Transportation Research Part E 210 (2026) 104802

Contents lists available at ScienceDirect

Transportation Research Part E

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

Dynamic vehicle dispatching for shared-and-autonomous-mobility services with adaptive request assignment

Jiangyan Huang, Min Xu *

Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, China

ARTICLE INFO

Keywords:
Shared autonomous vehicle
Ride-pooling
Dynamic vehicle dispatching
Hybridalgorithm

ABSTRACT

This study investigates a real-time vehicle dispatching problem for shared-and-autonomous-mobility (SAM) services that allow multiple passengers to share a ride. The objective is to optimize the real-time decision-making of the operator, and develop an online efficient algorithm to maximize the profit while ensuring service quality. In particular, we formulate the dynamic system with a series of static subproblems and continually optimize the vehicle dispatching solutions at each decision time point. Each static subproblem is formulated as a mixed-integer programming (MIP) model considering the maximum number of ride-pooling strangers and passenger satisfaction constraints. To solve the subproblem, we develop a customized hybrid algorithm that integrates an adaptive request assignment (ARA) scheme into the large neighborhood search (LNS) heuristic framework. Particularly, this method decomposes the multi-vehicle problem into single-vehicle problems and LNS iteratively identifies the optimal routing solution for each SAV. If overall profit does not improve after a certain number of iterations, the ARA scheme is invoked to adaptively reassign passenger requests to different vehicles. Numerical experiments are conducted to demonstrate the effectiveness of the proposed solution method against the benchmark approach and to examine the benefits of the SAM service model and the effect of passengers' flexibility time on system performance to derive management insights.

1. Introduction

Urban modernization has raised numerous concerns over prolonged traffic congestion, decreased transportation efficiencies, rising fuel consumption, and increased air pollution. Various forms of sustainable shared mobility services have emerged to address these pressing issues and reshape urban transportation. For example, over the past decades, there has been remarkable development in carsharing platforms, enabling individuals to rent cars for short periods (Yang et al., 2022). Effective vehicle relocations without significant labor costs are crucial to solving the spatial and temporal imbalances in vehicle distribution induced during carsharing operations. Meanwhile, ride-pooling services, pioneered by many transportation network companies (TNCs), are effective means to improve vehicle utilization and mobility efficiency. Individual travelers are pooled together based on similar itineraries and time schedules and share a vehicle for a trip (Agatz et al., 2012; Furuhata et al., 2013). Leveraging recent advancements in mobile technology, various on-demand ride-pooling programs such as UberPOOL, Lyft Line, and Didi Express Pool have been launched by utilizing dedicated for-hire drivers (Ke et al., 2020; Lu et al., 2022). However, these prevalent shared mobility solutions have been developed in

* Corresponding author at: Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, China.
E-mail addresses: jiangyan.huang@connect.polyu.hk (J. Huang), min.m.xu@polyu.edu.hk (M. Xu).

https://doi.org/10.1016/j.tre.2026.104802
Received 20 December 2024; Received in revised form 4 March 2026; Accepted 8 March 2026
Available online 16 March 2026

isolation, thus failing to fully capitalize on their combined potential advantages.

In recent years, advancements in self-driving technology have paved the way for consolidating and further enhancing current shared mobility services into a unified shared-and-autonomous-mobility (SAM) system that provides door-to-door services by utilizing centrally-controlled shared autonomous vehicles (SAVs). (Lokhandwala and Cai, 2018). Compared to traditional shared mobility business models, the inherent complete compliance characteristic of SAVs in such emerging SAM services enables more efficient vehicle management, as it eliminates the variability introduced by human drivers. This leads to more accurate and seamless vehicle dispatching operations that can better adapt to demand dynamics while reducing labor costs. Additionally, the flexibility of the SAV fleet facilitates customized services, including both pooled and solo rides, and improved levels of service tailored to passengers' preferences can be achieved. Despite these favorable merits, the ever-evolving SAM system with on-demand ride-pooling services has posed many decision-making challenges for service providers concerning service design, dynamic request assignment, vehicle routing, pooled trip arrangement, etc. The service quality should be ensured by balancing operational efficiency with passenger satisfaction, particularly when coordinating multiple rides into single-vehicle trips. These pooling arrangements necessitate careful consideration of passenger comfort and convenience associated with sharing space with strangers. To this end, it is imperative to develop advanced models and effective algorithms to guide the real-time operation of SAM services.

1.1. Literature review

The combination of shared mobility services with SAVs, characterized by high flexibility and reliability, is foreseeable to emerge as one of the promising directions in future mobility scenarios (Zhou and Roncoli, 2022). Previous studies have shown that autonomous vehicles can significantly reduce traditional private car ownership and alleviate road network congestion (Farhan and Chen, 2018; Huang et al., 2024; Levin, 2017; Narayanan et al., 2020). Relevant research on the SAM service operations using the SAV fleet already exists, and many of them have demonstrated significant benefits of ride-pooling opportunities in SAM services. (Farhan and Chen, 2018; Martinez and Viegas, 2017; Fagnant and Kockelman, 2018; Loeb et al., 2018; Ge et al., 2021; Lokhandwala and Cai, 2018; Zhang et al., 2015). For example, Fagnant and Kockelman (2018) investigated dynamic SAV operations with the ride-pooling option, employing a discrete-time agent-based simulation framework. They found that incorporating ride-pooling greatly contributed to the reduction in customers' waiting time and overall vehicle travel mileage, while avoiding new congestion problems. Similarly, Lokhandwala and Cai (2018) developed an agent-based model to explore ride-pooling services using both traditional vehicles (with drivers' shifts and breaks) and SAVs (available all day), considering individual heterogeneous preferences in sharing rides. A real-world investigation in New York City revealed that integrating SAVs and ride-pooling services could potentially reduce fleet size and carbon emissions. Other studies have examined the use of the electric SAV fleet in SAM services. Farhan and Chen (2018) examined the operation of electric SAVs based on an agent-based simulation model, and the results indicated that ride-pooling services decrease fleet size and the need for charging stations. Loeb et al. (2018) investigated the electric SAV operation performance by utilizing an agent-based simulator MATSim, focusing on the charging strategies and infrastructure decisions. Nevertheless, it is evident that most studies on dynamic SAV operations have concentrated on impact analysis using simulation frameworks with rule-based methods, rarely exploring the fields of operational-level decision-making optimization problems and algorithm design for SAM services. In our study, we will address a real-time SAV dispatching optimization problem for the SAM service considering ride-pooling options with the objective of maximizing the total profits of the service operator. The problem is referred to as the RT-SAVD for short thereafter.

The deployment of SAVs in SAM services enhances operational flexibility, potentially leading to improved system performance. This is achieved through continuous optimization of vehicle dispatching plans with seamless diversion and self-relocation to accommodate all user requests for pickups and drop-offs (Ma et al., 2017). In fact, the proposed RT-SAVD problem can be regarded as a specific variant of the dynamic dial-a-ride problem (DARP) or shared-taxi problem focusing on route planning. In this scenario, a fleet of SAVs is operated in an online manner to fulfill passenger requests with diverse origins and destinations. The dynamic DARP and shared-taxi problem are dynamic extensions of the static DARP. The static DARP, a variant of the pick-up and delivery problem with time windows, focuses on optimizing vehicle routes and schedules for people who are typically patients or the handicapped with specified pickup and drop-off locations, time windows, and maximum trip duration (Cordeau and Laporte, 2007). Unlike the static DARP, in which all the information of requests and vehicles are known in advance, the dynamic version involves unexpectedly arriving requests and decision-makers can adjust pre-planned routes as needed based on the new information. Notably, the dynamic DARP is considered deterministic if decision-makers have complete information about all ongoing and upcoming operations, except for unexpected new arrivals (Ho et al., 2018).

Among the literature on dynamic DARP, the decision-making problems typically focus on exploring the optimal accommodation of incoming requests on short notice and various metaheuristics and hybrid algorithms have been proposed (Attanasio et al., 2004; Berbeglia et al., 2010, 2012; Coslovich et al., 2006; Gendreau et al., 2006; Häll et al., 2015; Häme, 2011; Häme and Hakula, 2015; Maalouf et al., 2014; Marković et al., 2015; Souza et al., 2022). For example, Attanasio et al. (2004) introduced a parallel tabu search to efficiently insert newly arrived requests into existing routes and optimize routing solutions continuously. Coslovich et al. (2006) proposed a two-phase insertion algorithm utilizing route perturbances. The first stage, executed offline, involved the route neighborhood updating step to generate a collection of alternative feasible routes within the vicinity of the incumbent route, while the second stage, executed in real-time, provided a quick response to a newly arrived request using a simple insertion method to evaluate all possible insertions into the obtained feasible routes. Häme (2011) proposed an adaptive insertion algorithm with a priori clustering method to find the optimal solution. All possible insertion positions were evaluated for a specific request, and all resulting partial routes were preserved, particularly suitable for the highly restricted problem. Berbeglia et al. (2012) introduced a hybrid approach

combining the exact constraint programming with a tabu search algorithm to detect feasible insertions of new requests efficiently. Souza et al. (2022) investigated a dynamic DARP under a partially dynamic environment with both static requests known a priori and dynamically arrived ones over time. They developed a two-phase hybrid algorithm with the first phase dedicated to solving a static problem using the variable neighborhood search heuristic and the second phase responsible for dealing with the dynamic requests through a simple insertion heuristic. This method is designed in pursuit of operational cost minimization and customers' convenience maximization. Further extensions in dynamic DARP have also been explored by considering individual behaviors, stochastic factors, etc. (Azadeh et al., 2022; Hyytiä et al., 2012; Sayarshad and Chow, 2015; Sayarshad and Oliver Gao, 2018; Schilde et al., 2011, 2014; Tafreshian et al., 2021; Xiang et al., 2008). For example, Azadeh et al. (2022) proposed a choice-driven dynamic DARP integrating customer behaviors by incorporating assortment optimization with routing decisions. They proposed a pricing strategy while incorporating a choice model to offer personalized options of services (i.e., service type and pickup time) along with their corresponding price levels. This approach also generates possible assortments with associated probabilities for each new request. Schilde et al. (2014) investigated a dynamic and stochastic DARP considering the uncertainty of time-dependent travel speeds. Tafreshian et al. (2021) investigated an on-demand shuttle dispatching problem and developed an efficient data-driven two-phase algorithmic framework considering stochastic demand information.

The shared-taxi problem represents another novel variant of the DARP, which seeks to optimally allocate passengers to taxis and identify corresponding optimal routes in an online taxi-dispatch system (Mourad et al., 2019). Hosni et al. (2014) pioneered to formulate an MIP model for a generalized DARP that considered vehicles at various initial locations with onboard passengers, facilitating application within a dynamic system. The near-optimal solution was achieved through the application of the Lagrangian decomposition method, supplemented by two heuristic methods. To address dynamic scenarios, the proposed algorithm was periodically invoked and evaluated against a commercial solver. Given the large-scale nature and inherent dynamics of the shared-taxi problem, relevant studies primarily focused on developing efficient heuristic algorithms to produce high-quality vehicle routing solutions in real-time with good computation speed (Hua et al., 2022; Wang and Yang, 2019). To further explore extensions, Santos and Xavier (2013) investigated a dynamic shared-taxi problem by incorporating the monetary incentive and proposed an improved heuristic with a path-relinking technique to solve the problem. Santos and Xavier (2015) extended the previous study by incorporating the monetary incentive in a shared-taxi system and proposed an improved heuristic with a path relinking technique to solve the problem. Jung et al. (2016) proposed an advanced hybrid simulated annealing algorithm to solve the dynamic shared-taxi problem with the objective of maximizing occupancy rates and minimizing the travel time of passengers. Zhan et al. (2021) examined the collective advantages of the dynamic shared-taxi problem by maximizing the number of served customers and minimizing travel costs. A rolling horizon approach was adopted and each static subproblem was solved by an efficient adapted artificial bee colony algorithm integrated with a vantage-point tree technique for algorithm acceleration. Zhan et al. (2022) further considered electric vehicles (EVs) in the shared-taxi system and addressed two subproblems involving dynamic matching and EV charging respectively. These two subproblems are addressed concurrently, with the algorithm introduced by Zhan et al. (2021) and CPLEX solving them in parallel, utilizing a rolling horizon approach.

Additionally, other studies have tackled even larger situations and proposed efficient approximate methods. Alonso-Mora et al. (2017) proposed a dynamic vehicle dispatching approach to address a relaxed on-demand shared-taxi problem leveraging the concept of shareability networks to pair trips with vehicles. Their approach was evaluated using taxi data from Manhattan, demonstrating its effectiveness for large-scale applications. Later, Simonetto et al. (2019) developed a less computationally demanding algorithm, restricting the matching of only one passenger to a vehicle per optimization epoch and reducing the original problem into a linear assignment problem. Their findings revealed that the myopic optima preserved service quality comparable to the study by Alonso-Mora et al. (2017). Furthermore, Zhou and Roncoli (2022) extended the algorithmic framework from Simonetto et al. (2019) by incorporating the congestion prediction.

The above literature review shows that numerous studies have been conducted to investigate dynamic DARP or shared-taxi problems with different objectives, addressing challenges in efficient request assignment and vehicle dispatching by developing various optimization models and algorithms. However, the ride-pooling services considered in these studies often involve simultaneously serving two or more potential requests only respecting vehicle capacity constraints. The dynamic decision-making process focuses on finding optimal solutions to accommodate incoming requests without specific consideration of passengers' shared 'strangers' during their shared trips. In other words, the number of strangers sharing with each passenger is assumed to be unlimited in these studies. In reality, however, passengers might have the maximum ride-pooling stranger number preferences. For example, consider a passenger who is willing to share his/her trip with only one stranger. Suppose the concerned passenger has already been (or is being) transported with another passenger in the same vehicle. In that case, the service operator is not expected to continue pooling this passenger with additional passengers in its remaining journey, even if the vehicle capacity has sufficient capacity. Therefore, for the sake of service quality, it is recommended that SAM services impose restrictions on the number of strangers sharing rides with each request. Notably, the maximum number of ride-pooling strangers constraint distinguishes the SAM services from traditional dial-a-ride services. Additionally, given the inherent personal attributes and the characteristics of the ride-pooling trips such as detours and sharing trip duration, some passengers with potentially feasible ride-pooling options may be not satisfied with (i.e., may decline) the arranged shared trip. Their approval for sharing rides with strangers will inevitably affect the vehicle dispatching operations that provide pooled rides service, which is largely ignored in existing studies. To the best of our knowledge, previous research has not ever explored how to determine the real-time vehicle dispatching plans for the SAM services while incorporating ride-pooling stranger number limit and considering passengers' acceptance of sharing trips with others.

1.2. Objective and contributions

To bridge the research gap identified above, our study will investigate a RT-SAVD problem for SAM services in an online environment, where passenger requests are released dynamically. In particular, we impose constraints on the ride-pooling services by allowing each request only to be transported, not necessarily simultaneously, with a limited number of strangers during its trip. Individual acceptance of the shared trip is also considered based on a well-defined satisfaction measurement. In this context, satisfaction refers to the passenger's expected perceived quality of the offered pooled itinerary, determined by their personal attributes and the itinerary's characteristics (e.g., detour, shared-trip duration, and stranger exposure). This expected satisfaction is used to decide, for the current request, whether the passenger accepts the pooled option or instead chooses a solo ride (or rejects the offered shared option). Notably, some individual-specific attributes in the satisfaction function may also partially capture accumulated underlying long-run preferences and attitudes, which can be shaped by accumulated past ride-sharing experiences. The objective is to develop a fast and effective algorithm to support the decision-making of the service provider in terms of timely responding to newly arrived passenger requests, properly dispatching SAVs, and arranging satisfactory pooled trips in pursuit of profit maximization. To achieve this objective, we will propose a dynamic vehicle dispatching algorithmic framework based on the rolling horizon approach, which includes periodical optimization with a fixed time step over the entire planning horizon. For each optimization run, we will formulate a static SAV dispatching (S-SAVD) problem as an MIP model while considering the maximum number of ride-pooling strangers and passenger satisfaction. To solve each S-SAVD, we will develop a hybrid algorithm named ARA-LNS that integrates an adaptive request assignment (ARA) method into the traditional large neighborhood search (LNS) heuristic framework, allowing us to decompose the multi-vehicle S-SAVD problem into several single-vehicle problems to efficiently obtain the request assignment and corresponding vehicle routing plans. The contributions of this study can be summarized threefold.

- First, we propose a RT-SAVD problem for SAM services with ride-pooling options while incorporating the limit on the number of ride-pooling strangers and passenger satisfaction levels. We make the first attempt to integrate the service quality considerations into the real-time decision-making of the operator and develop an online efficient algorithmic framework with a series of S-SAVD subproblems to maximize the total profit. Each subproblem is formulated as an extension of the DARP.
- Second, we develop a tailored ARA-LNS solution algorithm for the proposed S-SAVD subproblem. The LNS iteratively identifies the optimal routing plan for each SAV, with the ARA scheme invoked if the overall profit for solutions for all SAVs fails to improve after certain iterations to adaptively reassign requests to different vehicles. The adaptive selection of the different assignment operators will guide the algorithm to efficiently search for good-quality vehicle dispatching solutions.
- Third, we conduct extensive numerical experiments to demonstrate the efficacy of our proposed solution method compared to the benchmark approach, to examine the benefits of the SAM service model, to analyze the impact of the flexibility in passengers' time windows on system performance, and to derive management insights.

The remainder of this study is organized as follows. Assumptions, notation and the description of the RT-SAVD problem are elaborated in Section 2. Section 3 presents the formulation for the dynamic vehicle dispatching problem by establishing the algorithmic framework and modeling the S-SAVD subproblem. Section 4 develops a customized ARA-LNS algorithm for solving the static subproblem. The efficacy of our proposed solution method and impact analysis are demonstrated in Section 5 through extensive numerical experiments. Finally, Section 6 summarizes the conclusions and future research of this study.

2. Assumptions, notations and problem statement

Consider a SAM service provider that offers door-to-door passenger transportation services using a fleet of homogeneous centrally controlled SAVs in an urban area over the operational period $[0, T]$. Let $\mathscr{R}$ and $\mathscr{V}$ denote sets of passenger requests and SAVs respectively. Passenger requests will dynamically arrive over time and space, and their information can only be known upon announcements through the SAM platform. Given the passenger request information, SAVs can be dispatched to provide pick-up and drop-off services. These requests are assumed to be ad-hoc orders, expected to be acknowledged by the SAM service platform within a specified timeframe. This acknowledgment indicates whether the requests will be confirmed to be accommodated by an SAV, although they are not required to be picked up by this confirmation deadline. In particular, we consider the SAM services that allow both solo rides and pooled rides among different passenger requests. To guarantee service quality, the satisfaction levels of pooled passenger requests are explicitly considered, reflecting their willingness to be arranged in specific pooled trips. To be applicable in an online context, the SAV dispatching plans will be generated and updated dynamically according to the latest known request information and provide services with pooled trip arrangements. Given a limited vehicle fleet size, our objective is to dispatch the SAVs in a real-time fashion to serve the dynamically arriving passenger requests during the planning horizon so as to maximize the total profit of the service operator. To fully present the RT-SAVD problem, the following three subsections will elaborate on the demand characterization, passenger satisfaction and SAV states during the dispatching procedure, respectively. The notation used throughout this study can be found in the Appendix.

2.1. Demand characterization

Each passenger request $r \in \mathscr{R}$ is characterized by request time, origin, destination, waiting time threshold for response, latest pickup and drop-off times, number of passengers, service charge and satisfaction. Specifically, all these attributes are represented by a

tuple $\left\{v_r^o, v_r^d, t_r^r, \delta_r, t_r^{lp}, t_r^{ld}, w_r\right\}$, indicating the pick-up location v_r^o, drop-off location v_r^d, request announcement time (earliest pick-up time) t_r^r, maximum waiting time δ_r to receive the response (i.e., maximum confirmation duration of service), latest pick-up time t_r^{lp} from origin, latest drop-off time t_r^{ld} at destination, and the number of passengers w_r. Note that each request can tolerate a maximum waiting time δ_r after its announcement to receive confirmation from the SAM service platform regarding the arrangement of vehicle for service. If the request r has not been arranged for any SAV by the time $\left(t_r^r + \delta_r\right)$, it will be considered canceled. Let G_r and $\widehat{G}_r = (1-\upsilon)\cdot G_r$ denote the service charge of the request r in a solo trip and a pooled trip respectively, where υ denotes the discount rate applied to the basic service charge to compensate for the requests sharing their trips with others. To maximize the operational objective under the current system state, some requests are allowed to remain unserved when they cannot be accommodated without violating service constraints (e.g., time windows, capacity, stranger limit, and satisfaction requirements) or when fleet capacity is insufficient.

Passengers sharing a trip are subject to a maximum limit on the number of ride-pooling strangers, denoted by Q. It is worth noting that even if the number of previously pooled strangers for a request does not exceed the pre-defined limit, the request may still decline a pooled trip arrangement due to satisfaction concerns. As discussed, to guarantee the level of service, each (potentially pooled) request is associated with a satisfaction level characterized by a function to incorporate passenger's acceptance of the ride-pooling options, which could be nonlinear. The satisfaction function of each request r is characterized by a value of time (VOT) p_r and a privacy-sensitivity value g_r which describes the degree of reluctance of the passenger to share with strangers. The passengers with the higher value of g_r indicate greater sensitivity to the number of strangers in the pooling arrangement. According to the empirical research by Lavieri and Bhat (2019), the passenger's acceptance of a shared ride will depend on both the increased travel times associated with the pick-up/drop-off of other passengers and the approval of strangers sharing their travel experience in the same vehicle.

We assume that passenger satisfaction largely depends on the impedances or benefits of ride-pooling in terms of the request's inherent attributes such as the VOT and privacy sensitivity and the pooling trip attributes such as the number of ride-pooling strangers, the duration of the shared ride and the additional travel time incurred due to pooling. In addition to the attributes of the concerned request, let $\overline{q}_r$ denote the number of strangers experienced by the request r in the shared trip. Therefore, the satisfaction function for a particular request r, denoted by $F_r(\cdot)$, can be defined as a multivariate function expressed as $F_r(\upsilon, p_r, g_r, \widehat{q}_r, \widehat{\zeta}_r, \widehat{\xi}_r)$, where p_r and g_r are the attributes of the concerned request, i.e., VOT and privacy-sensitivity, $\widehat{q}_r$ represents the ride-pooling stranger number, $\widehat{\zeta}_r$ corresponds to the ride-pooling duration (with other requests onboard), while $\widehat{\xi}_r$ denotes the extra (additional) travel time. To ensure service quality, a minimum passenger satisfaction threshold $\underline{F}$ is defined by the SAM service provider. That is to say, for the pooled trip arrangement, in addition to the time window, vehicle capacity and maximum ride-pooling stranger number constraint, the passenger satisfaction constraint should also be considered.

2.2. SAV states

Each SAV $h \in \mathscr{H}$ will be initialized at a specific location v_h^0 in the service area with no onboard passengers at the beginning of the period $[0, T]$. During the daily operation of an SAV $h \in \mathscr{H}$, it can be dispatched to serve multiple passenger requests simultaneously

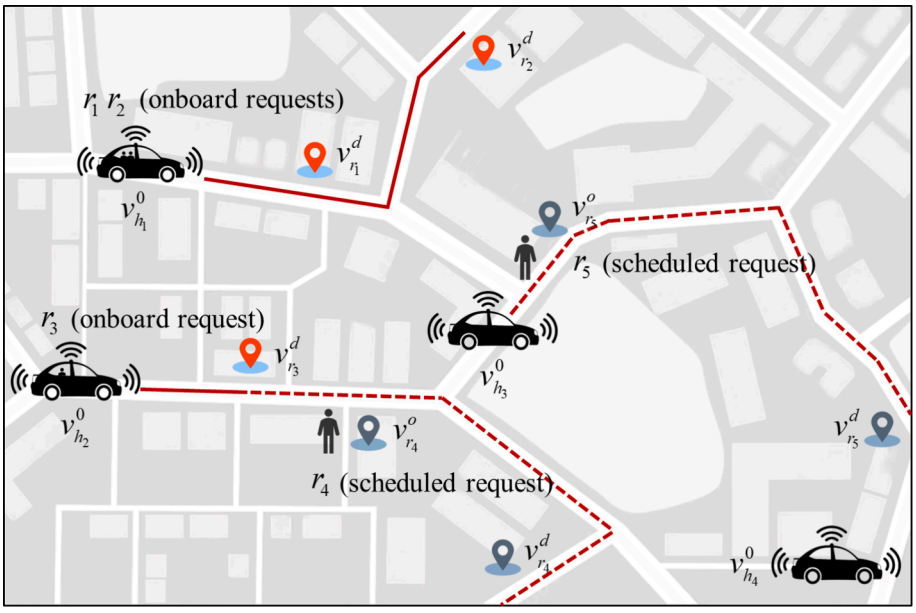

Fig. 1. SAV dispatching illustration.

subject to the carrying capacity W. Departing from the initial location, an SAV can transport many passenger requests between any pick-up and drop-off locations of these requests with the pooled version of service. Let $\tau_{v_r^o v_r^d}$ and $\kappa_{v_r^o v_r^d}$ denote travel time and incurred travel cost between the pick-up location and drop-off location of request $r \in \mathscr{R}$, respectively. Following this notation, the travel duration and travel cost from the initial location of an SAV v, i.e., v_h^0, to the pick-up location of request $r \in \mathscr{R}$ would be $\tau_{v_h^0 v_r^o}$ and $\kappa_{v_h^0 v_r^o}$, respectively.

In dynamic vehicle dispatching, an SAV may be in any of the following states: (i) transporting onboard passengers to their drop-off locations with additional scheduled requests; (ii) transporting onboard passengers to their drop-off locations without further scheduled requests; (iii) relocating (without onboard requests) to serve the next scheduled request en route; (iv) idle. Thus, the decision-making process for vehicle dispatching must consider the current states and positions of both SAVs and passenger requests in an online context. The SAV information, including their various states and positions, is continuously updated based on optimized routing plans throughout the operational period. Fig. 1 illustrates an SAV dispatching plan at a specific time, highlighting the various vehicle states involved. Red points represent drop-off nodes of onboard requests that the vehicles are already committed to serve and thus appear on the solid-line (current) routes. Blue points represent pick-up/drop-off nodes of scheduled requests, which are served along the dashed-line (planned) routes.

Given the dynamic passenger request demand, the RT-SAVD problem is to dynamically determine the optimal SAV routing plans that maximize the profit of the service operator such that: (i) each request is served by one SAV at most; (ii) the vehicle capacity and time window constraints of requests are respected; and (iii) the ride-pooling stranger number limit and passenger satisfaction threshold are not violated. The dispatching solutions of SAVs will be constantly adjusted and updated based on the information of the latest arrival of passengers.

3. RT-SAVD problem formulation

3.1. Dynamic algorithmic framework

To effectively deal with the passenger demand dynamics, we develop a dynamic vehicle dispatching algorithmic framework based on a rolling horizon approach to solving the RT-SAVD problem in temporal sequence over the entire planning horizon $[0, T]$. Instead of solving the problem over the whole planning horizon, this framework is designed to successively execute the planning at a given sequence of K time points, denoted by set $\mathscr{T} = \{t_0, t_1, t_2, \ldots, t_{k-1}, t_k, \ldots, t_K\}$, where $t_0 = 0$ and $t_K = T$. These time points are evenly spaced with the increment of a same time step Δt, i.e., $t_k = k\Delta t$, where $\Delta t = T/K$. For any iteration k at the *decision time instant* t_k, we will use a planning horizon that extends forward from the incumbent time instant and captures all currently known passenger request information. This framework aims to repeatedly slide the horizon forward by the time step Δt and address a series of S-SAVD subproblems to update the SAV dispatching solutions periodically. In particular, we impose the computation time limit λ for each S-SAVD subproblem solved at each decision time instant t_k to ensure the timely response to the passenger demand dynamics. In other words, by the time $t_k + \lambda$, a subproblem solution will be obtained for the implementation over the period $[t_k + \lambda, t_{k+1} + \lambda]$.

For each rolling iteration k, let $\mathscr{R}_k \in \mathscr{R}$ denote the set of *active* requests that can be considered in the corresponding S-SAVD subproblem at decision time instant t_k. Intuitively, these requests involve the ones considered in the previous static subproblem but have not been finalized (and can be reconsidered/rescheduled) by the time point $t_k + \lambda$ and the newly arrived passenger requests during the period $[t_{k-1}, t_k)$. To be specific, we have $\mathscr{R}_k = \left(\mathscr{R}_{k-1} \cup \widetilde{\mathscr{R}}_{k-1}\right) \backslash (\overline{\mathscr{R}}_{k-1} \cup \widehat{\mathscr{R}}_{k-1})$, where $\widetilde{\mathscr{R}}_{k-1}$ represents the set of dynamically *arrived* passenger requests during the period $[t_{k-1}, t_k)$; $\widehat{\mathscr{R}}_{k-1}$ indicates the *finished* requests that have already been served by the SAVs during the period $[t_{k-1} + \lambda, t_k + \lambda)$; and $\overline{\mathscr{R}}_{k-1}$ denotes the set of *expired* requests, including requests whose latest pick-up times fall within $[t_{k-1} + \lambda, t_k + \lambda)$ (and thus can no longer be feasibly served thereafter), and requests that have not been assigned to any vehicle by $t_r^r + \delta_r$, i.e., their maximum waiting time for confirmation has elapsed. In particular, the active requests involved in the first optimization run, i.e., $\mathscr{R}_0 = \emptyset$. In this way, the decisions made in each iteration based on the currently active requests $\mathscr{R}_k$ may be reconsidered at later decision periods as long as the associated requests have not been finished or expired, allowing the service provider to effectively handle the dynamics of the demand and obtain good-quality solutions.

For ease of presentation, we further divide the requests $\mathscr{R}_k$ in the incumbent S-SAVD problem into three sets: (1) *onboard* request set $\mathscr{O}_k$, (2) *confirmed* request set $\mathscr{C}_k$, and (3) *unscheduled* request set $\mathscr{N}_k$, i.e., $\mathscr{R}_k = \mathscr{O}_k \cup \mathscr{C}_k \cup \mathscr{N}_k$. Specifically, the set $\mathscr{O}_k$ includes the onboard passenger requests that have already been picked up and transported by specific pre-arranged SAVs. The set $\mathscr{C}_k$ includes the passenger requests that have been designated to be served by specific pre-assigned SAVs in the last rolling-horizon computation but have not been boarded yet. They have been confirmed service and cannot change their corresponding arranged SAVs but can still be rescheduled. The set $\mathscr{N}_k$ includes the passenger requests that are received within the current or the preceding time interval whose waiting times have not yet reached their maximum confirmation duration limit, thus allowing them to be (re)arranged.

In addition to the request information, the information for each SAV $h \in \mathscr{H}$, including the current vehicle location v_h, arrival time at the current location $\widehat{t}_h$, set of pre-assigned onboard and confirmed requests $\mathscr{S}_{k,h}$ and corresponding scheduled partial route information $\Theta_{k,h}$ consisting of a series of sequential pick-up and drop-off locations for requests in $\mathscr{S}_{k,h}$, is another important input to the incumbent S-SAVD sub-problem. Specifically, according to the SAV dispatching solution in iteration $(k-1)$, each vehicle might be in the course of being relocated to serve another passenger request, under the service of a trip with the solo request or pooled requests, or in the idle state at the end of the optimization run period for the incumbent S-SAVD sub-problem, i.e., time instant $t_k + \lambda$. Considering

the central-controlled characteristic of SAVs in the SAM services, we assume that a vehicle can be flexibly diverted to serve another passenger request if the vehicle either during the relocation operation to another passenger request or under the service for a request/pooled requests, i.e., with onboard passengers, by the time $t_k + \lambda$ as long as the number of accumulated pooled strangers for each concerned request does not exceed the pre-defined limit while the vehicle capacity constraint is respected. It worth noting that although SAVs may flexibly accommodate additional requests after assignment, the hard constraint on the latest pickup time and confirmation mechanism ensures that passengers will not suffer excessive or unreasonably long waits, thus balancing system flexibility and user expectations. The future dispatching plans of each SAV should consider the previous schedules related to the fixed arrangement of its onboard requests and confirmed requests. Therefore, once a solution to the S-SAVD subproblem in iteration $(k-1)$ is obtained and implemented at the time instant $t_k + \lambda$, the current vehicle location v_h, arrival time at the current location t_h, previously arranged request $\mathscr{S}_{k,h}$ as well as the corresponding scheduled partial route information $\Theta_{k,h}$ for the S-SAVD subproblem in iteration k will be known. In particular, for the first optimization run, the information of each SAV $h \in \mathscr{H}$ will be initialized as $v_h = v_h^0$, $\widehat{t}_h = t_0 + \lambda$, $\mathscr{S}_{0,h} = \varnothing$ and $\Theta_{0,h} = \varnothing$.

Given the information of the involved passenger requests $\mathscr{R}_k$ and the current state of each SAV $h \in \mathscr{H}$, the S-SAVD subproblem will be solved to determine the routing plans of SAVs in $\mathscr{H}$ to serve the request/request pairs in $\mathscr{R}_k$ so as to maximize the total profit of the SAM service operator over the current planning horizon. Each S-SAVD subproblem will be solved to guide the vehicle dispatching solutions update in response to incoming passenger requests during the period $(t_{k-1}, t_k]$. More and more passenger requests will be finalized to be served or expired without service as the operational horizon rolls forward. The real-time response to the newly arrived passenger requests can be realized by the short rolling time step Δt in the proposed dynamic vehicle dispatching rolling horizon framework.

3.2. S-SAVD subproblem model

According to the dynamic vehicle dispatching framework, the key is to efficiently address the S-SAVD subproblem. In this section, we formulated an MIP model for the S-SAVD subproblem to maximize the system profit while respecting the constraints on the time window, SAV capacity, maximum ride-pooling stranger number and passenger satisfaction. Consider the incumbent S-SAVD problem in iteration k with input of active passenger requests $\mathscr{R}_k = \mathscr{O}_k \cup \mathscr{C}_k \cup \mathscr{N}_k$ and current information of SAVs $\mathscr{H}$. For model building, we define a directed network $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where $\mathscr{V}$ is the set of nodes and $\mathscr{E}$ is the set of edges. Considering the different types of sets of requests, the set $\mathscr{V}$ is partitioned as $\mathscr{V} = \mathscr{V}\ominus_1 \cup \mathscr{V}\ominus_2 \cup \mathscr{V}\ominus_3 \cup \mathscr{V}\ominus_4$. $\mathscr{V}\ominus_1$ is the subset of nodes of the current vehicle locations, where $v_h \in \mathscr{V}\ominus_1$ denotes the node associated with the location of SAV h. $\mathscr{V}\ominus_2$ is the subset of the nodes denoting the drop-off locations of requests in set $\mathscr{O}_k$. $\mathscr{V}\ominus_3$ is the subset of the nodes denoting the pick-up locations of requests in sets $\mathscr{C}_k$ and $\mathscr{N}_k$. $\mathscr{V}\ominus_4$ is the subset of the nodes denoting the drop-off locations of requests in sets $\mathscr{C}_k$ and $\mathscr{N}_k$. Set $\mathscr{V}$ for all nodes will vary over time and have updated before the incumbent optimization invoking time t_k. Note that v_r^o and v_r^d denote the pick-up and drop-off node (index) of passenger request r. Each edge $(i,j) \in \mathscr{E}$ is associated with a travel time τ_{ij} and incurred travel cost κ_{ij}. For each request $r \in \mathscr{R}\ominus_k$, in addition to the request announcement time t_r^r, latest pick-up time t_r^{lp}, latest drop-off time t_r^{ld}, passenger number w_r, revenue of pooling $\widehat{G}_r$ and without pooling G_r, let q_r^* denote the recorded number of strangers that have previously shared with request r known by the time $t_k + \lambda$. Specifically, $q_r^* \geq 0, \forall r \in \mathscr{O}_k$ and $q_r^* = 0, \forall r \in \mathscr{C}_k \cup \mathscr{N}_k$. Additionally, for each request $r \in \mathscr{O}_k$, let h_r denote the SAV that is pre-arranged to the request r, ς_r^* denote the recorded (ride-pooling) duration (with other requests onboard) of the implemented (shared) trip up to the time $t_k + \lambda$ and t_r^* denote the past pick-up time instant in the (shared) trip. Note that confirmed and unscheduled requests have not been picked up by any vehicle and have no previously implemented pooling arrangement information.

We define binary decision variable x_{ij}^{rh} to denote whether request r travers on edge (i,j) onboard vehicle h; binary decision variable y_{ij}^h to denote whether the vehicle h travers on edge (i,j); binary decision variable d_{rh} to denote whether request r is picked up by vehicle h; binary decision variable z_r to denote whether request r is pooled with any other request during its trip; non-negative variable q_r to denote the number of strangers in the shared trip of request r (obtained in the incumbent subproblem); binary decision variable $\iota_{rr'h}$ to denote whether request r shares a ride with requests r' on vehicle h and this encounter is counted as new stranger request in this subproblem optimization epoch (i.e., it has not occurred in any earlier epoch); and non-negative variable u_{hi} to denote the time at which vehicle h reaches node i. Then the S-SAVD subproblem can be formulated as the following mixed-integer programming model:

[S-SAVD]

$$\max_{\{\boldsymbol{x},\boldsymbol{y},\boldsymbol{d},\boldsymbol{z},\boldsymbol{q},\boldsymbol{u}\}} \sum_{h\in\mathscr{H}}\sum_{r\in\mathscr{R}_k} \left(\widehat{G}_r z_r + G_r d_{rh}(1-z_r)\right) - \sum_{h\in\mathscr{H}}\sum_{i\in\mathscr{V}}\sum_{j\in\mathscr{V}} \kappa_{ij}\cdot y_{ij}^h \tag{1}$$

subject to

$$\sum_{j\in\mathscr{V}} x_{v_{hr},j}^{rh_r} = 1, \forall r \in \mathscr{O}_k \tag{2}$$

$$\sum_{j\in\mathscr{V}} x_{v_r^o,j}^{rh_r} = 1, \forall r \in \mathscr{C}_k \tag{3}$$

$$\sum_{j\in\mathscr{V}} x^{rh}_{j,v^d_r} = 1, \forall r \in \mathscr{O}_k \cup \mathscr{C}_k \tag{4}$$

$$\sum_{j\in\mathscr{V}} x^{rh}_{j,v^o_r} = \sum_{j\in\mathscr{V}} x^{rh}_{v^o_r,j} - d_{rh}, \forall r \in \mathscr{N}_k, h \in \mathscr{H} \tag{5}$$

$$\sum_{j\in\mathscr{V}} x^{rh}_{j,v^d_r} = \sum_{j\in\mathscr{V}} x^{rh}_{v^d_r,j} + d_{rh}, \forall r \in \mathscr{N}_k, h \in \mathscr{H} \tag{6}$$

$$\sum_{j\in\mathscr{V}} x^{rh}_{ji} = \sum_{j\in\mathscr{V}} x^{rh}_{ij}, \forall i \in \mathscr{V}\backslash\left(\mathscr{V}_1 \cup \left\{v^o_r, v^d_r\right\}\right), r \in \mathscr{R}_k, h \in \mathscr{H} \tag{7}$$

$$\iota_{rr'h} \le 1 - \rho_{rr'}, \forall r \in \mathscr{R}_k, r \ne r', h \in \mathscr{H} \tag{8}$$

$$\sum_{j\in\mathscr{V}}\sum_{i\in\mathscr{V}} x^{rh}_{ij} x^{r'h}_{ij} \le M\iota_{rr'h}, \forall r \in \mathscr{R}_k, r \ne r', h \in \mathscr{H} \tag{9}$$

$$q_r = \sum_{h\in\mathscr{H}}\sum_{r'\in\mathscr{R}\backslash r\ominus} \iota_{rr'h} w_{r'} \le Q - q^*_r, \forall r \in \mathscr{R}_k \tag{10}$$

$$F_r\left(v, p_r, g_r, q_r, \sum_{h\in\mathscr{H}}\sum_{r'\in\mathscr{R}\backslash r\ominus}\sum_{j\in\mathscr{V}}\sum_{i\in\mathscr{V}} x^{rh}_{ij} x^{r'h}_{ij}\tau_{ij}, \sum_{h\in\mathscr{H}} d_{rh}\left(u_{hv^d_r} - u_{hv^o_r} - \tau_{v^o_r v^d_r}\right)\right) \ge \underline{F} z_r, \forall r \in \mathscr{C}_k \cup \mathscr{N}_k \tag{11}$$

$$F_r\left(v, p_r, g_r, q_r + q^*_r, \zeta^*_r + \sum_{h\in\mathscr{H}}\sum_{r'\in\mathscr{R}\backslash r\ominus}\sum_{j\in\mathscr{V}}\sum_{i\in\mathscr{V}} x^{rh}_{ij} x^{r'h}_{ij}\tau_{ij}, \sum_{h\in\mathscr{H}} d_{rh}\left(u_{hv^d_r} - t^*_r - \tau_{v^o_r v^d_r}\right)\right) \ge \underline{F} z_r, \forall r \in \mathscr{C}_k \tag{12}$$

$$\sum_{r\in\mathscr{R}} x^{rh}_{ij} w_r \le W_h y^h_{ij}, \forall i, j \in \mathscr{V}, h \in \mathscr{H} \tag{13}$$

$$z_r = \min\left\{1, q_r + q^*_r\right\}, \forall r \in \mathscr{R}_k \tag{14}$$

$$\sum_{h\in\mathscr{H}} d_{rh} \le 1, \forall r \in \mathscr{N}_k \tag{15}$$

$$\sum_{j\in\mathscr{V}} y^h_{ji} \le 1, \forall i \in \mathscr{V}, h \in \mathscr{H} \tag{16}$$

$$\sum_{j\in\mathscr{V}} y^h_{ij} \le 1, \forall i \in \mathscr{V}, h \in \mathscr{H} \tag{17}$$

$$\sum_{j\in\mathscr{V}} y^h_{ij} \le \sum_{j\in\mathscr{V}} y^h_{ji}, \forall i \in \mathscr{V}\backslash\mathscr{V}_1, h \in \mathscr{H} \tag{18}$$

$$\sum_{j\in\mathscr{V}} y^h_{ji} \le 0, \forall i \in \mathscr{V}_1, h \in \mathscr{H} \tag{19}$$

$$\sum_{h\in\mathscr{H}}\sum_{j\in\mathscr{V}} y^h_{ij} \le 1, \forall i \in \mathscr{V}_1 \tag{20}$$

$$u_{hj} - u_{hi} \ge \tau_{ij} - M\left(1 - y^h_{ij}\right), \forall i, j \in \mathscr{V}, h \in \mathscr{H} \tag{21}$$

$$t^r_r d_{rh} \le u_{hv^o_r} \le t^{lp}_r d_{rh}, \forall r \in \mathscr{C}_k \cup \mathscr{N}_k, h \in \mathscr{H} \tag{22}$$

$$u_{hv^d_r} \le t^{ld}_r d_{rh}, \forall r \in \mathscr{R}_k, h \in \mathscr{H} \tag{23}$$

$$x^{rh}_{ij} \in \{0,1\},\ y^h_{ij} \in \{0,1\},\ d_{rh} \in \{0,1\},\ z_r \in \{0,1\},\ q_r \ge 0, u_{hi} \ge 0,\ \forall i, j \in \mathscr{V}, r \in \mathscr{R}_k, h \in \mathscr{H} \tag{24}$$

$$\rho_{rr'} \in \{0,1\},\ \forall r \in \mathscr{R}_k, r' \ne \mathscr{R}\backslash\{r\}, h \in \mathscr{H} \tag{25}$$

The objective function (1) is formulated to maximize the total profit of the SAM service, calculated as the total revenue of the involved requests minus the operating costs of the SAV fleet. Constraint (2) ensures that onboard requests in set carried by the corresponding

pre-arranged SAVs leave their current locations. Constraint (3) guarantees that all confirmed requests in are picked up from their pick-up locations by their pre-arranged SAVs. Constraint (4) ensures that both onboard and confirmed requests in are delivered to their designated drop-off locations by their pre-arranged SAVs. Constraints (5) and (6) guarantee that once an unscheduled request is assigned to a specific SAV, it will be picked up from the designated pick-up location and finally transported to the drop-off location. Constraint (7) expresses the flow balance indicating that each request entering the node (other than its pick-up/currently boarded vehicle location and drop-off location) will always leave from the same node.

We further introduce a known carry-over indicator at the beginning of each epoch , i.e., binary variable and means request has already encountered another request in an eaelier epoch (hence, should not be counted again as a "new stranger" for in epoch). Therefore, Constraint (8) ensures that if shares with the same as in the previous time step (so), then and this co-rider contributes zero to the new stranger count in the current epoch. Constraint (9) links the shared-arc measure, i.e., sharing occurs, to the counting indicator, where is a valid upper bound (e.g., or the maximum number of arcs in a feasible route). Constraint (10) imposes the limit of the maximum ride-pooling stranger number for each request. In addition, the update of the carry-over indicator after the epoch is , which is used for the input for the next epoch . Constraints (11) and (12) ensure the satisfaction threshold of pooled request in their shared trips are not exceeded. Constraint (13) guarantees that the number of passengers carried by each SAV remains within the vehicle capacity limit. Constraint (14) ensures that each request is accommodated in a pooled trip sharing with at least one stranger. Constraint (15) means no unscheduled request may be served by more than one SAV. Constraints (16) and (17) guarantee that an SAV cannot simultaneously arrive at a node from multiple sources or depart from a single source to multiple nodes. Constraint (18) guarantees that each SAV will either traverse the node (i.e.,) or end its trip at the drop-off node (i.e.,) after starting from the current location. Constraint (19) guarantees that each SAV can only leave from its current location without revisiting any current locations of all the vehicles. Constraint (20) ensures that only one SAV is permitted to depart from any existing vehicle location within set . Constraint (21) defines the time instant of each SAV visiting each node, where denotes a sufficiently large value. Time window conditions for each request are specified in Constraints (22) and (23). Constraint (24) and Constraint (25) defines the feasible domain of each variable.

4. Solution method

As discussed, we can easily obtain the overall dynamic vehicle dispatching solutions if we can address each static subproblem periodically. Thus, we will not detail the derivation of the dynamic solution based on continuously updated static vehicle dispatching plans. To efficiently solve the [S-SAVD] model, we develop a customized hybrid algorithm, referred to as the ARA-LNS, which integrates an ARA scheme into the LNS heuristic framework. The LNS, widely utilized in many vehicle routing problems including variants like DARP, iteratively improves a solution by destroying and repairing the current one by exploring a large solution space through removal and insertion operators. Despite its efficacy for vehicle routing-related problems, it is still not very efficient to directly use the LNS for solving the multiple-vehicle dispatching problem that requires high computational speed considering the real-time context and large problem scale. To this end, we extend the traditional LNS algorithm by incorporating an ARA scheme to decompose the optimization of the multiple-SAV dispatching problem into several single-SAV dispatching problems. The LNS aims to effectively identify the optimal routing solution for each SAV. The embedded ARA scheme involves the selection and application of various assignment operators to assign passenger requests to suitable SAVs, thus filtering the requests to be scheduled for each SAV. In particular, the selection probabilities of assignment operators will be updated adaptively based on the solution's past performance. This assignment-based request mechanism not only efficiently guides the search for the optimal solution within the solution space but also enables parallel optimization for single-SAV dispatching problems, thus accelerating the overall optimization process.

4.1. ARA-LNS algorithm structure

Given the current request and SAV information, the ARA-LNS algorithm begins by constructing initial routing solutions to accommodate unscheduled requests based on the previously scheduled partial routes (for onboard and confirmed requests) of each SAV. In each iteration, the LNS heuristic is employed to improve the current routing solution of each SAV individually by using removal and insertion operators. During the search process, if the total objective of the single SAV dispatching solutions has not improved for a certain number of iterations during the search process, the ARA scheme will be invoked to execute the reassignment process for all removed requests by selecting and applying different assignment operators. Subsequently, each SAV routing solution will be improved based on the newly assigned request set until the assignment scheme is called again. The newly generated SAV routing solutions will be evaluated to be potentially accepted as the incumbent solution and optimal solution. Furthermore, the performance of these SAV routing solutions will also provide feedback to adaptively adjust the selection probability of assignment operators in the ARA scheme. This iterative procedure will continue until a specific stop criterion is reached.

The pseudocode of the proposed ARA-LNS algorithm is outlined in **Algorithm 1**. Before initialization, the potentially available SAVs for each unscheduled request $\mathcal{N}_k$ are filtered based on the current vehicle information in set $\mathcal{H}$. We suppose that all SAVs can divert to directly serve unscheduled requests. For each unscheduled request, the pick-up time window requirement will be verified for all SAVs, disregarding other constraints, to roughly filter out the impossible vehicle arrangements. This process generates the possible request-SAV mapping Ω to define the assignment of each request to possible SAVs (see Line 1). Essentially, this mapping context reflects that each unscheduled request may have multiple vehicle choices to be arranged, which will set the foundation for the subsequent request assignment procedures. Then, the algorithm generates the initial solution s_0 based on our proposed assignment operators and greedy insertion heuristic, which is described in detail in Subsection 4.2. For ease of presentation, the solution s_0 is defined

to be composed of $s_0^h \ominus$, $\forall h \in \mathscr{H}$, where the superscript h is added in the notation to denote the single-SAV solution. The incumbent solution $s^h \ominus$ and optimal solution $s_b^h \ominus$ for each SAV are initialized to be the initial solution $s_0^h \ominus$, and the corresponding solutions for all SAVs, i.e., s and s_b, can be obtained accordingly (see Lines 2–3). We will also initialize the selection probability of the assignment operators denoted by $\mathscr{A}$. Meanwhile, both the iteration number counter, denoted by N, and the counter of the total number of consecutive iterations that an assignment operator does not achieve a better solution, denoted by μ, are initialized to be 0 (see Line 4).

Algorithm 1. **Pseudocode of the ARA-LNS algorithm.**

Input: Active request set $\mathcal{R}_k = \mathcal{O}_k \cup \mathcal{C}_k \cup \mathcal{N}_k$, vehicle set $\mathcal{H}\left(v_h, t_h, \mathcal{S}_{k,h}, \Theta_{k,h}\right)$

Output: Optimal SAV dispatching solution s_b

1 Filter possible request-SAV mapping Ω;
2 Initialize s_0 composed of s_0^h, $s^h \leftarrow s_0^h$, $s_b^h \leftarrow s_0^h$;
3 $s \leftarrow \left\{s^h\right\}_{h\in\mathcal{H}}$; $s_b \leftarrow \left\{s_b^h\right\}_{h\in\mathcal{H}}$; // initialize the incumbent and best solution
4 $N \leftarrow 0$, $\mu \leftarrow 0$, initialize selection probability of assignment operators $\mathcal{A}$;
5 **While** $\neg$ ($N == N_{\max}$ $\vee$ elapsed CPU time exceeds $U_{\max}$) do
6 **For** each $h \in \mathcal{H}$ do
7 $s_d^h \leftarrow Removal(s^h)$ and put removed requests into $\mathcal{M}_h$;
8 **EndFor**
9 **If** $\mu > \mu_{\max}$, then // whether re-assign the request
10 Mix the removed requests $\mathcal{M} \leftarrow \left\{\mathcal{M}_h\right\}_{h\in\mathcal{H}}$, $\mu \leftarrow 0$;
11 $\left\{\mathcal{M}_h\right\}_{h\in\mathcal{H}} \leftarrow \boldsymbol{ARA}\left(\mathcal{A}, \mathcal{M}, \left\{s_d^h\right\}_{h\in\mathcal{H}}\right)$;
12 **If** the end of every φ times of assignment, then // assignment segment
13 $\mathcal{A} \leftarrow \boldsymbol{PrbUpd}(\mathcal{A})$ and initialize scores;
14 **EndIf**
15 **EndIf**
16 **For** each $h \in \mathcal{H}$ do
17 $s^{h'} \leftarrow Insertion(s_d^h, \mathcal{M}_h)$; // obtain (repaired) new solution;
18 **EndFor**
19 $s' \leftarrow \left\{s^{h'}\right\}_{h\in\mathcal{H}}$;
20 **If** s' improves upon s, then
21 $\mu \leftarrow \mu + 1$;
22 **EndIf**
23 **If** $AcpIcm(s', s)$ is true, then
24 $s \leftarrow s'$;
25 **EndIf**
26 **If** $AcpOpt(s', s_b)$ is true, then
27 $s_b \leftarrow s'$;
28 **EndIf**
29 $\mathcal{A} \leftarrow \boldsymbol{ScrUpd}(\mathcal{A})$; // Update scores of the selected assignment operator;
30 $N \leftarrow N + 1$;
31 **EndWhile**

After initialization, the algorithm loop is launched to iteratively improve the incumbent solution for each SAV by implementing the ARA-embedded LNS (see Lines 5–31). The removal operator is used to the incumbent solution for each SAV h to generate a destroyed solution s_d^h and the removed requests are stored in set $\mathscr{M}_h$ (see Lines 6–8). When the non-improvement counter μ is larger than the maximum number of consecutive attempts $\mu_{\max}$, the request reassignment process of the ARA scheme will be triggered (see Lines 9–15), which is detailed in Subsection 4.3. This assignment process operates on the mixed removed requests in set $\mathscr{M}$ using five

assignment operators $\mathscr{A}$ to update the request bank $\mathscr{M}_h$ for each $h \in \mathscr{H}$ and the non-improvement counter μ is reset to 0. (see Lines 10–11). Additionally, $PrbUpd(\bullet)$ is the subfunction used to update the selection probability of assignment operators based on their historical performance after the ARA scheme has been called for φ times (see Line 13). Then the insertion operator will be used to select the requests in $\mathscr{M}_h$ for each SAV h and insert them into its previously destroyed solution, thus obtaining the new solution s' composed of $s^{h'} \ominus, \forall h \in \mathscr{H}$ (see Lines 16–19). If the solution fails to find a better solution in the incumbent iteration, the non-improvement counter μ will be incremented by 1 (see Lines 20–22).

As for the solution evaluation, two evaluation subfunctions $AcpIcm(\bullet)$ and $AcpOpt(\bullet)$ are used to determine whether to accept the new solution as the incumbent solution and optimal solution respectively (see Lines 23–28). The subfunctions will return 'true' if specific criteria are satisfied. The subfunction $AcpIcm(\bullet)$ is defined to be the simulated annealing accept criterion, which accepts a solution $s^{h'} \ominus$ given the incumbent solution $s^h \ominus$ with probability $\exp\left[-\left(f_{profit}(s^h \ominus) - f_{profit}(s^{h'} \ominus)\right)/\ell\right]$, where $f_{profit}(\bullet)$ denotes the objective value for the particular solution, obtained from the objective function expressed by Eq. (1), and $\ell > 0$ is the temperature. The initial temperature is determined based on the method proposed by Ropke and Pisinger (2006) and the temperature will be iteratively decreased within the simulated annealing process based on a cooling rate $0 < \nu < 1$ such that $\ell \leftarrow \nu\ell$. The subfunction $AcpOpt(\bullet)$ is defined to accept the new solution as the optimal one if it is better than the optimal one, i.e., $f_{profit}(s^{h'}) > f_{profit}(s_b^h \ominus)$. $ScrUpd(\bullet)$ is the subfunction used to update the score of the currently selected assignment operator based on the newly generated solution (see Line 29). The iteration number N is increased by one at the end of each loop (see Line 30). The loop terminates when it either reaches the maximum iteration count, denoted by $N_{\max}$, or exceeds the CPU time threshold, represented by $U_{\max}$ (see Line 5).

In what follows, we will elaborate on the SAV dispatching routing plan generation and the design of the ARA scheme in Subsections 4.2 and 4.3. respectively.

4.2. SAV dispatching plan generation

Initial solution

We will use our proposed adaptive assignment operators and a simple greedy heuristic to construct an initial solution based on the partial routes of all SAVs. To construct the initial solution in a fast way, we will first use each assignment operator (which will be elaborated in detail in Subsection 4.3) to allocate unscheduled requests among SAVs. Then, for each assignment operator, we will construct the initial solution for each vehicle individually based on its partial route. Since LNS is not highly sensitive to the initial solution, we use a simple greedy insertion heuristic to generate the initial routing solution for each SAV. Specifically, the assigned unscheduled requests are sorted in the ascending order of their latest pick-up time and are inserted in the current constructed route of the corresponding vehicle according to this order. Each request will be inserted into the position that brings the maximum profit improvement under all the constraints stated in Section 4. For example, the scheduled partial route solution of an SAV $h \in \mathscr{H}$ is represented as $\Theta_{k,h} := \left(v_{r_1}^d, v_{r_2}^d, v_{r_3}^o, v_{r_3}^d\right)$, indicating the sequence of pick-up and drop-off nodes of the pre-assigned onboard requests r_1, $r_2 \in \mathscr{S}_{k,h}$ and confirmed request $r_3 \in \mathscr{S}_{k,h}$. For the assigned unscheduled r_4, we try to insert the $v_{r_4}^o$ and $v_{r_4}^d$ in all possible positions that do not change the current ordering, i.e., $\left(v_{r_4}^o, v_{r_4}^d, v_{r_1}^d, v_{r_2}^d, v_{r_3}^o, v_{r_3}^d\right)$, $\left(v_{r_4}^o, v_{r_1}^d, v_{r_4}^d, v_{r_2}^d, v_{r_3}^o, v_{r_3}^d\right)$, …. The profit improvement of each feasible insertion can be computed and the insertion that yields the highest profit improvement will be selected.

Removal and insertion procedure

In each search iteration, the removal and insertion operators in the LNS heuristic will be applied to generate a new routing solution in the neighborhood of the incumbent solution for each SAV $h \in \mathscr{H}$. The removal operator randomly selects and removes the m percentage of requests from the current routing solution to diversify the search. In order to ensure the service of the pre-arranged onboard and confirmed requests, we have $\mathscr{S}_{k,h} \notin \mathscr{M}_h$. In other words, these removed requests, which are grouped into set $\mathscr{M}_h$, only comprising unscheduled requests in $\mathscr{N}_k$ and become candidates to be selected and reinserted into the destroyed solution. We employ a greedy insertion operator to construct a new routing plan for SAV h based on the destroyed solution. This operator performs iteratively to insert one candidate request from the removed request set $\mathscr{M}_h$ at a time to repair the destroyed solution. Each request is associated with a *best insertion position* in the current repaired routing solution, in which this insertion in the route brings the maximum profit improvement while ensuring the feasibility of the solution. In each iteration, the request with the highest profit improvement will be chosen and inserted in the route at its best insertion position. This process repeats until all requests are considered. Note that if the best insertion position for a request results in a profit decrease, the request will be removed completely, indicating that the service of this request will be rejected to make the service profitable.

Our proposed insertion heuristic will be implemented by additionally checking the feasibility of the ride-pooling arrangement. For example, let us consider the insertion of a request $r \in \mathscr{M}_h$ into a partially constructed routing solution Υ. Intuitively, we will check all the possible insertion positions for both pick-up and drop-off nodes of request r, iterating from the first position till the last one in the possible insertion position sequence. For a specific pick-up node insertion position, we will first examine whether inserting this drop-off node insertion would exceed the ride-pooling stranger number constraint for all potentially influenced requests or vehicle capacity constraints. If either constraint is violated, the checking process for the following drop-off node insertion positions can be stopped, as all the subsequent drop-off node insertion positions will definitely lead to violations. If no violations occur, we will proceed to assess the feasibility of time windows and check the passenger satisfaction with the proposed insertion.

Following the above idea, the procedure of inserting the candidate request $r \in \mathscr{M}_h$ into the partial routing solution Υ with ride-pooling feasibility checking is outlined in Algorithm 2. For ease of presentation, let I denote the last possible insertion position for

evaluating the insertion of the pick-up node of request r and let $\mathscr{C}_r$ denote the set of previously shared requests of request r. For each combination of the potential insertion positions, i.e., i and j, of the pick-up node and drop-off node, a new solution Υ_r^{ij} is generated and the corresponding profit improvement Δf_r^{ij} is initialized to $-\infty$ (see Lines 1–6). The potentially affected sequence partial route $\overline{\Upsilon}_r^{ij}$ will be identified to check the feasibility of capacity and ride-pooling stranger number constraints (see Lines 7–17). If the time window and passenger satisfaction constraints are not violated, the profit improvement Δf_r^{ij} for this insertion will be updated. The insertion position (i^*, j^*) that yields the highest profit improvement $\Delta f_r^{i^*j^*}$ is selected to insert request r into Υ, thus generating new partial solution $\Upsilon_r^{i^*j^*}$ (see Line 24). Let $TT^{(v_1,\ldots,v_n)}$ denote the total travel time when an SAV sequentially arrives nodes $(v_1, \ldots, v_n)$. In mathematical terms, a request r can be served by vehicle h in the newly generated routing solution only if the time window constraints $t_h + TT^{(v_h,\ldots,v_r^o)} \le t_r^{lp}$ and $t_h + TT^{(v_h,\ldots,v_r^d)} \le t_r^{ld}$ are respected. For satisfaction feasibility constraints, it is essential to consider previous pooling information regarding the accumulated pooled duration ζ_r^* and pick-up time t_r^* of the previously unfinished pooled trip in the satisfaction calculation for onboard requests.

Algorithm 2 . Insertion algorithm with ride-pooling feasibility check.

1 **For** $i \in \{1, 2, ..., I\}$ do
2 Insert v_r^o to Υ in positions i and generate Υ_r^i;
3 **If** time window constraint is violated for v_r^o, then
4 Break;
5 **EndIf**
6 **For** $j \in \{i+1, i+2, ..., I+1\}$ do
7 Insert v_r^d to Υ_r^i in positions j and generate Υ_r^{ij}; $\Delta f_r^{ij} \leftarrow -\infty$;
8 $\overline{\Upsilon}_r^{ij} \leftarrow \left(v_r^o, ..., v_r^d\right)$; // potentially affected node sequence
9 **For** v in $\overline{\Upsilon}_r^{ij}$ do
10 **If** capacity constraint at node v is not respected, then
11 Break;
12 **EndIf**
13 Set r_v as the associated request of node v; $\mathcal{Q}_{r_v}^{ij} \leftarrow \mathcal{Q}_{r_v}$;
14 Add r_v in $\mathcal{Q}_{r_v}^{ij}$ if r_v is not in $\mathcal{Q}_{r_v}^{ij}$;
15 **If** $\left|\mathcal{Q}_{r_v}^{ij}\right| > Q$, then
16 Break;
17 **EndIf**
18 **EndFor**
19 **If** Υ_r^{ij} is feasible for time window and passenger satisfaction constraints, then
20 $\Delta f_r^{ij} \leftarrow f_{profit}(\Upsilon_r^{ij}) - f_{profit}(\Upsilon)$
21 **EndIf**
22 **EndFor**
23 **EndFor**
24 $(i^*, j^*) \leftarrow \arg\max \left\{\Delta f_r^{ij}\right\}_{i \in \{1,2,...,I\}, j \in \{i+1, i+2, ..., I+1\}}$; // best insertion position
25 Return $\Upsilon_r^{i^*j^*}$, $\Delta f_r^{i^*j^*}$

4.3. Adaptive request assignment scheme

As we have mentioned earlier, the adaptive request assignment scheme will be invoked when no further improvement of the total profit of all SAVs' routing solutions can be achieved for a certain number of iterations. Different assignment operators will be selected according to their respective selection probabilities and then utilized for allocating each unscheduled request to an SAV. Note that the selected SAV for a request must be among the filtered possible SAVs assigned to this request, as indicated in the mapping context.

Assignment operators

Five assignment operators will be proposed to determine the suitable assignment of each unscheduled request in the set of mixed removed requests $\mathscr{M}$ to a specific vehicle. These operators will define different measures for assignment and the request bank of each vehicle, i.e., $\mathscr{M}_h$, $\forall \ominus \in \mathscr{K}$, will be updated after checking all unscheduled requests using the assignment operator.

A1 (random assignment): This operator randomly assigns each request to an SAV to improve the diversification of the search.

A2 (distance assignment): This operator aims to assign each unscheduled request to the nearest SAV. A distance level is defined to measure the spatial closeness between a request and an SAV by calculating the distance between the current location of an SAV and the origin of the concerned request. For each request, this distance criterion will be used to obtain the distance level of all SAVs. The SAV with the lowest distance level will be chosen to determine the assignment of this request.

A3 (violation assignment): This operator is designed to assign each unscheduled request based on the overlapping length of pick-up time windows among all the requests pre-assigned to the SAV. A violation level is introduced to measure the average overlap between the concerned request and all other requests assigned to an SAV. A smaller overlapping length between the time windows of requests suggests a higher potential for inserting this request into the route (possibly facilitating ride-pooling with other requests) while avoiding time infeasibilities and improving the routing solutions for vehicles. The violation level of a request r to another request r' concerning the overlap of the pick-up time window is calculated as

$$\theta_r^v(r') = \frac{\max\{0, \min\{t_r^{lp}, t_{r'}^{lp}\} - \max\{t_r^e, t_{r'}^e\}\}}{t_r^{lp} - t_r^e} \tag{26}$$

The violation level of assigning a request r to the SAV h can be calculated as

$$\theta_{rh}^v = \frac{\sum_{r' \in \widetilde{\mathscr{S}}_h} \theta_r^v(r')}{\left|\widetilde{\mathscr{S}}_h\right|} \tag{27}$$

where $\widetilde{\mathscr{S}}_h$ denotes the currently assigned requests to the SAV h excluding the onboard requests. Each request will be assigned to the vehicle with the minimum violation level. Note that if there is currently no assigned request to the vehicle, the violation level is set to 0.

A4 (matching assignment): Inspired by the study of Hou et al. (2018), this operator aims to assign an unscheduled request to an SAV by evaluating the matching level among its pre-assigned requests. We first define the two-request matching level measurement as

$$\theta_{rr'}^m = \frac{\min\left\{l_{v_r^o v_r^d}, l_{v_{r'}^o v_{r'}^d}\right\}}{T_{\min}(rr')} \tag{28}$$

where $T_{\min}(rr')$ denotes the feasible trip minimum travel distance among all the feasible trips that connect the pick-up and drop-off nodes of two requests under the time window and customers' satisfaction constraints (if any). It can be seen that $\theta_{rr'}^m$ can measure the rerouting change that should be made when inserting the request r' into the existing route with request r already arranged. If no feasible trip can be found, the value of the matching level will be set to 0.

The two-request matching measurement can be extended to evaluate the matching level of assigning an unscheduled request r to an SAV h, denoted by θ_{rh}^m. It is determined by the average matching level with all assigned requests to the concerned vehicle and expressed as follows:

$$\theta_{rh}^m = \frac{\sum_{r' \in \widetilde{\mathscr{S}}_h} \theta_{rr'}^m}{\left|\widetilde{\mathscr{S}}_h\right|} \tag{29}$$

where $\widetilde{\mathscr{S}}_h$ denotes the currently assigned requests for SAV h. Each request will be assigned to the vehicle with the maximum matching level. Note that although this operator only measures the two-request pooling trips for matching level evaluation, it is expected that this matching-level-oriented assignment still finds out requests for each SAV that have the potential to accept more shared requests during their trips and sets the foundation for generating better routing solutions.

A5 (observation-based assignment): This operator aims to assign each unscheduled request to an SAV based on historical experience gained from the previous performance of the algorithm. Specifically, for every request assigned to an SAV, the operator keeps track of a limited collection of objective values over the latest several iterations, i.e., $\mathscr{K} = \left\{f_1^h, \ldots, f_{|\mathscr{K}|}^h\right\}$. The observation level of assigning request r to an SAV h, denoted by θ_{rh}^o, is determined by the average observed value of the profit set across all $|\mathscr{K}|$ iterations and is calculated as:

$$\theta_{rh}^{o} = \frac{\sum_{i=1}^{|\mathscr{K}|} f_i^h}{|\mathscr{K}|} \tag{30}$$

where the recorded $|\mathscr{K}|$ is set to be not exceeding a maximum number of recorded iterations $|\mathscr{K}|_{\max}$. Each request will be assigned to the vehicle with the minimum observation level.

Adaptive strategy design

Drawing inspiration from the Pisinger and Ropke (2007) where different operators are used with a frequency corresponding to their historic performance based on the roulette wheel selection principle, we introduce a similar adaptive mechanism for selecting the assignment operator in our proposed request assignment scheme. Specifically, we define a segment for the request assignment process that consists of φ times of assignments that are independent of the overall algorithm iteration number. Each assignment operator is associated with an iteration-specific score and a segment-specific weight. The score is iteratively updated to measure the recent performance of the selected assignment operator. A higher score indicates a successful assignment achieved by the corresponding operator. At the end of an assignment segment, the weight of the operators will be updated based on the cumulative score within this segment, thus updating the selection probability of a particular assignment operator. Let π_a denote the recorded score of assignment operator $a \in \mathscr{A}$ at last assignment in a specific segment (at the iteration when the corresponding non-improvement counter μ reaches $\mu_{\max}$) and ϖ_a denote the weight of the operator in this segment. Then the weight of the assignment operator a for the next segment can be updated according to the following expression:

$$\varpi_a = (1-\psi)\varpi_a + \psi \frac{\pi_a}{\sum_{a\in\mathscr{A}} \pi_a} \tag{31}$$

where ψ reflects the importance of historical information on the algorithm performance. The assignment operator selection probability is computed by $\frac{\varpi_a}{\sum_{a\in\mathscr{A}} \varpi_a}$, derived from the roulette wheel selection method proposed in Prins (2004). Note that all the assignment operators are equally weighted in the first assignment segment.

The score of the selected assignment operator is incremented based on the newly generated solutions in each iteration. At the beginning of each assignment segment, the scores of all operators are initialized to zero. The update implementation of the score π_a for the selected operator $a \in \mathscr{A}$ varies in each iteration, depending on different increment levels, i.e., σ_1, σ_2, σ_3 and σ_4, associated with the newly generated solutions as follows:

- $\pi_a \leftarrow \pi_a + \sigma_1$: The newly generated solution is accepted as the new global optimum.
- $\pi_a \leftarrow \pi_a + \sigma_2$: The newly generated solution is accepted and better than the incumbent one but worse than the optimal one.
- $\pi_a \leftarrow \pi_a + \sigma_3$: The newly generated solution is accepted and worse than the incumbent one.
- $\pi_a \leftarrow \pi_a + \sigma_4$: The newly generated solution is rejected.

5. Numerical experiments

This section presents the results of extensive numerical experiments conducted on randomly generated instances. First, we will introduce the test instance generation and experimental settings. Second, we will evaluate the efficacy of the proposed solution method by conducting a comparative analysis with a benchmark approach. Finally, we will explore the potential benefits of the SAM service and analyze the effects of the flexibility in time windows of passenger requests on the system performance. The solution algorithm is coded with Python on a personal computer with Intel (R) Core (TM) i7, 2.80 GHz CPU, 16.0 GB RAM.

5.1. Instance generation and parameter settings

Due to the absence of benchmark instances for SAM services in the previous research, instances used for our numerical experiments will be randomly generated. Considering that our proposed problem falls under the variant of dynamic DARP, the generation of random instances in this study is inspired by the benchmark data of DARP proposed by Cordeau (2006). Assuming that all the instances are created within a 20 km × 20 km square service area. The pick-up and drop-off locations of $|\mathscr{R}|$ passenger requests, i.e., v_r^o and v_r^d, $\forall r \in \mathscr{R}$, and the initial locations of $|\mathscr{K}|$ SAVs, i.e., v_h^0, $\forall h \in \mathscr{K}$, are randomly generated in this square region following the uniform distribution. The Euclidean distance is used to calculate the distance between two locations. It is assumed that the average travel speed for each SAV is $V = 40$ km/hr, and each standard SAV is designed to comfortably accommodate up to four passengers, i.e., $W_h = 4$. These operational paramters are widely adopted simplifying assumptions in large-scale simulation/algorithmic evaluations to keep instances controlled and comparable (Santi et al., 2014; Alonso-Mora et al., 2017; Narayanan et al., 2020). Let $dis(i,j)$ denote the distance from location i to j. Then the travel time between the pick-up location and drop-off location of request r can be calculated as $\tau_{ij} = dis(i,j)/V$. Travel time from the pick-up to the drop-off location of request r can be computed by $\tau_{ij} = dis(i,j)/V$. Considering the per-unit traveling distance operating cost of SAVs at \$0.2/km (Fagnant and Kockelman, 2018; Johnson and Walker, 2016; Stephens et al., 2016), travel cost from i to j can be determined by $\kappa_{ij} = 0.2 \cdot dis(i.j)$. As for the planning horizon, two different scenarios are considered and designed to capture distinct commuter travel patterns: a peak-hour operational period and a full-day operational period. For the peak-hour scenario, the planning horizon is set to 3 h, representing typical busy periods during the operational day, such as the morning rush from 7:00 a.m. to 10:00 a.m. and the afternoon rush from 4:00p.m. to 7:00p.m. When measured in minutes,

this planning horizon is defined as [0,180]. For each passenger request $r \in \mathscr{R}$, the earliest pick-up time, i.e., t_r^r, is generated randomly, with a 25% chance of falling within the interval [0,60), a 50% probability of being within [60,120], and a 25% likelihood from the interval (120,180]. In addition to the peak-hour scenario, a full-day planning horizon is considered to simulate travel patterns throughout an entire operational day, including both peak and off-peak periods. This full-day horizon spans 12 h, represented as [0,720]. For this instance, the earliest pick-up times are generated with a similar probabilistic distribution: 25% of requests fall within the interval [0,120] (morning peak), 50% within [120,600] (off-peak period), and the remaining 25% within [600,720] (evening peak).

In addition, the maximum confirmation duration δ_r is set to 10 min. The corresponding latest pick-up time of the request r is $t_r^{lp} = t_r^r + \chi(r)$, where $\chi(r)$ denotes a *flexibility time* that is drawn from a uniform distribution $u(5,30)$. The latest arrival time is determined by considering the latest pick-up time, i.e., $t_r^{ld} = t_r^{lp} + \tau_{v_r^o v_r^d}$. Note that we treat t_r^{ld} (and the derived t_r^{lp}) as operational time windows used for routing feasibility in SAV dispatch, rather than as deterministic passenger cancellation thresholds. Actually, it refers to a tolerance (slack) parameter for in-vehicle delay / detour (and/or total trip time), not a waiting deadline. We consider single-passenger and two-passenger requests in our experiments, indicating that w_r is randomly set to 1 or 2. The maximum number of ride-pooling strangers is defined to align with the typical scenario of pooling two orders, as commonly observed in current ride-sourcing services, i.e., $Q = 2$. We assume that the SAM service operator charges each request based on the direct travel distance between its origin and destination, excluding any detours due to sharing. Based on the previous literature on the pricing and cost parameters commonly assumed in the exsiting shared-mobility/ride-sourcing literature (Bösch et al., 2018; Hazan et al., 2016; Litman, 2017; Bimpikis et al., 2019), the unit service charge for the solo-ride trip is give at \$1/km, and the service charge of the request in a solo-ride trip is calculated by $G_r = 1 \cdot dis(v_r^o, v_r^d)$. We further define that the request pooled with others enjoys a 10% discount rate, i.e., $\upsilon = 0.1$, and therefore, the service charge of the request in a pooled-ride trip is $\widehat{G}_r = 0.9 \cdot G_r$. Additionally, based on some behavioral-related literatures (U.S. Department of Transportation, 2025; Tirachini, 2020), the VOT of each request, i.e., p_r, will be chosen at random from set {0.1, 0.2, 0.3, …,1}. Similarly, the privacy-sensitivity of each request, e.g., g_r, is determined by randomly selecting an integer from set {1, 2, 3, 4, 5}, with each value of sensitivity level having an equal probability of being chosen. Without loss of generality, the multivariate satisfaction function is represented as (Finn, 2011; Fang et al., 2021; Lavieri and Bhat, 2019; Naumov and Keith, 2023; Xu, 2025):

$$F_r(\upsilon, p_r, g_r, \widehat{q}_r, \widehat{\zeta}_r, \widehat{\xi}_r) = 1 - \frac{\upsilon^2}{5} - \frac{p_r^2}{5} - \frac{g_r/5 \times \widehat{q}_r/Q}{5} - \frac{(\widehat{\zeta}_r/\widehat{\zeta}_{\max})^2}{5} - \frac{(\widehat{\xi}_r/\widehat{\xi}_{\max})^2}{5} \tag{32}$$

where $\widehat{\zeta}_{\max}$ and $\widehat{\xi}_{\max}$ denote the upper limits for ride-pooling duration and extra travel time of request r respectively, and both are set to 30 min. The minimum passenger satisfaction value $\underline{F}$ is set to 0.6.

Regarding the rolling horizon framework, the predefined increment time between two consecutive decision time instant and the computational time for each static subproblem, i.e., Δt and λ, are set to 120 s and 15 s, respectively. As for the algorithm-related parameters, they were determined via a pilot tuning process on representative instances based on standard metaheuristic practices and fixed for all experiments (Pisinger and Ropke, 2018; Ropke and Pisinger, 2006; Vidal et al., 2014). Specifically, the percentage of requests to be removed in the neighborhood search is set to 25%. The parameters of the score increment, i.e., σ_1, σ_2, σ_3 and σ_4, are set to 6, 4, 1 and 0.2. The simulated annealing accept criterion parameter ν is set to 0.9954. The segment size and assignment frequency parameters of the ARA method, i.e., φ and $\mu_{\max}$, are set to 5 and 5, respectively. The parameter $|\mathscr{K}|_{\max}$ in assignment operator A5 is set to 10. The stopping criteria of the ARA-LNS algorithm are set to $N_{\max} = 300$ and $U_{\max} = 15$ s.

Table 1
Comparison of the proposed solution method and the benchmark method for peak-hour instances.

#SAV	#Request	Benchmark method		Our method		Comparison			
		Obj	#SR	Obj	#SR	Diff_Obj AbsGap	RelGap	Diff_#SR AbsGap	RelGap
25	150	834.9	117.2	915.6	124.2	80.7	9.7%	7.0	6.0%
25	180	936.9	134.0	1028.5	141.6	91.6	9.8%	7.6	5.7%
25	210	1032.2	146.0	1155.6	154.2	123.4	12.0%	8.2	5.6%
25	240	1095.1	160.2	1229.5	168.3	134.4	12.3%	8.1	5.1%
25	270	1147.6	166.8	1298.3	175.4	150.7	13.1%	8.6	5.2%
Avg.		1009.3	144.8	1125.5	152.7	116.1	11.4%	7.9	5.5%
40	150	932.9	132.3	1034.4	140.6	101.5	10.9%	8.3	6.3%
40	180	1069.9	152.9	1223.8	164.2	153.9	14.4%	11.3	7.4%
40	210	1217.4	176.8	1420.9	188.9	203.5	16.7%	12.1	6.8%
40	240	1324.6	186.9	1543.3	202.4	218.7	16.5%	15.5	8.3%
40	270	1440.7	204.5	1708.6	226.8	267.9	18.6%	22.3	10.9%
Avg.		1197.1	170.7	1386.2	184.6	189.1	15.4%	13.9	7.9%
55	150	1015.8	138.8	1165.1	149.6	149.4	14.7%	10.8	7.8%
55	180	1122.0	160.6	1320.6	176.4	198.6	17.7%	15.8	9.8%
55	210	1314.9	180.4	1543.0	198.2	228.1	17.3%	17.8	9.9%
55	240	1461.9	203.4	1737.1	223.8	275.3	18.8%	20.4	10.0%
55	270	1554.0	217.2	1865.9	246.4	311.8	20.1%	29.2	13.4%
Avg.		1293.7	180.1	1526.3	198.9	232.6	17.7%	18.8	10.2%

Unless stated otherwise, all the above parameters will remain unchanged throughout the numerical experiments.

5.2. Comparison with the benchmark approach

To evaluate the overall performance of the proposed algorithm in terms of the solution quality, we compare the results obtained by our proposed algorithm against the results of the benchmark method that follows the same rolling horizon framework proposed in section 3 but employs a different algorithm to solve the static subproblem for each rolling-horizon computation. Specifically, it constructs the initial solution for all vehicles based on their partial solutions by employing the insertion heuristic proposed in the study of Potvin and Rousseau (1993). Then, the LNS algorithm, utilizing the same removal and insertion operators as our proposed method, is directly implemented on all vehicle routes to explore improved routing solutions, disregarding the adaptive assignment procedures. Since the fleet size and the number of passenger request arrivals are expected to influence the computational efficiency of the solution method, we consider a total of 15 scenarios with different numbers of SAVs and numbers of passenger request combinations, i.e., $|\mathscr{K}| \in \{30, 45, 60\}$ and $|\mathscr{R}| \in \{150, 180, 210, 240, 270\}$. Given the specific SAV fleet size and request number combination, 10 test instances are randomly generated with the aforementioned parameters and the average results are reported. As such, a total of 150 instances will be used to assess the proposed solution method. Similarly, we also investigate larger full-day instances, where the number of passenger requests is set to {520, 640, 760, 880, 1000}, to evaluate the performance of the solution method under more extensive demand scenarios.

Table 1 presents a comparison of the performance between the proposed solution method and the benchmark method for various instances, differentiated by varying number of SAV (#SAV) and number of requests (#Request) combinations. For both approaches, we report the system profitability (Obj) and the number of served requests (#SR). We also represent the difference, i.e., the absolution gap (AbsGap) and the relative gap in percentage (RelGap), in terms of the objective value (Diff_Obj) and the number of served requests (Diff_#SR) between the proposed solution method and the benchmark approach to have a more intuitive comparison. For the peak-hour instances in Table 1, the proposed solution method consistently outperforms the benchmark approach across all scenarios, achieving higher profitability and serving more requests. The absolute differences in objective value range from 80.7 to 232.6, while the number of served requests improves by up to 18.8. This indicates that the proposed method with the well-designed adaptive assignment procedure can bring about apparent advantages over the benchmark approach. Furthermore, the performance gap becomes more significant as the fleet size increases. For instances with 30 SAVs, the average improvements in objective value and served requests are 116.1 and 7.9, respectively. However, for instances with 60 SAVs, these averages rise significantly to 232.6 and 18.8. This observation also holds for their relative gaps. For instances with 60 SAVs, the average relative gaps in objective value and served request number reach as high as 17.7% and 10.2%, respectively, whereas for instances with 30 SAVs, these values are only 11.4% and 5.5%. That is to say, the performance improvement between the proposed solution method and the benchmark method becomes more apparent, on average, under the scenarios with a larger number of vehicles. This suggests that our proposed request assignment strategy integrated into the search process can notably achieve improvements in solution quality, especially for instance involving a larger fleet size. The primary reason is that the adaptive pre-assignment procedure can effectively derive more feasible and potentially profitable request-SAV assignment for each vehicle, thus more likely generating high-quality routes. Similarly, Table 2 illustrates the superior performance of the proposed solution method for full-day instances, where the improvements are even more significant under larger request demand. The absolute differences in objective value range from 235.8 to 1573.6, while the improvements in the number of served requests range from 19.9 to 150.7. The relative gaps in objective value for full-day instances with 60 SAVs can reach as high as 31.1%, highlighting the proposed method's effectiveness in handling large-size scenarios. Overall, the results from both peak-hour

Table 2
Comparison of the proposed solution method and the benchmark method for full-day instances.

#SAV	#Request	Benchmark method		Our method		Comparison			
		Obj	#SR	Obj	#SR	Diff_Obj AbsGap	RelGap	Diff_#SR AbsGap	RelGap
25	520	2646.2	375.5	2882.0	395.4	235.8	8.9%	19.9	5.3%
25	640	3081.0	416.6	3512.2	442.3	431.2	14.0%	25.7	6.2%
25	760	3150.1	458.2	3675.7	495.1	525.6	16.7%	36.9	8.1%
25	880	3172.3	465.2	3776.6	507.2	604.3	19.0%	42.0	9.0%
25	1000	3265.2	488.1	3921.4	531.1	656.2	20.1%	43.0	8.8%
Avg.		3063.0	440.7	3553.6	474.2	490.6	15.7%	33.5	7.5%
40	520	3077.2	440.1	3659.7	482.3	582.5	18.9%	42.2	9.6%
40	640	3697.1	516.6	4465.1	570.9	768.0	20.8%	54.3	10.5%
40	760	3966.7	572.9	4831.1	650.1	864.4	21.8%	77.2	13.5%
40	880	4260.2	602.2	5269.7	685.2	1009.5	23.7%	83.0	13.8%
40	1000	4446.8	630.7	5416.8	725.5	970.0	21.8%	94.8	15.0%
Avg.		3889.6	552.5	4728.5	622.8	648.9	21.4%	70.3	12.5%
55	520	3265.4	446.6	3849.7	497.1	852.2	17.9%	50.5	11.3%
55	640	3735.2	526.2	4503.7	600.6	768.6	20.6%	74.4	14.1%
55	760	4202.6	597.7	5166.3	679.8	963.7	22.9%	82.1	13.7%
55	880	4943.6	688.1	6151.0	790.0	1207.4	24.4%	101.9	14.8%
55	1000	5051.9	718.1	6625.5	868.8	1573.6	31.1%	150.7	21.0%
Avg.		4239.7	595.3	5259.2	687.3	1073.1	23.4%	91.9	15.0%

and full-day instances highlight the better performance of our proposed solution method compared to the benchmark method in solving the RT-SAVD problem, and demonstrate the scalability and robustness of the proposed solution method.

5.3. Impact analysis

In this subsection, we will first evaluate the benefit of the SAM service by comparing it with the services without the ride-pooling option, referred to as SAMw/oP. We will then investigate how the passengers' flexibility time affects the performance of SAM services.

Impact of SAM service model

Considering that peak-hour scenarios are representative in capturing the key characteristics and dynamics of the system, we will assess these random instances discussed in Subsection 6.2 for the SAM and SAMw/op services to evaluate the benefit of the ride-pooling option. Under each combination of SAV number and request number, average results for five randomly generated instances will be reported. To be specific, in addition to the profit and served request number, we will also evaluate the average waiting time (AvgWat) and served request ratio (SerRat). Differences in average waiting time, profit and served request number are also reported in terms of both absolute gap and relative gaps in percentage. Table 3 tabulated the results of the comparison of these two services.

As expected, the ride-pooling option in an SAM service can significantly improve the total profit and served request ratio for all instances. For some instances, e.g., #SAV = 30 and #Request = 270, the total profit improvement is as high as 176.5, and the average relative increment gap reaches up to 15.9%. Furthermore, the introduction of ride-pooling features does not affect the average passenger waiting time, with a relative reduction gap approximating 7.4% on average for all instances. This results from the ride-pooling feature, which facilitates the consolidation of multiple passengers to share the journey, effectively improving the utilization rate of vehicles and reducing passenger waiting time. In contrast, in non-pooling services, each request needs to wait for an SAV to pick them up, leading to a longer waiting time.

For a given SAV fleet size, the profit increment shows an increasing trend as the request number rises. For example, with 45 SAVs in operation, the profit gaps between the two types of services are only 65.1 and 87.6 instances with 150 and 180 respectively, while the gap becomes 150.1 when number of requests reaches up to 270. Additionally, larger instances generally exhibit a greater relative profit gap compared to smaller ones, which can be explained by the higher revenue generated from serving more requests. These findings suggest that SAM services tend to be more beneficial for larger instances. Another interesting finding is that the profit increment in SAM services decreases as the number of vehicles grows. This observation is also evident in the relative gap in profit for most instances. It suggests that even without ride-pooling options, requests can still have more opportunities to be feasibly accommodated in an SAV's route under a larger fleet size, thus making SAMw/oP services achieve more comparable results to SAM services. This aligns with the decreased difference in the served request number between the two types of services. Hence, it is advisable for SAM service providers to carefully consider the SAV resources distribution in order to make full use of the advantages of the ride-pooling mode and enhance system profitability.

Impact of flexibility time

To explore how the flexibility time of passenger requests influences the SAM service system performance, we will set up scenarios with five different flexibility time ranges. Specifically, we will vary the uniform distribution of the flexibility time parameter of passenger requests while keeping other parameters consistent with those given in section 6.1. The average value of the flexibility time distribution varies incrementally, with each step adding 5 min, resulting in five distinct settings for the distribution: $\{u(5,10), u(10,15), u(15,20), u(20,25), u(25,30)\}$. The flexibility time parameter, i.e.,$\chi(r)$, will be drawn from these different uniform distributions. A distribution setting characterized by a higher average value indicates that, on average, passengers have greater temporal flexibility. This implies that passengers are more tolerant of longer detours during their journeys.

In order to have a comprehensive evaluation, we will examine the system performance of the SAM services for various peak-hour scenarios involving different combinations of the SAV number (selected from {30, 45, 60}) and the request number (selected from {150, 210, 270) under different flexibility time distribution settings. For a particular distribution setting, we will report the average results for 10 randomly generated instances. In addition to the aforementioned performance metrics, we additionally present the number of pooled requests (#PR) and the pooled requests ratio (PolRat). The overall results are reported in Table 4. To facilitate comparison, we further visualize the variations of different performance metrics based on a specific SAV number, i.e., #SAV = 60, and all the request number combinations in Figs. 2-7.

We can observe from Fig. 2 that the profit steadily increases as the flexibility time increases for all tested scenarios. This finding aligns with the observations in the observed growth in both served request numbers and pooled request numbers under the growing flexibility time by looking further into Fig. 4 and Fig. 6. The results are within our expectation: passengers' willingness to accept later arrival time at their destinations allows the SAM services to serve more passengers before their latest departure time and have more flexibility to further create more cost-saving pooling arrangements, thus finding more profitable SAV routing plans and achieving profit improvement for the service provider. Nevertheless, the flexibility time increase will inevitably result in a prolonged average waiting time. Fig. 3 illustrates that as flexibility time increases, the average waiting time steadily increases, showing a roughly linear trend. For example, the average waiting time increases from 4.99 min to 17.55 min for the scenarios of 150 requests when the flexibility time distribution increases from $u(5,10)$ to $u(25,30)$.

Particularly, it is also encouraging to observe that the average increment of profit for every additional 5 min of the average value of the flexibility time distribution is relatively larger when the demand is higher. For instance, for the scenarios with 45 SAVs, the corresponding profit will averagely increase by 50.5 and 112.7 as the flexibility time distribution changes 5 min incremental step for scenarios with 150 to 270 requests. The results imply that the overall profit for the SAM services will be further boosted if a larger number of requests are happy to accept delayed arrival times. In comparison, the average increment of profit becomes lower under the

Table 3
Comparison of the SAMw/oP and SAM service.

#SAV	#Request	SAMw/oP				SAM				Comparison					
		Profit($)	AvgWat(min)	#SR	SerRat	Profit($)	AvgWat(min)	#SR	SerRat	Diff_Profit		Diff_AvgWat		Diff_#SR	
										AbsGap	RelGap	AbsGap	RelGap	AbsGap	RelGap
25	180	840.1	12.28	115.2	76.8%	915.6	10.98	124.2	82.8%	75.4	9.0%	−1.30	−10.6%	9.0	7.8%
25	210	939.0	12.45	130.0	72.2%	1028.5	11.31	141.6	78.7%	89.5	9.5%	−1.14	−9.2%	11.6	8.9%
25	240	1046.3	12.78	142.8	68.0%	1155.6	11.58	154.2	73.4%	109.3	10.4%	−1.20	−9.4%	11.4	8.0%
25	270	1091.2	13.01	155.1	64.6%	1229.5	12.01	168.3	70.1%	138.4	12.7%	−1.00	−7.7%	13.2	8.5%
25	300	1121.8	13.58	160.0	59.3%	1298.3	12.61	175.9	65.1%	176.5	15.7%	−0.97	−7.1%	15.9	9.9%
40	180	969.3	10.21	134.2	89.5%	1034.4	9.23	140.6	93.7%	65.1	6.7%	−0.98	−9.6%	6.4	4.8%
40	210	1136.2	10.71	155.8	86.6%	1223.8	9.87	164.2	91.2%	87.6	7.7%	−0.84	−7.8%	8.4	5.4%
40	240	1307.6	11.08	177.9	84.7%	1420.9	10.25	188.9	90.0%	113.3	8.7%	−0.83	−7.5%	11.0	6.2%
40	270	1415.0	11.48	190.4	79.3%	1543.3	10.63	202.4	84.3%	128.2	9.1%	−0.85	−7.4%	12.0	6.3%
40	300	1558.5	11.65	211.4	78.3%	1708.6	10.95	226.8	84.0%	150.1	9.6%	−0.70	−6.0%	15.4	7.3%
55	180	1102.8	10.12	143.9	95.9%	1165.1	9.36	149.6	99.7%	62.3	5.6%	−0.76	−7.5%	5.7	4.0%
55	210	1246.7	10.22	169.2	94.0%	1320.6	9.65	176.4	98.0%	73.9	5.9%	−0.57	−5.6%	7.2	4.3%
55	240	1430.7	10.45	188.2	89.6%	1543.0	9.81	198.2	94.4%	112.3	7.8%	−0.64	−6.1%	10.0	5.3%
55	270	1611.8	10.64	212.4	88.5%	1737.1	10.09	223.8	93.3%	125.4	7.8%	−0.55	−5.2%	11.4	5.4%
55	300	1756.2	10.85	232.5	86.1%	1865.9	10.35	246.4	91.3%	109.7	6.2%	−0.50	−4.6%	13.9	6.0%

Table 4
Impact of the flexibility time on the system performance with different SAV and request number combinations.

Distribution	#SAV	#Request	Profit ($)	AvgWat (min)	#SR	SerRat	#PR	PolRat
$u(5,10)$	150	30	835.9	4.99	102.4	68.3%	3.2	3.1%
$u(10,15)$	150	30	960.9	6.83	116.0	77.3%	4.8	4.1%
$u(15,20)$	150	30	1057.0	9.81	131.0	87.3%	17.2	13.1%
$u(20,25)$	150	30	1115.2	13.41	141.2	94.1%	30.4	21.5%
$u(25,30)$	150	30	1141.9	17.55	148.4	98.9%	47.6	32.1%
$u(5,10)$	210	30	929.7	5.08	119.6	57.0%	8.6	7.2%
$u(10,15)$	210	30	1069.5	7.96	135.6	64.6%	14.2	10.5%
$u(15,20)$	210	30	1175.1	10.52	152.4	72.6%	25.6	16.8%
$u(20,25)$	210	30	1265.1	14.06	169.8	80.9%	49.2	29.0%
$u(25,30)$	210	30	1307.1	17.31	180.2	85.8%	84.0	46.6%
$u(5,10)$	270	30	997.0	4.99	128.4	47.6%	10.6	8.3%
$u(10,15)$	270	30	1145.8	6.83	150.0	55.6%	22.3	14.9%
$u(15,20)$	270	30	1299.9	9.81	172.3	63.8%	41.8	24.3%
$u(20,25)$	270	30	1397.5	13.41	197.0	73.0%	65.6	33.3%
$u(25,30)$	270	30	1458.2	17.55	210.5	78.0%	91.4	43.4%
$u(5,10)$	150	45	983.3	4.48	117.4	78.3%	2.1	1.8%
$u(10,15)$	150	45	1080.8	6.25	129.4	86.3%	3.2	2.5%
$u(15,20)$	150	45	1138.0	8.72	143.4	95.6%	12.4	8.6%
$u(20,25)$	150	45	1166.8	12.50	148.1	98.7%	31.6	21.3%
$u(25,30)$	150	45	1185.2	16.38	149.6	99.7%	41.6	27.8%
$u(5,10)$	210	45	1200.0	4.68	149.2	71.0%	6.4	4.3%
$u(10,15)$	210	45	1325.7	6.36	165.8	79.0%	12.4	7.5%
$u(15,20)$	210	45	1414.7	9.35	186.0	88.6%	24.8	13.3%
$u(20,25)$	210	45	1494.3	12.68	197.1	93.9%	54.8	27.8%
$u(25,30)$	210	45	1538.8	17.21	202.4	96.4%	73.2	36.2%
$u(5,10)$	270	45	1327.2	4.71	181.6	67.3%	7.0	3.9%
$u(10,15)$	270	45	1470.6	6.59	192.1	71.1%	15.4	8.0%
$u(15,20)$	270	45	1605.0	10.12	216.2	80.1%	29.6	13.7%
$u(20,25)$	270	45	1706.8	13.73	244.2	90.4%	68.0	27.8%
$u(25,30)$	270	45	1778.1	17.62	255.0	94.4%	86.6	34.0%
$u(5,10)$	150	60	1043.8	4.20	129.8	86.5%	0.9	0.7%
$u(10,15)$	150	60	1121.9	5.49	139.6	93.1%	2.5	1.8%
$u(15,20)$	150	60	1164.7	7.89	147.4	98.3%	14.2	9.6%
$u(20,25)$	150	60	1192.6	10.18	149.5	99.7%	24.1	16.1%
$u(25,30)$	150	60	1208.8	15.53	150.0	100.0%	31.0	20.7%
$u(5,10)$	210	60	1264.9	4.27	165.6	78.9%	3.2	1.9%
$u(10,15)$	210	60	1382.1	5.99	177.7	84.6%	6.8	3.8%
$u(15,20)$	210	60	1471.0	8.88	194.2	92.5%	22.6	11.6%
$u(20,25)$	210	60	1538.8	12.63	205.6	97.9%	50.2	24.4%
$u(25,30)$	210	60	1569.6	16.90	207.0	98.6%	58.8	28.4%
$u(5,10)$	270	60	1720.2	4.39	204.2	75.6%	5.1	2.5%
$u(10,15)$	270	60	1856.5	6.09	218.5	80.9%	10.4	4.8%
$u(15,20)$	270	60	1972.2	8.95	235.4	87.2%	28.8	12.2%
$u(20,25)$	270	60	2034.5	12.75	254.4	94.2%	65.8	25.9%
$u(25,30)$	270	60	2071.8	17.20	260.2	96.4%	82.8	31.8%

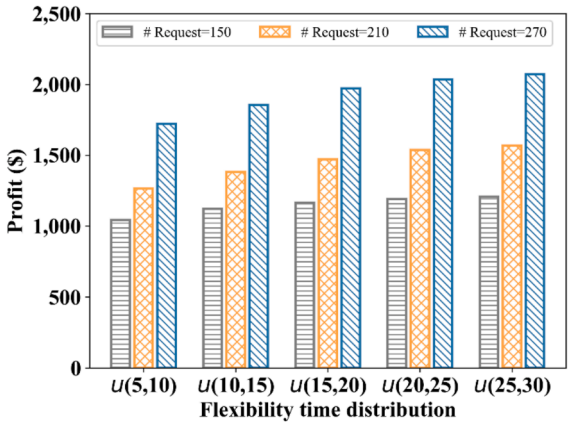

Fig. 2. Impact of the flexibility time on the total profit.

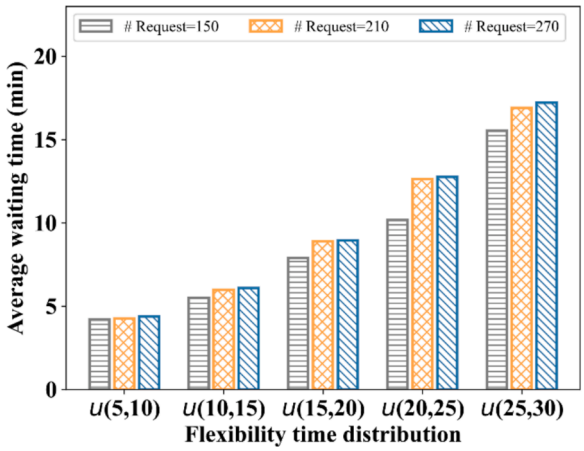

Fig. 3. Impact of the flexibility time on the average waiting time.

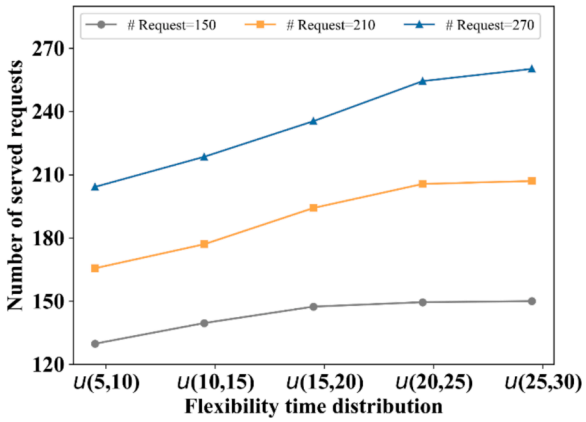

Fig. 4. Impact of the flexibility time on the number of served requests.

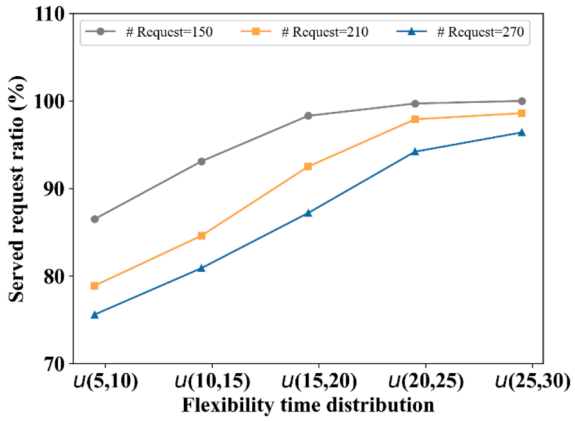

Fig. 5. Impact of the flexibility time on the served request ratio.

increased number of SAVs. For example, for the scenarios of 270 requests, the profit will averagely increase by 115.3 and 87.9 with an additional 5 min of flexibility time for fleets of 30 and 60 SAVs respectively. This suggests a diminishing beneficial effect of the increasing flexibility time on the total profit when the SAV fleet size expands.

Fig. 4 and Fig. 5 show the variations in the number of served requests and the served request ratio with the increase of passengers' flexibility time, respectively. Notably, the number of served requests increases rapidly from 204.2 to 260.2 for the scenarios with the

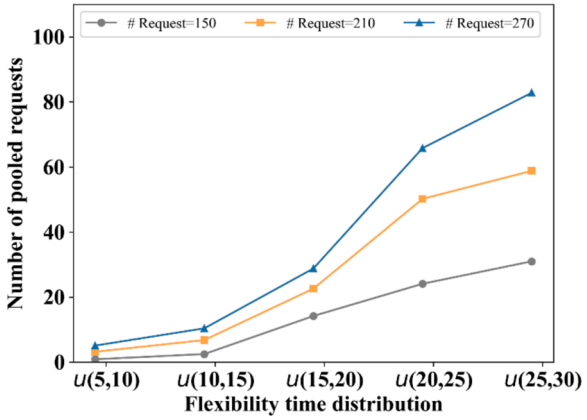

Fig. 6. Impact of the flexibility time on the number of pooled requests.

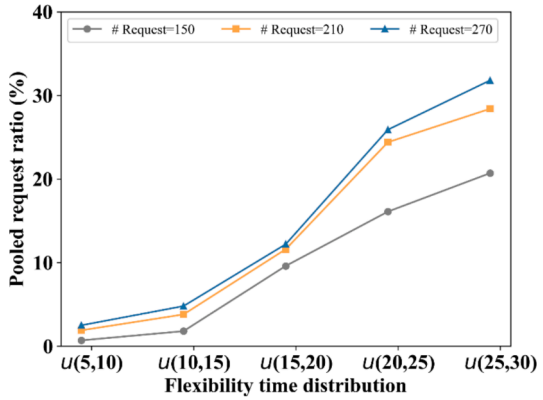

Fig. 7. Impact of the flexibility time on the pooled request ratio.

number of requests being 270, and the corresponding served request ratio increases from 75.6% to 96.4%. Additionally, according to Fig. 4 and Fig. 5, when the market adoption of SAM services remains at a comparatively low level such as #Request = 150, the increment is marginal (around 5.1 on average calculation for #SAV = 60 scenarios). More specifically, for the scenarios of 60 SAVs, the increase rate of the served request number for 150 requests keeps decreasing with the increase of passengers' flexibility time, while the increase rate for the request number of 210 or 270 first rises and then drops. This may be attributed to a large SAV fleet in relation to demand, i.e., 60 SAVs for 150 requests, enabling the SAM services to meet the majority of the demand with relatively small flexibility time. Therefore, even if the time window becomes more flexible, the service capacity of the fleet is relatively stable and no significant growth for the served requests will be produced. Comparatively, when the ratio of the SAV fleet to demand is low, it is not possible to fully meet all demand within the tight time window. As flexibility time increases, the fleet gains more opportunities to match passenger requests, thereby increasing the number of passengers served. However, with continued increases in flexibility time, once the service capacity of the SAV fleet saturates, further changes in flexibility time no longer significantly affect the number of served requests, resulting in a decreased growth rate.

Regarding the variations in the number of pooled requests and pooled request ratio, Fig. 6 and Fig. 7 show that both variations show a similar upward trend to the variations of served request number and served request number ratio. It is worth noting that the number of pooled requests averagely reaches as high as 82.8 with the corresponding pooled ratio of 31.8% when the number of requests is 270. In particular, an average increase of 5 min in flexibility time for passengers is expected to significantly boost the number of pooled requests and this contribution is even more significant under larger-scale instances. For example, when the flexibility time distribution changes from $u(5,10)$ to $u(25,30)$, the number of pooled requests will averagely increase by 7.5 for the scenario with 150 requests, while the average increments become 13.9 and 19.4 for the scenarios with additional 60 and 120 number of requests, i.e., 210 and 270, respectively. We also notice that the increase rates of both pooled request number and pooled request ratio become smaller for 210 and 270 requests when the flexibility time distribution exceeds $u(20,25)$. To be specific, the pooled request number and its ratio for the first two scenarios increase sharply when the flexibility time distribution changes from $u(5,10)$ to $u(20,25)$, but the increment becomes smaller when the flexibility time distribution increases further. This finding is consistent with the changes in the increase rate observed in the number of requests served and their corresponding ratios. It also indicates that a prolonged flexibility time can positively affect the pooling probability among potential requests, but the improvement in pooling would become limited when the flexibility time

exceeds approximately 20 min. In comparison, for the larger instance with 270 requests, it remains in the rapid growth phase, with an increased rate due to not yet reaching the pooling limit. This suggests that further increases in flexibility time can continue to unlock pooling potential, resulting in a higher number of pooled passengers and an increased pooling rate.

Impact of customer satisfaction

To better understand the role of the proposed multivariate satisfaction function, we conduct an sensitivity study based on a specific instance (i.e., #SAV = 45 and #Request = 270) that selectively removes individual disutility components in the function form to explore which terms are most binding under the satisfaction threshold constraint and how they affect the resulting pooling–service–profit trade-offs. Table 5 reports averaged results over five runs for the selected instance. Different scenarios of No_VOT, No_privacy, No_extraTT and No_sharedDur indicate four variants of the satisfaction function used in the instance, constructed by removing the VOT term, the privacy/stranger-exposure penalty term, the extra-travel-time penalty term, and the shared ride-pooling duration penalty term in the function respectively. For the baseline scenario (i.e., using the satisfaction function in its full form), the system serves 226.8 requests with a pooling ratio of 13.4%, an average waiting time of 10.96 min, and a profit of 1708.6. VOT and shared-ride duration are the dominant drivers that suppress pooling: compared with the baseline pooling ratio of 13.4% (# PR = 30.4), removing VOT increases pooling to 21.1% (#PR = 47.2) and removing shared-duration increases pooling to 21.7% (#PR = 48.8). In contrast, removing privacy or extra travel time yields only moderate pooling gains, suggesting these components are less frequently binding than VOT/shared-duration in this instance. In addition, higher pooling does not necessarily translate into improved system-level outcomes under the current pricing scheme. The profit decreases slightly from the baseline, while average waiting time increases from 10.96 to 11.06–11.21 min and service rate drops from 84.0% to 81.7%–83.2%. This is possibly because when satisfaction penalties are relaxed, the optimizer can form more pooled matches, but a larger share of served demand is then charged at the discounted price. In this instance, the revenue loss induced by discounting appears to outweigh the operational gains from higher vehicle utilization. At the same time, pooling introduces stronger spatiotemporal coupling, as a single vehicle must satisfy multiple pickup and drop-off constraints simultaneously. This reduces dispatch flexibility and can introduce extra scheduling friction, which is consistent with the slight increase in average waiting time and the marginal decrease in service rate. Moreover, removing the extra-travel-time penalty is the most disruptive to service stability, yielding the lowest service rate and profit. This suggests that detour-related disutility helps prevent excessive route distortion and maintains more stable feasible schedules. Overall, the results highlight that satisfaction function does not merely cosmetic “filter" modeling choice for pooling. Different satisfaction components constrain different dimensions of pooling feasibility. It serves as a regulating mechanism that balances operational performance (profit/service efficiency) against user experience by controlling how aggressively the system pursues pooling under the satisfaction threshold.

We further conduct a sensitivity analysis on the satisfaction threshold $\underline{F}$. As is shown in Table 6 and Fig. 8, when $\underline{F}$ is relaxed from 0.6 to 0.5, the pooling ratio increases from 13.4% (#PR = 30.4) to 16.8% (#PR = 38.4); relaxing it further to 0.1 raises pooling to 21.4% (#PR = 49.6). In contrast, once $\underline{F}$ exceeds 0.7, pooling significantly drops and the pooling ratio is only 6.5% with pooling request number being 14.6. With the threshold further increasing, the pooling phenomenon essentially vanishes. This steep decline indicates that, in this instance, many potential shared rides yield satisfaction values clustered around the 0.6–0.8 region and a small increase in the minimum acceptable satisfaction can therefore invalidate a large fraction of otherwise feasible matches. In addition, Fig. 9 shows that changes in $\underline{F}$ alter the set of feasible pooled matches, which leads to the corresponding variations in both profit and service rate changes accordingly. Specifically, relaxing the threshold increases both demand coverage and profit: profit rises from 1708.6 at $\underline{F} = 0.6$ to 1729.0 at $\underline{F} = 0.5$ and 1772.0 at $\underline{F} = 0.1$, while the service rate improves from 84.0% (#SR = 226.8) to 84.8% (#SR = 228.9) and 86.0\% (#SR = 232.1). Conversely, tightening the threshold to $\underline{F} = 0.9$ reduces profit to 1639.9 and the service rate

Table 5
Performance comparison under different scenarios for different satisfaction function.

Scenarios	Profit($)	AvgWat(min)	#SR	SerRat	#PR	PolRat
Baseline	1708.6	10.96	226.8	84.0%	30.4	13.4%
No_VOT	1692.8	11.08	224.2	83.0%	47.2	21.1%
No_privacy	1688.2	11.06	222.2	82.3%	41.2	18.5%
No_extraTT	1687.8	11.09	220.6	81.7%	41.2	18.7%
No_sharedDur	1693.7	11.21	224.6	83.2%	48.8	21.7%

Table 6
Sensitivity analysis results under different satisfaction thresholds.

$\underline{F}$	Profit($)	AvgWat(min)	#SR	SerRat	#PR	PolRat
0.1	1772.0	11.15	232.1	86.0%	49.6	21.4%
0.2	1768.6	11.09	232.3	86.0%	48.8	21.0%
0.3	1758.7	11.09	231.5	85.7%	44.9	19.4%
0.4	1742.3	11.08	230.2	85.3%	43.2	18.8%
0.5	1729.0	11.00	228.9	84.8%	38.4	16.8%
0.6	1708.6	10.96	226.8	84.0%	30.4	13.4%
0.7	1688.4	10.88	223.9	82.9%	14.6	6.5%
0.8	1645.9	10.78	218.4	80.9%	0.2	0.1%
0.9	1639.9	10.56	215.9	80.0%	0.0	0.0%

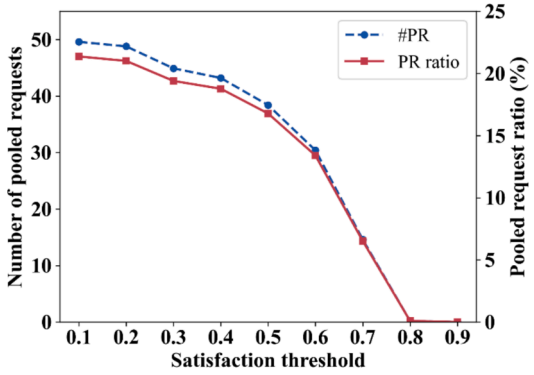

Fig. 8. Sensitivity analysis of pooled request number and pooled request ratio to the satisfaction threshold.

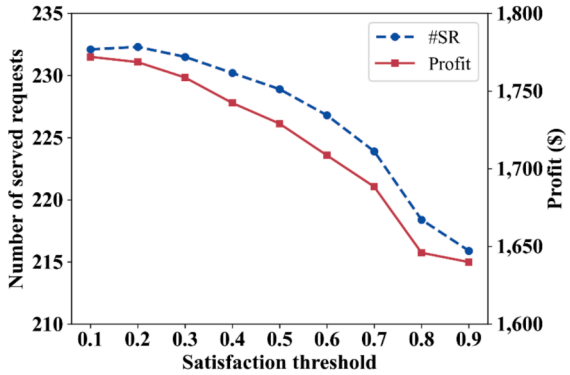

Fig. 9. Sensitivity analysis of served request number and profits to the satisfaction threshold.

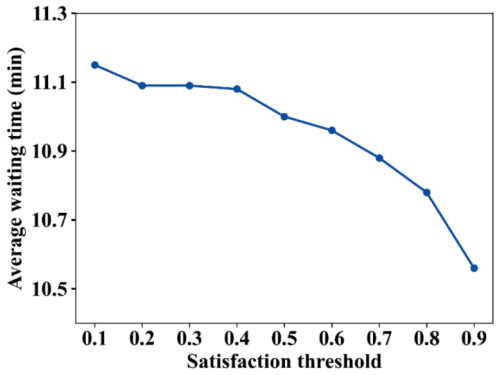

Fig. 10. Sensitivity analysis of average waiting time to the satisfaction threshold.

to 80.0% (#SR = 215.9). Meanwhile, as is depicted in Fig. 10, average waiting time varies within a narrow range and slightly decreases as $\underline{F}$ increases. This suggests a nuanced trade-off: stricter satisfaction requirements filter out more shared rides and also remove some service opportunities, which reduces overall throughput and profit. At the same time, fewer shared rides reduce multi-rider pickup and drop-off coupling, which slightly eases dispatching for the remaining requests and leads to a small reduction in waiting time.

6. Conclusion

In this paper, we investigate the RT-SAVD problem for the SAM service in a dynamic environment to timely respond to newly arrived passenger requests, properly determine the SAV dispatching, and arrange satisfactory pooled trips in pursuit of profit maximization. We develop a dynamic vehicle dispatching framework based on the rolling horizon approach, in which a series of S-SAVD subproblems are solved at a given set of consecutive time points using all known information up to the decision epoch to periodically

update the SAV dispatching solutions. The final demand serving and pooling arrangement decisions of each passenger request will not be made until necessitated by the deadline of the request. For each static subproblem, we propose an MIP model to optimize the vehicle dispatching plans by maximizing the total profit while respecting the maximum ride-pooling stranger number and passenger satisfaction constraints. The current states of SAVs, onboard requests and confirmed requests are considered in the formulation.

To efficiently solve the proposed S-SAVD model, we develop a customized iterative hybrid algorithm, named ARA-LNS, that integrates an ARA scheme into the LNS heuristic framework, allowing us to decompose the multi-vehicle problem into several single-vehicle problems and obtain good-quality solutions efficiently. The main idea of the proposed algorithm is that the LNS is employed to iteratively identify the optimal routing plan for each SAV, and if the total profit for all vehicle routing solutions has not improved for certain iterations, the ARA scheme will be invoked and reassign the requests to different SAVs by adaptively selecting the assignment operators. To evaluate the efficacy of the proposed solution method, we have conducted extensive numerical experiments on randomly generated instances. The results show that the ARA-LNS can achieve better performance over the benchmark approach. Moreover, we examine the impact of the SAM services by comparing them with the case without ride-pooling option, i.e., SAMw/oP services, and results indicate that the SAM services significantly improve the total profit of the system. We also analyze the impact of the flexibility time of passenger requests on the performance of the SAM services. The findings show that the increase in time flexibility can lead to larger profits and more number of served requests and pooled requests, but prolonged average waiting time of passengers.

Future research can be explored in several aspects. First, this study investigated a dynamic and deterministic SAV dispatch problem without considering probabilistic or forecasted features of future demand. The stochastic information, e.g., the uncertain passenger requests, can be incorporated into the SAV dispatching would offer more alignment with reality. The era of big data and the advancement of machine learning methods may also help to predict potential requests and improve SAV pre-scheduling decision-making in SAM services. Second, this study did not explicitly consider empty-vehicle repositioning/rebalancing. It is an important operational component in real-world SAV systems and can significantly improve response times under spatially imbalanced demand. In future work, repositioning can be incorporated by adding a periodic rebalancing module (or additional operators in the proposed search framework) that proactively dispatches idle vehicles to zones with high historical demand intensity or to locations suggested by real-time/forecasted demand, while accounting for the additional empty mileage and operating costs. Third, with the increasing penetration of the EV market driven by declining battery prices and the growing availability of charging facilities, EVs are expected to become more economically and environmentally competitive compared to conventional vehicles. Therefore, it would also be interesting to explore the deployment of the electric SAV fleet, or a mixed fleet comprising both EVs and conventional vehicles, with a particular focus on the optimization of charging strategies. Fourth, designing a reasonable compensation scheme that incentivizes the passengers to accept pooled trips while minimizing the costs of the SAM service provider presents another research direction worth future exploration. Lastly, the integration of the SAV fleet into existing urban transport networks necessitates the consideration of congestion effects within SAM service operations to derive more robust solutions. Moreover, additional consideration in the economic, social, and environmental aspects will offer more research opportunities to improve the sustainability of SAM services.

CRediT authorship contribution statement

Jiangyan Huang: Writing – original draft, Software, Methodology, Conceptualization. **Min Xu:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of competing interest

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

Acknowledgement

This work is supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 15210620).

Appendix. . Notations

T	Duration of the entire planning horizon
$\mathscr{R}$	Set of all passenger requests
$\mathscr{R}_k$	Set of active passenger requests known up to decision time instant t_k for static subproblem in rolling iteration k
$\mathscr{O}_k$	Set of onboard requests for static subproblem in rolling iteration k
$\mathscr{C}_k$	Set of confirmed requests for static subproblem in rolling iteration k
$\mathscr{N}_k$	Set of unscheduled requests for static subproblem in rolling iteration k
v_r^o	Pick-up location of request r
v_r^d	Drop-off location of request r
t_r^r	Request announcement time of request r
δ_r	Maximum confirmation duration of request r

(*continued on next page*)

(*continued*)

t_r^{lp}	Latest pick-up time from origin of request r
t_r^{ld}	Latest drop-off time at destination of request r
w_r	The number of passengers of request r
G_r	Service charge of request r in a solo trip
$\widehat{G}_r$	Service charge of request r in a pooled
Q	Maximum pooling stranger number
p_r	Value of time of request r
g_r	Privacy-sensitivity value of request r
$\widehat{q}_r$	Ride-pooling stranger number of request r
$\widehat{\zeta}_r$	Ride-pooling duration (with other requests onboard) of request r
$\widehat{\xi}_r$	Extra (additional) travel time in the pooled trip of request r
$\underline{F}$	Minimum passenger satisfaction threshold
v_h^0	Initial location of SAV h at the beginning of the planning horizon
t_k	Decision time instant for each rolling iteration k
λ	Computational time limit for each static subproblem
Δt	Rolling time step in the rolling horizon framework
$\mathscr{H}$	Set of all SAVs
v_h	Current location of SAV h for the static subproblem
$\widehat{t}_h$	Arrival time of SAV h at the current location
$\mathscr{S}_{k,h}$	Pre-assigned onboard requests and confirmed requests of SAV h for the static subproblem
$\Theta_{k,h}$	Scheduled partial route information of SAV h for the static subproblem
$\mathscr{G}$	Directed network for the static subproblem
$\mathscr{V}$	Set of all nodes in network for the static subproblem
$\mathscr{E}$	Set of edges in network for the static subproblem
$\mathscr{V}\ominus_1$	Subset of nodes of the current vehicle locations for the static subproblem
$\mathscr{V}\ominus_2$	Subset of the nodes denoting the drop-off locations of requests in set $\mathscr{C}_k$
$\mathscr{V}\ominus_3$	Subset of the nodes denoting the pick-up locations of requests in set $\mathscr{C}_k$ and $\mathscr{N}_k$
$\mathscr{V}\ominus_4$	Subset of the nodes denoting the drop-off locations of requests in set $\mathscr{C}_k$ and $\mathscr{N}_k$
τ_{ij}	Travel time between location i and j
κ_{ij}	Travel cost between location i and j
q_r^*	Recorded number of strangers that have previously shared with the request r
ζ_r^*	Recorded (ride-pooling) duration (with other requests onboard) of the implemented (shared) trip of request r
t_r^*	Past pick-up time instant in the (shared) trip of request r
x_{ij}^{rh}	Whether request r travers on edge $(i.j)$ onboard of vehicle h
y_{ij}^{h}	Whether the vehicle h travers on edge $(i.j)$
d_{rh}	Whether request r is picked up by vehicle h
z_r	Whether request r is pooled with any other request during its trip
$\iota_{rr'h}$	Whether request r shares a ride with requests r' on vehicle h and this encounter is counted as new stranger request
$\rho_{rr'}$	Whether request r has already encountered requests r' in a previous epoch, thereby preventing them from being redundantly counted as "new strangers" in the current epoch.
q_r	Number of strangers in the shared trip of request r
u_{hi}	Time instant at which node i is reached by vehicle h
Ω	Possible request-SAV mapping
s_0	Initial solution composed of $s_0^h \ominus, \forall h \in \mathscr{H}$
s	Incumbent solution composed of $s^h \ominus, \forall h \in \mathscr{H}$
s_b	Best solution composed of $s_b^h \ominus, \forall h \in \mathscr{H}$
s'	Newly generated candidate solution of $s^{h'} \ominus, \forall h \in \mathscr{H}$
N	Iteration number counter
μ	Counter of the total number of consecutive iterations that an assignment operator does not achieve a better solution
$\mu_{\max}$	Maximum continuous attempts that an assignment operator does not achieve a better solution
$\mathscr{A}$	Set of all assignment operators
m	Percentage of removed requests in the removal operator
$\mathscr{M}$	Set of all removed requests
$\mathscr{M}_h$	Set of removed requests for vehicle h
φ	Size of the segment for the ARA scheme
ℓ	Temperature in the simulated annealing accept criterion
ν	Cooling rate in the simulated annealing accept criterion
$N_{\max}$	Maximum iteration number stop criterion
$U_{\max}$	Elapsed CPU time stop criterion
θ_{rh}^v	Violation level of assigning a request r to the SAV h
θ_{rh}^m	Matching level of assigning an unscheduled request r to an SAV h
θ_{rh}^o	Observation level of assigning request r to an SAV h
$l_{v_r^o v_r^d}$	Direct travel distance of request r from origin to destination
$\widetilde{\mathscr{T}}_h$	Currently assigned requests for SAV h in assignment operators A3 or A4
$\lvert\mathscr{K}\rvert$	Number of iterations for recording the previous obtained profit values in assignment operator A5
$\lvert\mathscr{K}\rvert_{\max}$	Maximum number of recorded iterations in assignment operator A5
ϖ_a	Weight of the assignment operator $a \in \mathscr{A}$
π_a	Recorded score of assignment operator $a \in \mathscr{A}$
ψ	Importance parameter of the historical information on the algorithm performance in the adaptive weight adjustment strategy of the ARA scheme

Data availability

Data will be made available on request.

References

Agatz, N., Erera, A., Savelsbergh, M., Wang, X., 2012. Optimization for dynamic ride-sharing: a review. Eur. J. Oper. Res. 223 (2), 295–303.

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

Attanasio, A., Cordeau, J.-F., Ghiani, G., Laporte, G., 2004. Parallel Tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. Parallel Comput. 30 (3), 377–387.

Azadeh, S.S., Atasoy, B., Ben-Akiva, M.E., Bierlaire, M., Maknoon, M.Y., 2022. Choice-driven dial-a-ride problem for demand responsive mobility service. Transp. Res. B Methodol. 161, 128–149.

Berbeglia, G., Cordeau, J.-F., Laporte, G., 2010. Dynamic pickup and delivery problems. Eur. J. Oper. Res. 202 (1), 8–15.

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

Bimpikis, K., Candogan, O., Saban, D., 2019. Spatial pricing in ride-sharing networks. Oper. Res. 67 (3), 744–769.

Bösch, P.M., Becker, F., Becker, H., Axhausen, K.W., 2018. Cost-based analysis of autonomous mobility services. Transp. Policy 64, 76–91.

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

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

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

Fagnant, D., Kockelman, K., 2018. Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin Texas. Transportation 45 (1), 143–158.

Fang, D., Xue, Y., Cao, J., Sun, S., 2021. Exploring satisfaction of choice and captive bus riders: an impact asymmetry analysis. Transp. Res. Part D: Transp. Environ. 93, 102798.

Farhan, J., Chen, T.D., 2018. Impact of ridesharing on operational efficiency of shared autonomous electric vehicle fleet. Transp. Res. Part C Emerging Technol. 93, 310–321.

Finn, A., 2011. Investigating the non-linear effects of e-service quality dimensions on customer satisfaction. J. Retail. Consum. Serv. 18 (1), 27–37.

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

Ge, Q., Han, K., Liu, X., 2021. Matching and routing for shared autonomous vehicles in congestible network. Transp. Res. Part e: Logist. Transp. Rev. 156, 102513.

Gendreau, M., Guertin, F., Potvin, J.-Y., Séguin, R., 2006. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Transp. Res. Part C Emerging Technol. 14 (3), 157–174.

Häll, C.H., Lundgren, J.T., Voß, S., 2015. Evaluating the performance of a dial-a-ride service using simulation. Public Transp. 7 (2), 139–157.

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

Häme, L., Hakula, H., 2015. A maximum cluster algorithm for checking the feasibility of dial-a-ride instances. Transp. Sci. 49 (2), 295–310.

Hazan, J., Lang, N., Ulrich, P., Chua, J., Doubara, X., & Steffens, T. (2016). Will autonomous vehicles derail trains?.

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. Transp. Res. B Methodol. 111, 395–421.

Hosni, H., Naoum-Sawaya, J., Artail, H., 2014. The shared-taxi problem: Formulation and solution methods. Transp. Res. B Methodol. 70, 303–318.

Hou, L., Li, D., Zhang, D., 2018. Ride-matching and routing optimisation: Models and a large neighbourhood search heuristic. Transp. Res. Part e: Logist. Transp. Rev. 118, 143–162.

Hua, S., Zeng, W., Liu, X., Qi, M., 2022. Optimality-guaranteed algorithms on the dynamic shared-taxi problem. Transp. Res. Part e: Logist. Transp. Rev. 164, 102809.

Huang, K., Liu, C., Zhang, C., Liu, Z., Hu, H., 2024. Shared autonomous vehicle operational decisions with vehicle movement and user travel behaviour. Travel Behav. Soc. 37, 100848.

Hyytiä, E., Penttinen, A., Sulonen, R., 2012. Non-myopic vehicle and route selection in dynamic DARP with travel time and workload objectives. Comput. Oper. Res. 39 (12), 3021–3030.

Johnson, C., & Walker, J. (2016). Peak car ownership: the market opportunity of electric automated mobility services. *Rocky Mountain Institute and Mobility Transformation.*.

Jung, J., Jayakrishnan, R., Park, J.Y., 2016. Dynamic shared-taxi dispatch algorithm with hybrid-simulated annealing. Comput. Aided Civ. Inf. Eng. 31 (4), 275–291.

Ke, J., Yang, H., Li, X., Wang, H., Ye, J., 2020. Pricing and equilibrium in on-demand ride-pooling markets. Transp. Res. B Methodol. 139, 411–431.

Lavieri, P.S., Bhat, C.R., 2019. Modeling individuals' willingness to share trips with strangers in an autonomous vehicle future. Transp. Res. A Policy Pract. 124, 242–261.

Levin, M.W., 2017. Congestion-aware system optimal route choice for shared autonomous vehicles. Transp. Res. Part C Emerging Technol. 82, 229–247.

Litman, T. (2017). Autonomous vehicle implementation predictions.

Loeb, B., Kockelman, K.M., Liu, J., 2018. Shared autonomous electric vehicle (SAEV) operations across the Austin, Texas network with charging infrastructure decisions. Transp. Res. Part C Emerging Technol. 89, 222–233.

Lokhandwala, M., Cai, H., 2018. Dynamic ride sharing using traditional taxis and shared autonomous taxis: a case study of NYC. Transp. Res. Part C Emerging Technol. 97, 45–60.

Lu, C.C., Diabat, A., Li, Y.T., Yang, Y.M., 2022. Combined passenger and parcel transportation using a mixed fleet of electric and gasoline vehicles. Transp. Res. Part e: Logist. Transp. Rev. 157, 102546.

Ma, J., Li, X., Zhou, F., Hao, W., 2017. Designing optimal autonomous vehicle sharing and reservation systems: a linear programming approach. Transp. Res. Part C Emerging Technol. 84, 124–141.

Maalouf, M., MacKenzie, C.A., Radakrishnan, S., Court, M., 2014. A new fuzzy logic approach to capacitated dynamic Dial-a-Ride problem. Fuzzy Set. Syst. 255, 30–40.

Marković, N., Nair, R., Schonfeld, P., Miller-Hooks, E., Mohebbi, M., 2015. Optimizing dial-a-ride services in Maryland: benefits of computerized routing and scheduling. Transp. Res. Part C Emerging Technol. 55, 156–165.

Martinez, L.M., Viegas, J.M., 2017. Assessing the impacts of deploying a shared self-driving urban mobility system: an agent-based model applied to the city of Lisbon, Portugal. Int. J. Transp. Sci. Technol. 6 (1), 13–27.

Mourad, A., Puchinger, J., Chu, C., 2019. A survey of models and algorithms for optimizing shared mobility. Transp. Res. B Methodol. 123 (C), 323–346.

Narayanan, S., Chaniotakis, E., Antoniou, C., 2020. Shared autonomous vehicle services: a comprehensive review. Transp. Res. Part C Emerging Technol. 111, 255–293.

Naumov, S., Keith, D., 2023. Optimizing the economic and environmental benefits of ride-hailing and pooling. Prod. Oper. Manag. 32 (3), 904–929.

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

Pisinger, D., & Ropke, S. (2018). Large neighborhood search. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of metaheuristics.*, pp. 99–127.

Potvin, J.-Y., Rousseau, J.-M., 1993. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. Eur. J. Oper. Res. 66 (3), 331–340.

Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Comput. Oper. Res. 31 (12), 1985–2002.

Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. 40 (4), 455–472.
Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S.H., Ratti, C., 2014. Quantifying the benefits of vehicle pooling with shareability networks. Proc. Natl. Acad. Sci. 111 (37), 13290–13294.
Santos, D. O., & Xavier, E. C. (2013). Dynamic taxi and ridesharing: A framework and heuristics for the optimization problem. *Twenty-Third International Joint Conference on Artificial Intelligence*, 2885–2891.
Santos, D.O., Xavier, E.C., 2015. Taxi and ride sharing: a dynamic dial-a-ride problem with money as an incentive. Expert Syst. Appl. 42 (19), 6728–6737.
Sayarshad, H.R., Chow, J.Y.J., 2015. A scalable non-myopic dynamic dial-a-ride and pricing problem. Transp. Res. B Methodol. 81, 539–554.
Sayarshad, H.R., Oliver Gao, H., 2018. A scalable non-myopic dynamic dial-a-ride and pricing problem for competitive on-demand mobility systems. Transp. Res. Part C Emerging Technol. 91, 192–208.
Schilde, M., Doerner, K.F., Hartl, R.F., 2011. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Comput. Oper. Res. 38 (12), 1719–1730.
Schilde, M., Doerner, K.F., Hartl, R.F., 2014. Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem. Eur. J. Oper. Res. 238 (1), 18–30.
Simonetto, A., Monteil, J., Gambella, C., 2019. Real-time city-scale ridesharing via linear assignment problems. Transp. Res. Part C Emerging Technol. 101, 208–232.
Souza, A.L., Bernardo, M., Penna, P.H., Pannek, J., Souza, M.J., 2022. Bi-objective optimization model for the heterogeneous dynamic dial-a-ride problem with no rejects. Optim. Lett. 16 (1), 355–374.
Stephens, T. S., Gonder, J., Chen, Y., Lin, Z., Liu, C., & Gohlke, D. (2016). *Estimated bounds and important factors for fuel use and consumer costs of connected and automated vehicles* (No. NREL/TP-5400-67216). National Renewable Energy Lab.(NREL), Golden, CO (United States).
Tafreshian, A., Abdolmaleki, M., Masoud, N., Wang, H., 2021. Proactive shuttle dispatching in large-scale dynamic dial-a-ride systems. Transp. Res. B Methodol. 150, 227–259.
Tirachini, A., 2020. Ride-hailing, travel behaviour and sustainable mobility: an international review. Transportation 47 (4), 2011–2047.
U.S. Department of Transportation. (2025). *Benefit-cost analysis guidance for discretionary grant programs.* https://www.transportation.gov/mission/office-secretary/office-policy/transportation-policy/benefit-cost-analysis-guidance.
Vidal, T., Crainic, T.G., Gendreau, M., Prins, C., 2014. A unified solution framework for multi-attribute vehicle routing problems. Eur. J. Oper. Res. 234 (3), 658–673.
Wang, H., Yang, H., 2019. Ridesourcing systems: a framework and review. Transp. Res. B Methodol. 129, 122–155.
Xiang, Z., Chu, C., Chen, H., 2008. The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. Eur. J. Oper. Res. 185 (2), 534–551.
Xu, M., 2025. A branch-and-cut-and-price algorithm for shared mobility considering customer satisfaction. Comput. Oper. Res. 177, 106998.
Yang, J., Hu, L., Jiang, Y., 2022. An overnight relocation problem for one-way carsharing systems considering employment planning, return restrictions, and ride sharing of temporary workers. Transp. Res. Part e: Logist. Transp. Rev. 168, 102950.
Zhan, X., Szeto, W. Y., & (Michael) Chen, X. (2022). A simulation–optimization framework for a dynamic electric ride-hailing sharing problem with a novel charging strategy. *Transportation Research Part E: Logistics and Transportation Review*, *159*, 102615.
Zhan, X., Szeto, W.Y., Shui, C.S., Chen, X.(Michael), 2021. A modified artificial bee colony algorithm for the dynamic ride-hailing sharing problem. Transp. Res. Part e: Logist. Transp. Rev. 150, 102124.
Zhang, W., Guhathakurta, S., Fang, J., Zhang, G., 2015. The performance and benefits of a shared autonomous vehicles based dynamic ridesharing system: an agent-based simulation approach. Transp. Res. Board 94th Annual Meet. 15, 2919.
Zhou, Z., Roncoli, C., 2022. A scalable vehicle assignment and routing strategy for real-time on-demand ridesharing considering endogenous congestion. Transp. Res. Part C Emerging Technol. 139, 103658.