Delft University of Technology

The flexible airport bus and last-mile ride-sharing problem
Math-heuristic and metaheuristic approaches

He, Ping; Jin, Jian Gang; Schulte, Frederik

DOI
10.1016/j.tre.2024.103489

Publication date
2024

Document Version
Final published version

Published in
Transportation Research Part E: Logistics and Transportation Review

Citation (APA)
He, P., Jin, J. G., & Schulte, F. (2024). The flexible airport bus and last-mile ride-sharing problem: Math-heuristic and metaheuristic approaches. *Transportation Research Part E: Logistics and Transportation Review*, *184*, Article 103489. https://doi.org/10.1016/j.tre.2024.103489

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Transportation Research Part E 184 (2024) 103489

Contents lists available at ScienceDirect

Transportation Research Part E

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

The flexible airport bus and last-mile ride-sharing problem: Math-heuristic and metaheuristic approaches

Ping He [a], Jian Gang Jin [a,*], Frederik Schulte [b]

[a] *School of Ocean and Civil Engineering, and MOE Key Laboratory of Marine Intelligent Equipment and System, Shanghai Jiao Tong University, Shanghai, China*

[b] *Faculty of Mechanical Engineering, Delft University of Technology, Delft, The Netherlands*

A R T I C L E I N F O

Keywords:
Flexible airport bus
Last-mile ride-sharing
Routing
Double decomposition math-heuristic
Logic-based Benders decomposition
Column generation

A B S T R A C T

Airport buses play a crucial role in addressing the last-mile problem of air travel, especially in cities and countries lacking inner-city rail transit systems. Nevertheless, airport buses are currently witnessing a decline in ridership due to drawbacks such as long departure intervals, inflexible stops, and considerable distances between stops. Consequently, delivering high-quality airport bus services has become a pressing concern for public transport operators. Motivated by new flexible buses and ride-sharing services, this paper explores a flexible airport bus service that integrates ride-sharing services for passengers traveling from bus stops to their destinations. This problem entails integrated decisions involving bus stop selection, passenger assignment to drop-off bus stops, as well as bus and ride-sharing routing. Accordingly, this problem presents more challenges in decision-making than traditional flexible bus or ride-sharing routing problems. We first develop an arc-based mixed-integer linear programming model. Subsequently, we design a double decomposition math-heuristic algorithm that builds upon logic-based Benders decomposition and column generation algorithms to obtain a near-optimal solution within practical computation time limits for practical-scale instances. Additionally, we implement an adaptive large neighborhood search algorithm to evaluate the solution quality of this math-heuristic algorithm and to solve large-scale instances. To validate the effectiveness of both the model and the algorithms, we conduct numerical experiments using instances derived from Shenzhen airport bus lines. The experimental results demonstrate that the flexible service mode offers significant advantages in reducing both passenger ride time and vehicle mileage over traditional airport bus or taxi modes.

1. Introduction

Airport buses have the advantages of higher transport capacities, lower transport costs, and fewer transport emissions than travel modes such as taxis or private cars (Lu et al., 2016a; Chen et al., 2017; Hu et al., 2023; Liu et al., 2023a). This makes them a practical solution to the last-mile challenge in air travel, referring to the journey from the airport to the final destinations of passengers (Ma et al., 2023a). Public transit agencies in countries or cities with underdeveloped inner-city rail transit systems widely embrace airport buses to bridge the last-mile gap in air travel (Currie and Wallis, 2008; Chen et al., 2017; Ma et al., 2023a). Additionally, airport buses are indispensable during nighttime hours when metro and other rapid transit systems either cease operations or reduce departure frequencies. Fig. 1 illustrates three airport bus lines (NA1, A2, and A3) in Shenzhen, operated by Shenzhen Bus Group.

* Correspondence to: Dongchuan 800, Shanghai Jiao Tong University, Shanghai 200240, China.
E-mail address: jiangang.jin@sjtu.edu.cn (J.G. Jin).

https://doi.org/10.1016/j.tre.2024.103489
Received 6 December 2023; Received in revised form 22 February 2024; Accepted 8 March 2024
Available online 19 March 2024
1366-5545/© 2024 Elsevier Ltd. All rights reserved.

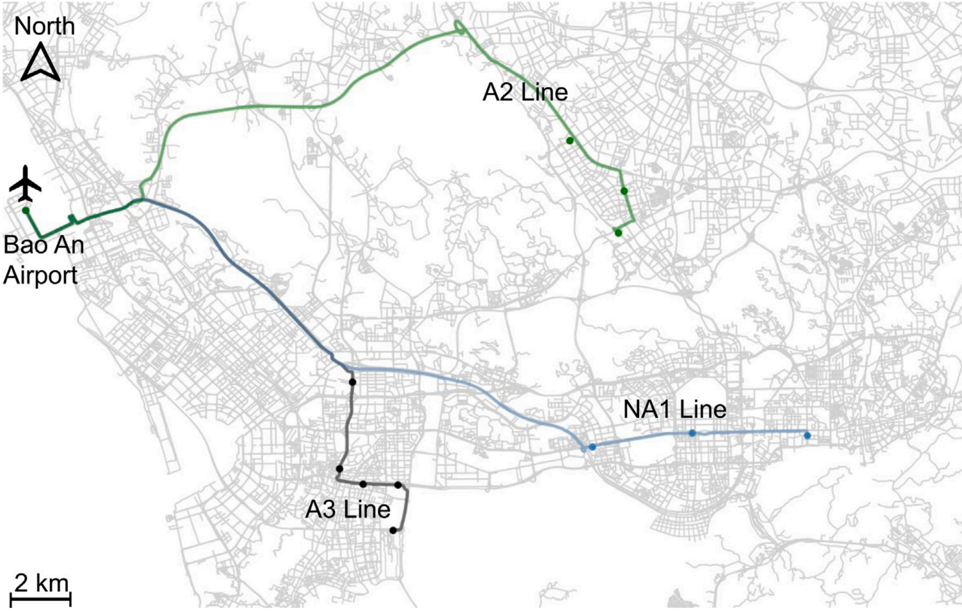

Fig. 1. Shenzhen airport bus lines NA1, A2, and A3.

These lines are primarily deployed to meet the travel needs of passengers, particularly those who are cost-sensitive, heading to or from the airport. This is especially beneficial during periods when the metro or alternative public transit options are not in operation.

However, the traditional operation mode of airport buses exhibits a low quality of service due to its characteristics of inflexible stops, routes, and departure frequencies. In particular, airport buses typically operate with a small number of stops, usually around five, and have long departure intervals, often ranging from 20 to 60 min. Furthermore, these bus lines are longer in length, typically exceeding 30 km. These factors contribute to long waiting times for airport bus services and the considerable distances between bus stops and passenger destinations, which can reach up to 5 km. Taking the NA1 line in Fig. 1 as an example, its departure interval is 30 min, the one-way distance from the airport to its last stop is about 40 km, and there are only three stops on the line. As a result, airport bus services frequently offer passengers clearly suboptimal experiences, leading to a decline in ridership (Akar, 2013).

To enhance the service quality of airport buses, operators such as Shenzhen Bus Group and Beijing Airport Bus Group are exploring the deployment of flexible or customized buses. These services aim to reduce passenger ride times, wait times, and last-mile travel distances from bus stops to final destinations by optimizing the bus departure time and route based on passenger reservation information, such as flights and destinations. However, this operational model still falls short of providing passengers with door-to-door transportation services. As a result, some passengers still face significant last-mile challenges between bus stops and their final destination, underscoring the need for further improvements in the quality of flexible airport bus services.

Ride-sharing, recently identified as a high-quality and cost-effective mode of transportation that not only reduces emissions and traffic congestion but also provides door-to-door travel services, is regarded as an effective solution to the first- and last-mile problems (Bian and Liu, 2019a; Chen et al., 2020; Bian et al., 2022; He et al., 2023). Motivated by these practices, in this work, we combine the flexible airport bus service with ride-sharing services from each bus stop to passenger destinations. This service mode aims to provide a high-quality and cost-effective last-mile service for air travelers by planning bus stops, the bus route, and ride-sharing routes based on the destinations of passengers. Additionally, to make this service attractive to business travelers in air travel, the model prioritizes shorter ride times for business passengers in the service design.

To the best of our knowledge, this is the first study investigating the integrated optimization problem involving flexible bus and ride-sharing services. Previous studies focused solely on either flexible buses or ride-sharing services for offering last-mile connections with intercity transit hubs. The flexible bus with ride-sharing services aims to provide cost-effective and high-quality solutions to this long-distance last-mile travel challenge from an intercity transit hub, such as an airport. As it involves integrated decisions on bus stop selection, request assignment, bus routing, ride-sharing matching and routing, this problem is more complex than related last-mile problems. To obtain high-quality solutions for this complex problem, we propose a novel math-heuristic algorithm that builds upon the framework of the logic-based Benders decomposition algorithm. In addition, we embed a column generation algorithm into the Benders algorithm to enhance the efficiency of the solution approach. As a result, we refer to this math-heuristic algorithm as a double decomposition algorithm.

Our contributions in this paper are outlined as follows:

(1) We define a novel integrated optimization problem concerning the flexible airport bus with last-mile ride-sharing services. This provides a new approach for addressing the last-mile travel challenges of intercity transit hubs and could be helpful in improving the service quality of airport buses, thus increasing their ridership.

(2) We formulate a new mixed-integer linear programming model for the proposed integrated optimization problem. The objective of this model is to minimize total operating costs by incorporating decisions, including designing a flexible bus route, assigning each request to a drop-off bus stop, and selecting the last-trip service – either ridesharing or walking – from the drop-off stop for each request. Consequently, the model provides a foundation for modeling related flexible bus routing problems with last-trip connection services from bus stops.
(3) We propose a novel double decomposition math-heuristic algorithm that builds upon the logic-based Benders decomposition algorithm and the column-generation algorithm, with analytical Benders cuts proposed to expedite this algorithm. This algorithm is capable of obtaining high-quality solutions within a reasonable amount of computation time. This is a general-purpose solution framework and may be extended to other flexible bus routing problems with last-trip services from bus stops, last-mile delivery with crowd-shipping problems (Macrina et al., 2020), and truck-and-trailer routing problems (Derigs et al., 2013).
(4) We develop an adaptive large neighborhood search (ALNS) algorithm to assess the solution quality of the math-heuristic algorithm. This algorithm is more efficient in computation time, so it can also be applied to quickly adjust the solution in response to changes in demand. For instance, in scenarios where some passengers do not show up at the airport pickup stop. Again, the ALNS can be applied to other flexible bus routing problems with larger scales.
(5) We conduct numerical experiments based on real-world cases from Shenzhen airport buses. The results demonstrate that the proposed service model is efficient in reducing total ride times for passengers and mileages traveled by vehicles. When compared to traditional airport bus or taxi services, the proposed model offers high-quality and cost-effective services, providing the first evidence for related future projects in the real world.

The remainder of this paper is organized as follows: Section 2 reviews related studies and positions our work. Section 3 elaborates on the investigated problem, including the problem definition and an arc-based mixed integer programming model. In Section 4, we introduce the double decomposition math-heuristic algorithm and the adaptive large neighborhood search algorithm. Section 5 describes the numerical experiments conducted. Conclusions and future research directions are summarized in Section 6.

2. Literature review

We review two streams of studies closely related to our work: the airport flexible bus services and the first- or last-mile ride-sharing services.

2.1. Airport flexible bus services

Flexible buses, characterized by adaptable stops and routes for each trip, are widely used to deliver demand-responsive and door-to-door services in low-density areas, providing a higher service quality compared to traditional buses (Kim and Schonfeld, 2014; Estrada et al., 2021; Lee et al., 2022). Consequently, numerous studies delved into optimizing the service of flexible or demand-responsive buses. For instance, Kong et al. (2018) proposed a two-stage method to generate dynamic bus routes for offering last-mile service towards metro stations. Their approach involves analyzing resident travel behaviors, predicting travel demands using crowdsourced shared bus data, and employing a dynamic programming algorithm to generate flexible and dynamic bus routes. Estrada et al. (2021) conducted cost and user performance analyses for on-demand flexible bus and taxi systems, revealing that flexible vehicle layouts are preferable in areas with low demand densities. Lu et al. (2016b) introduced a flexible feeder bus routing model with the objective of minimizing total bus travel time. They also developed a three-stage algorithm to find near-optimal solutions to this problem within a reasonable amount of computation time. In another study, Ma et al. (2023b) investigated a dynamic bus routing problem with stochastic passenger demand. They leveraged two-stage stochastic programming to model it and developed a clustering-based adaptive large neighborhood search algorithm to solve it. Additionally, Zhen et al. (2023) explored a customized on-demand bus routing problem and developed a column generation algorithm to identify high-quality solutions.

Current studies on optimizing flexible bus service primarily focus on conventional bus lines, with limited attention given to special scenarios, such as airport buses. Airport buses exhibit unique characteristics, serving a wider area with relatively low passenger travel demand, resulting in a low demand density within the service area of airport bus lines. Consequently, the flexible bus operation mode is deemed suitable for airport buses. For instance, Chen et al. (2017) proposed a dynamic programming algorithm and an artificial bee colony method to determine an optimal flexible bus route for suburban passengers accessing the airport, with the objective of minimizing the total travel time. Yu et al. (2020) introduced a dynamic airport bus line generation and vehicle scheduling method using multi-source transportation data to enhance the station coverage and passenger demand compatibility of flexible airport buses. Ma et al. (2023a) proposed an improved clustering method to identify fixed stops and a batch response strategy to select demand-responsive stops for matching dynamic requests.

However, flexible airport buses have longer route distances and fewer stops than conventional flexible buses. This distinction could potentially result in a situation where drop-off stops are distant from passenger destinations, leading to passenger dissatisfaction with airport bus services. As a result, bridging this last-mile gap between bus stops and passenger destinations is crucial for improving the overall service quality of airport buses.

2.2. First-/last-mile ride-sharing services

Long travel distances and a lack of transit connections between the origins or destinations of passengers and public transit stations, such as bus, metro, and train stations, as well as airports, are known as first-mile and last-mile challenges (Wang and

Odoni, 2016; Wang, 2019; Chen et al., 2020; Abe, 2021). In recent years, emerging travel modes, such as ride-sharing and ride-pooling, have been introduced to address these challenges by providing high-quality and cost-effective transportation connections to and from transit stations (Shaheen and Chan, 2016; Stiglic et al., 2018; Bian and Liu, 2019a,b; Ma et al., 2019; Chen et al., 2020; He et al., 2023). As a result, an increasing number of studies are focusing on the design and optimization of first- and last-mile ride-sharing services. The overarching goal of these studies is to enable riders to connect with transit systems in a more efficient, convenient, and cost-effective manner. For instance, Stiglic et al. (2018) investigated an integrated system of ride-sharing and public transit systems, finding that it is beneficial for addressing first- and last-mile travel issues as well as increasing the usage of public transport. Bian and Liu (2019a,b) proposed a novel mechanism for first-mile ride-sharing services and a solution pooling approach for it. Ma et al. (2019) introduced a ride-sharing strategy for on-demand mobility service operators to provide first- and last-mile services. Chen et al. (2020) explored the application of autonomous vehicles to address the first-mile ride-sharing problem and developed a cluster-based heuristic algorithm for it.

Current first-mile or last-mile ride-sharing services typically connect stops on traditional transit lines or routes with fixed stops and schedules. Consequently, they cannot simultaneously optimize feeder and trunk line transportation to enhance service quality and transport efficiency. This means that the integrated optimization of public transit lines and first-mile or last-mile ride-sharing services has not been thoroughly investigated. However, in the context of designing flexible airport bus services, there is an opportunity to integrate and optimize the bus route with last-mile ride-sharing services. This integrated optimization has the potential to provide more efficient travel services, lower operating costs for transit agencies, and improve the quality of service for riders.

In summary, numerous studies have explored flexible bus services and last-mile ride-sharing services from transit stations. However, as of yet, there has been a lack of exploration into the integrated optimization of airport flexible bus and last-mile ride-sharing services aimed at providing high-quality and cost-effective door-to-door services for intercity airline travelers. This study seeks to fill this gap in existing research.

3. Flexible airport bus with last-mile ride-sharing services

In this section, we first introduce the flexible airport bus problem integrated with last-mile ride-sharing services. Subsequently, we formulate a mixed-integer linear programming model for it.

3.1. Problem description

Suppose a public transit agency operates a flexible airport bus line within a designated service area with the goal of providing high-quality transportation services for passengers traveling from the airport to their homes, offices, or other destinations, as illustrated in Fig. 2. After arrival at the airport, passengers whose destinations fall within this service area can reserve the flexible bus service by providing the location of destinations, passenger numbers, and the earliest times they can reach the bus pickup point. Upon receiving a sufficient number of requests, the agency assigns requests with similar arrival times at the pickup point to a bus and then makes the following decisions: assigning each request to a drop-off bus stop, designing a flexible bus route, and arranging last-mile ride-sharing services for riders at the same drop-off stop.

The primary objective of the transit agency is to minimize the overall operating costs associated with delivering this flexible service. Additionally, for some passengers, the walking distance from the assigned drop-off stop to their final destination is acceptable, referred to as a comfortable walking distance (ξ). In such cases, the operator can decide whether or not to provide ride-sharing services for these passengers based on its operational objectives. To enhance service quality, transport service operators can employ the value-of-time coefficient to translate passenger trip time into a cost term. This allows for the reduction of passenger travel time as much as possible with the objective of minimizing total costs. Moreover, to make this service attractive to business travelers seeking shorter ride times or greater comfort, the agency can prioritize shorter ride times for business passengers in the service design.

The transit agency can pre-determine bus departure times based on passenger reservation information, subsequently notifying passengers accordingly. As a result, passengers only need to board the bus before the designated departure time, which eliminates the need to factor in passenger waiting times at the airport. Furthermore, we assume that the operator can schedule an adequate number of cars, such as self-owned cars or taxis, in accordance with established ride-sharing schemes before the bus arrives at each stop. With cars capable of arriving ahead of the bus, passengers utilizing ride-sharing services have no waiting times at bus stops. Consequently, the total ride time for each passenger consists of only two components: the time spent on the first leg (i.e., time taking the bus), and the time spent on the second leg of the journey, either by taking a taxi or walking.

To facilitate understanding of this problem, we present an illustrative example involving 14 requests and 5 available bus stops within the coverage area of the bus line, as depicted in Fig. 2. The bus departs from the airport stop, makes stops at S1, S3, and S5, and then returns to the airport. At stop S1, requests 1, 2, 3, 4, 5, and 6 disembark from the bus. Requests 2, 3, and 4 are transported to their respective destinations by car 1 via ride-sharing services, while requests 5 and 6 are assigned to car 2. Given the proximity of S1 to the final destination of request 1 (R1) and the cost-effectiveness of providing ride-sharing services for R1, R1 reaches the destination on foot. Similarly, requests 7, 8, 9, and 10 alight from the bus at stop S3. R9 and R10 are subsequently offered ride-sharing services via car 3 to reach their final destinations. Requests 11, 12, 13, and 14 get off at stop S5, and they head to their final destination via ride-sharing services provided by car 4. It is worth noting that, despite the fact that S5 is close to the final destination of R11, ride-sharing is the preferred option in this scenario due to its ability to reduce total costs.

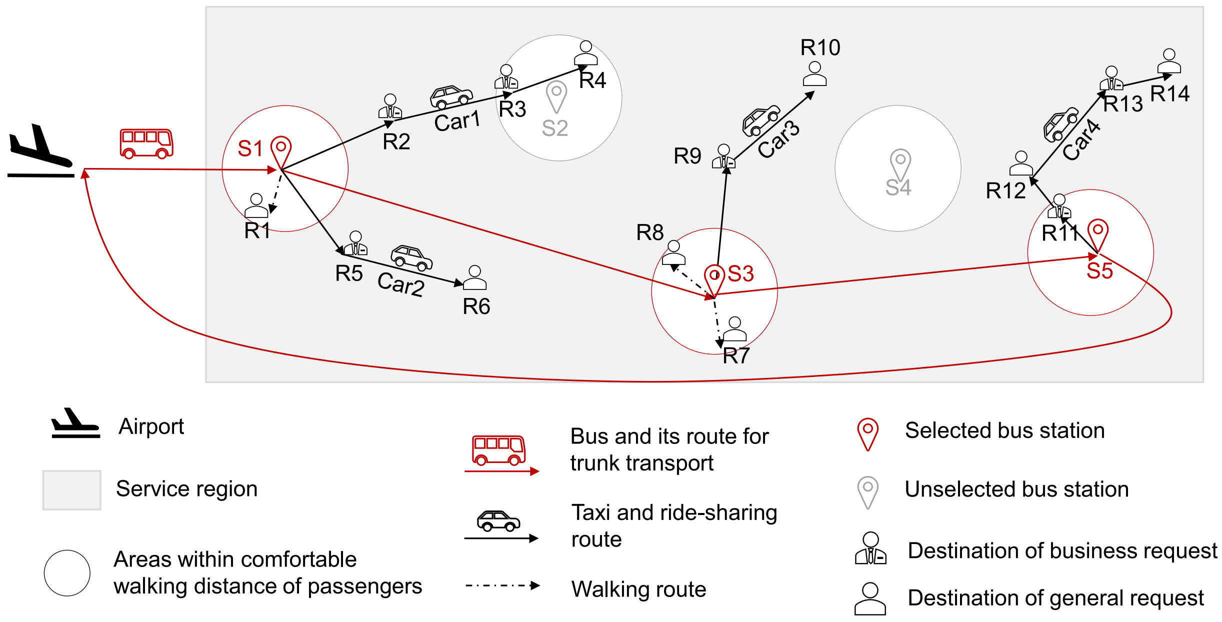

Fig. 2. An illustrative example of the flexible airport bus with last-mile ride-sharing service.

Table 1
Sets, parameters, and variables.

Notations	Descriptions
Sets	
$\mathcal{N}_C^b$	Set of business requests
$\mathcal{N}_C^g$	Set of general requests
$\mathcal{N}_C$	Set of total requests, $\mathcal{N}_C = \mathcal{N}_C^b \cup \mathcal{N}_C^g$
$\mathcal{N}_S$	Set of drop-off bus stations
$\mathcal{N}_S^+$	Set of drop-off bus stops and the hub stop at the airport, $\mathcal{N}_S^+ = \mathcal{N}_S \cup \{0\}$
Parameters	
n_d	A dummy sink node for each car route
d_{ij}	Distance between nodes i and j
v_{bus}/v_{car}	Bus or car travel speed (km/h)
v_{walk}	Walk speed (km/h)
c_{bus}/c_{car}	Unit travel cost of the bus or car ($/km)
s_{bus}/s_{car}	Service time at each bus or car stop for dropping off passengers (min)
c_{fix}	Fixed cost for each scheduled car ($)
ξ	Comfortable walking distance (m)
q	Number of seats in each car
q_m	Number of passengers for request m
p_b/p_g	Value-of-time cost for each business or general request ($/min)
a_0	Scheduled bus departure time
M	A sufficiently large positive constant
Variables	
$x_{ij} \in \{0,1\}$	1 if the bus traverses arc(i,j), $\forall i,j \in \mathcal{N}_S^+$; otherwise, 0.
$y_{mn}^s \in \{0,1\}$	1 if a car departing from stop s traverses arc(m,n), $\forall m,n \in \mathcal{N}_S \cup \mathcal{N}_C$; otherwise, 0.
$w_m^s \in \{0,1\}$	1 if request m gets off the bus at bus stop s; otherwise, 0.
$z_m^s \in \{0,1\}$	1 if request m gets off at stop s and walks to its destination; otherwise, 0.
$\pi_m \in \mathbb{Z}^+$	Cumulative number of passengers dropped off before the car leaves request node m.
$a_i \geq 0$	Arrival time at node i, $i \in \mathcal{N}_S \cup \mathcal{N}_C$.
$\Delta_m \geq 0$	Ride time for request m.

3.2. Notations

Let $\mathcal{N}$ denote the set of all nodes, consisting of a bus stop at the airport, denoted as 0, flexible bus stops $\mathcal{N}_S$, destinations of requests $\mathcal{N}_C$, and a dummy node to denote the final destination for each car. Let $\mathcal{A}$ denote the set of arcs connecting the nodes in $\mathcal{N}$, and $\mathcal{A} = \{(i,j)|i,j \in \mathcal{N}, i \neq j\}$. Then, this problem can be defined on a directed graph $G = (\mathcal{N}, \mathcal{A})$. Other parameters and variables for defining the model can be found in Table 1.

3.3. Arc-based MILP model

With these notations, we can formulate this problem as an arc-based mixed-integer linear programming model.

$$min \sum_{i\in\mathcal{N}_S^+}\sum_{j\in\mathcal{N}_S^+} c_{bus}d_{ij}x_{ij} + \sum_{s\in\mathcal{N}_S}\sum_{m\in\mathcal{N}_S\cup\mathcal{N}_C}\sum_{n\in\mathcal{N}_C} c_{car}d_{mn}y^s_{mn} + \sum_{s\in\mathcal{N}_S}\sum_{n\in\mathcal{N}_C} c_{fix}y^s_{sn} + \sum_{m\in\mathcal{N}_C^b} p_b\Delta_m + \sum_{m\in\mathcal{N}_C^g} p_g\Delta_m \tag{1}$$

subject to:

$$\sum_{j\in\mathcal{N}_S^+\setminus\{i\}} x_{ij} = \sum_{j\in\mathcal{N}_S^+\setminus\{i\}} x_{ji} \le 1, \forall i \in \mathcal{N}_S^+ \tag{2}$$

$$\sum_{s\in\mathcal{N}_S} w^s_m = 1, \forall m \in \mathcal{N}_C \tag{3}$$

$$w^s_m \le \sum_{j\in\mathcal{N}_S^+} x_{sj}, \forall m \in \mathcal{N}_C, \forall s \in \mathcal{N}_S \tag{4}$$

$$a_s \ge a_0 + d_{0s}/v_{bus} - M(1 - x_{0s}), \forall s \in \mathcal{N}_S \tag{5}$$

$$a_s \ge a_z + d_{zs}/v_{bus} - M(1 - x_{zs}), \forall z, s \in \mathcal{N}_S \tag{6}$$

$$\sum_{n\in\mathcal{N}_C\cup\{n_d\}} y^s_{mn} = y^s_{sm} + \sum_{n\in\mathcal{N}_C} y^s_{nm} \le w^s_m, \forall s \in \mathcal{N}_S, \forall m \in \mathcal{N}_C \tag{7}$$

$$\sum_{s\in\mathcal{N}_S} (\sum_{n\in\mathcal{N}_C\cup\{n_d\}} y^s_{mn} + z^s_m) = 1, \forall m \in \mathcal{N}_C \tag{8}$$

$$z^s_m \le w^s_m, \forall s \in \mathcal{N}_S, \forall m \in \mathcal{N}_C \tag{9}$$

$$z^s_m(d_{sm} - \xi) \le 0, \forall s \in \mathcal{N}_S, \forall m \in \mathcal{N}_C \tag{10}$$

$$\pi_n \ge q_n + \pi_m - M(1 - \sum_{s\in\mathcal{N}_S} y^s_{mn}), \forall m \in \mathcal{N}_C, n \in \mathcal{N}_C \cup \{n_d\} \tag{11}$$

$$q_n \le \pi_n \le q, \forall n \in \mathcal{N}_C \tag{12}$$

$$a_n \ge a_s + d_{sn}/v_{car} + s_{bus} - M(2 - y^s_{sn} - w^s_n), \forall n \in \mathcal{N}_C, \forall s \in \mathcal{N}_S \tag{13}$$

$$a_n \ge a_s + d_{sn}/v_{walk} + s_{bus} - M(2 - z^s_m - w^s_n), \forall n \in \mathcal{N}_C, \forall s \in \mathcal{N}_S \tag{14}$$

$$a_n \ge a_m + d_{mn}/v_{car} + s_{car} - M(1 - \sum_{s\in\mathcal{N}_S} y^s_{mn}), \forall m, n \in \mathcal{N}_C \tag{15}$$

$$\Delta_n \ge a_n - a_0, \forall n \in \mathcal{N}_C \tag{16}$$

$$x_{ij}, y^s_{mn}, w^s_m, z^s_m \in \{0, 1\}; \pi_m \in \mathbb{Z}^+; a_i, \Delta_m \ge 0 \tag{17}$$

The objective function (1) aims to minimize total costs for providing flexible airport bus and last-mile ride-sharing services. These costs encompass transportation expenses for the bus and cars, fixed costs for scheduled cars, and ride time costs for each business and general request. To ensure a higher level of service quality for business requests, the value-of-time cost coefficient associated with business requests is assigned a higher value than that of general requests.

Constraints (2) determine whether a bus stop will be visited and ensure that the inflow and outflow of each bus stop are conserved. Constraints (3) guarantee that each request is matched with a bus stop. Constraints (4) impose that a request can be dropped off at a stop only if the stop is on the bus route. Constraints (5) and (6) denote the arrival time at each bus stop. Constraints (7) ensure that the inflow and outflow are conserved for a request if it is provided with a last-mile ride-sharing service at a bus stop. Constraints (8) denote that the last-mile trip from the bus stop to the destination of each request is either serviced by ride-sharing or matched with walking. Constraints (9) and (10) enforce that a request can walk from the bus stop to its final destination only if it is assigned to this bus stop and the distance is within the comfortable walking distance. Constraints (11) represent the count of passengers dropped off before the car leaves a node. Constraints (12) denote that the cumulative number of dropped-off passengers is greater than the passenger number of each request and meets the vehicle capacity constraint. Constraints (13)–(15) determine the arrival time of each request at its final destination. Constraints (16) identify the ride time for each request. Constraints (17) are the domains of variables.

4. Solution methodology

This problem comprises two main components: a bus routing and request assignment problem and a last-mile ride-sharing service design problem, which is similar to the open vehicle routing problem. It is worth noting that the routing problem has been established as NP-hard (Toth and Vigo, 2014). Consequently, our problem can also be classified as NP-hard, as both the bus routing and last-mile ride-sharing problems fall into the routing problem category. Given the problem characteristics, we propose a double-decomposition math-heuristic algorithm, aiming to obtain high-quality solutions. This algorithm builds upon a logic-based Benders decomposition algorithm and a column-generation algorithm. The column generation focuses on solving the last-mile service design problem and is embedded in the logic-based Benders decomposition algorithm framework. The logic-based Benders algorithm addresses the bus routing and request assignment based on the information provided by the last-mile ride-sharing sub-problem. To assess the solution quality of the math-heuristic algorithm, we develop an adaptive large neighborhood search (ALNS) algorithm. Sections 4.1 and 4.2 will introduce and discuss these two algorithms in depth.

4.1. Math-heuristic algorithm

The proposed math-heuristic is implemented in the logic-based decomposition framework and decomposes the original MILP model into a master problem (MP) and a series of slave problems (SPs). The MP aims to select bus stops, determine the bus route, and assign requests to selected bus stops. Meanwhile, each slave problem (SP), corresponding to a selected bus stop, focuses on identifying the best last-mile services for requests assigned to that stop. After solving the SP of each selected stop with the column generation algorithm, the logic-based Benders cuts are generated and incorporated into the MP model in subsequent iterations. This iterative procedure repeats until a predefined stopping criterion is met. In the subsequent subsections, we will present detailed explanations of the MP, SP, and the generation of Benders cuts, respectively.

4.1.1. Master problem

The MP serves as a relaxation model of the original MILP model, focusing solely on bus routing and request assignment without considering last-mile services. In this context, we introduce two key variables: Δ'_m and c_s. Δ'_m represents the ride time of request m for taking the bus trip, and c_s denotes the total costs incurred for providing the last-mile services for requests assigned to stop s. Then, the MP can be formulated as follows:

$$[\boldsymbol{MP}]\min \sum_{i\in\mathcal{N}_S^+}\sum_{j\in\mathcal{N}_S^+} c_{bus}d_{ij}x_{ij} + \sum_{m\in\mathcal{N}_C^b} p_b\Delta'_m + \sum_{m\in\mathcal{N}_C^g} p_g\Delta'_m + \sum_{s\in\mathcal{N}_S} c_s \tag{18}$$

subject to:

$$(2)\text{–}(6) \tag{19}$$

$$\Delta'_m \geq a_s - a_0 - M\left(1 - w_m^s\right) + s_{bus}, \forall m \in \mathcal{N}_C, \forall s \in \mathcal{N}_S \tag{20}$$

$$c_s \geq \text{ Optimality Cuts}, \forall s \in \mathcal{N}_S \tag{21}$$

The objective function (18) minimizes total costs, which consist of bus transportation costs, ride time costs for requests to take the bus, and last-mile service costs for each request. Constraints (20) identify the ride time for each passenger to take the bus. Constraints (21) are the optimality cuts, which are generated by solving the SPs and then integrated into the MP in an iterative manner. The incorporation of these cuts aims to enhance the lower bound estimation for c_s.

4.1.2. Strengthen the MP

The lower bound of c_s in the MP model is relatively weak, which can negatively impact the computational efficiency of the MP. To strengthen the lower bound, we propose two valid inequalities. Let $\mathcal{N}'_C$ denote those requests that are within a comfortable walking distance from stop s.

$$c_s \geq \sum_{m\in\mathcal{N}_C} p_m\frac{d_{sm}}{v_{car}}w_m^s + c_{fix}\left\lceil \sum_{m\in\mathcal{N}_C\setminus\mathcal{N}'_C}\frac{w_m^s q_m}{q}\right\rceil + c_{car}\max_{m\in\mathcal{N}_C}\left\{w_m^s d_{sm}\right\}, \forall s \in \mathcal{N}_S \tag{22}$$

$$c_s \geq \sum_{m\in\mathcal{N}_C} p_m\frac{d_{sm}}{v_{car}}w_m^s + c_{fix}\left\lceil \sum_{m\in\mathcal{N}_C\setminus\mathcal{N}'_C}\frac{w_m^s q_m}{q}\right\rceil + c_{car}\left\lceil \sum_{m\in\mathcal{N}_C\setminus\mathcal{N}'_C}\frac{w_m^s q_m}{q}\right\rceil\min_{m\in\mathcal{N}_C}\left\{d_{sm}\right\}, \forall s \in \mathcal{N}_S \tag{23}$$

At each stop, the total cost associated with passenger last-mile trips comprises three components: travel time costs for each request, fixed costs for scheduled cars, and transportation costs. Consequently, we can establish a lower bound for c_s by determining a lower bound value for each of these components.

To obtain a lower bound of travel time costs, we can multiply the minimum travel time of each request by the corresponding value-of-time cost factor, denoted as $\sum_{m\in\mathcal{N}_C} p_m\frac{d_{sm}}{v_{car}}w_m^s$ in inequalities (22) and (23).

Regarding the lower bound on total fixed costs, we can multiply the fixed cost for each car by the minimum number of cars required. To determine this vehicle number, we first identify the total number of passengers who are not within comfortable walking distance from the bus stop to their respective destinations. This passenger number is then divided by the car capacity to ascertain the minimum number of cars needed.

Since the vehicles are homogeneous and have the same transportation cost per kilometer, we can determine a lower bound of transportation cost by multiplying the lower bound of transportation distance by the unit transportation cost. Travel distances are known to adhere to the triangular inequality. For instance, the distance between nodes a and b is less than the sum of the distance between nodes a and c and the distance between nodes c and d. Thus, we can use the farthest distance between stop s and request nodes, among the requests assigned to stop s, as a lower bound for the total transportation distance of cars.

Additionally, based on the requests assigned to stop s, we can estimate a lower bound on the number of cars required to serve them. Multiplying the shortest transportation distance between stop s and requests by the lower bound on the number of vehicles provides another lower bound on the total transportation distance. Therefore, we can obtain two lower bounds for transportation distances, the last part in inequalities (22) and (23), respectively.

4.1.3. Slave problem

After solving the MP model, which provides a bus routing and requests assignment scheme based on the values of variables x_{ij}, a_s, w_m^s, and $\Delta'_m \left(i, j \in \mathcal{N}_S^+, s \in \mathcal{N}_S\right.$, and $\left. m \in \mathcal{N}_C\right)$, we can proceed to tackle the slave problem (SP). The SP, associated with each selected bus stop and requests assigned to it, is a variant of the open vehicle routing problem. Here, we define a subset $\mathcal{N}_C^s$ representing the requests assigned to stop s, $\mathcal{N}_C^s = \left\{m \mid w_m^s = 1, \forall m \in \mathcal{N}_C\right\}$. Let binary variable $y_{mn} \left(\forall m, n \in \mathcal{N}_C^s\right)$ denote whether a car travels from request m to request n. Specifically, y_{mn} takes the value 1 if a car traverses from request m to request n. Additionally, we define a binary variable $z_m \left(\forall m \in \mathcal{N}_C^s\right)$, which takes the value 1 if request m is required to walk to its final destination. Let $\Delta''_m \left(\forall m \in \mathcal{N}_C^s\right)$ represent the travel time from stop s to the destination of request m. With these definitions, the sub-problem for bus stop $s\,(SP(s))$ can be formulated as follows:

$$[\boldsymbol{SP(s)}] \quad \min \sum_{m \in \mathcal{N}_C^s \cup \{s\}} \sum_{n \in \mathcal{N}_C^s} c_{car} d_{mn} y_{mn} + \sum_{n \in \mathcal{N}_C^s} c_{fix} y_{sn} + \sum_{m \in \mathcal{N}_C^s \cap \mathcal{N}_C^b} p_b \Delta''_m + \sum_{m \in \mathcal{N}_C^s \cap \mathcal{N}_C^g} p_g \Delta''_m \tag{24}$$

subject to:

$$\sum_{n \in \mathcal{N}_C^s \cup \{n_d\}} y_{mn} + z_m = 1, \forall m \in \mathcal{N}_C^s \tag{25}$$

$$\sum_{n \in \mathcal{N}_C^s \cup \{n_d\}} y_{mn} = \sum_{n \in \mathcal{N}_C^s} y_{nm} + y_{sm} \leq 1, \forall m \in \mathcal{N}_C^s \tag{26}$$

$$z_m \left(d_{sm} - \xi\right) \leq 0, \forall m \in \mathcal{N}_C^s \tag{27}$$

$$\pi_n \geq q_n + \pi_m - M\left(1 - y_{mn}\right), \forall m \in \mathcal{N}_C^s, \forall n \in \mathcal{N}_C^s \cup \left\{n_d\right\} \tag{28}$$

$$q_n \leq \pi_n \leq q, \forall n \in \mathcal{N}_C^s \tag{29}$$

$$a_n \geq a_s + d_{sn}/v_{car} + s_{bus} - M\left(1 - y_{sn}\right), \forall n \in \mathcal{N}_C^s \tag{30}$$

$$a_n \geq a_s + d_{sn}/v_{walk} + s_{bus} - M\left(1 - z_m\right), \forall n \in \mathcal{N}_C^s \tag{31}$$

$$a_n \geq a_m + d_{mn}/v_{car} + s_{car} - M\left(1 - y_{mn}\right), \forall m, n \in \mathcal{N}_C^s \tag{32}$$

$$\Delta''_m \geq a_m - a_s - s_{bus}, \forall m \in \mathcal{N}_C^s \tag{33}$$

The objective function (24) aims to minimize total costs associated with the last-mile trip for requests assigned to stop s. These costs consist of transportation costs, fixed costs associated with scheduled cars, and ride time costs for each business and general request.

Constraints (25) denote that the last-mile trip for each request can be either serviced by a car or by walking. Constraints (26) ensure the conservation of inflow and outflow for a request if it is provided with a last-mile ride-sharing service. Constraints (27) enforce that a request can walk from the bus stop to its final destination only if the distance is within comfortable walking distance. Constraints (28) represent the total number of passengers dropped off before the car leaves a request node. Constraints (29) denote that the cumulative number of dropped-off passengers is greater than the passenger number of each request and satisfies the vehicle capacity constraint. The arrival time of each request at the final destination is determined by constraints (30)–(32). Constraints (33) identify the travel time of the last-mile trip for each request.

The sub-problem requires repeated solving and is NP-hard. As a result, obtaining high-quality solutions quickly for practical instances is crucial. To achieve this goal, we resort to the column-generation algorithm. This choice is motivated by the fact that the sub-problem shares similarities with the open vehicle routing problem, and the column-generation algorithm is well-known for quickly identifying near-optimal solutions to routing-related problems. The details regarding the column-generation procedure can be found in Appendix.

4.1.4. Analytical benders cut

After solving the $SP(s)$, we can obtain the total costs for stop s, denoted by c_s^*. Then, we can derive generic Benders optimality cuts, as shown in inequality (34).

$$c_s \geq c_s^* \left(\sum_{m \in \mathcal{N}_C^s} w_m^s - \left| \mathcal{N}_C^s \right| + 1 \right), \forall s \in \mathcal{N}_S \tag{34}$$

Cuts (34) indicate that if the same set of requests $\mathcal{N}_C^s$ is assigned to stop s in subsequent iterations, the total cost for stop s is at least c_s^*. Note that if the element in set $\mathcal{N}_C^s$ changes in the subsequent iterations, cuts (34) become non-binding. As a result, cuts (34) are termed no-good cuts due to their limited effectiveness in improving the lower bound for c_s. This limitation stems from the fact that the cut only incorporates cost information about requests assigned to s in the incumbent solution. As such, we present analytical Benders cuts that provide both total cost information for the incumbent solution and approximate cost information for solutions related to this solution.

$$c_s \geq c_s^* + \sum_{m \in \mathcal{N}_C \setminus \mathcal{N}_C^s} \Theta_m w_m^s - \sum_{m \in \mathcal{N}_C^s} \Xi_m \left(1 - w_m^s\right), \forall s \in \mathcal{N}_S \tag{35}$$

$$\Theta_m = \min\{A, B, C\} \tag{36}$$

$$A = \left(\frac{\min_{n \in \mathcal{N}_C^s} \{d_{mn}\}}{v_{car}} + 2s_{car} \right) p_m \tag{37}$$

$$B = c_{fix} + d_{sm} c_{car} + \frac{d_{sm}}{v_{car}} p_m \tag{38}$$

$$C = \begin{cases} \frac{d_{sm}}{v_{walk}} p_m, & if\ d_{sm} < \xi \\ M, & otherwise \end{cases} \tag{39}$$

$$\Xi_m = \max\{D, E, F\} \tag{40}$$

$$D = 2 \max_{n \in \mathcal{N}_C^s} \{d_{mn}\} c_{car} + \max_{n \in \mathcal{N}_C^s} \left\{ \frac{d_{mn}}{v_{car}} \right\} p_m + 2\left(q - q_m\right) \left(\max_{n \in \mathcal{N}_C^S} \left\{ \frac{d_{mn}}{v_{car}} \right\} + s_{car} \right) p_b \tag{41}$$

$$E = c_{fix} + d_{sm} c_{car} + \frac{d_{sm}}{v_{car}} p_m \tag{42}$$

$$F = \begin{cases} \frac{d_{sm}}{v_{walk}} p_m, & if\ d_{sm} < \xi \\ 0, & otherwise \end{cases} \tag{43}$$

Cuts (35) are analytical Benders cuts, which means that the total cost for stop s is at least c_s^* if the same set of requests $\mathcal{N}_C^s$ is assigned to stop s in the subsequent iterations. In cuts (35), Θ_m represents the minimum increase in total costs if any request m outside the set $\mathcal{N}_C^s$ is assigned to the stop s during subsequent iterations, and Ξ_m denotes the maximum decrease in total costs if any request m within the set $\mathcal{N}_C^s$ is assigned to another stop during subsequent iterations.

Θ_m is the minimum cost increase among three travel modes for request m from the stop to the destination. These modes include: request m sharing the ride with other requests in set $\mathcal{N}_C^s$, m not sharing the ride with other passengers, and m walking from stop s to the final destination. "A" represents the minimum increase in ride time costs when request m completes the last-mile trip in ride-sharing mode with passengers in set $\mathcal{N}_C^s$. In this mode, when request m shares the trip with the nearest request n in set $\mathcal{N}_C^s$, request m is delivered before request n, and the shared trip does not change the route of request n, the total cost increase is minimized. "B" signifies the minimum cost increase if request m does not share the ride with other requests. This includes vehicle fixed costs, transportation costs, and ride time costs. "C" denotes the minimum cost increase when request m walks from stop s to the final destination.

Similarly, Ξ_m is the maximum cost decrease among the three aforementioned travel modes for request m. "D" represents the maximum cost reduction achieved by removing request m from the ride-sharing route if it shares the ride with other passengers. Here, $max_{n \in \mathcal{N}_C^s}\{d_{mn}\}$ signifies the maximum distance between request m and the other requests in set $\mathcal{N}_C^s$. Subsequently, we denote $2\max_{n \in \mathcal{N}_C^s}\{d_{mn}\} c_{car}$ as the maximum reduction in transportation costs after removing request m from the ride-sharing route, $\max_{n \in \mathcal{N}_C^s}\left\{\frac{d_{mn}}{v_{car}}\right\} p_m$ as the maximum reduction in travel time costs for request m, and $2\left(q - q_m\right)\left(max_{n \in \mathcal{N}_C^S}\left\{\frac{d_{mn}}{v_{car}}\right\} + s_{car}\right) p_b$ as the maximum reduction in travel time costs for other passengers sharing the ride with request m. The meanings of "E" and "F" are similar to "B" and "C", respectively.

4.1.5. Algorithm enhancement

The linear relaxation of the MP model is weak, making it challenging to obtain the optimal solution with solvers like CPLEX or GUROBI, primarily due to the inclusion of big-M constraints (5) and (6). To mitigate this challenge, one approach is to remove these two constraints from the MP model, solve its relaxed version directly, and then check whether these constraints are met. However, this strategy may result in the generation of sub-tour routes, as constraints (5) and (6), which prevent sub-tours, have been relaxed. To address the potential issue of sub-tours, we incorporate the following sub-tour elimination cut into the MP model whenever a sub-tour is identified. The model is then re-solved, and the process is repeated until no sub-tour is found. In addition, to enhance the efficiency of incorporating this cut, we utilize the callback function, specifically the "lazy constraint callback", in off-the-shelf solvers like CPLEX.

$$\sum_{(i,j)\in ST} (1 - x_{ij}) \geq 1 \tag{44}$$

Here, ST represents the set of arcs comprising a sub-tour.

4.1.6. Outline of the math-heuristic algorithm

The math-heuristic algorithm, as outlined in Algorithm 1, is implemented in the logic-based Benders decomposition framework and utilizes the *branch-and-check* technique. The *branch-and-check* approach is an advanced technique for implementing the logic-based Benders decomposition algorithm (Tran et al., 2016; Li et al., 2023). This approach employs a single *branch-and-bound* search tree, in which Benders cuts are generated upon finding an integer feasible solution, and these cuts are added to the nodes that remain unfathomed. Specifically, at each tree node where an integer solution to the MP is identified, the algorithm scrutinizes for sub-tours. If sub-tours are present, sub-tour cuts are added to resolve it. Upon identifying an integer solution without sub-tours, the corresponding SP for each stop is solved using the column generation algorithm. Then, the SP solution is leveraged to generate analytical Benders cuts, which are incorporated into the unfathomed tree nodes. This iterative process continues until an optimal solution is identified or a predefined time limit is reached. The time limit for this procedure is defined as the time interval between identifying two consecutive integer solutions, set to 100 s.

Algorithm 1: Math-heuristic algorithm

1 Formulate the MP presented in Section 4.1.1;
2 Strengthen the MP with inequalities introduced in Section 4.1.2;
//Solve the MP using the branch-and-cut algorithm within a branch-and-bound (BB) search tree:
3 **while** *unexplored BB nodes exist, and the time limit is not exceeded* **do**
4   Solve the MP of a BB node;
5   **if** *an integer solution for the MP is identified* **then**
6   Verify if the solution includes sub-tours by utilizing the "callback" function of an off-the-shelf solver, such as CPLEX;
7   **if** *the solution contains sub-tours* **then**
            Add sub-tour elimination cuts (44) in the current BB node, then re-solve it, and **go to** Step 5;
8   **else**
9   Solve $SP(s)$ for each bus stop via the column generation algorithm;
10   Generate Benders cuts and incorporate them into unexplored tree nodes, and **go to** Step 4;
11 Output the best solution.

4.2. ALNS algorithm

To evaluate the solution quality of the math-heuristic method, this subsection introduces an adaptive large neighborhood search algorithm for the studied problem. The core idea of ALNS involves iteratively improving solutions by applying destroy and repair operators. The destroy operators break down a portion of the solution, while the repair operators aim to rebuild the destroyed solutions. During each iteration, a destroy and a repair operator are randomly selected to generate new solutions until the stopping criterion is met. The selection of operators is based on their historical success in improving solution quality or finding new high-quality solutions. A new solution is accepted according to the Metropolis rule. The primary steps of the ALNS algorithm are described in Pisinger and Ropke (2007). In the following sections, we will first introduce the idea of generating an initial solution and then describe seven destroy operators as well as five repair operators.

4.2.1. Initial solution

We generate the initial solution using a two-phase construction algorithm. In the first phase, each request is assigned to its nearest bus stop, and the bus route is then generated using the nearest neighbor heuristic algorithm. In the second phase, the last-mile trip is generated for requests assigned to each stop. The detailed procedure is presented in Algorithm 2.

4.2.2. Destroy operators

In the following, we introduce a total of seven destroy operators, categorized as small and large destroy operators. The small destroy operators only focus on removing some requests, while the large destroy operators also alter the configuration of the bus route. The set of small destroy operators consists of *random removal, worst removal, related removal*, and *route removal*. Further details

Algorithm 2: Initial solution generation algorithm

```
//Phase 1: assign each request to a bus stop and generate a bus route
Calculate the direct travel distance to each bus station for each request;
Assign each request to its nearest bus station;
Generate the bus route with the nearest neighbor algorithm for each bus stop assigned with requests;
//Phase 2: generate a last-mile leg for each request
for each station do
    while the number of un-inserted requests assigned to it is greater than 0 do
        Generate an empty route starting from the station;
        while the number of matched riders is less than the car capacity do
            Pick out the request nearest to the last node of the route from un-inserted requests assigned to the station,
             and add this request to the tail of the route;
            if the route is infeasible then
                Remove this request from the route and break this while;
        Add the route into the route set for the bus stop;
Output the routes of the bus and cars.
```

on these small removal operators can be found in the work by Pisinger and Ropke (2007). Additionally, we introduce three large removal operators: *station close removal, station open removal*, and *station swap removal*.

Station close removal. This operator randomly selects a bus station and removes all requests assigned to it. Then, the information regarding the route and arrival time at each bus stop is updated.

Station open removal. This operator opens a random bus stop from those not selected in the current bus route. Following this, a certain number of requests closest to the stop are removed.

Station swap removal. This operator randomly closes one bus stop and opens another stop using the roulette wheel method, with the distance to the closed stop serving as a criterion. Subsequently, the bus route and arrival time are updated.

4.2.3. Repair operators

For this problem, we introduce five repair operators: *random insert, greedy insert, regret-2 insert, regret-3 insert*, and *best insert*.

Random insert. For each removed request, a car route is randomly chosen, and an attempt is made to insert the request into it if feasible. If there is no feasible insertion, a new car route may be generated for this request. Alternatively, if the distance between the stop and the destination is within comfortable walking distance, the request may not be provided with the last-mile travel service, which depends on the cost of employing a new car and walking to the destination.

Best insert. The location with the lowest cost increase for insertion is determined for each removed request, and the request is inserted into that location.

Regarding the basic *greedy insert, regret-2 insert*, and *regret-3 insert* operators, we refer to Pisinger and Ropke (2007).

It is important to note that when the walking distance is within the comfortable walking range, the walking cost from the stop to the destination for each request must be calculated before insertion. If the walking cost is lower than the insertion cost, the request will not be offered a last-mile travel service and will need to walk to the destination. This consideration ensures that the decision to provide last-mile travel service is cost-effective when compared to walking.

5. Numerical experiments

In this section, we conduct computational experiments to test the performance of the proposed methods based on the case of Shenzhen, China. These procedures are coded in C++, and the MILP, MP, and RMP models are solved using CPLEX 12.8. All experiments are run on a PC with 16 GB of RAM and a 3.0 GHz CPU.

5.1. Instances generation and parameters setting

We select the coverage area of Shenzhen airport bus lines A3 and NA1, as shown in Fig. 3, as the service area of the flexible airport bus. Flexible bus stops are selected from the stops of these two bus lines and express bus lines in this area. Passenger destinations within this service region are randomly generated, with a ratio of 1:3 for business requests and general requests. Fig. 3 provides an illustrative example for an instance involving 20 flexible bus stops and 20 requests within the service area. The ALNS algorithm is executed with a maximum of 20,000 iterations, and other parameter values align with those specified by Pisinger and Ropke (2007).

The car capacity is set at four seats, and each request consists of 1 or 2 riders, with probabilities of 75% and 25%, respectively. The value-of-time coefficient is \$0.3 per minute for general requests and \$0.4 per minute for business requests. According to the statistical data (Victoria Transport Policy Institute, 2022), we take the transportation costs to be \$0.2 per kilometer for the bus and \$1.6 per kilometer for cars, respectively. Thanks to the implementation of policies such as bus priority lanes in Shenzhen, we

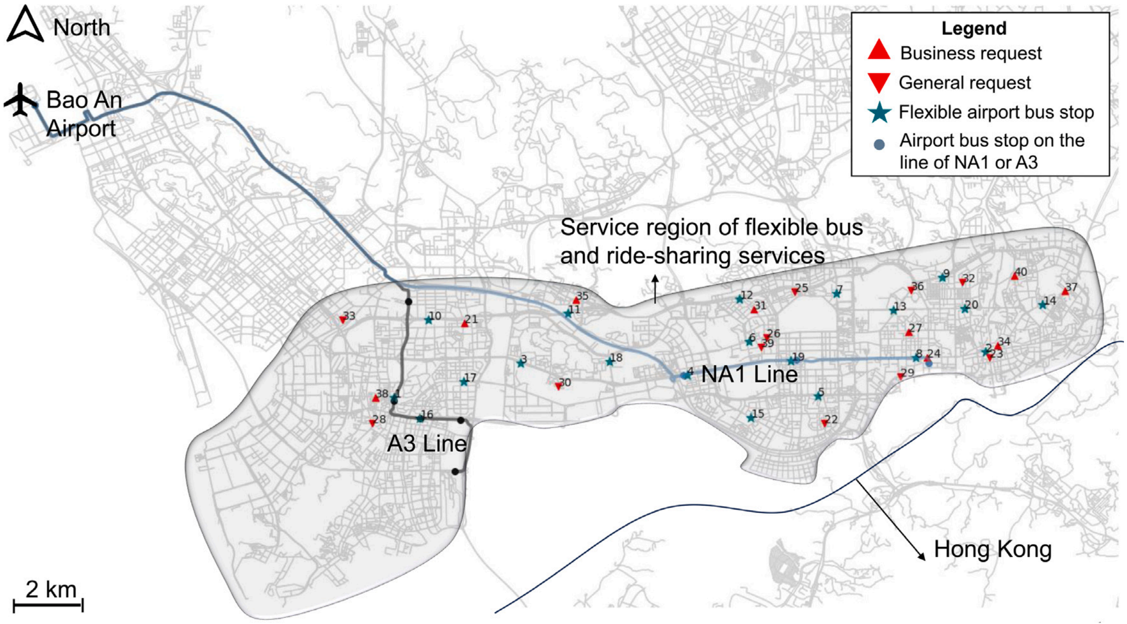

Fig. 3. The service area and some sample requests for flexible airport bus and ride-sharing services.

consider the speeds of the flexible bus and cars to be 45 km/h and 30 km/h, respectively. Drop-off times at the bus stop and request node are 2 min and 1 min, respectively. The comfortable walking distance is one kilometer, with a walking speed of 6 kilometers per hour.

5.2. Computational performances

To evaluate the solution quality and computational performance of two heuristic algorithms, we compare their computational results on three sets of instances: small-scale, practical-scale, and large-scale instances. Small-scale instances, derived from simplified real-world scenarios, serve as a baseline to evaluate whether an algorithm can achieve optimality in these settings. Practical-scale instances are more representative of real-world scenarios and are utilized to test the applicability of algorithms in authentic situations. For instance, in the operation of airport buses, seating capacities typically range from 17 to 50. To further evaluate the algorithms' scalability and applicability to more complex scenarios, we generate a set of larger-scale instances, varying the number of requests from 60, 70, and 80, as well as the number of stops from 20, 25, and 30.

5.2.1. Small-scale instances

We first compare the computational results, as presented in Table 2, among CPLEX, ALNS, and the math-heuristic algorithm using 18 small-scale instances. Each instance is named in the form of "S*R*", with "S" and "R" representing the number of flexible stops and requests, respectively. Columns 2, 5, and 7 denote the objective value for the best solution identified by CPLEX, the ALNS algorithm, and the math-heuristic algorithm. Column 3 represents the optimality gap between the lower bound solution and the best integer solution found by the CPLEX solver. Columns 4, 6, and 8 denote the computation time for obtaining solutions with these methods. Columns 9 and 10 indicate the gap in objective values between solutions identified by the ALNS algorithm and the math-heuristic algorithm in comparison to those of CPLEX.

As shown in Table 2, both the ALNS and math-heuristic algorithms achieve optimal solutions for most small-scale instances. The math-heuristic algorithm outperforms the ALNS algorithm in terms of solution quality, with a 0.4% lower average Gap. In terms of computation time, the ALNS algorithm is the fastest, with an average computation time of only 2.71 s. The math-heuristic algorithm requires about 47 s on average. However, CPLEX requires a longer computation time, averaging around 5133 s.

5.2.2. Practical-scale instances

To further evaluate the computational performance of the math-heuristic method on practical-scale instances, we compare its results with those of the ALNS algorithm on 21 instances, as presented in Table 3. Columns 2 and 3 represent the best and average objective values identified by running the ALNS ten times, respectively, while column 5 denotes the objective value found by the math-heuristic algorithm. Columns 4 and 6 indicate the computation times of ALNS and math-heuristic algorithms, respectively. Columns 7, 8, and 9 show the optimality gap between the objective values identified by these methods.

In terms of solution quality, the math-heuristic algorithm significantly outperforms the ALNS algorithm. More specifically, the average optimality gap between the math-heuristic solution and the best solution of the ALNS algorithm is −0.29%, and the average

Table 2
Computational performances for small-scale instances.

Instances	*CPLEX*			*ALNS*		*MH*		$Gap^{compare}$	
	Obj	*Gap*	*Time*	Obj^{best}	*Time*	*Obj*	*Time*	*Gap*1	*Gap*2
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
S4R5	157.32	0.00	3.91	157.32	0.95	157.32	1.52	0.00	0.00
S4R7	200.59	0.00	141.92	200.59	1.53	200.59	1.90	0.00	0.00
S4R9	253.36	12.22	7200.01	253.36	2.10	253.36	5.01	0.00	0.00
S4R11	294.06	52.56	7200.03	294.06	2.79	294.06	10.02	0.00	0.00
S4R13	349.79	61.87	7200.03	347.89	3.46	347.89	9.83	−0.54	−0.54
S4R15	400.36	71.26	7200.03	400.36	4.82	400.36	12.96	0.00	0.00
S5R5	157.32	0.00	6.51	157.32	1.12	157.32	2.93	0.00	0.00
S5R7	200.59	0.00	619.17	200.59	1.62	200.59	7.49	0.00	0.00
S5R9	251.60	46.27	7200.03	253.36	2.21	251.60	24.39	0.70	0.00
S5R11	295.53	66.69	7200.05	295.02	2.71	295.53	14.47	−0.17	0.00
S5R13	351.79	65.68	7207.13	347.89	3.54	347.89	17.39	−1.11	−1.11
S5R15	400.36	68.88	7200.05	400.36	5.37	400.36	16.39	0.00	0.00
S6R5	143.40	0.00	24.05	144.84	1.11	143.40	4.34	1.00	0.00
S6R7	190.27	0.00	5183.59	195.93	1.50	190.27	42.31	2.97	0.00
S6R9	230.94	58.78	7200.03	235.43	2.32	230.94	59.95	1.94	0.00
S6R11	274.84	68.13	7200.05	280.61	2.91	276.41	256.04	2.10	0.57
S6R13	330.20	67.46	7200.08	332.13	3.68	334.87	166.31	0.58	1.41
S6R15	384.97	71.73	7200.50	386.90	4.95	386.28	186.05	0.50	0.34
Average	**270.41**	**39.53**	**5132.62**	**271.33**	**2.71**	**270.50**	**46.63**	**0.44**	**0.04**

*$Gap1 = [(5) - (2)]/(2) \times 100\%$; $Gap2 = [(7) - (2)]/(2) \times 100\%$.

Table 3
Computational performances for practical-scale instances.

Instances	*ALNS*			*MH*		$Gap^{compare}$		
	Obj^{best}	$Obj^{average}$	*Time*	*Obj*	*Time*	*Gap*1	*Gap*2	*Gap*3
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
S10R20	483.79	490.27	10.37	475.15	234.80	1.34	−1.79	−3.09
S10R25	589.69	595.73	18.44	573.99	329.32	1.02	−2.66	−3.65
S10R30	705.23	731.87	30.51	689.42	90.64	3.78	−2.24	−5.80
S10R35	814.27	838.71	62.17	802.19	103.05	3.00	−1.48	−4.35
S10R40	915.10	954.33	68.23	930.88	134.21	4.29	1.72	−2.46
S10R45	1019.62	1067.93	101.41	1023.76	203.76	4.74	0.41	−4.14
S10R50	1147.76	1179.47	133.63	1151.60	869.61	2.76	0.33	−2.36
S15R20	479.52	491.38	12.31	493.27	371.09	2.47	2.87	0.38
S15R25	593.11	629.40	22.36	573.55	524.44	6.12	−3.30	−8.87
S15R30	707.08	744.27	33.31	688.19	392.88	5.26	−2.67	−7.54
S15R35	837.66	856.66	54.41	837.22	756.45	2.27	−0.05	−2.27
S15R40	926.94	945.96	77.64	907.85	322.26	2.05	−2.06	−4.03
S15R45	1032.32	1047.58	102.91	1033.48	312.63	1.48	0.11	−1.35
S15R50	1140.91	1193.94	128.85	1145.90	1009.61	4.65	0.44	−4.02
S20R20	478.74	503.72	12.11	471.59	386.82	5.22	−1.49	−6.38
S20R25	589.50	620.12	21.75	605.73	477.87	5.19	2.75	−2.32
S20R30	704.05	719.07	31.16	722.22	869.23	2.13	2.58	0.44
S20R35	839.66	850.29	52.51	825.03	819.04	1.27	−1.74	−2.97
S20R40	939.94	961.62	76.47	933.31	993.21	2.31	−0.71	−2.94
S20R45	1043.83	1059.42	126.76	1035.26	1247.33	1.49	−0.82	−2.28
S20R50	1135.59	1174.18	178.02	1176.64	1243.45	3.40	3.61	0.21
Average	**815.44**	**840.76**	**64.54**	**814.11**	**556.75**	**3.15**	**−0.29**	**−3.32**

*$Gap1 = [(3) - (2)]/(2) \times 100\%$; $Gap2 = [(5) - (2)]/(2) \times 100\%$; $Gap3 = [(5) - (3)]/(3) \times 100\%$.

optimality gap between the math-heuristic algorithm and the average solution of the ALNS algorithm is −3.32%. Regarding the computational time, the ALNS algorithm performs better than the math-heuristic algorithm, with an average computation time of 64.54 s, compared to 556.75 s for the math-heuristic. Hence, the ALNS can be used to swiftly modify the solution in the case that some passengers do not show up at the scheduled bus departure time.

Based on the results of the two sets of instances, it is clear that the math-heuristic algorithm achieves excellent solution quality for small-scale and practical-scale instances for flexible airport buses, albeit at the cost of longer computation time compared to ALNS.

5.2.3. Large-scale instances

To verify the applicability of the algorithm in other application scenarios with more requests and flexible stops, we generated a set of large-scale instances with flexible bus stops of 20, 25, and 30, and request numbers of 60, 70, and 80, respectively. The

Table 4
Computational performances for large-scale instances.

Instances	*ALNS*			*MH*	
	Obj^{best}	$Obj^{average}$	*Time*	*Obj*	*Time*
(1)	(2)	(3)	(4)	(5)	(6)
60R20S	1451.43	1484.56	294.29	1471.30	2051.06
60R25S	1455.86	1479.88	341.01	1478.29	2562.94
60R30S	1428.14	1476.61	359.42	1507.86	5729.24
70R20S	1675.13	1715.06	490.73	1742.16	>7200.00
70R25S	1646.12	1713.54	519.14	1678.70	3909.40
70R30S	1662.46	1738.81	554.46	1724.08	>7200.00
80R20S	1863.21	1908.16	719.16	1961.62	>7200.00
80R25S	1884.82	1923.66	765.09	2163.78	4985.63
80R30S	1833.72	1906.25	866.39	2256.78	6314.22

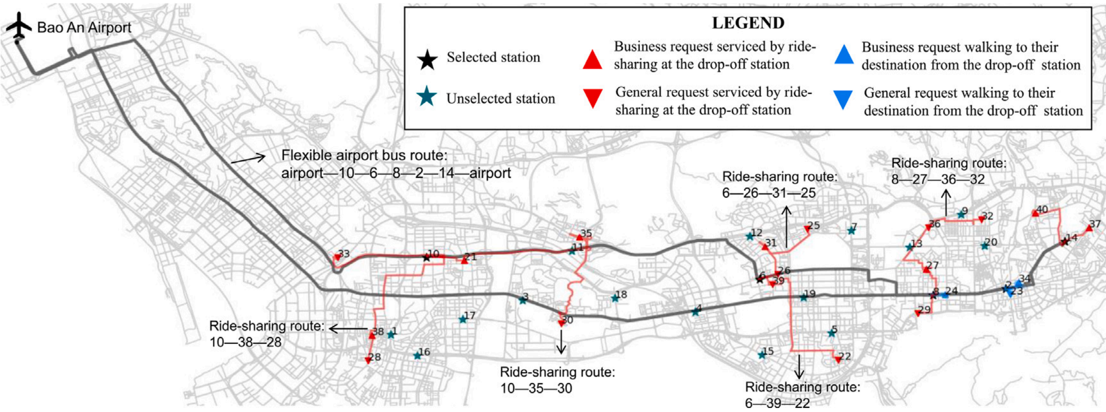

Fig. 4. Flexible bus and ride-sharing service scheme for scenario 1.

computational results of the ALNS algorithm and the MH algorithm are shown in Table 4, from which it can be seen that, as the scale increases, ALNS is still able to obtain high-quality solutions in an efficient computational time. For example, the ALNS algorithm is able to solve the instance with 80 requests as well as 30 bus stops in 866 s. The numerical results show that these algorithms, particularly ALNS, can be applied to larger-scale flexible bus routing problems than airport bus scenarios.

5.3. Flexibility of the service mode

To demonstrate the flexibility of this service model in bus route design and last-mile ride-sharing schemes, we randomly generate two scenarios with identical numbers and locations of flexible bus stops, each comprising 20 randomly generated demand points. Then, we obtain near-optimal solutions for both scenarios using the math-heuristic algorithm and visualize their service plans in Figs. 4 and 5.

5.3.1. Scenario 1

In this scenario, the airport bus makes sequential stops at stations 10, 6, 8, 2, and 14. Ten cars are scheduled to provide last-mile services associated with these stops, with five of them providing ride-sharing services and five offering non-sharing services. Additionally, three requests, 23, 24, and 34, will walk to their final destinations.

5.3.2. Scenario 2

In this scenario, the bus makes stops at 17, 19, 2, and 20 in order, and only stop 2 is also included in the bus route of scenario 1. Eight cars are used to provide last-mile services, with six of them providing ride-sharing services and two offering non-sharing services. Additionally, there is only one request that must walk to its final destination. From the service schemes of the two scenarios, we can observe that this mode is flexible and can dynamically adjust the optimal bus route and last-mile services based on the demand situation.

5.4. Benefits and drawbacks of flexible airport bus with ride-sharing services

To analyze the benefits and drawbacks of this service mode, we utilize six groups of randomly generated requests from Shenzhen airport buses NA1 and A3 lines to compare the total travel time of each request and the total vehicle travel distance with traditional airport bus mode, as well as the taxi or ride-hailing mode. Each group consists of fifty requests.

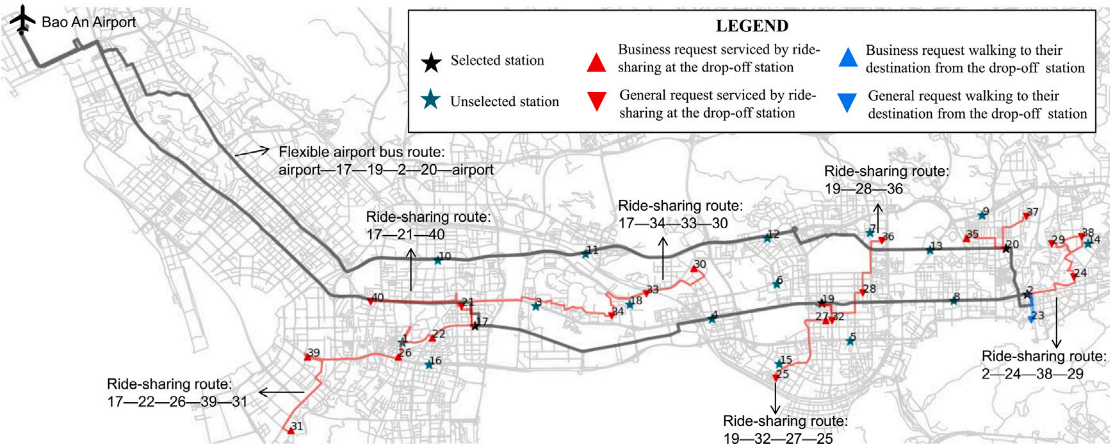

Fig. 5. Flexible bus and ride-sharing service scheme for scenario 2.

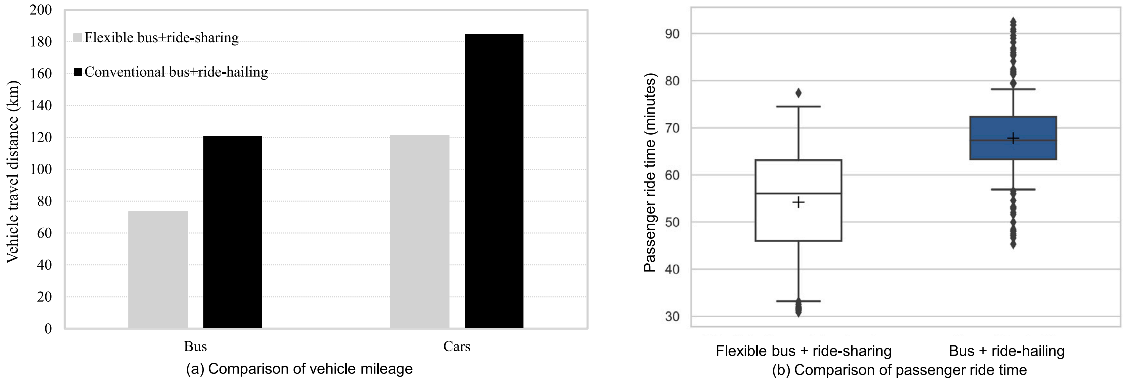

Fig. 6. Comparison with conventional airport bus services coupled with ride-hailing.

5.4.1. Comparison with conventional airport bus services coupled with ride-hailing

We begin by comparing the flexible airport bus service with the traditional airport bus plus ride-hailing service model, examining the differences in passenger travel time and vehicle travel distance between the two models. For this purpose, we use two Shenzhen airport bus lines, NA1 and A3, as representatives of traditional airport bus lines. The NA1 and A3 lines, along with their stops, are illustrated in Fig. 3. The fixed departure times of NA1 and A3 lines are 30 and 60 min, respectively. In the traditional model, we assign an optimal drop-off bus stop for each request and assume that each passenger is matched to a car through online ride-hailing before arriving at the drop-off point, without waiting time for transferring to the car at the drop-off bus stop. Detailed results of the comparison are presented in Fig. 6. The average vehicle mileage traveled by buses and cars for the six groups of demand scenarios is depicted in Fig. 6(a), while Fig. 6(b) compares passenger travel times on flexible buses and conventional buses for all passengers.

As illustrated in Fig. 6, the flexible service mode reduces both bus and car travel distances compared to the conventional mode. Specifically, the travel distances of the flexible bus and conventional airport buses (on NA1 and A3 lines) are 73.38 km and 120.88 km, respectively, signifying a 39.30% reduction in bus travel distance. Furthermore, vehicle travel distances in ride-sharing and ride-hailing modes are 121.28 km and 184.92 km, respectively, indicating a 34.42% reduction in car travel distances in the flexible mode.

With respect to passenger ride times, the average ride time for these two modes is 54 min and 68 min, respectively, which means that the new mode can achieve an average reduction of 20.59% in passenger ride times.

5.4.2. Comparison with ride-hailing mode

We then compare this flexible bus service with the ride-hailing mode to analyze the differences in passenger travel times and vehicle travel distances. We assume that passenger waiting times for a car at the airport are zero for the ride-hailing service. Figs. 7(a) and 7(b) compare the average vehicle mile as well as passenger travel time associated with the flexible bus mode and taxi mode for the six demand scenarios, respectively.

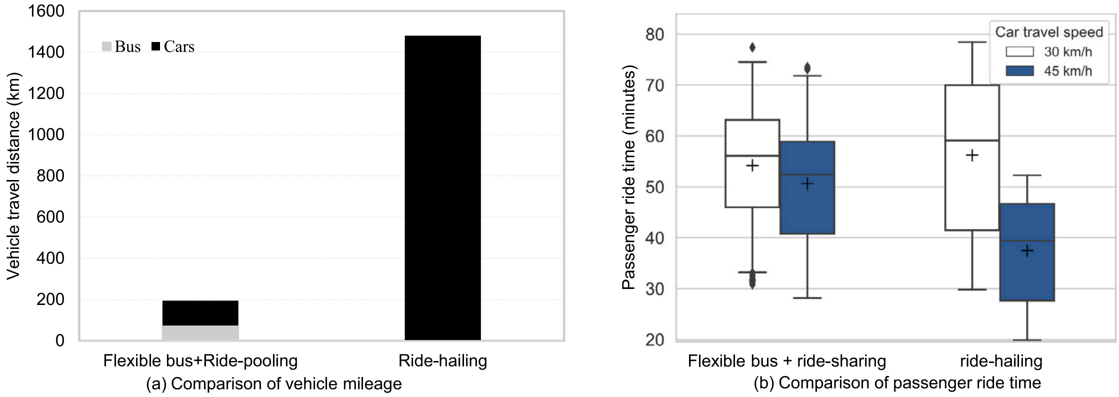

Fig. 7. Comparison with ride-hailing mode.

As illustrated in Fig. 7(a), the average vehicle travel distances for the flexible model are reduced by 86.86% compared to the ride-hailing model. Regarding the total ride time for each passenger, we compare the flexible mode with ride-hailing under two car travel speeds, 30 km/h and 45 km/h, with results depicted in Fig. 7(b). According to the results, at a speed of 30 km/h, the new model reduces the average passenger trip time from 56.21 min to 54.20 min. At a speed of 45 km/h, the new model increases the average passenger trip time from 37.47 min to 50.65 min. Results indicate that in cities with bus priority lanes, where the average speed of buses typically exceeds that of cars, the use of this flexible mode is more advantageous, resulting in shorter ride times for passengers compared to ride-hailing services.

6. Conclusion and future research

Airport buses play a critical role in a sustainable and cost-efficient last-mile of air travel, but they struggle with long departure intervals, inflexible stops, and considerable distances between stops. In this work, we investigate the integrated optimization problem concerning flexible bus and ride-sharing services. The primary goal is to deliver a high-quality and cost-effective last-mile service for passengers departing from the airport and heading to their final destinations. We optimize this service from the perspective of a public transit agency, with the objective of minimizing total operational costs. We first build a mixed-integer linear programming model, then decompose the problem into a flexible bus route optimization master problem and a last-mile ride-sharing service optimization sub-problem associated with each visited bus stop. Based on this decomposition, we design a double decomposition math-heuristic algorithm to obtain a high-quality solution. To evaluate the solution quality of the math-heuristic algorithm, we implement an adaptive large neighborhood search heuristic algorithm. Finally, we conduct numerical experiments based on the coverage and stop information from two Shenzhen airport bus lines, as well as randomly generated demand information, to verify the effectiveness of the model and algorithm.

The results of numerical experiments demonstrate the multiple benefits of this flexible airport bus and last-mile ride-sharing service. First, it dynamically designs bus routes and last-mile travel plans based on passenger demand, effectively reducing passenger travel time and thus enhancing the service quality of airport buses. This benefit is more pronounced in cities with bus priority lanes, which ensure that buses travel at a faster speed than private cars or cabs. Second, when compared to traditional airport buses and taxi/ride-hailing services, this flexible service efficiently reduces the number of vehicles used and total mileage, potentially helping mitigate traffic congestion and lower emissions. Although we consider instances from Shenzhen, it is important to note that this flexible airport bus service model holds promise for deployment in other megacities such as Osaka, Paris, Montreal, and Los Angeles.

When conducting experiments, we assume that flexible airport bus service providers can schedule sufficient cars to offer last-mile services to passengers before the bus arrives at each stop. Future research could delve into the design of this service under limited car availability. Additionally, we assume that cars are either owned by public transportation agencies or hired on an ad hoc basis, which incurs relatively high operating costs. It would be worthwhile for future work to investigate how the service could be designed by incorporating free-floating cars operated by transportation network companies and, accordingly, how revenue would be distributed between public transportation agencies and transportation network companies. Moreover, we assume that all passengers would accept the travel service offered by the provider without considering whether passengers would be willing to accept it. Future research could also investigate the impact of passenger choice behavior, or preferences for different service options, on the design of this service. Lastly, leveraging artificial intelligence technologies, such as machine learning (Liu et al., 2022) and language models (Qu et al., 2023; Liu et al., 2023b), can enhance the efficiency of the algorithm by aiding in the selection of bus stops and matching stops with passengers.

CRediT authorship contribution statement

Ping He: Writing – original draft, Validation, Software, Methodology, Conceptualization. **Jian Gang Jin:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization. **Frederik Schulte:** Writing – review & editing, Conceptualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the National Natural Science Foundation of China [Grant 72122014, 72061127003]. The first author very much appreciates the support of the China Scholarship Council [Grant 202106230253].

Appendix. Column generation procedure

A.1. A set partitioning formulation

The $SP(s)$ model aims to find the best last-mile service combinations for passengers getting off at stop s. This model can be decomposed into a sub-problem to determine promising car routes – consisting of ride-sharing and non-sharing routes – and a master problem, which selects the best combination between car routes and walking legs from the stop to the respective passenger destinations.

Let $\mathbb{R}$ represent the set of car routes, and let a binary decision variable $\delta_r (r \in \mathbb{R})$ denote whether the route $r \in \mathbb{R}$ is selected. For a given route r, let ϵ_r denote total costs. ϵ_r consists of transportation costs, ride time costs for requests matched with the route, and the fixed cost of a car. The binary parameter χ_n^r is set to one if request n ($n \in \mathcal{N}_C^s$) is matched by route r and zero otherwise. Similarly, let binary variable Γ_n denote whether request n has to walk to the destination from stop s and μ_n denote the total costs for walking from the stop s to the destination of requests n. Note that the μ_n is set to an extremely large constant M if the destination of request n is not within the comfortable walking distance from stop s. Then, the set partitioning model is formulated as

$$[MP_{CG}]\ min \sum_{r \in \mathbb{R}} \epsilon_r \delta_r + \sum_{n \in \mathcal{N}_C^s} \mu_n \Gamma_n \tag{A.1}$$

subject to:

$$\sum_{r \in \mathbb{R}} \chi_n^r \delta_r + \Gamma_n = 1, \forall n \in \mathcal{N}_C^s \tag{A.2}$$

$$\delta_r, \Gamma_n \in \{0, 1\}, \forall r \in \mathbb{R}, \forall n \in \mathcal{N}_C^s \tag{A.3}$$

The objective function (A.1) minimizes the total costs of selected car routes and walking legs. Constraints (A.2) denote each request must be matched with one kind of last-mile solution. Constraints (A.3) are the integrality constraint.

Enumerating all trips in the MP_{CG} formulation is impractical due to the exponential number of complete routes. Instead, a common approach is to relax the binary decision variable δ_r into a continuous variable, and then solve the restricted master problem (RMP_{CG}), focusing on a subset $\overline{\mathbb{R}}$ of $\mathbb{R}$ using the column generation algorithm. To identify promising routes for the RMP_{CG}, the pricing sub-problem (PSP_{CG}) is solved.

A.2. Pricing sub-problem

The pricing sub-problem of the RMP_{CG} model aims to find routes with negative reduced costs that can be incorporated into the current set of columns ($\overline{\mathbb{R}}$) and can be formulated as an elementary shortest path problem with resource constraints (ESPPRC). Let $\pi_n (n \in \mathcal{N}_C^s)$ be the dual variables of constraints (A.2). Then, the reduced cost of route r is $\overline{\epsilon_r} = \epsilon_r - \sum_{n \in \mathcal{N}_C^s} \chi_n^r \pi_n$. Let y_{mn} be a binary variable that denotes whether the arc (m, n) is traversed. Therefore, the sub-problem can be formulated as:

$$[PSP_{CG}]\ min\ \overline{\epsilon_r} = \sum_{m \in \mathcal{N}_C^s} \sum_{n \in \mathcal{N}_C^s} c_{car} d_{mn} y_{mn} + c_{fix} + \sum_{n \in \mathcal{N}_C^s \cap \mathcal{N}_C^b} p_b \Delta_n'' + \sum_{n \in \mathcal{N}_C^s \cap \mathcal{N}_C^g} p_g \Delta_n'' - \sum_{n \in \mathcal{N}_C^s} \chi_n^r \pi_n \tag{A.4}$$

subject to:

$$\sum_{n\in\mathcal{N}_C^s\cup\{n_d\}} y_{mn} + \Gamma_m \leq 1, \forall m \in \mathcal{N}_C^s \tag{A.5}$$

(26)–(33) (A.6)

The objective function (A.4) minimizes the total costs of a car route, comprising transportation costs, fixed costs, ride time costs, and dual value costs. Constraints (A.5) denote that each request disembarking at stop s can be matched with at most one kind of last-mile solution.

The ESPPRC is a well-known NP-hard problem (Dror, 1994). The common method for solving the ESPPRC as well as its variants to optimality is to employ the labeling algorithm. Despite this PSP_{CG} problem involving ride time costs and having the characteristic that each route ends with a dummy node, it does not introduce additional complexity for the application of the classical labeling algorithm. As a result, we do not elaborate on how to use the labeling algorithm to solve this problem. Readers seeking more details about the labeling algorithm can refer to Righini and Salani (2006).

Building upon the master problem, sub-problem, and labeling algorithm described above, the implementation of the column generation algorithm for obtaining last-mile travel solutions is straightforward. For more details about the column generation algorithm, we recommend interested readers to read this book by Desaulniers et al. (2006).

References

Abe, R., 2021. Preferences of urban rail users for first- and last-mile autonomous vehicles: Price and service elasticities of demand in a multimodal environment. Transp. Res. C 126, 103105.

Akar, G., 2013. Ground access to airports, case study: Port columbus international airport. J. Air Transp. Manag. 30, 25–31.

Bian, Z., Bai, Y., Liu, X., Wang, B., 2022. An online hybrid mechanism for dynamic first-mile ridesharing service. Transp. Res. C 138, 103585.

Bian, Z., Liu, X., 2019a. Mechanism design for first-mile ridesharing based on personalized requirements part I: Theoretical analysis in generalized scenarios. Transp. Res. B 120, 147–171.

Bian, Z., Liu, X., 2019b. Mechanism design for first-mile ridesharing based on personalized requirements part II: Solution algorithm for large-scale problems. Transp. Res. B 120, 172–192.

Chen, J., Wang, S., Liu, Z., Wang, W., 2017. Design of suburban bus route for airport access. Transp. A: Transp. Sci. 13 (6), 568–589.

Chen, S., Wang, H., Meng, Q., 2020. Solving the first-mile ridesharing problem using autonomous vehicles. Comput.-Aided Civ. Infrastruct. Eng. 35 (1), 45–60.

Currie, G., Wallis, I., 2008. Effective ways to grow urban bus markets–a synthesis of evidence. J. Transp. Geogr. 16 (6), 419–429.

Derigs, U., Pullmann, M., Vogel, U., 2013. Truck and trailer routing—problems, heuristics and computational experience. Comput. Oper. Res. 40 (2), 536–546.

Desaulniers, G., Desrosiers, J., Solomon, M.M., 2006. Column Generation. vol. 5, Springer Science & Business Media.

Dror, M., 1994. Note on the complexity of the shortest path models for column generation in VRPTW. Oper. Res. 42 (5), 977–978.

Estrada, M., Salanova, J.M., Medina-Tapia, M., Robusté, F., 2021. Operational cost and user performance analysis of on-demand bus and taxi systems. Transp. Lett. 13 (3), 229–242.

He, P., Jin, J.G., Schulte, F., Trépanier, M., 2023. Optimizing first-mile ridesharing services to intercity transit hubs. Transp. Res. C 150, 104082.

Hu, Y., Luo, X., Bai, D., 2023. Passenger congestion alleviation in large hub airport ground-access system based on queueing theory. Transp. B: Transp. Dyn. 11 (1), 257–278.

Kim, M.E., Schonfeld, P., 2014. Integration of conventional and flexible bus services with timed transfers. Transp. Res. B 68, 76–97.

Kong, X., Li, M., Tang, T., Tian, K., Moreira-Matias, L., Xia, F., 2018. Shared subway shuttle bus route planning based on transport data analytics. IEEE Trans. Autom. Sci. Eng. 15 (4), 1507–1520.

Lee, E., Cen, X., Lo, H.K., 2022. Scheduling zonal-based flexible bus service under dynamic stochastic demand and time-dependent travel time. Transp. Res. E 168, 102931.

Li, Y., Cote, J.-F., Coelho, L.C., Zhang, C., Zhang, S., 2023. Order assignment and scheduling under processing and distribution time uncertainty. European J. Oper. Res. 305 (1), 148–163.

Liu, Y., Jia, R., Ye, J., Qu, X., 2022. How machine learning informs ride-hailing services: A survey. Commun. Transp. Res. 2, 100075.

Liu, Z., Wang, Q., Sigler, D., Kotz, A., Kelly, K.J., Lunacek, M., Phillips, C., Garikapati, V., 2023a. Data-driven simulation-based planning for electric airport shuttle systems: A real-world case study. Appl. Energy 332, 120483.

Liu, Y., Wu, F., Liu, Z., Wang, K., Wang, F., Qu, X., 2023b. Can language models be used for real-world urban-delivery route optimization? The Innovation 4 (6).

Lu, J., Yang, Z., Timmermans, H., Wang, W., 2016a. Optimization of airport bus timetable in cultivation period considering passenger dynamic airport choice under conditions of uncertainty. Transp. Res. C 67, 15–30.

Lu, X., Yu, J., Yang, X., Pan, S., Zou, N., 2016b. Flexible feeder transit route design to enhance service accessibility in urban area. J. Adv. Transp. 50 (4), 507–521.

Ma, J., Chen, X., Xing, Z., Zhang, Y., Yu, L., 2023a. Improving the performance of airport shuttle through demand-responsive service with dynamic fare strategy considering mixed demand. J. Air Transp. Manag. 112, 102459.

Ma, T.-Y., Rasulkhani, S., Chow, J.Y.J., Klein, S., 2019. A dynamic ridesharing dispatch and idle vehicle repositioning strategy with integrated transit transfers. Trans. Res. E 128, 417–442.

Ma, W., Zeng, L., An, K., 2023b. Dynamic vehicle routing problem for flexible buses considering stochastic requests. Transp. Res. C 148, 104030.

Macrina, G., Pugliese, L.D., Guerriero, F., Laporte, G., 2020. Crowd-shipping with time windows and transshipment nodes. Comput. Oper. Res. 113, 104806.

Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. Comput. Oper. Res. 34 (8), 2403–2435.

Qu, X., Lin, H., Liu, Y., 2023. Envisioning the future of transportation: Inspiration of ChatGPT and large models. Commun. Transp. Res. 3.

Righini, G., Salani, M., 2006. Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Discrete Optim. 3 (3), 255–273.

Shaheen, S., Chan, N., 2016. Mobility and the sharing economy: Potential to facilitate the first- and last-mile public transit connections. Built Environ. 42 (4), 573–588.

Stiglic, M., Agatz, N., Savelsbergh, M., Gradisar, M., 2018. Enhancing urban mobility: Integrating ride-sharing and public transit. Comput. Oper. Res. 90, 12–21.

Toth, P., Vigo, D., 2014. Vehicle Routing: Problems, Methods, and Applications. SIAM.

Tran, T.T., Araujo, A., Beck, J.C., 2016. Decomposition methods for the parallel machine scheduling problem with setups. INFORMS J. Comput. 28 (1), 83–95.

Victoria Transport Policy Institute, 2022. Transportation cost and benefit analysis II - vehicle costs. URL https://www.vtpi.org/tca/tca0501.pd, Online.

Wang, H., 2019. Routing and scheduling for a last-mile transportation system. Transp. Sci. 53 (1), 131–147.
Wang, H., Odoni, A., 2016. Approximating the performance of a "last mile" transportation system. Transp. Sci. 50 (2), 659–675.
Yu, H., Lv, W., Liu, H., Fu, X., Xiao, R., 2020. A dynamic line generation and vehicle scheduling method for airport bus line based on multi-source big travel data. Soft Comput. 24, 6329–6344.
Zhen, L., He, X., Wang, S., Wu, J., Liu, K., 2023. Vehicle routing for customized on-demand bus services. IISE Trans. 1–18.