\text{pick}} = t_{s(r)}^{\text{dep}}, \quad \forall r \in R_1 \cup R_3 \tag{25}$$

$$t_r^{\text{drop}} = t_{d(r)}^{\text{arr}}, \quad \forall r \in R_1 \cup R_2 \tag{26}$$

$$t_r^{\text{pick}} \leqslant My_{d(r)}, \quad \forall r \in R_2 \tag{27}$$

$$t_r^{\text{pick}} \geqslant t_{c(r)}^{\text{dep}} - M(1-y_{d(r)}), \quad \forall r \in R_2 \tag{28}$$

$$t_r^{\text{pick}} \leqslant t_{c(r)}^{\text{dep}} + M(1-y_{d(r)}), \quad \forall r \in R_2 \tag{29}$$

$$t_r^{\text{drop}} \leqslant My_{s(r)}, \quad \forall r \in R_3 \tag{30}$$

$$t_r^{\text{drop}} \geqslant t_{c(r)}^{\text{arr}} - M(1-y_{s(r)}), \quad \forall r \in R_3 \tag{31}$$

$$t_r^{\text{drop}} \leqslant t_{c(r)}^{\text{arr}} + M(1-y_{s(r)}), \quad \forall r \in R_3 \tag{32}$$

$$t_r^{\text{drop}} \geqslant t_r^{\text{pick}}, \quad \forall r \in R \tag{33}$$

Table 2 Table of notations

Name	Description	Unit
Index		
r	Riding request index, $r \in R$	
k	DAS route index, $k \in K$	
i, j	Compulsory stop or optional stop index, $i, j \in V$	
Function		
$f(k,p)$	pth compulsory stop on route k, $f(k,p) \in F_k$, $k \in K$	
$s(r)$	Pick-up stop index of O2O or O2C riding request r, $s(r) \in N$, $r \in R_1 \cup R_3$	
$d(r)$	Drop-off stop index of O2O or C2O riding request r, $d(r) \in N$, $r \in R_1 \cup R_2$	
$c(r)$	The "corresponding compulsory stop" which is the pick-up stop of C2O request r or the drop-off stop of O2C request r, $c(r) \in F$, $r \in R_2 \cup R_3$	
$l(r)$	The "corresponding route" of C2O or O2C riding request r, $l(r) \in K$, $r \in R_2 \cup R_3$	
Set		
R_1	Set of O2O riding requests	
R_2	Set of C2O riding requests	
R_3	Set of O2C riding requests	
R	Set of three types of riding requests, $R = R_1 \cup R_2 \cup R_3$	
K	Set of DAS routes in a region, $K = \{1, 2, ..., \lvert K \rvert\}$	
F_k	A sequence set of all compulsory stops of route k, $k \in K$, $F_k = \{f(k,1), f(k,2).., f(k,p), ..., f(k,P)\}$; P = number of compulsory stops in each DAS route	
F	Set of compulsory stops of all routes, $F = \underset{k \in K}{\cup} F_k$	
N	Set of all optional stops issued as a pick-up stop or a drop-off stop in requests	
V	Set of all stops including compulsory stops and issued optional stops, $V = F \cup N$	
F^0	Set of starting and ending compulsory stops of all routes	
Parameter		
M	A large positive number	
$T_{i,j}$	Rectilinear travel time from stop i to stop j, $i, j \in V$	min
$A_{i,k}$	EDT of compulsory stop i, $i \in F_k$, $k \in K$	min
$B_{i,k}$	LDT of compulsory stop i, $i \in F_k$, $k \in K$	min
Q_r	Reserved boarding time of O2O or O2C riding request r, $r \in R_1 \cup R_3$	min
N_r	Number of passengers for request r, $r \in R$	
λ_1	Cost value of bus travel time	\$/min
λ_2	Cost value of passenger in-vehicle time	\$/min
λ_3	Cost value of passenger waiting time	\$/min
λ_4	Penalty for rejecting a passenger	\$/passenger
Decision Variable		
$y_{i,k}$	1, stop i is assigned to route k; 0, otherwise	
y_i	1, stop i is assigned to a route; 0, otherwise	
$x_{i,j,k}$	1, stop j is visited after stop i on route k; 0, otherwise	
$t_{i,k}^{\text{arr}}$	Arrival time at stop i on route k	min
$t_{i,k}^{\text{dep}}$	Departure time at stop i on route k	min
t_i^{arr}	Arrival time at stop i	min
t_i^{dep}	Departure time at stop i	min
t_r^{pick}	Pick-up time of request r	min
t_r^{drop}	Drop-off time of request r	min

(*Continued*)

Name	Description	Unit
Auxiliary Variable		
$g_{i,k}$	An auxiliary binary variable for the decision of $t_{i,k}^{\text{dep}}$	
h_i	An auxiliary binary variable for the decision of $t_{s(r)}^{\text{dep}}$, $r \in R_1 \cup R_3$	
μ_r	An auxiliary real variable in the linearization of the objective function	

$$t_{i,k}^{\text{arr}} \leqslant M y_{i,k}, \quad \forall i \in V, \quad \forall k \in K, \tag{34}$$

$$t_{i,k}^{\text{dep}} \leqslant M y_{i,k}, \quad \forall i \in V, \quad \forall k \in K, \tag{35}$$

$$y_{i,k},\ x_{i,j,k},\ g_{i,k},\ h_i \in \{0,1\}, \tag{36}$$

$$t_{i,k}^{\text{arr}},\ t_{i,k}^{\text{dep}} \geqslant 0. \tag{37}$$

The objective function (1) aims to minimize the total cost, which is comprised of four key components: the cost of bus travel time, the cost of passenger in-vehicle time, the cost of passenger waiting time, and penalties for rejected requests.

Request assignment and routing constraints. We use $y_{i,k}$ to assign all stops, including compulsory stops, to routes. Constraint (2) ensures that each compulsory stop is assigned to its designated route. Constraint (3) limits each stop to be assigned to at most one route. Constraint (4) requires that pick-up and drop-off stops for O2O requests be assigned to the same route. Constraint (5) enforces that the drop-off stop in a C2O request must be assigned to its "corresponding route". Similarly, Constraint (6) applies the same rule for O2C requests. For simplicity, this study introduces y_i, which indicates whether a stop is assigned to a route. Constraint (7) defines the relationship between y_i and $y_{i,k}$. Constraint (8) ensures flow equilibrium: for all stops, except for the starting and ending stops of routes, if stop i is assigned to a route, its incoming and outgoing degree must each equal 1; otherwise, they must be 0. Constraints (9)–(10) define the degrees for the starting and ending stops of each route. The outgoing degree of a starting stop is 1 with an incoming degree of 0, while the incoming degree of an ending stop is 1 with an outgoing degree of 0.

Time constraints. Constraints (11)–(12) define the relationship between the arrival and departure times at two consecutive stops. Additionally, constrain (11) functions as a subtour elimination constraint in the vehicle routing problem (VRP). For simplicity, this study introduces t_i^{arr} and t_i^{dep} to represent the arrival and departure times at stop i, respectively. Constraint (13) gives the relationship between t_i^{arr} and $t_{i,k}^{\text{arr}}$, similarly addressed in constraint (14). Constraints (15)–(19) determine the departure time at each compulsory stop. Constraint (15) restricts that departure occurs after the arrival time. Constraints (16)–(17) restrict that the departure time falls within the designated time window. Constraints (18)–(19) restrict that the departure time equals either the arrival time or the EDT. Constraints (20)–(23) calculate the departure time at the pick-up stops of O2O and O2C requests. Constraint (20) restricts that the departure time is later than the arrival time. Constraint (21) restricts that if a request is selected, the departure time is later than the reserved boarding time. Constraints (22)–(23) enforce that the departure time equals either the arrival time or the reserved boarding time. Constraint (24) ensures that the departure time equals the arrival time at the drop-off stop of an O2O or C2O request. Constraints (25)–(32) calculate the pick-up and drop-off times for all requests. Constraint (25) restricts that the pick-up time of an O2O or O2C request equals the departure time at its pick-up stop. Constraint (26) restricts that the drop-off time of an O2O or C2O request equals the arrival time at its drop-off stop. Constraints (27)–(29) restrict that if a C2O request is selected, its pick-up time matches the departure time at its "corresponding compulsory stop". Constraints (30)–(32) restrict that if an O2C request is selected, its drop-off time matches the arrival time at its "corresponding compulsory stop". Constraint (33) ensures that the drop-off time is later than the pick-up time for each request, confirming that the pick-up stop is visited before the drop-off stop. Finally, Constraints (34)–(35) ensure that both the arrival and departure times at a stop on route k are zero if the stop is not assigned to route k.

Decision variable ranges. Constraints (36)–(37) define the ranges of binary and nonnegative variables.

We reformulate the objective function by introducing an auxiliary variable w_r to linearize the nonlinear term $\max\{0, t_r^{\text{pick}} - Q_r\}$ in Eq. (1), as shown in the following equations.

$$\begin{aligned}\min\ & \lambda_1 \sum_{k\in K}\sum_{i\in V}\sum_{j\in V} T_{i,j}x_{i,j,k} + \lambda_2 \sum_{r\in R} N_r(t_r^{\text{drop}} - t_r^{\text{pick}}) \\ & + \lambda_3 \sum_{r\in R_1\cup R_3} N_r\mu_r + \lambda_4[\sum_{r\in R_1\cup R_3} N_r(1-y_{s(r)}) \\ & + \sum_{r\in R_2} N_r(1-y_{d(r)})],\end{aligned} \tag{38}$$

$$\mu_r \geqslant t_r^{\text{pick}} - Q_r, \quad \forall r \in R_1 \cup R_3, \tag{39}$$

$$\mu_r \geqslant 0, \quad \forall r \in R. \tag{40}$$

4 Solution method

VRP is classified as NP-hard. The model presented here can be regarded as a specific variant of the VRP, particularly when certain constraints are relaxed, thereby retaining its NP-hard designation. Exact algorithms, such as branch-and-bound, can efficiently determine optimal solutions for small instances. However, as the size of the instances increases, the time required for solutions grows exponentially. Our experiments, as detailed in Section 5, demonstrate that when the number of requests rises significantly, the solution time using the branch-and-cut algorithm in the GUROBI solver can extend to several days.

Compounding the challenge, the DAS necessitates rapid route planning and prompt responses to passengers, as reservation replies are time-sensitive. Consequently, a heuristic algorithm is more appropriate than an exact algorithm in this context. Among the numerous heuristic algorithms developed for various combinatorial optimization challenges, ALNS has gained prominence and demonstrated effectiveness (Ropke and Pisinger, 2006; Pisinger and Ropke, 2007; Sacramento et al., 2019) due to its capacity to explore more promising areas of the search space. Therefore, we employ ALNS to efficiently solve the model. In addition, simulated annealing is incorporated to facilitate the acceptance of current solutions, enabling the algorithm to escape local optima.

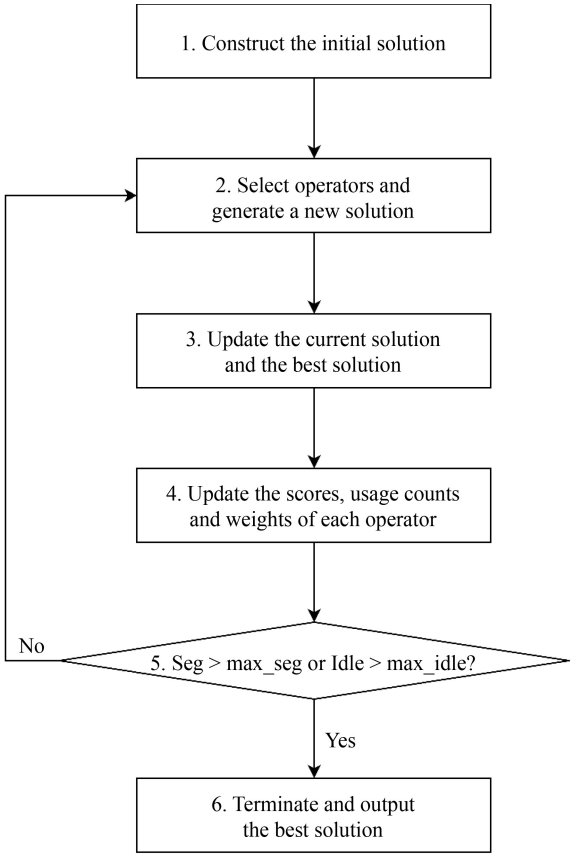

Fig. 2 Flowchart of the ALNS.

The ALNS process consists of six steps, as illustrated in Fig. 2.

Step 1: Initialize input parameters. Construct the initial solution using the customized method described in Section 4.1.

Step 2: Select a pair of destroy and repair operators based on their respective weights, and apply them to generate a new solution. Details regarding the operators can be found in Section 4.2.

Step 3: Update the current and best solutions according to the acceptance criteria outlined in Section 4.3.

Step 4: Update the scores and usage counts of the operators selected in the current iteration. If the current iteration concludes a time segment, adjust the weights of all operators. Refer to Section 4.4 for the weight adjustment rules.

Step 5: If the ALNS reaches the maximum number of iterations or if the count of consecutive iterations that fail to improve the best-known solution exceeds the predetermined limit, proceed to Step 6; otherwise, return to Step 2.

Step 6: Terminate the ALNS. Output the best solution.

The overall framework of the proposed ALNS algorithm is summarized in Algorithm 1. The customized components of this algorithm are reflected in two key aspects: (1) the construction of the initial solution via a tailored heuristic algorithm and (2) the specific destroy and repair rules for the operators. Further details regarding these aspects can be found in Sections 4.1 and 4.2.

4.1 The construction of the initial solution

This section corresponds to Step 1 of Fig. 2. The MDRP involves route design, with each route consisting of a sequence of compulsory stops. A solution derived from the ALNS contains $|K|$ routing sets that represent these sequences. Furthermore, the MDRP entails request selection, where some requests may be rejected, resulting in a rejected set that contains all requests that were not accepted. Consequently, a comprehensive solution comprises both $|K|$ routing sets and the rejected set. To address the MDRP, it is necessary to determine whether to accept or reject a request, and if accepted, to decide where to insert it. The MDRP includes three types of requests, which leads to three different representations within the solution. Since C2O and O2C requests include compulsory stops that are already part of the solution, their representations differ from those of O2O requests. An O2C (C2O) request is represented by its pick-up (drop-off) stop, while an O2O request is represented by both its pick-up and drop-off stops. In the rejected set, each request is represented by a single stop for simplicity; specifically, O2O and O2C requests are represented by their pick-up stops, while C2O requests are represented by their drop-off stops.

As illustrated in Fig. 3, the construction of the initial

Algorithm 1: ALNS algorithm:

Input: data file, $\max\_seg$, $\max\_iter$, $\max\_idle$, T_{start}, τ, σ_1, σ_2, σ_3, σ_4, w^-, w^+, π^-, π^+, θ^-, θ^+, ρ

generate initial solution $x_{initial}$ based on the heuristic algorithm proposed in Section 4.1
$x_{best} = x_{initial}$
$x_{cur} = x_{initial}$
$seg \leftarrow 1$
$idle = 0$
$T \leftarrow T_{start}$
do until $seg < \max\_seg$
$iter \leftarrow 1$
while $iter < \max\_iter$ **do**
select destroy operator i and repair operator j based on roulette-wheel mechanism with destroy probability w^- and repair probability w^+
$x_{new} = destroy - and - repair(x_{cur})$
if x_{new} is better than x_{best} then $idle \leftarrow 0$
else $idle \leftarrow idle + 1$
end if
if $idle \leftarrow \max\_idle$ **then** $iter \leftarrow \max\_iter$, $seg \leftarrow \max\_seg$
end if
if x_{new} is better than x_{best} **then** $x_{best} \leftarrow x_{new}$, $x_{cur} \leftarrow x_{new}$, $\sigma^* = \sigma_1$
else if x_{new} is better than x_{cur} **then** $x_{cur} \leftarrow x_{new}$, $\sigma^* = \sigma_2$
else if x_{new} is accepted by Metropolis criterion **then** $x_{cur} \leftarrow x_{new}$, $\sigma^* = \sigma_3$
else if x_{new} is rejected **then** $\sigma^* = \sigma_4$
end if
$\pi_i^- \leftarrow \pi_i^- + \sigma^*$
$\pi_j^+ \leftarrow \pi_j^+ + \sigma^*$
$\theta_i^- \leftarrow \theta_i^- + 1$
$\theta_j^+ \leftarrow \theta_j^+ + 1$
$T \leftarrow \tau \times T$
$iter \leftarrow iter + 1$
end while
for each w_i^- **do**
$w_i^- \leftarrow \rho w_i^- + (1-\rho)\pi_i^- / \theta_i^-$
end for
for each w_j^+ **do**
$w_j^+ \leftarrow \rho w_j^+ + (1-\rho)\pi_j^+ / \theta_j^+$
end for
$seg \leftarrow seg + 1$
loop
return x_{best}

solution occurs in three steps. First, $|K|$ routing sets are generated according to the sequence of compulsory stops that must be visited. An empty set is initialized to represent the rejected set. Second, a set of candidate requests is generated, with each request represented by one of its stops. Third, requests are inserted into their minimum cost positions according to the order of generation for an instance. The insertion cost is calculated as the cost after insertion minus the cost before insertion. For this insertion process, the MDRP stipulates that both stops of each request must be included in the same route, with the drop-off stop occurring after the pick-up stop. Consequently, different request types require specific insertion logic. O2O requests can be inserted into any route and at any position, whereas C2O and O2C requests must be inserted into their designated routes, with C2O optional stops placed after and O2C optional stops placed before their corresponding compulsory stops. If the insertion of a request violates time windows at compulsory stops, the request is removed and transferred to the rejected set.

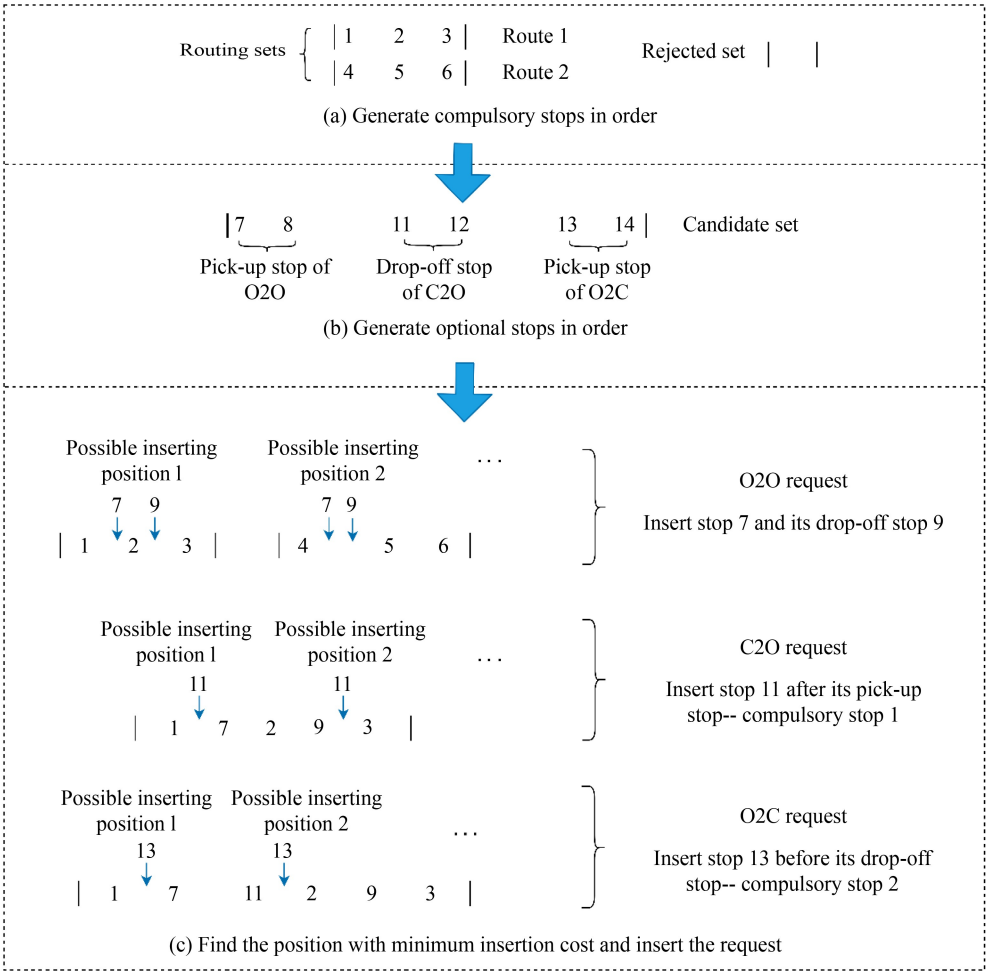

Fig. 3 An example of the construction for the initial solution.

Upon the completion of this process, where all requests have either been inserted or rejected, we obtain an initial solution.

4.2 Destroy and repair operators

The ALNS heuristic employs three destroy operators and two repair operators. The destroy operators are responsible for removing several previously inserted requests, while the repair operators focus on reinserting them into more advantageous positions to improve solution quality.

4.2.1 Destroy operators

The inputs of the destroy operators are $|K|$ complete routes and a rejected set Φ. Given the number of total removed requests q, the outputs are $|K|$ partial routes and $q-|\Phi|$ requests removed from the original routes together with the rejected set Φ. The $q-|\Phi|$ removed requests and rejected requests in Φ form the unplaced request set U, which are the candidate requests in the subsequent repair process. The three destroy operators are illustrated as follows.

Random removal operator. The random removal operator eliminates a selection of requests from the routes randomly, resulting in significant alterations to request assignments and vehicle routing. This process enhances search diversity and increases the likelihood of discovering new solutions. Figure 4 demonstrates the request removal process.

Worst removal operator. To minimize total costs, this operator targets requests situated in less advantageous positions for removal, enabling their placement in more appropriate locations through repair operators. Requests with higher insertion costs are deemed more likely to be incorrectly positioned, and thus they are prioritized for removal. The worst removal operator selects one request at a time using a roulette mechanism, where the probabilities correspond to the costs of the inserted requests. As illustrated in Fig. 5, the "inserted set" refers to all requests currently integrated into the routes. The incorporation of randomness through roulette adds variability to the operator and diversifies the search.

Related removal operator. The related removal

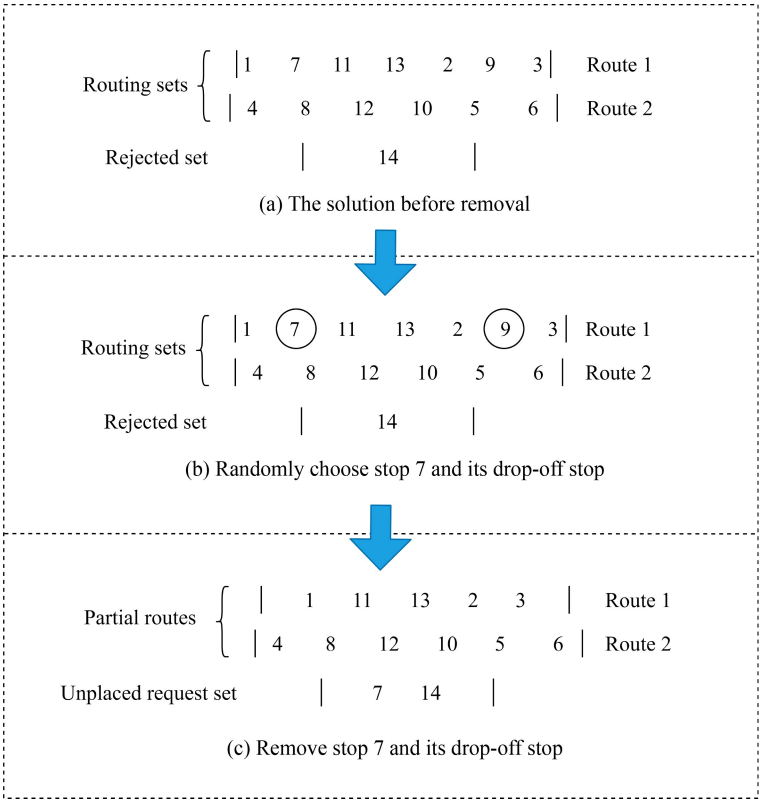

Fig. 4 An example of the random removal operator.

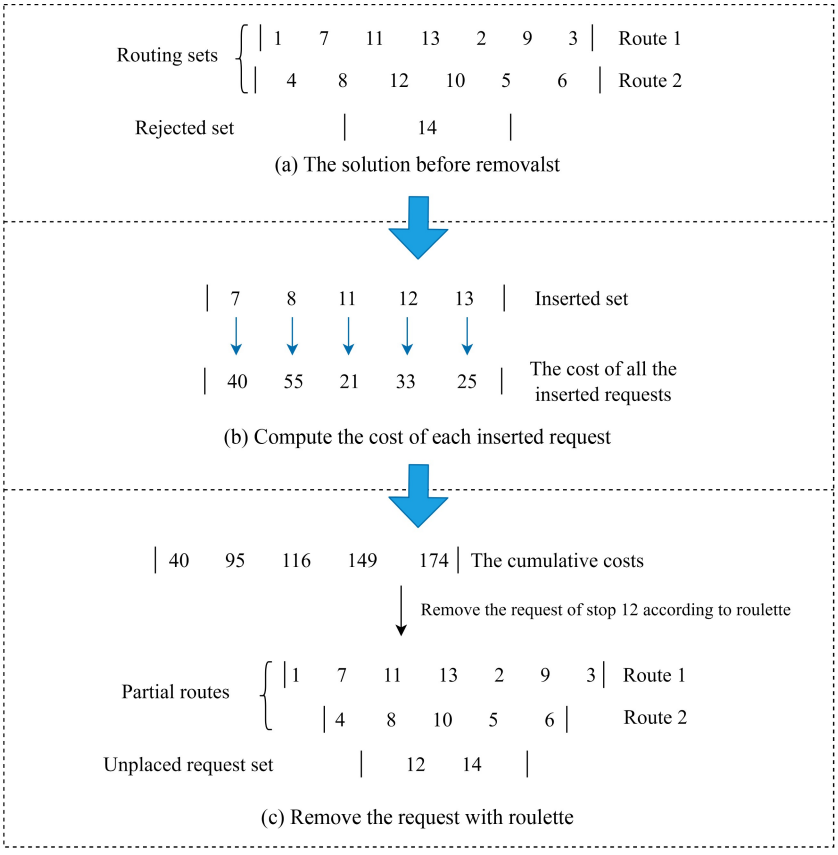

Fig. 5 An example of the worst removal operator.

operator removes a set of related requests that are geographically close or have similar boarding times. Because these requests can be more easily interchanged within routes to identify improved solutions. A distance metric and a time metric are used to assess the relatedness between requests. For request r and request r', the distance metric is: $D_{s(r),s(r')} + D_{d(r),d(r')}$, where $D_{i,j}$ is the rectilinear distance between stop i and stop j; the time metric is: $|Q_r - Q_{r'}|$. The two metrics are weighted by α and β, respectively. Hence, the relatedness is expressed as

$$\gamma_{r,r'} = \alpha[D_{s(r),s(r')} + D_{d(r),d(r')}] + \beta[|Q_r - Q_{r'}|]. \quad (41)$$

Clearly, a smaller $\gamma_{r,r'}$ indicates a higher relatedness between the two requests. The possibilities of roulette are inversely proportional to the relatedness. As Fig. 6 shows, the algorithm begins by randomly selecting a request r to initiate the selected list. Following the initial removal, the related removal operator selects a new request from the list, calculates its relatedness with the remaining inserted requests, and removes one request from the inserted set to the selected list using roulette.

4.2.2 Repair operators

The repair operators insert requests from U in $|K|$ partial routes. When the insertion of a request violates the time windows at compulsory stops, it is moved to the rejected set. This paper utilizes two repair operators: the basic greedy operator and the regret operator.

Basic greedy operator. Requests with lower insertion costs are more likely to be positioned correctly. The basic greedy operator repeatedly inserts the request with the lowest cost. Specifically, let $\Delta f_{r,k}$ denote the added value of the objective value incurred by inserting request r at the cheapest position in route k. Set $\Delta f_{r,k} = \infty$ if request r cannot be inserted into route k. Then we calculate

$$(r,k) = \arg \min_{r \in U, k \in K} \Delta f_{r,k}, \quad (42)$$

and insert request r into route k at its minimum cost position.

Regret operator. The MDRP involves multiple routes. We must determine the assignment of requests to routes while considering look-ahead information. The regret operator prioritizes assigning requests with high

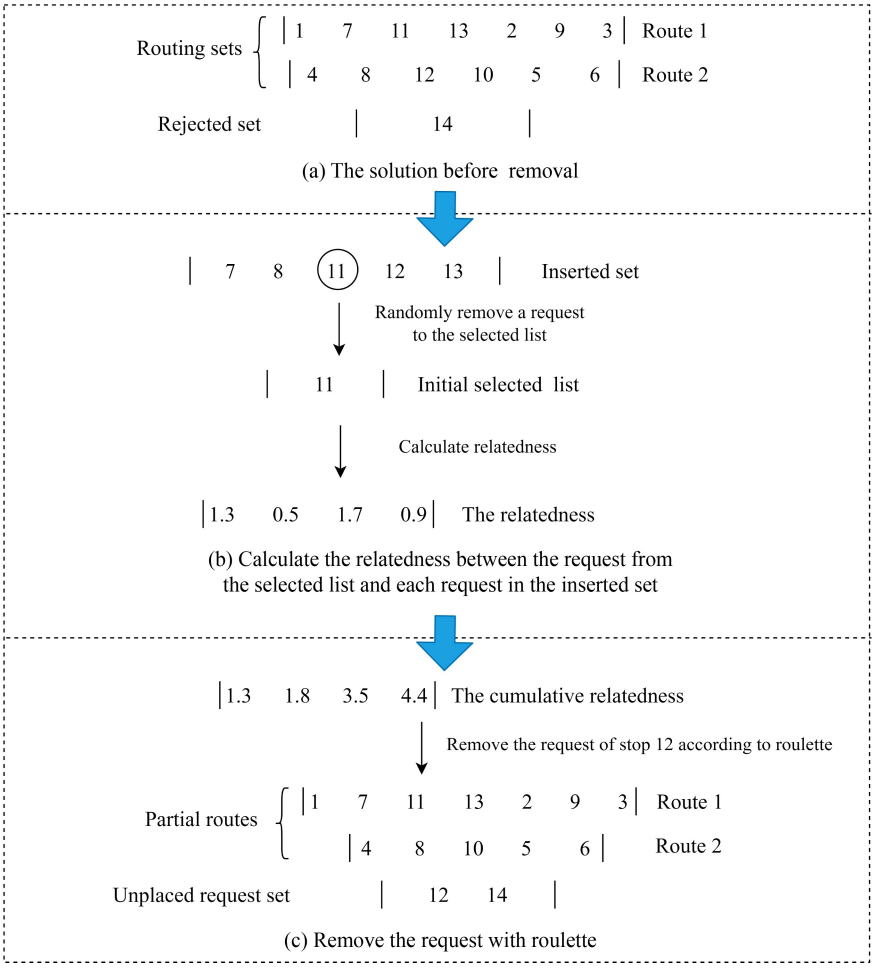

Fig. 6 An example of the first remove of the related removal operator.

subsequent insertion costs to their lowest-cost route, thereby preventing challenges in later insertions. Let $\Delta f_{r,m}^{th}$ denote the added value of the objective value incurred by inserting request r at the minimum cost position in its mth cheapest route. For example, $\Delta f_{r,2}^{th}$ denotes the added value of the objective value by inserting request r in its second cheapest route. The regret value of request r is defined as: $\Delta f_{r,2}^{th} - \Delta f_{r,1}^{th}$. In each iteration, we calculate

$$r = \arg\max_{r \in U} (\Delta f_{r,2}^{th} - \Delta f_{r,1}^{th}) \tag{43}$$

and insert request r at the minimum cost position in its cheapest route. This process repeats until no more requests can be inserted.

4.3 Master local search framework

Simulated annealing is used as the local search framework at the master level. In each iteration, a candidate solution x' is accepted given the current solution x with probability $e^{-(F(x')-F(x))/T}$, where $T > 0$ is the temperature and $F(x)$ denotes the objective value of solution x. An exponential cooling rate $\tau (0 < \tau < 1)$ is used to decrease T from the initial temperature T_{start}. To calculate T_{start}, we refer to the method outlined by Ropke and Pisinger (2006), which adjusts based on the instance size. Initially, we calculate the modified cost of the initial solution by setting the rejection cost to zero. Next, this study determines the starting temperature so that a solution ω percent worse than the initial solution is accepted with a probability of 0.5, where ω is the start temperature control parameter. The rejection cost is disregarded to prevent an excessively high starting temperature when the initial solution includes rejected requests.

4.4 Adaptive weights adjustment

As shown in Step 4 of Fig. 2, the selection of destroy and repair operators in each iteration is determined by a roulette mechanism, weighted by the values $w_{i,j}$ of each operator. Operators with higher weights are more likely to be selected, reflecting their contribution to improving the solution. The entire iterative process is divided into several time segments, each consisting of 10 iterations, with weights updated at the end of each segment. A score for each operator is collected in each segment and used to calculate its weight. The score $\pi_{i,j}$ of operator i in time segment j is adjusted based on parameters related to the new solution x_{new}, as shown in Table 3.

We calculate $\pi_{i,j}$ as follows:

$$\pi_{i,j} = \sigma_1 n_{i,j}^1 + \sigma_2 n_{i,j}^2 + \sigma_3 n_{i,j}^3 + \sigma_4 n_{i,j}^4, \tag{44}$$

where $n_{i,j}^1$ represents the times operator i is called and generated new global best solutions in time segment j, corresponding to the description of σ_1 in Table 3. The other three notations have similar meanings. At the end of segment j we calculate the weight of operator i as follows:

$$w_{i,j+1} = (1-\rho) w_{i,j} + \rho \frac{\pi_{i,j}}{\theta_{i,j}}, \tag{45}$$

where $\theta_{i,j}$ represents the times operator i is called in time segment j and ρ is the reaction factor controlling the responsiveness of the weight adjustment algorithm. A larger ρ gives greater weight to the most recent segment's score compared to past scores.

5 Numerical experiments

The objectives of the experiments detailed in this section are to: (1) fine-tune the ALNS heuristics for improved performance; (2) conduct comparisons to highlight the advantages of the proposed model and the efficiency of the ALNS algorithm; and (3) perform sensitivity analyses to examine the effects of various parameters on system performance. The algorithms presented in this paper are implemented using MATLAB on a machine equipped with an Intel Core i9-12900KF CPU. The ALNS algorithm is executed ten times for each test instance to acquire average values.

5.1 Parameter values

This section describes the generation of test instances used in the numerical experiments. Some parameters of the generated instances are based on the metropolitan transit authority (MTA) Line 646 flex-route transit service, which has been widely cited in previous studies (Quadrifoglio et al., 2007; Quadrifoglio et al., 2008; Qiu et al., 2014a; Qiu et al., 2014b; Zheng et al., 2018b; Zheng et al., 2019). The parameter values are listed in

Table 3 Score Adjustment Parameters

Parameter	Description
σ_1	The last remove-insert operation results in a new global best solution x_{new}.
σ_2	The last remove-insert operation results in a solution x_{new} whose cost is better than the cost of current solution x_{cur}.
σ_3	The last remove-insert operation results in a solution x_{new} whose cost is worse than the cost of current solution x_{cur}, but the solution is accepted.
σ_4	The last remove-insert operation results in a solution x_{new} whose cost is worse than the cost of current solution x_{cur}, and the solution is rejected.

Table 4. As shown in Fig. 7, the DAS operates within a rectangular region with two parallel routes evenly distributed longitudinally. Each route has nine compulsory stops evenly spaced along the base route, with the first and last stops serving as terminals. A bus serves each DAS route. The proportions of the three request types are η_1, η_2, and η_3. All requests outside the compulsory stops are assumed to be uniformly distributed throughout the service area. Considering the route planning for DAS during peak hours, it is reasonable to set the bus's full trip duration T_r to approximately one hour. Additionally, the time window width δ at all compulsory stops is set to 2 min, except at the origin stop, where both the EDT and LDT are zero (Crainic et al., 2005). Similar to Crainic et al. (2010), travel time between stops is calculated using Euclidean distances, and travel time is estimated based on the average vehicle speed V_b (Millward et al., 2013). For the unit time cost, we refer to Li et al. (2022). Lastly, the penalty for request rejection is set to a high value to minimize the system's rejection rate.

Table 4 Parameter values

Parameter	Value	Unit
L	10	miles
W	1.5	miles
K	2	
P	9	
V_b	25	miles/h
T_r	64	min
δ	2	min
$\eta_1/\eta_2/\eta_3$	$>0.5/<0.25/<0.25$	
λ_1	0.3	\$/min
λ_2	0.3	\$/min
λ_3	0.4	\$/min
λ_4	50	\$/passenger

5.2 Parameter tuning

In this section, the parameters of the ALNS algorithm are tuned to improve its performance. This study chooses $\max\_gen = 30$, $\max\_iter = 10$ and $\max\_idle = 100$ based on prior experience. Nine parameters require tuning, namely, α in the related removal operator, τ and ω in simulated annealing, σ_1, σ_2, σ_3, σ_4 and ρ in the weight adjustment algorithm, ξ that is the percentage of requests removed in each iteration. Following the method outlined by Ropke and Pisinger (2006), our tuning procedure is as follows: First, we adjust one parameter within a predefined range (determined through an ad hoc trial-and-error process) while keeping the other parameters constant. We then proceed to the next parameter, utilizing the values derived from the previous tuning. This process continues until all parameters have been optimized. A total of 30 instances are generated for the calibration phase, taking into account 10 demand levels evenly distributed from 12 to 39 passengers, with three instances for each demand level. The parameter tuning results are $(\alpha,\tau,\omega,\sigma_1,\sigma_2,\sigma_3,\sigma_4,\rho,\xi)$ = (0.3, 0.94, 1.05, 10, 2, 3, 1, 0.7, 0.7). The detailed parameter tuning process is presented in Appendix A.

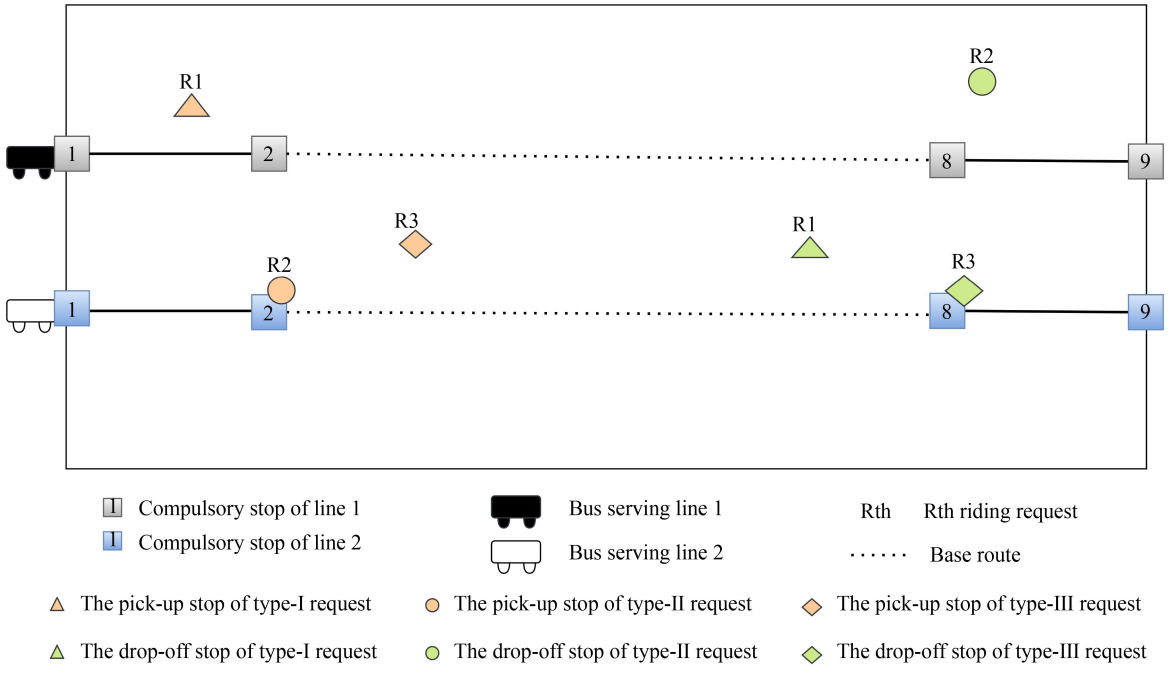

Fig. 7 An instance of the demand-adaptive system.

5.3 Comparison between ALNS and exact algorithms

To evaluate the effectiveness of the ALNS algorithm, numerical experiments are performed to compare its performance against the exact branch-and-cut algorithm utilized in the commercial solver GUROBI. The objective function values produced by both algorithms across various demand levels are assessed, maintaining a maximum computational time limit of one hour. Since DAS typically operates in low-demand areas, a single route is assumed with a maximum passenger demand of 25 passengers per 40-min trip (Zheng et al., 2019). Consequently, for a larger region with two routes, a maximum demand of 40 passengers per hour is deemed reasonable.

The results in Table 5 reveal that the runtime of the exact algorithm escalates exponentially with increased passenger demand. For example, at a demand level of 8, the computation time for the exact algorithm already exceeds one hour. In scenarios of moderate to high demand levels, the exact algorithm necessitates substantial computational resources. Conversely, the ALNS algorithm consistently resolves instances of varying sizes within 250 s, thereby fulfilling the high responsiveness requirements of DAS. Furthermore, the objective values calculated by the ALNS algorithm are equal to or even less than that of the exact algorithm within the one-hour timeframe, with an average cost reduction of 10.48%. Consequently, this study concludes that the ALNS algorithm demonstrates commendable performance in terms of both solution quality and computational efficiency.

5.4 Comparison between ALNS and other heuristic algorithms

To further illustrate the efficacy of our customized ALNS, we compare it with the large neighborhood search (LNS) algorithm and genetic algorithm (GA), both of which are commonly employed in DAS optimization (Guo et al., 2018; Galarza Montenegro et al., 2021). Additionally, two classic heuristic algorithms for continuous optimization are tested: particle swarm optimization (PSO) and differential evolution (DE). Six randomly generated instances at various demand levels are used to perform the experiment.

Table 6 demonstrates that the ALNS algorithm outperforms other heuristic approaches regarding objective value within the same timeframe, achieving average savings of 7.12%, 11.51%, 49.4%, and 48.09%, respectively. Conversely, PSO and DE exhibit the poorest performance. This is largely because these algorithms are more suited to continuous optimization problems, whereas the 0−1 variables in our mixed-integer linear model present challenges for them. Both the LNS and GA outperform PSO and DE, as they are more suitable for combinatorial optimization tasks and can develop customized solutions and iterative operators specific to the MDRP. However, both LNS and GA still lag behind ALNS. The ALNS algorithm utilizes a broader range of operators to diversify the search process and can adaptively select and deploy more effective operators, thereby expanding the search space and enhancing the likelihood of discovering superior solutions.

5.5 Benefits of multiple routes

To analyze the advantages of incorporating multiple routes for DAS, this study compares the results between the single-route DAS (hereinafter referred to as SR in the figures) and the proposed multi-route DAS (hereinafter referred to as MR). Two route layouts are considered: one is the parallel layout discussed in Section 5.1, and the other features an intersection, as illustrated in Fig. 8. In

Table 5 Comparison between exact and heuristic algorithms

Demand	GUROBI results		ALNS results		Gap	TG/TA[b)]
	Obj[a)] ($)	Time (s)	Obj[a)] ($)	Time (s)		
4	168.05	30.61	168.05	18.00	0.00%	1.70
6	130.88	52.52	130.88	18.62	0.00%	2.82
8	305.67	3600	305.67	18.88	0.00%	190.72
10	407.13	3600	407.13	19.24	0.00%	187.10
15	845.01	3600	844.64	26.95	−0.04%	133.60
20	552.56	3600	551.26	37.76	−0.24%	95.35
25	1135.49	3600	1129.68	52.30	−0.51%	68.84
30	3900.00	3600	1100.02	92.26	−71.79%	39.02
35	2617.11	3600	2006.11	147.14	−23.35%	24.47
40	2352.18	3600	2143.85	246.83	−8.86%	14.58
Avg					−10.48%	75.82

Note: a) Objective value; b) The runtime ratio of the exact algorithm to the ALNS algorithm.

Table 6 Comparison between ALNS and LNS/GA/PSO/DE algorithms at different demand levels

Demand	ALNS	LNS		GA		PSO		DE	
	Obj[a] ($)	Obj[a] ($)	Gap	Obj[a] ($)	Gap	Obj[a] ($)	Gap	Obj[a] ($)	Gap
15	844.56	926.26	9.67%	854.80	1.21%	968.36	14.66%	948.87	12.35%
20	550.19	562.36	2.21%	601.17	9.27%	823.92	49.75%	797.79	45.00%
25	1130.66	1144.31	1.21%	1230.47	8.83%	1558.06	37.80%	1583.55	40.06%
30	1113.51	1287.08	15.59%	1337.99	20.16%	2245.84	101.69%	2125.68	90.90%
35	2003.78	2156.81	7.64%	2414.48	20.50%	3003.59	49.90%	3026.34	51.03%
40	2146.81	2284.24	6.40%	2342.09	9.10%	3061.78	42.62%	3203.19	49.21%
Avg			7.12%		11.51%		49.40%		48.09%

Note: a) Objective value.

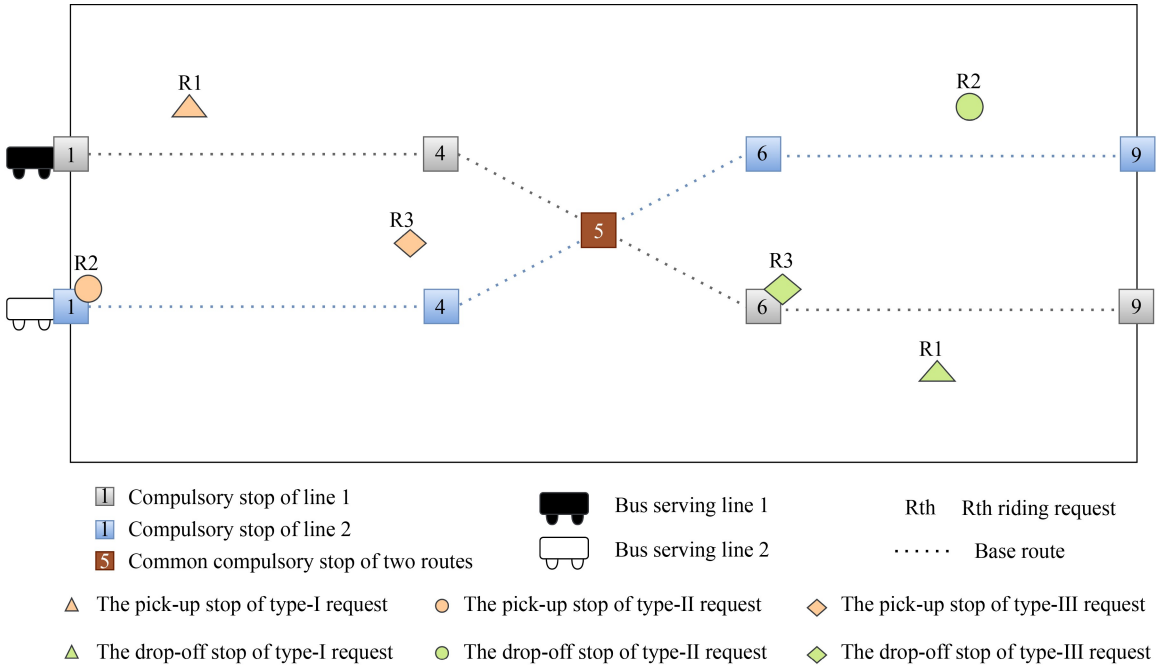

Fig. 8 An instance of the demand-adaptive system with an intersection.

the subsequent figures, the single-route DAS in the parallel layout is termed SR-P, while the intersecting layout is designated SR-I. Similarly, MR-P and MR-I represent the multi-route DAS in parallel and intersecting layouts, respectively.

In the single-route scenario, requests are pre-assigned to each route based on distance prior to optimization. Various demand levels are tested to reflect fluctuations in actual demand, with requests generated randomly within the service area. This study evaluates and compares system performance using the following indicators: (1) the rejection rate, defined as the percentage of passengers rejected by the system; (2) the average in-vehicle time per passenger (IVT); (3) the average waiting time per passenger before boarding the bus (WT); (4) the travel time for buses on the road (TT); and (5) the total cost, representing the objective value of the model. Indicators (1), (2), and (3) are essential for assessing the service level of the DAS and measuring its operational efficiency.

The results are illustrated in Figs. 9–12. In comparison to the single-route DAS, the multi-route DAS demonstrates the capability to accommodate a greater number of passengers (refer to Fig. 9) while incurring a lower total cost (Fig. 10). It is important to note that the single-route DAS pre-assigns requests based on distance before optimization, a technique that has been commonly employed in previous studies (Crainic et al., 2005; Quadrifoglio et al., 2007; Quadrifoglio et al., 2008). If a request cannot be accommodated by a particular route, it is subsequently rejected. In contrast, our model facilitates the assignment of requests among routes during the optimization phase. Consequently, the multi-route DAS achieves lower rejection rates. It can be observed that rejection rates escalate with increasing demand levels in both scenarios; however, the multi-route DAS consistently maintains a

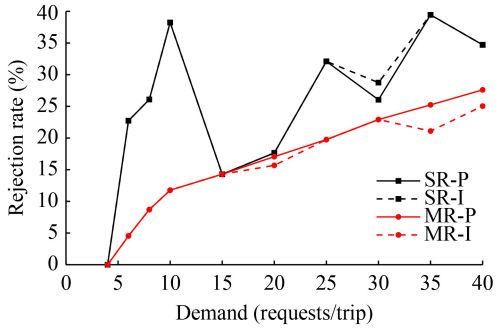

Fig. 9 Rejection rates under different demand levels.

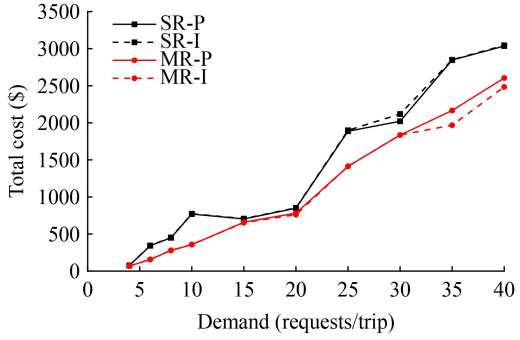

Fig. 10 Total costs under different demand levels.

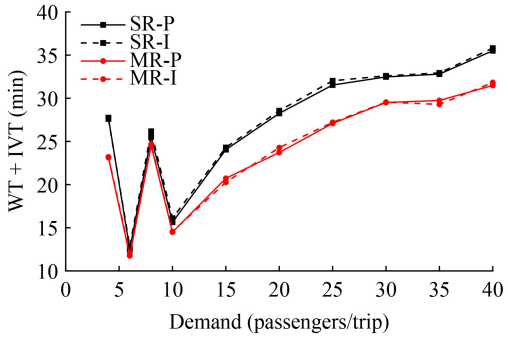

Fig. 11 WT + IVT under different demand levels.

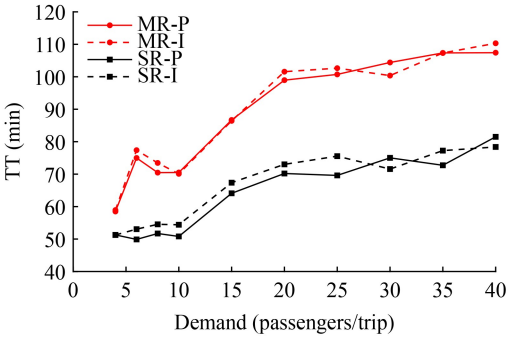

Fig. 12 TT under different demand levels.

lower rejection rate than the single-route DAS across all demand levels. Notably, for the case involving 10 requests, the rejection rate for the single-route DAS reaches as high as 40%, whereas only 10% of requests are rejected in the multi-route scenario.

Additionally, our findings indicate that the performance of our model improves as the complexity of the DAS route layouts increases. As illustrated in Fig. 9, with the increase in route layout complexity, the discrepancy in rejection rates between single-route DAS and multi-route DAS becomes even more pronounced. Similarly, the difference in total costs between the two systems also grows, as Fig. 10 shows. For instance, in the case with 35 requests, the multi-route DAS operating within an intersecting layout can accommodate 4.13% more passengers and further reduce total costs by 9.15% compared to the parallel layout. Furthermore, the total waiting time and in-vehicle time per passenger is reduced (Fig. 11) in the multi-route DAS compared to the single-route DAS at every demand level. However, as a result of serving more requests, the overall travel time associated with the multi-route DAS increases (Fig. 12).

These findings indicate that the incorporation of multiple routes enhances service levels while simultaneously reducing costs.

5.6 Sensitivity analysis

This section presents the results of sensitivity analyses aimed at examining the effects of various parameters on system performance. Given that both route layouts yield similar results in Section 5.5, only the parallel layout is considered in this analysis. All experiments are conducted at a demand level of 20 requests per trip.

Table 7 presents the effect of varying the unit travel time cost λ_1 on system performance indicators. It can be seen that the rejection rate remains unchanged as λ_1 increases. Meanwhile, TT decreases, while IVT and WT both show an upward trend. In our model, a very high penalty is applied for rejecting requests, significantly increasing the objective value if requests are rejected. As a result, small increases in λ_1 do not lead to more rejected requests. An interesting observation is that TT decreases despite the rejection rate staying constant. This indicates that the model serves the same requests but adjusts the routes as λ_1 rises. To control total cost, buses tend to alter routes to reduce overall travel time. However, shorter routes change the visiting sequence of requests, leading to increased IVT and WT.

Table 8 illustrates the effect of varying the unit in-vehicle time cost λ_2 on system performance indicators. As observed in Table 7, the rejection rate remains unchanged as λ_2 increases. However, IVT decreases, while TT and WT both rise. The high penalty for rejecting requests ensures that the rejection rate stays constant. As λ_2 increases, the model prioritizes minimizing passengers' in-vehicle time, prompting buses to adjust routes to shorten the distance between pick-up and drop-off stops. This leads to an increase in total travel distance, thereby raising TT. Additionally, the later arrival times at pick-up

Table 7 Sensitivity analysis on the unit travel time cost

λ_1 ($/min)	Rejection rate (%)	TT (min)	IVT (min)	WT (min)
0.1	0.16	106.52	20.90	3.65
0.2	0.16	106.52	20.90	3.65
0.3	0.16	101.80	20.55	3.98
0.4	0.16	101.80	20.55	3.98
0.5	0.16	101.80	20.55	3.98
0.6	0.16	95.46	20.59	4.09
0.7	0.16	84.42	21.01	4.15
0.8	0.16	80.34	21.12	4.25

Table 8 Sensitivity analysis on the unit in-vehicle time cost

λ_2 ($/min)	Rejection rate (%)	TT (min)	IVT (min)	WT (min)
0.1	0.16	84.02	21.64	3.69
0.2	0.16	84.02	21.64	3.69
0.3	0.16	84.02	21.64	3.69
0.4	0.16	95.03	18.59	5.89
0.5	0.16	100.53	17.04	7.49
0.6	0.16	100.53	17.04	7.49
0.7	0.16	100.53	17.04	7.49
0.8	0.16	100.53	17.04	7.49

stops result in longer WT.

Table 9 illustrates how varying the unit waiting time cost λ_3 affects system performance indicators. The rejection rate remains unchanged as λ_3 increases. Meanwhile, WT and TT generally decrease, while IVT shows an upward trend. Similar to previous analyses, the high penalty for rejecting requests keeps the rejection rate constant. As λ_3 increases, the model prioritizes reducing WT by picking up passengers earlier, which results in a longer IVT. TT generally decreases because, with higher λ_3, buses minimize detours, aiming to pick up passengers more quickly and reduce WT.

Table 10 shows the effect of varying the unit cost of rejecting requests λ_4 on system performance indicators. The rejection rate decreases as λ_3 increases, while TT, IVT and WT all rise. When λ_4 is low, the cost of serving requests is higher than rejecting them, leading to a high rejection rate. As λ_4 increases, the model accepts more requests, resulting in buses taking more detours and extending their routes, which in turn increases TT, IVT, and WT.

In the next set of experiments, as shown in Table 11, we vary the average vehicle speed V_b. In real-life operations, average speed may fluctuate due to factors such as inclement weather. The results show that the rejection rate decreases as V_b increases. Meanwhile, IVT increases, WT decreases, while TT fluctuates around 99 min. The decrease in rejection rate with increasing V_b is straightforward, as buses can transport more passengers within the same time frame. When the same number of requests is served at higher speeds, TT tends to decrease. However, if more requests are accepted as V_b increases, TT fluctuates. IVT rises because buses take more detours to accommodate additional requests, while WT decreases due to earlier arrivals at pick-up stops.

Finally, this study examines the effects of varying time window width δ on system performance. For C2C passengers who do not reserve DAS and wait at compulsory stops, a waiting time between 1 and 5 min is considered reasonable. Therefore, nine different time window widths are tested, as shown in Table 12. The results show that the rejection rate decreases as δ increases. Meanwhile, TT and WT both have an upward trend, while IVT decreases. The time windows at compulsory stops are

Table 9 Sensitivity analysis on the unit waiting time cost

λ_3 ($/min)	Rejection rate (%)	TT (min)	IVT (min)	WT (min)
0.1	0.16	101.04	15.40	10.63
0.2	0.16	97.44	17.09	8.18
0.3	0.16	95.74	18.44	5.93
0.4	0.16	95.74	18.44	5.93
0.5	0.16	100.17	18.96	5.40
0.6	0.16	93.12	21.18	3.69
0.7	0.16	93.12	21.18	3.69
0.8	0.16	93.12	21.18	3.69

Table 10 Sensitivity analysis on the unit cost of rejecting requests

λ_4 ($/passenger)	Rejection rate (%)	TT (min)	IVT (min)	WT (min)
2	0.75	72.89	13.58	1.49
4	0.75	72.89	13.58	1.49
6	0.53	73.62	14.48	0.18
8	0.53	73.62	14.48	0.18
10	0.53	73.62	14.48	0.18
12	0.37	80.78	15.88	1.65
14	0.29	90.73	18.24	1.75
16	0.18	98.34	19.18	3.85

Table 11 Sensitivity analysis on the average vehicle speed

V_b (miles/h)	Rejection rate (%)	TT (min)	IVT (min)	WT (min)
23	0.18	100.05	18.62	5.11
23.5	0.18	95.71	19.57	4.27
24	0.18	96.07	19.34	4.20
24.5	0.18	97.87	19.13	4.14
25	0.16	102.29	20.52	4.00
25.5	0.10	97.31	20.86	4.79
26	0.10	102.59	21.07	3.96
26.5	0.10	100.65	20.93	3.90
27	0.10	98.79	20.81	3.85

Table 12 Sensitivity analysis on the time window width

δ (min)	Rejection rate (%)	TT (min)	IVT (min)	WT (min)
1	0.18	96.46	20.24	3.93
1.5	0.18	96.21	19.89	3.89
2	0.18	90.11	19.68	3.71
2.5	0.16	100.76	19.71	3.70
3	0.10	98.73	21.12	4.62
3.5	0.10	99.24	18.12	6.01
4	0.10	100.83	17.73	5.06
4.5	0.10	100.83	17.53	4.80
5	0.10	101.31	17.24	4.78

shared by their two consecutive segments, and a wider time window provides greater flexibility in time allocation for each segment. This allows the model to serve more requests by utilizing the additional slack time, explaining the lower rejection rate and higher TT. IVT decreases because buses can depart from compulsory stops earlier, leading to earlier drop-offs for each request. WT rises as buses tend to make more detours to accommodate additional requests with the increased time window widths, resulting in later arrivals at pick-up stops.

6 Conclusions

DAS integrates the cost-efficiency of FRT with the flexibility of DAR, is highly promising to operate in low demand areas. It can also collaborate with mass transit as a feeder system to solve the first/last mile problem in large cities. To solve the optimal routing and request selection problem for multiple service routes in DAS, this paper proposes a MILP model and a tailored ALNS algorithm to solve the model efficiently. Experiments are conducted to demonstrate the advantages of the proposed model and the efficiency of the ALNS algorithm. The results are summarized as follows.

(1) The tailored ALNS algorithm outperforms the exact algorithm in terms of both solution time and quality. All sizes of instances can be solved by an ALNS algorithm up to 250 s and at solution costs equal to or even less than those of the exact algorithm within the one-hour timeframe. The average gap between ALNS and exact algorithm is 10.48%.

(2) The tailored ALNS algorithm outperformed LNS, GA, PSO, and DE. The average objective value is reduced by 7.12%, 11.51%, 49.4%, and 48.09%, respectively.

(3) The proposed MILP model and tailored ALNS algorithm provide improvements in the effectiveness of the system. The multi-route DAS comparatively handles more passengers than the single-route DAS while serving them at lower total costs.

(4) Sensitivity analyses reveal the effects of various inputs parameter-wise on system performance. The three unit time costs mostly did reduce their respective time costs while at the same time raising other time costs. The greater the unit costs of rejecting requests, the lower the rejection rate, letting more passengers to be served. Speed increase saves time and gives more passengers a chance to be served. When a wider time window is allowed, the bus planner is given flexibility in time allocation for segments, hence it can serve a higher number of passengers.

For future research, there are several interesting directions. For example, one could explore how to determine the time window at each compulsory stop while considering potential requests at optional stops. Additionally, studies have shown that appropriate incentive pricing mechanisms can facilitate traffic control, promote cooperative behavior, and improve resource allocation (Bi et al., 2021; Bi et al., 2022). Accordingly, researchers could design a differential pricing mechanism for optional and compulsory stops to encourage users to board at compulsory stops and reduce bus detours.

Acknowledgements We sincerely thank the editor and reviewers for their valuable comments to improve this manuscript.

Competing Interests The authors declare that they have no competing interests.

Appendix A. Tuning the values of parameters

We conduct sensitivity analyses to compare the algorithm's performance across different values of specific parameters. The "percentage gap" is the evaluation indicator, as shown in Fig. A1. The following describes the method used to calculate it. Let I denote the set of instances, with index i. The level set, which is the set of alternative values for the parameter to be tuned, is denoted by J, with index j. $a_{i,j}$ denotes the objective value for level j of instance i. The minimum objective value among all levels for instance i, denoted by a_i^*, is calculated as follows:

$$a_i^* = \min_{j \in J} a_{i,j}. \tag{A1}$$

For instance i, $e_{i,j}$ denotes the relative difference between the objective value at level j and the minimum objective value among all levels, which is calculated as follows:

$$e_{i,j} = \frac{a_{i,j} - a_i^*}{a_i^*}. \tag{A2}$$

The sum of relative differences across all instances for level j, denoted by e_j, represents the "percentage gap" and is calculated as follows:

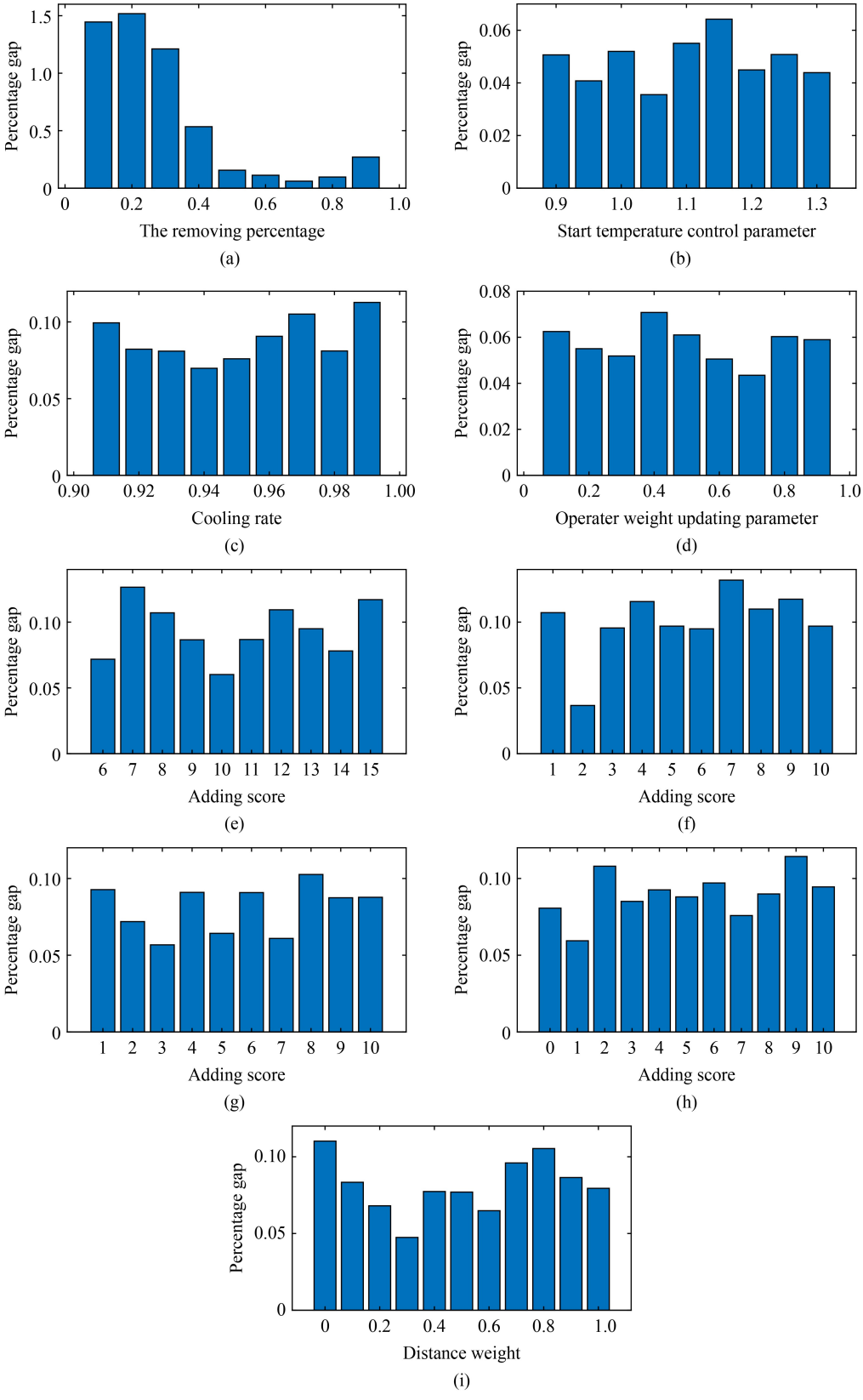

Fig. A1 The sensitivity analysis results of nine parameters.

$$e_j = \sum_{i \in I} e_{i,j}. \tag{A3}$$

The parameters being analyzed include: (a) ξ: the percentage of requests removed by the removal operators; (b) ω: the start temperature control parameter; (c) τ: the cooling rate in simulated annealing; (d) ρ: the weight reaction parameter; (e)–(h) σ_1, σ_2, σ_3, σ_4: the score adjustment parameters; (i) α: the distance weight in the related removal operator.

References

Becker J, Teal R, Mossige R (2013). Metropolitan transit agency's experience operating general-public demand-responsive transit. Transportation Research Record: Journal of the Transportation Research Board, 2352(1): 136–145

Bi H, Shang W L, Chen Y, Wang K, Yu Q, Sui Y (2021). GIS aided sustainable urban road management with a unifying queueing and neural network model. Applied Energy, 291: 116818

Bi H, Shang W L, Chen Y, Yu K, Ochieng W Y (2022). An incentive based road traffic control mechanism for Covid-19 pandemic alike emergency preparedness and response. IEEE Transactions on Intelligent Transportation Systems, 23(12): 25092–25105

Chen P W, Nie Y M (2017). Analysis of an idealized system of demand adaptive paired-line hybrid transit. Transportation Research Part B: Methodological, 102: 38–54

Crainic T G, Errico F, Malucelli F, Nonato M (2010). Designing the master schedule for demand-adaptive transit systems. Annals of Operations Research, 194(1): 151–166

Crainic T G, Malucelli F, Nonato M, Guertin F (2005). Meta-heuristics for a class of demand-responsive transit systems. INFORMS Journal on Computing, 17(1): 10–24

Errico F, Crainic T G, Malucelli F, Nonato M (2013). A survey on planning semi-flexible transit systems: Methodological issues and a unifying framework. Transportation Research Part C, Emerging Technologies, 36: 324–338

Errico F, Crainic T G, Malucelli F, Nonato M (2021). The single-line design problem for demand-adaptive transit systems: A modeling framework and decomposition approach for the stationary-demand case. Transportation Science, 55(6): 1300–1321

Fittante S R, Lubin A (2016). Adapting the Swedish service route model to suburban transit in the United States. Transportation Research Record: Journal of the Transportation Research Board, 2563(1): 52–59

Fu L (2002). Planning and design of flex-route transit services. Transportation Research Record: Journal of the Transportation Research Board, 1791(1): 59–66

Guo R, Guan W, Zhang W (2018). Route design problem of customized buses: Mixed integer programming model and case study. Journal of Transportation Engineering. Part A: Systems, 144(11)

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

Jin W, Du H, Wu W (2023). Semi-flexible demand responsive transit scheduling based on ALNS-TS algorithm. Journal of Shenzhen University Science and Engineering, 40(4): 425–434

(Edward) Kim M, Levy J, Schonfeld P (2019). Optimal zone sizes and headways for flexible-route bus services. Transportation Research Part B: Methodological, 130: 67–81

Koffman D (2004). Operational experiences with flexible transit services (No. 53). Washington, D.C.: Transportation Research Board

Li M, Tang J (2023). Simulation-based optimization considering energy consumption for assisted station locations to enhance flex-route transit. Energy, 277: 127715

Li M, Tang J, Zeng J, Huang H (2023a). A Kriging-based optimization method for meeting point locations to enhance flex-route transit services. Transportmetrica. B, Transport Dynamics, 11(1): 1281–1310

Li X, Huang J, Guan Y, Li Y, Yuan Y (2022). Electric demand-responsive transit routing with opportunity charging strategy. Transportation Research Part D, Transport and Environment, 110: 103427

Li X, Liu W, Qiao J, Li Y, Hu J (2023b). An enhanced semi-flexible transit service with introducing meeting points. Networks and Spatial Economics, 23(3): 487–527

Li X, Quadrifoglio L (2009). Optimal zone design for feeder transit services. Transportation Research Record: Journal of the Transportation Research Board, 2111(1): 100–108

Liu X, Qu X, Ma X (2021). Improving flex-route transit services with modular autonomous vehicles. Transportation Research Part E, Logistics and Transportation Review, 149: 102331

Lu B, He X, Diao S, Shu Q (2020). Study on coordinated scheduling of multi-route flexible buses in urban periphery in off-peak period. Journal of Highway and Transportation Research and Development, 37(5): 131–139

Malucelli F, Nonato M, Pallottino S (1999). Demand adaptive systems: some proposals on flexible transit1. In Operational Research in Industry. London: Palgrave Macmillan UK

Millward H, Spinney J, Scott D (2013). Active-transport walking behavior: destinations, durations, distances. Journal of Transport Geography, 28: 101–110

Galarza Montenegro B D, Sörensen K, Vansteenwegen P (2021). A large neighborhood search algorithm to optimize a demand-responsive feeder service. Transportation Research Part C, Emerging Technologies, 127: 103102

Galarza Montenegro B D, Sörensen K, Vansteenwegen P (2022). A column generation algorithm for the demand-responsive feeder service with mandatory and optional, clustered bus-stops. Networks, 80(3): 274–296

Galarza Montenegro B D, Sörensen K, Vansteenwegen P (2023). A demand‐responsive feeder service with a maximum headway at mandatory stops. Networks, 83(1): 100–130

Nourbakhsh S M, Ouyang Y (2012). A structured flexible transit system for low demand areas. Transportation Research Part B: Methodological, 46(1): 204–216

Palmer K, Dessouky M, Abdelmaguid T (2004). Impacts of management practices and advanced technologies on demand responsive transit systems. Transportation Research Part A, Policy and Practice,

38(7): 495–509

Pang M, Chen M, Zhang N (2017). Scheduling optimization of intelligent public transport system based on MAST. Journal of Transportation Systems Engineering and Information Technology, 17(1): 143–163

Pei M, Lin P, Liu R, Ma Y (2019). Flexible transit routing model considering passengers' willingness to pay. IET Intelligent Transport Systems, 13(5): 841–850

Pisinger D, Ropke S (2007). A general heuristic for vehicle routing problems. Computers & Operations Research, 34(8): 2403–2435

Potts J F, Marshall M A, Crockett E C, Washington J (2010). A guide for planning and operating flexible public transportation services. Washington, D.C. Transportation Research Board

Qiu F, Li W, Haghani A (2014a). A methodology for choosing between fixed-route and flex-route policies for transit services. Journal of Advanced Transportation, 49(3): 496–509

Qiu F, Li W, Zhang J (2014b). A dynamic station strategy to improve the performance of flex-route transit services. Transportation Research Part C, Emerging Technologies, 48: 229–240

Quadrifoglio L, Dessouky M M, Ordóñez F (2008). Mobility allowance shuttle transit (MAST) services: MIP formulation and strengthening with logic constraints. European Journal of Operational Research, 185(2): 481–494

Quadrifoglio L, Dessouky M M, Palmer K (2007). An insertion heuristic for scheduling mobility allowance shuttle transit (MAST) services. Journal of Scheduling, 10(1): 25–40

Quadrifoglio L, Li X (2009). A methodology to derive the critical demand density for designing and operating feeder transit services. Transportation Research Part B: Methodological, 43(10): 922–935

Ropke S, Pisinger D (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science, 40(4): 455–472

Sacramento D, Pisinger D, Ropke S (2019). An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. Transportation Research Part C: Emerging Technologies, 102: 289–315

Sipetas C, Gonzales E J (2021). Continuous approximation model for hybrid flexible transit systems with low demand density. Transportation Research Record: Journal of the Transportation Research Board, 2675(8): 198–214

Tang C, Liu J, Ceder A, Jiang Y (2023). Optimisation of a new hybrid transit service with modular autonomous vehicles. Transportmetrica A: Transport Science, 20(2)

Yang H, Cherry C R, Zaretzki R, Ryerson M S, Liu X, Fu Z (2016). A GIS-based method to identify cost-effective routes for rural deviated fixed route transit. Journal of Advanced Transportation, 50(8): 1770–1784

Zhao J, Dessouky M (2008). Service capacity design problems for mobility allowance shuttle transit systems. Transportation Research Part B: Methodological, 42(2): 135–146

Zheng Y, Li W, Qiu F (2018a). A methodology for choosing between route deviation and point deviation policies for flexible transit services. Journal of Advanced Transportation, 2018: 1–12

Zheng Y, Li W, Qiu F (2018b). A slack arrival strategy to promote flex-route transit services. Transportation Research Part C, Emerging Technologies, 92: 442–455

Zheng Y, Li W, Qiu F, Wei H (2019). The benefits of introducing meeting points into flex-route transit services. Transportation Research Part C, Emerging Technologies, 106: 98–112