CIRRELT-2025-10

Dynamic Shipment-to-Service Matching for Interurban Synchromodal Transport Systems with Shared Resources

Wenjing Guo
Teodor Gabriel Crainic
Michel Gendreau
Wenfeng Li
Walter Rei

March 2025

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

Dynamic Shipment-to-Service Matching for Interurban Synchromodal Transport Systems with Shared Resources

Wenjing Guo[1,2], Teodor Gabriel Crainic[1,3,*], Michel Gendreau[1,4], Wenfeng Li[2], Walter Rei[1,3]

1. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
2. Wuhan University of Technology, Wuhan, China
3. School of Management, Université du Québec à Montréal
4. Department of Mathematical and Industrial Engineering, Polytechnique Montréal

Abstract. This paper investigates a dynamic shipment-to-service matching problem on interurban transport networks in which a synchromodal platform aims to provide optimal decisions on the acceptance of dynamic shipment requests received from shippers and dynamic service offers received from carriers, shipment-to-service assignments, shipment itineraries, and service time schedules. Each shipment could be matched with multiple services for a multimodal itinerary, each service could be matched with multiple shipments to build loads. Additionally, services have limited capacities on designated routes, which may be served by one or multiple vehicles using the same or different modes of transport. Besides, we consider time-varying handling and storage capacities at transshipment nodes. This paper develops a mixed integer linear programming model to formulate the matching problem. Due to the computational complexity, a preprocessing-based adaptive large neighborhood search algorithm is designed to solve the problem. Shipments' feasible itineraries are preprocessed to avoid route generation in each iteration of the algorithm. To address dynamic events, a rolling horizon framework is employed, allowing for the re-optimization of active requests and offers at each decision time. Extensive numerical experiments are conducted to verify the validity of the proposed approaches.

Keywords: Interurban freight transportation; Synchromodal platform; Dynamic shipment-to-service matching; Mixed integer linear programming model; Preprocessing-based adaptive

Acknowledgments: This research was supported by the National Natural Science Foundation of China under Grants 72301203 and 62173263. While working on the project, the second author held the UQAM Chair on Intelligent Logistics and Transportation Systems Planning, and was Adjunct Professor, Department of Computer Science and Operations Research, Université de Montréal, while the fifth held the Canada Research Chair in Stochastic Optimization of Transport and Logistics Systems. The authors gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) through its Collaborative Research and Development, Discovery, Acceleration, and Engage grant programs, and the support of Fonds de recherche du Québec through their CIRRELT infrastructure grants.

---

* Corresponding author:teodorgabriel.crainic@cirrelt.net

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2025

1 Introduction

Synchromodal transport systems that facilitate the movement of freights over multimodal networks based on mode-free booking, resource sharing, differentiated fare classes, and real-time planning are increasingly recognized in response to the trend towards sustainability (SteadieSeifi et al., 2014; Giusti et al., 2019; Archetti et al., 2022; Sakti et al., 2023; Zhang et al., 2025). Currently, interurban freight transport is dominated by trucks, which causes high transportation costs, high empty drives, and heavy carbon emissions (Larsen et al., 2023). With the challenges of global warming, green transportation modes, such as high-speed railways and drones, become increasingly appealing for time-sensitive and high-valued shipments (Zhen et al., 2024; Madani et al., 2024). The integration of network connectivity is revolutionizing interurban transport systems and providing diverse mobility options, including trains, barges, trucks, planes, drones, autonomous vehicles, etc., as shown in Figure 1. These advancements promise a more reliable, efficient, seamless, and sustainable experience for interurban freight transportation. Nonetheless, this evolution makes the transport planning problem more complex with the consideration of scheduled services and flexible services, line services (with multiple intermediate stops) and shuttle services, contracted services (i.e., the services with known schedules and capacities) and spot services (i.e., the services that are unknown before their announcements), and the synchronization of transshipments between different services.

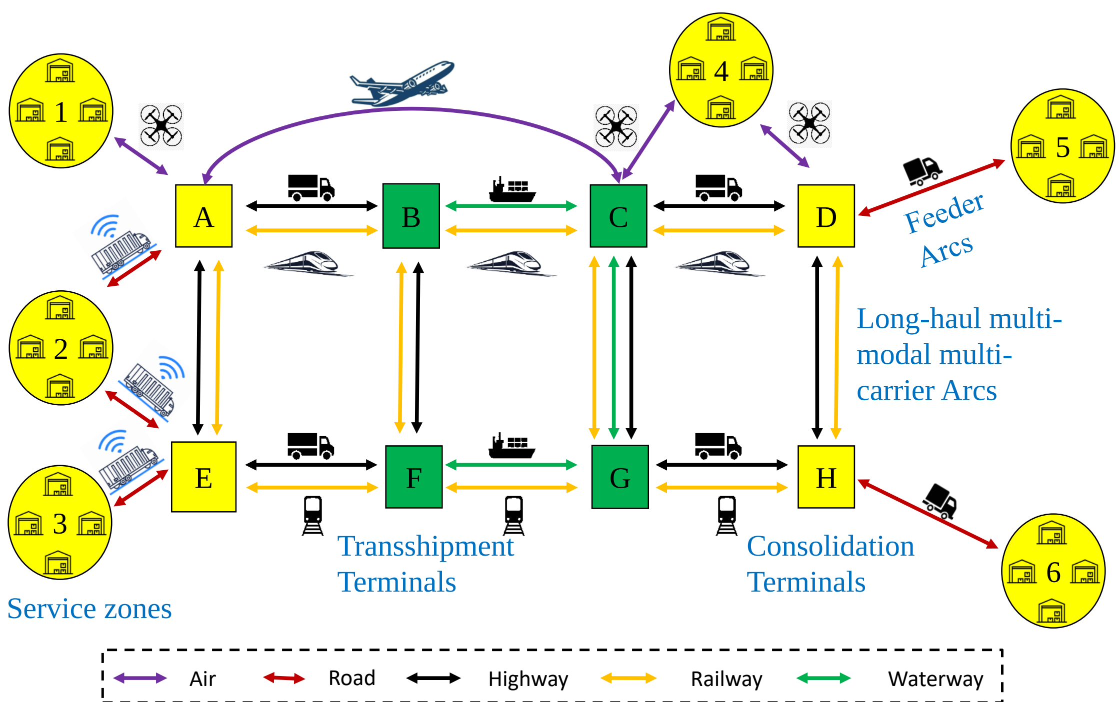


Figure 1: An illustrative example of interconnected interurban multimodal networks.

With the rapid development of information and communication technologies, online freight platforms that provide matches between shippers and carriers are flourishing (Miller et al., 2020; Park et al., 2023; Yang et al., 2024; Wu et al., 2024), such as Quicargo, Uber Freight, Amazon Mechanical Turk, and Meituan. These online platforms could provide efficient logistics services by consolidating shipments from different shippers and sharing resources among different carriers. In this paper, we investigate a dynamic shipment-to-service matching problem of a synchromodal platform for interurban freight transportation, as shown in Figure 2. On the one side of the platform, many shippers (e.g., producers, wholesalers, and distributors) make shipment requests for cost and time-efficient transportation of their product loads. Each accepted shipment needs to be transported from a given shipper location to a consignee location within given time windows. On the other side, many carriers (e.g., railway companies, truckload carriers) make service offers for transport capacities and request profitable loads. Each service provides a limited transport capacity on a specific route with or without time schedules, served by one or multiple vehicles with the same or different modes. The platform receives requests and offers continuously over time and optimizes in time and space the selection of shipment requests and service offers, shipment-to-service assignments, shipment itineraries, and service schedules through consolidation of shipments from different shippers into the same vehicles and synchronization of activities in an interconnected multimodal transport network. The recent developments in information technologies such as cloud computing and Internet of Things allow real-time monitoring of shipments' and vehicles' status and information sharing among stakeholders, which facilitates the adoption of such a platform in practice.

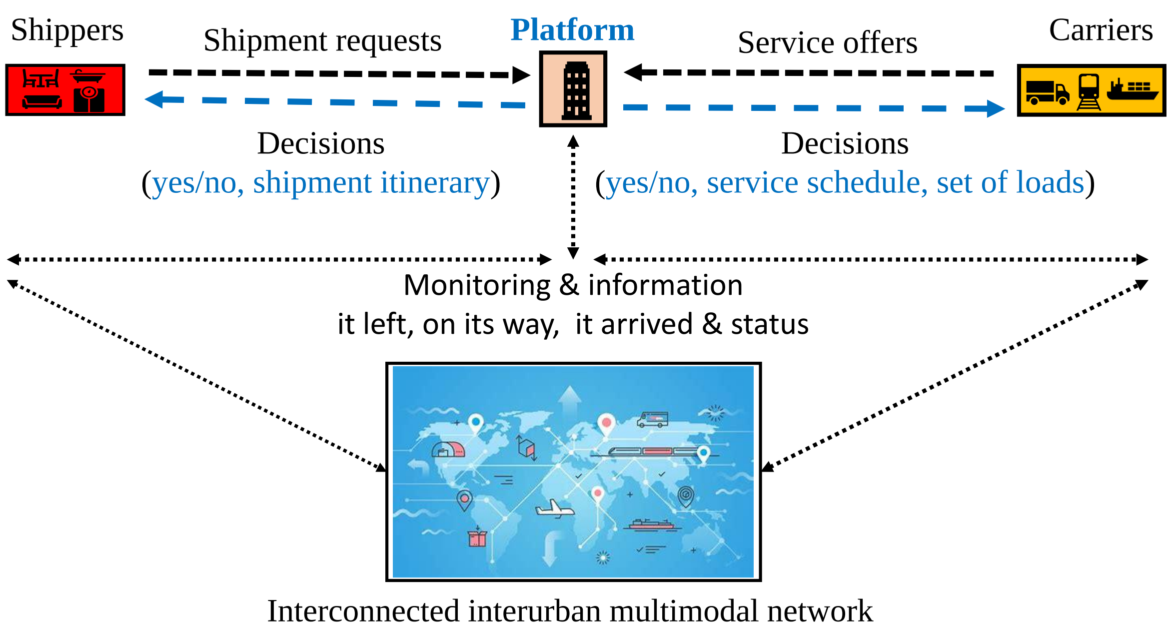


Figure 2: A synchromodal platform for interurban freight transportation.

In the literature, most of the studies focus either on dynamic pickup and delivery problems within road networks (Berbeglia et al., 2010; Ferrucci and Bock, 2014; Arslan et al., 2019; Ulmer et al., 2021; Mousavi et al., 2024; Stoia et al., 2025) or on dynamic transport planning problems within multimodal networks (Li et al., 2015; Qu et al., 2019; Guo et al., 2020, 2021; Rivera and Mes, 2022; Zhang et al., 2023; Larsen et al., 2023; Filom and Razavi, 2025). However, none of the existing studies addresses the dynamic shipment-to-service matching within interurban multimodal networks. To the best of our knowledge, this is the first paper that considers the selection of shipment requests and service offers simultaneously in multimodal transport systems at the operational level. To bridge this gap in the literature, the first contribution of this paper is the development of a mathematical model that integrates the decisions on the acceptance or rejection of shipment requests and service offers, along with decisions on shipment-to-service assignments, shipment itineraries, and service time schedules. Additionally, in consolidation and transshipment terminals, the resource for handling and storage operations is limited, so we consider time-varying capacities in the matching model. Besides, requests and offers arrive at the platform dynamically, the decisions made at each time are not all to be put into practice. In this paper, we develop a rolling horizon framework to control the implementation and re-optimization of decisions when new requests and offers are received. To produce good-quality solutions rapidly at each decision epoch, a preprocessing-based adaptive large neighborhood search (P-ALNS) algorithm is designed to solve the optimization problem. Finally, we conduct extensive numerical experiments to evaluate the performance of the rolling horizon approach in comparison to a first-come-first-serve approach that does not consider re-optimization, and assess the efficiency of the P-ALNS algorithm in terms of computation time and solution quality in comparison to the CPLEX solver.

The remainder of this paper is organized as follows: Section 2 reviews related works. Section 3 describes the dynamic shipment-to-service matching problem. Section 4 formulates the mathematical model. Section 5 designs the heuristic algorithm. Section 6 presents the rolling horizon framework. The experimental settings and results are provided in Section 7. Section 8 concludes and gives future research directions.

2 Related works

This paper investigates a dynamic shipment-to-service matching problem for interurban synchromodal transport systems. Four types of decisions are included in the dynamic matching process: acceptance decision of shipment requests and service offers, shipment-to-service assignment, shipment itineraries, and service schedules. The articles related to this paper can be divided into two groups: pickup and delivery problems with transshipment and synchromodal transport planning.

2.1 Pickup and delivery problems with transshipment

The pickup and delivery problem with transshipment (PDPT) aims to find the optimal routes for a fleet of capacitated vehicles to satisfy shipment requests in which requests can be transferred from one vehicle to another at transshipment nodes (Mitrović-Minić and Laporte, 2006). Qu and Bard (2012) developed a greedy randomized adaptive search procedure for the PDPT which considers transshipment between different aircraft. Rais et al. (2014) introduced a mixed-integer linear programming model for the PDPT involving a fleet of heterogeneous vehicles with different capacities. Wolfinger and Salazar-González (2021) presented an arc-based mixed-integer formulation for the PDPT, accommodating a fleet of heterogeneous vehicles, and proposed a branch-and-cut algorithm to solve it. Lyu and Yu (2023) introduced a new mixed-integer linear programming model for the PDPT with a fleet of heterogeneous vehicles and requests' time windows, and the time synchronization between different vehicles at transshipment nodes was considered.

All the above studies on PDPT involve transshipment among a fleet of vehicles with flexible routes and time schedules. To explore the feasibility of using available public transportation vehicles for freight transport, Ghilas et al. (2016a,b, 2018) proposed the transshipment between a fleet of pickup and delivery vehicles and scheduled public transportation lines (e.g., bus, train, and metro) that operate according to predetermined routes and schedules in urban transportation. Their models account for time and spatial coordination between flexible vehicles and scheduled services at transshipment nodes, as well as shipment consolidation considering vehicles' capacity limitations. However, in their model, public transportation lines are formulated as arcs with only an origin and a destination, which prohibits shipments' loading and unloading at intermediate stops. Zhen et al. (2023, 2024) investigated the transshipment between a fleet of trucks and high-speed railways (HSR) in interurban transportation by integrally addressing vehicle arrangement, station selection, freight allocation, and the optimization of HSR freight routes. In their model, the routes and time schedules of HSR are both flexible, and the pickup and delivery operations of shipment requests are restricted by time points.

Motivated by the growth of digital platforms in freight markets, several recent papers have studied dynamic pickup and delivery problems (PDP) for freight transportation where demand requests and supply offers arrive dynamically. Arslan et al. (2019) investigated a dynamic PDP for a crowdsourced delivery platform that automatically creates matches between parcel delivery tasks and ad hoc drivers. Zhang et al. (2022) introduced an online crowdsourced truck delivery problem where orders arrive online and truck carriers decide whether to supply a group of available trucks to serve the order. Su et al. (2023) addressed a dynamic PDP with crowdsourced bids and transshipment in last-mile delivery, where all requests can be satisfied by either using their own vehicle fleet or outsourcing with a small compensation to crowdshippers through transshipment facilities. Behrendt et al. (2024) researched a capacity planning problem for a crowdsourced delivery platform that operators as an intermediary between consumers who require delivery tasks and couriers who make these deliveries.

2.2 Synchromodal transport planning

Synchromodality can be regarded as a further evolution of intermodality by involving different and strongly integrated transportation modes (Giusti et al., 2019). It aims to provide efficient, reliable, and sustainable services through the coordination and co-operation of stakeholders, the synchronization of operations, mode-free booking, and real-time planning (Sakti et al., 2023; Zhang et al., 2025). In synchromodal transportation, dynamic updating of transport plans based on real-time information plays a key role in differentiating synchromodality from other paradigms. The most common dynamic events are the arrival of new shipment requests, but service offers, travel times, and transfer capacities are possible dynamics as well. In the literature, Li et al. (2015) presented a receding horizon intermodal container flow control approach to deal with the dynamic transport demands and dynamic traffic conditions in hinterland transportation. van Riessen et al. (2016) designed a decision tree to derive real-time decision rules for suitable allocation of shipment requests to services. Qu et al. (2019) proposed a re-planning framework to re-plan shipment routes and service schedules when uncertainties cause deviations from the original plan. Guo et al. (2020) developed a rolling horizon approach to handle shipment requests that arrive dynamically in a synchromodal matching platform for hinterland transportation. Rivera and Mes (2022) proposed an algorithm based on approximate dynamic programming to tackle the curse of dimensionality of a Markov decision process model for anticipatory freight selection in intermodal long-haul round-trips. Larsen et al. (2023) developed a real-time co-planning framework for decentralized synchromodal transport with two decision makers. Zhang et al. (2023) developed an online deep reinforcement learning approach to re-plan vehicle routes and shipment itineraries in response to service time uncertainties. Labarthe et al. (2024) proposed a model-based decision support approach for on-demand freight delivery services in urban areas enabled by synchromodality and synergy in passenger and freight mobility. Filom and Razavi (2025) presented a learning-based robust optimization framework for synchromodal freight transportation to drive data-driven explainable decisions.

2.3 Summary

Table 1 summarizes the formulation characteristics and methodologies of relevant literature. Among these studies, the work by Guo et al. (2021) is the most comparable to this paper. In contrast to Guo et al. (2021), this paper addresses the selection of both shipment requests and service offers. Additionally, we consider shipment requests' pickup and delivery time windows instead of time points, and establish departure time windows for flexible services. Furthermore, this paper investigates interurban transport systems that involve both line services (such as high-speed railways in long-haul transportation) and shuttle services (such as drones in feeder arcs). Importantly, we also incorporate time-varying transshipment capacity limitations, and treat both shipment requests and service offers as dynamic events.

Table 1: The formulation characteristics and methodologies of relevant literature.

Articles	Formulation characteristics							Methodologies	
	Network	Decisions[1]	Time windows	Service type	Time schedules	Transshipment capacity limitations	Dynamic events	Dynamic approach[2]	Optimization algorithm[3]
Pickup and delivery problem with transshipment									
Qu and Bard (2012)	Urban, air+air	VR, SR	Pickup, delivery	A fleet of aircraft	Flexible	Unlimited			GRASP
Rais et al. (2014)	urban road+road	VR, SR	Pickup, delivery	A fleet of vehicles	Flexible	Unlimited			MILP
Ghilas et al. (2016a)	Urban, Road+metro	VR, SR	Pickup, delivery	Shuttle services, a fleet of vehicles	Scheduled, flexible	Unlimited			ALNS
Ghilas et al. (2018)	Urban, Road+metro	VR, SR	Pickup, delivery	Shuttle services, a fleet of vehicles	Scheduled, flexible	Unlimited			BP
Zhen et al. (2023)	Interurban, road+HSR	VA, SIS, FA, SR		A fleet of vehicles and HSR	Flexible	Time-constant			Meta-heuristic
Zhen et al. (2024)	Interurban, road+HSR	VA, SIS, FA, SR	Time points	A fleet of vehicles and HSR	Flexible	Time-constant			Meta-heuristic
Synchromodal transport planning									
Qu et al. (2019)	Inland, synchro-modal	SR, SS	Time points	Line services	Scheduled	Unlimited	Regular disturbances	RPF	CPLEX solver
Guo et al. (2020)	Inland, synchro-modal	SSA, SR, SS	Time points	Shuttle services	Scheduled, flexible	Unlimited	Shipment requests	RHA	HA
Guo et al. (2021)	Global, synchro-modal	ASR, SSA, SR, SS	Time points	Shuttle services	Scheduled, flexible	Unlimited	Shipment requests, travel times	RHA	HA
Rivera and Mes (2022)	Inland, synchro-modal	SR	Time points	Shuttle services	Scheduled	Unlimited	Containers	MDP	ADP
Larsen et al. (2023)	Inland, synchro-modal	SSA, SR, VR, SS	Time points	Shuttle services	Scheduled, flexible	Unlimited	Shipment requests	RHA	HA
Zhang et al. (2023)	Inland, synchro-modal	SSA, SR, VR	Pickup, delivery	A fleet of vehicles	Time windows	Unlimited	Service times	DRL	ALNS
Labarthe et al. (2024)	Urban, synchro-modal	SR	Time points	Line services, shuttle services	Scheduled	Time-constant			HA
Filom and Razavi (2025)	Inland, synchro-modal	ASR, SSA, SR	Time points	Shuttle services	Scheduled	Unlimited	Shipment requests	LROF	HA
This paper	Interurban, synchro-modal	ASR, ASO, SSA, SR, SS	Pickup, delivery	Line services, shuttle services	Scheduled, flexible	Time-varying	Shipment requests, service offers	RHA	P-ALNS

[1] VR: Vehicle routing; SR: Shipment routing; VA: Vehicle arrangement; SIS: Selection of intermodal station; FA: freight allocation; SS: Service scheduling; SSA: Shipment-to-service assignment; ASR: Acceptance of shipment requests; ASO: Acceptance of service offers

[2] RPF: Re-planning framework; RHA: Rolling horizon approach; MDP: Markov decision process; DRL: Deep reinforcement learning; LROF: Learning-based robust optimization framework;

[3] GRASP: Greedy randomized adaptive search procedure; MILP: Mixed-integer linear programming; ALNS: Adaptive large neighborhood search; BP: Branch-and-price; HA: Heuristic algorithm; ADP: Approximate dynamic programming; P-ALNS: Preprocessing-based adaptive large neighborhood search

3 Problem definition

In this section, first, we define the notation of service zones, consolidation and transshipment terminals, shipment requests, and service offers. After that, we define the system states and active events at any decision time.

3.1 Transshipment nodes

Let $\mathcal{Z}$ stand for the set of service zones where to pick up or deliver shipments. Zones can be urban areas or any geographically, administratively or commercially relevant area. In this paper, we do not consider the vehicle routing problem within zones and focus on the level of interurban transport planning by aggregating demands in the same zone. For zone $z \in \mathcal{Z}$, define:

- t^{P}_{zm}: Pickup/delivery time with mode $m \in \mathcal{M}$ at zone z;
- c^{P}_{zm}: Pickup/delivery cost per volume with mode $m \in \mathcal{M}$ at zone z.

3.2 Terminals

Let Θ be the set of consolidation and transshipment terminals where shipments can be directly transferred from one vehicle to another with the same or different mode, or be temporarily stored at terminals waiting for further transportation. While consolidation terminals link feeder arcs and long-haul arcs, transshipment terminals link between long-haul transport arcs. Notice that more than one consolidation terminal may service a single zone, particularly for large zones with important populations, industrial density, or access to several major modal infrastructures. A large city may thus have several terminals linking it to different geographical regions, or be serviced simultaneously by a maritime or river port, an airport, one or several rail yards, and several truck carriers. For terminal $i \in \Theta$, define:

- $t^{\mathtt{L}}_{im}$: Loading/unloading time with mode $m \in \mathcal{M}$ at terminal i;
- $c^{\mathtt{L}}_{im}$: Loading/unloading cost per volume with mode $m \in \mathcal{M}$ at terminal i;
- $c^{\mathtt{W}}_{i}$: Storage cost per volume per time unit at terminal i;
- $u^{\mathtt{L}}_{i}$: Maximum loading and unloading capacity at terminal i;
- $u^{\mathtt{W}}_{i}$: Maximum storage capacity at terminal i.

3.3 Shipment requests

Let $\mathfrak{R}$ be the set of shipment requests received from shippers. For request $\mathfrak{r} \in \mathfrak{R}$, define:

- $o(\mathfrak{r})$: Origin, $o(\mathfrak{r}) \in \mathcal{Z}$, the shipper facility within a service zone;
- $d(\mathfrak{r})$: Destination, $d(\mathfrak{r}) \in \mathcal{Z}$, the consignee facility within a service zone;
- $u(\mathfrak{r})$: Shipment volume, the number of packages/objects/containers which is consistent with the loading unit;
- $\alpha^{\mathtt{A}}(\mathfrak{r})$: Announce time, the time when the platform receives the request;
- $[\alpha^{\mathtt{R}}(\mathfrak{r}), \beta^{\mathtt{R}}(\mathfrak{r})]$: Pickup time window;
- $[\alpha^{\mathtt{E}}(\mathfrak{r}), \beta^{\mathtt{E}}(\mathfrak{r})]$: Early delivery time window;
- $[\alpha^{\mathtt{L}}(\mathfrak{r}), \beta^{\mathtt{L}}(\mathfrak{r})]$: Late delivery time window;
- $[\beta^{\mathtt{E}}(\mathfrak{r}), \alpha^{\mathtt{L}}(\mathfrak{r})]$: Target delivery time window;
- $\rho(\mathfrak{r})$: Fare, the revenue received from shippers if request $\mathfrak{r}$ is accepted;
- $\psi^{\mathtt{E}}(\mathfrak{r})$: Early delivery penalty per time unit;
- $\psi^{\mathtt{L}}(\mathfrak{r})$: Late delivery penalty per time unit.

Figure 3 shows the timeline of shipment request $\mathfrak{r} \in \mathfrak{R}$. We define $\alpha^{\mathtt{A}}(\mathfrak{r}) \leq \alpha^{\mathtt{R}}(\mathfrak{r}) \leq \beta^{\mathtt{R}}(\mathfrak{r}) \leq \alpha^{\mathtt{E}}(\mathfrak{r}) \leq \beta^{\mathtt{E}}(\mathfrak{r}) \leq \alpha^{\mathtt{L}}(\mathfrak{r}) \leq \beta^{\mathtt{L}}(\mathfrak{r})$. In the problem setting, we consider a synchromodal platform that offers differentiated fare classes as incentives to motivate shippers opt for mode-free booking. For each origin-destination pair, the platform provides multiple fare classes tailored to the needs of shippers. Each fare class is defined by a combination of fare price, lead time (i.e, the time between earliest pickup time $\alpha^{\mathtt{R}}(\mathfrak{r})$ and latest delivery time $\beta^{\mathtt{L}}(\mathfrak{r})$), and penalty cost for early and later delivery. For high valued and urgent cargoes, shippers can choose a fare class with a higher price, a shorter lead time, and a higher penalty cost; for regular products, shippers may prefer a fare class with a lower price, a longer lead time, and a lower penalty cost.

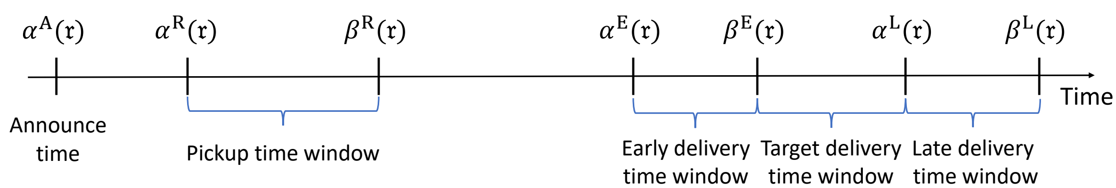


Figure 3: Timeline for shipment request $\mathfrak{r}$.

Shipment requests can be divided into two groups: contractual requests $\mathfrak{R}^{\text{contract}}$ received from long-term contracts and spot requests $\mathfrak{R}^{\text{spot}}$ received in real-time. For a contractual request $\mathfrak{r} \in \mathfrak{R}^{\text{contract}}$, its announce time $\alpha^{\mathtt{A}}(\mathfrak{r}) = 0$, and all the attributes are known in advance. Conversely, for a spot request $\mathfrak{r} \in \mathfrak{R}^{\text{spot}}$, its attributes are known to the system after its announce time $\alpha^{\mathtt{A}}(\mathfrak{r}) > 0$, the platform must decide whether to accept or reject it.

3.4 Service offers

Let $\mathfrak{O}$ be the set of service offers proposed by carriers. In the problem setting, we consider carriers have a fleet of vehicles with various modes (such as trucks and drones) operating between service zones and consolidation terminals, or have capacities on container slots or cargo space on multimodal corridors (such as barges, trains, or airplanes) in long-haul transportation. Carriers offer capacity for interurban transportation by offering time-scheduled or time flexible services, line services or shuttle services, as shown in Figure 4. The platform does not manage carriers fleets and operations, it concentrates on the acceptance or rejection of these offers and optimizing comprehensively the assignment of shipment requests to service offers.

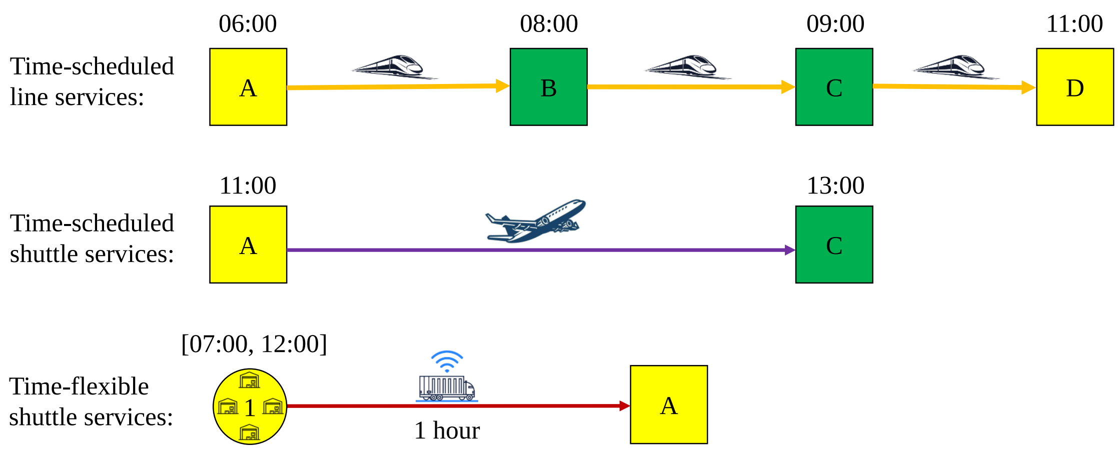


Figure 4: An illustration example of different service types in interurban transportation.

For service offer $\mathfrak{o} \in \mathfrak{O}$, define:

- type($\mathfrak{o}$): Service type, 1 if $\mathfrak{o}$ is a time-scheduled service, 0 otherwise;
- $\Pi(\mathfrak{o})$: Route of service $\mathfrak{o}$, i.e., the sequence of service segments, $\Pi(\mathfrak{o}) = \{\pi_l(\mathfrak{o}) \mid l = 1, \ldots, |\Pi(\mathfrak{o})|\}$; for line services, $|\Pi(\mathfrak{o})| \geq 2$; for shuttle services, $|\Pi(\mathfrak{o})| = 1$;
- $o_l(\mathfrak{o})$: Origin of segment $\pi_l(\mathfrak{o})$, $l = 1, \ldots, |\Pi(\mathfrak{o})|$;

- $d_l(\mathfrak{o})$: Destination of segment $\pi_l(\mathfrak{o})$, $l = 1, \ldots, |\Pi(\mathfrak{o})|$;
- $m_l(\mathfrak{o})$: Mode of segment $\pi_l(\mathfrak{o})$, $l = 1, \ldots, |\Pi(\mathfrak{o})|$;
- $u_l(\mathfrak{o})$: Maximum capacity of segment $\pi_l(\mathfrak{o})$, $l = 1, \ldots, |\Pi(\mathfrak{o})|$;
- $\alpha^{\mathtt{A}}(\mathfrak{o})$: Announce time, the time when the platform receives the offer;
- $[\alpha_l^{\mathtt{R}}(\mathfrak{o}), \beta_l^{\mathtt{R}}(\mathfrak{o})]$: Departure time window of segment $\pi_l(\mathfrak{o})$ at its origin $o_l(\mathfrak{o})$; for time scheduled service offer $\mathfrak{o} \in \{\mathfrak{O}|\text{type}(\mathfrak{o}) = 1\}$, $\alpha_l^{\mathtt{R}}(\mathfrak{o}) = \beta_l^{\mathtt{R}}(\mathfrak{o})$;
- $\tau_l(\mathfrak{o})$: Travel time of segment $\pi_l(\mathfrak{o})$, $l = 1, \ldots, |\Pi(\mathfrak{o})|$;
- $f(\mathfrak{o})$: Fixed cost of service $\mathfrak{o}$;
- $c_l(\mathfrak{o})$: Variable transportation cost of segment $\pi_l(\mathfrak{o})$ per volume, $l = 1, \ldots, |\Pi(\mathfrak{o})|$.

Service offers can also be divided into two groups: contractual offers $\mathfrak{O}^{\text{contract}}$ received from long-term contracts and spot offers $\mathfrak{O}^{\text{spot}}$ received in real-time. For a contractual offer $\mathfrak{o} \in \mathfrak{O}^{\text{contract}}$, its announce time $\alpha^{\mathtt{A}}(\mathfrak{o}) = 0$, and all the attributes are known in advance. Conversely, for a spot offer $\mathfrak{o} \in \mathfrak{O}^{\text{spot}}$, its attributes are known after its announce time $\alpha^{\mathtt{A}}(\mathfrak{o}) > 0$, the platform must decide whether to accept or reject it.

3.5 System states and active events

The evolution of the synchromodal platform is indexed by a discrete time variable $t \in [0, ..., \infty)$. At time 0, the platform has contractual shipment requests and service offers received from long-term contracts. At time $t \in (0, ..., \infty)$, the platform receives spot shipment requests and service offers from the spot market. We denote time period $t \in [1, ..., \infty)$ as the duration from time $t-1$ to time t. Requests and offers received during time period t will be kept until decision time $t \in [1, ..., \infty)$. At any time t, decisions are made over a planning horizon T. The planning horizon is formed of a number of consecutive periods, starting at time t, and finishing at time $t+T$.

At time $t \in \{0, 1, ...\}$, decisions are made based on the current system state. The system state consists of the status of requests, offers, and terminals.

- Let binary parameter statu1($\mathfrak{r}$) record request $\mathfrak{r}$'s decision status: 1 if request $\mathfrak{r} \in \mathfrak{R}$ is accepted, 0 otherwise; let binary parameter statu2($\mathfrak{r}$) record request $\mathfrak{r}$'s operation status: 1 if request $\mathfrak{r} \in \mathfrak{R}$ is picked up from its origin zone, 0 otherwise.
- Let binary parameter statu3($\mathfrak{o}$) record offer $\mathfrak{o}$'s decision status: 1 if offer $\mathfrak{o} \in \mathfrak{O}$ is accepted, 0 otherwise. We denote $u_l^t(\mathfrak{o})$ as the available transport capacity of segment $\pi_l(\mathfrak{o})$ at decision time t, $\mathfrak{o} \in \mathfrak{O}, l \in \{1, ..., |\Pi(\mathfrak{o})|\}$.

- We denote $u_i^{\mathrm{L},t,p}$ as the available loading and unloading capacity at terminal $i \in \Theta$ during time period $p \in \{t+1, ..., t+T\}$ at decision time t; let $u_i^{\mathrm{W},t,p}$ be the available storage capacity at terminal $i \in \Theta$ during time period $p \in \{t+1, ..., t+T\}$ at decision time t.

At time t, the active requests over the planning horizon T consists of two groups: accepted requests not yet picked up and new requests. Let $\dot{\mathfrak{R}}^t$ be the set of accepted requests but not yet picked up, $\dot{\mathfrak{R}}^t = \{\mathfrak{r} \in \mathfrak{R} : \alpha^{\mathtt{A}}(\mathfrak{r}) \le t-1, \mathrm{statu1}(\mathfrak{r}) = 1, \mathrm{statu2}(\mathfrak{r}) = 0\}$; let $\tilde{\mathfrak{R}}^t$ be the set of new requests received during time period t, $\tilde{\mathfrak{R}}^t = \{\mathfrak{r} \in \mathfrak{R}^{\mathrm{spot}} : t-1 < \alpha^{\mathtt{A}}(\mathfrak{r}) \le t, \mathrm{statu1}(\mathfrak{r}) = 0, \mathrm{statu2}(\mathfrak{r}) = 0\}$. We denote active requests $\mathfrak{R}^t = \dot{\mathfrak{R}}^t \cup \tilde{\mathfrak{R}}^t$.

At time t, the active offers over the planning horizon T consists of two groups: accepted offers and new offers. Let $\dot{\mathfrak{O}}^t$ represent the set of accepted offers, $\dot{\mathfrak{O}}^t = \{\mathfrak{o} \in \mathfrak{O} : \alpha^{\mathtt{A}}(\mathfrak{o}) \le t-1, \mathrm{statu3}(\mathfrak{o}) = 1\}$; let $\tilde{\mathfrak{O}}^t$ be the set of new offers received during time period t, $\tilde{\mathfrak{O}}^t = \{\mathfrak{o} \in \mathfrak{O}^{\mathrm{spot}} : t-1 < \alpha^{\mathtt{A}}(\mathfrak{o}) \le t, \mathrm{statu1}(\mathfrak{o}) = 0\}$. We denote active offers $\mathfrak{O}^t = \dot{\mathfrak{O}}^t \cup \tilde{\mathfrak{O}}^t$.

4 Optimization model

In this section, we design a mixed integer linear programming model to formulate the shipment-to-service matching problem at each decision time. At time t, the platform decides to accept or reject newly received shipment requests $\tilde{\mathfrak{R}}^t$ and offers $\tilde{\mathfrak{O}}^t$, and decides on shipment-to-service assignments, service schedules, and shipment itineraries for active requests $\mathfrak{R}^t$ and offers $\mathfrak{O}^t$:

- *Acceptance decisions.* Let $y_{\mathfrak{r}}^t$ be the binary variable which is 1 if request $\mathfrak{r} \in \tilde{\mathfrak{R}}^{\mathfrak{t}}$ is accepted at time t, otherwise 0; let $y_{\mathfrak{o}}^t$ be the binary variable which is 1 if offer $\mathfrak{o} \in \tilde{\mathfrak{O}}^t$ is accepted at time t, otherwise 0.

- *Assignment decisions.* Let $x_{\mathfrak{r}\pi_l(\mathfrak{o})}^t$ be the binary variable which is 1 if shipment $\mathfrak{r}$ is assigned to segment $\pi_l(\mathfrak{o})$ at time t, 0 otherwise, $\mathfrak{r} \in \mathfrak{R}^t$, $\mathfrak{o} \in \mathfrak{O}^t$, $l \in \{1, \ldots, |\Pi(\mathfrak{o})|\}$; let $z_{\mathfrak{r}i\pi_l(\mathfrak{o})\pi_{l'}(\mathfrak{o}')}^t$ be the binary variable which is 1 if shipment $\mathfrak{r}$ will be transferred at terminal $i \in \Theta$ between segment $\pi_l(\mathfrak{o})$ and segment $\pi_{l'}(\mathfrak{o}')$ decided at time t, $\mathfrak{o} \neq \mathfrak{o}'$.

- *Service schedules.* For service $\mathfrak{o} \in \{\mathfrak{O}^t | \mathrm{type}(\mathfrak{o}) = 0\}, \pi_l(\mathfrak{o}) \in \Pi(\mathfrak{o})$, let $D_l(\mathfrak{o})$ be the departure time of segment $\pi_l(\mathfrak{o})$ at its origin $o_l(\mathfrak{o})$; let $A_l(\mathfrak{o})$ be the arrival time of segment $\pi_l(\mathfrak{o})$ at its destination $d_l(\mathfrak{o})$.

- *Shipment itineraries.* Given the assigned service segments and service schedules, shipment itineraries can be calculated. Let $\Gamma_{\mathfrak{r}}^{\mathrm{pickup}}$ represent the planned time that shipment $\mathfrak{r} \in \mathfrak{R}^t$ will be picked up at its origin; let $\Gamma_{\mathfrak{r}}^{\mathrm{delivery}}$ be the planned time that shipment $\mathfrak{r}$ will be delivered at its destination; let $\Gamma_{\mathfrak{r}i}^{\mathrm{unload}}$ be the planned time that shipment $\mathfrak{r}$ will be started unloading at terminal $i \in \Theta$; let $\Gamma^{\text{store}}_{\mathfrak{r}i}$ be the planned time that shipment $\mathfrak{r}$ will be started storage at terminal i; let $w_{\mathfrak{r}i}$ be the storage time of shipment $\mathfrak{r}$ at terminal i; let $\Gamma^{\text{load}}_{\mathfrak{r}i}$ be the planned time that shipment $\mathfrak{r}$ will be started loading at terminal i;let $\Gamma^{\text{depart}}_{\mathfrak{r}i}$ be the planned time that shipment $\mathfrak{r}$ will be departed from terminal i. Let $g^{\text{unload},p}_{\mathfrak{r}i}$ be the binary variable which equals 1 if shipment $\mathfrak{r}$ will be unloaded at terminal i during time period $p \in \{t+1, ..., t+T\}$; let $g^{\text{store},p}_{\mathfrak{r}i}$ be the binary variable which equals 1 if shipment $\mathfrak{r}$ will be stored at terminal i during time period $p \in \{t+1, ..., t+T\}$; let $g^{\text{load},p}_{\mathfrak{r}i}$ be the binary variable which equals 1 if shipment $\mathfrak{r}$ will be loaded at terminal i during time period $p \in \{t+1, ..., t+T\}$. Figure 5 shows an illustrative example of a shipment itinerary in time and space.

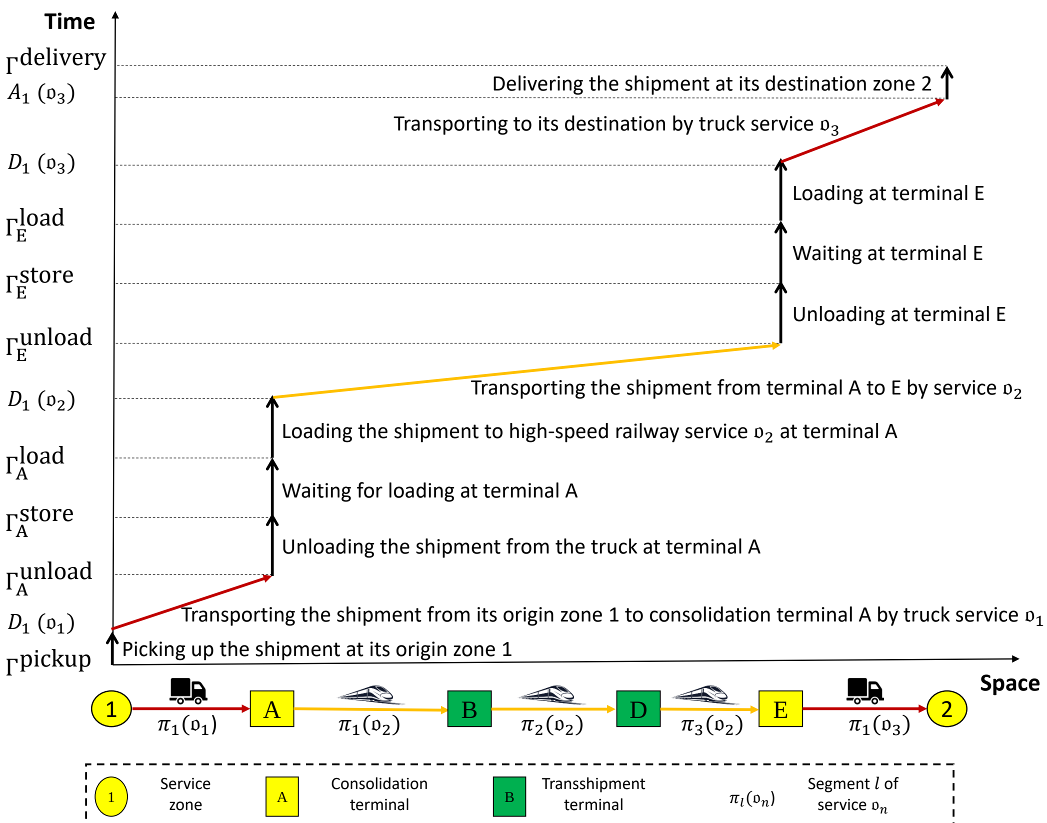


Figure 5: An illustrative example of a shipment itinerary in time and space.

The objective function at each decision time is to maximize the total profits over the planning horizon, including: the fare for accepted requests; the fixed costs for accepted offers; the transportation costs; the pickup costs at origin zones; the delivery costs at destination zones; the unloading and loading costs at consolidation and transshipment terminals; the storage costs at consolidation and transshipment terminals; the penalty costs for early delivery of shipments; the penalty costs for later delivery of shipments. The mixed integer linear programming model at time t is defined as follows:

$$\begin{aligned}
\max \quad & \sum_{\mathfrak{r}\in\tilde{\mathfrak{R}}^t} \rho(\mathfrak{r}) y^t_{\mathfrak{r}} - \sum_{\mathfrak{o}\in\tilde{\mathfrak{O}}^t} f(\mathfrak{o}) y^t_{\mathfrak{o}} - \sum_{\mathfrak{r}\in\mathfrak{R}^t}\sum_{\mathfrak{o}\in\mathfrak{O}^t}\sum_{l=1}^{|\Pi(\mathfrak{o})|} c_l(\mathfrak{o}) u(\mathfrak{r}) x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} \\
& - \sum_{\mathfrak{r}\in\mathfrak{R}^t}\sum_{\mathfrak{o}\in\mathfrak{O}^t}\sum_{l:o_l(\mathfrak{o})=o(\mathfrak{r})} c^{\mathtt{P}}_{o(\mathfrak{r})m_l(\mathfrak{o})} u(\mathfrak{r}) x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} - \sum_{\mathfrak{r}\in\mathfrak{R}^t}\sum_{\mathfrak{o}\in\mathfrak{O}^t}\sum_{l:d_l(\mathfrak{o})=d(\mathfrak{r})} c^{\mathtt{P}}_{d(\mathfrak{r})m_l(\mathfrak{o})} u(\mathfrak{r}) x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} \\
& - \sum_{\mathfrak{r}\in\mathfrak{R}^t}\sum_{i\in\Theta}\sum_{\mathfrak{o}\in\mathfrak{O}^t}\sum_{\mathfrak{o}'\in\mathfrak{O}^t}\sum_{l=1}^{|\Pi(\mathfrak{o})|}\sum_{l'=1}^{|\Pi(\mathfrak{o}')|} \left(c^{\mathtt{L}}_{im_l(\mathfrak{o})} + c^{\mathtt{L}}_{im_{l'}(\mathfrak{o}')}\right) u(\mathfrak{r}) z^t_{\mathfrak{r}i\pi_l(\mathfrak{o})\pi_{l'}(\mathfrak{o}')} \\
& - \sum_{\mathfrak{r}\in\mathfrak{R}^t}\sum_{i\in\Theta} c^{\mathtt{W}}_i u(\mathfrak{r}) w_{\mathfrak{r}i} - \sum_{\mathfrak{r}\in\mathfrak{R}^t} \psi^{\mathtt{E}}(\mathfrak{r}) \check{\Gamma}^{\text{delivery}}_{\mathfrak{r}} - \sum_{\mathfrak{r}\in\mathfrak{R}^t} \psi^{\mathtt{L}}(\mathfrak{r}) \hat{\Gamma}^{\text{delivery}}_{\mathfrak{r}}
\end{aligned} \tag{1}$$

where

$$\check{\Gamma}^{\text{delivery}}_{\mathfrak{r}} \geq \beta^{\mathtt{E}}(\mathfrak{r}) - \Gamma^{\text{delivery}}_{\mathfrak{r}}, \quad \forall \mathfrak{r} \in \mathfrak{R}^t, \tag{2}$$

$$\hat{\Gamma}^{\text{delivery}}_{\mathfrak{r}} \geq \Gamma^{\text{delivery}}_{\mathfrak{r}} - \alpha^{\mathtt{L}}(\mathfrak{r}), \quad \forall \mathfrak{r} \in \mathfrak{R}^t. \tag{3}$$

subject to

- Assignment constraints:

$$y^t_{\mathfrak{r}} \leq \sum_{\mathfrak{o}\in\mathfrak{O}^t, \pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}), o_l(\mathfrak{o})=o(\mathfrak{r})} x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} \leq 1, \quad \forall \mathfrak{r} \in \tilde{\mathfrak{R}}^t, \tag{4}$$

$$y^t_{\mathfrak{r}} \leq \sum_{\mathfrak{o}\in\mathfrak{O}^t, \pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}), d_l(\mathfrak{o})=d(\mathfrak{r})} x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} \leq 1, \quad \forall \mathfrak{r} \in \tilde{\mathfrak{R}}^t. \tag{5}$$

Constraints (4) ensure that new shipment $\mathfrak{r} \in \tilde{\mathfrak{R}}^t$ must be assigned to a service that will depart from its origin $o(\mathfrak{r})$ if request $\mathfrak{r}$ is accepted by the platform at time t. Constraints (5) ensure that shipment $\mathfrak{r} \in \tilde{\mathfrak{R}}^t$ must be assigned to a service that will arrive at its destination $o(\mathfrak{r})$ if request $\mathfrak{r}$ is accepted by the platform at time t.

$$\sum_{\mathfrak{o}\in\mathfrak{O}^t, \pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}), o_l(\mathfrak{o})=o(\mathfrak{r})} x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} = 1, \quad \forall \mathfrak{r} \in \dot{\mathfrak{R}}^t, \tag{6}$$

$$\sum_{\mathfrak{o}\in\mathfrak{O}^t, \pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}), d_l(\mathfrak{o})=d(\mathfrak{r})} x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} = 1, \quad \forall \mathfrak{r} \in \dot{\mathfrak{R}}^t. \tag{7}$$

Constraints (6-7) ensure that accepted shipment $\mathfrak{r} \in \dot{\mathfrak{R}}^t$ must be assigned to a service that will depart from its origin $o(\mathfrak{r})$ and a service that will arrive to its destination $d(\mathfrak{r})$.

$$\sum_{\mathfrak{o}\in\mathfrak{O}^t, \pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}), d_l(\mathfrak{o})=o(\mathfrak{r})} x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} = 0, \quad \forall \mathfrak{r} \in \mathfrak{R}^t, \tag{8}$$

$$\sum_{\mathfrak{o}\in\mathfrak{O}^t, \pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}), o_l(\mathfrak{o})=d(\mathfrak{r})} x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} = 0, \quad \forall \mathfrak{r} \in \mathfrak{R}^t. \tag{9}$$

Constraints (8) forbid a shipment enters its origin. Constraints (9) forbid a shipment leaves its destination.

$$\sum_{\mathfrak{o}\in\mathfrak{O}^t,\pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}),d_l(\mathfrak{o})=i} x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} = \sum_{\mathfrak{o}'\in\mathfrak{O}^t,\pi_{l'}(\mathfrak{o}')\in\Pi(\mathfrak{o}'),o_{l'}(\mathfrak{o}')=i} x^t_{\mathfrak{r}\pi_{l'}(\mathfrak{o}')},\ \forall\mathfrak{r}\in\mathfrak{R}^t, i\in\Theta. \tag{10}$$

Constraints (10) ensure flow conservation of shipments at consolidation and transshipment terminals.

$$z^t_{\mathfrak{r}i\pi_l(\mathfrak{o})\pi_{l'}(\mathfrak{o}')} \geq x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} + x^t_{\mathfrak{r}\pi_{l'}(\mathfrak{o}')} - 1,\ \forall\mathfrak{r}\in\mathfrak{R}^t, i\in\Theta, \mathfrak{o},\mathfrak{o}'\in\mathfrak{O}^t, \mathfrak{o}\neq\mathfrak{o}', \pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}), \pi_{l'}(\mathfrak{o}')\in\Pi(\mathfrak{o}'), d_l(\mathfrak{o})=o_{l'}(\mathfrak{o}')=i. \tag{11}$$

$$z^t_{\mathfrak{r}i\pi_l(\mathfrak{o})\pi_{l'}(\mathfrak{o}')} \leq x^t_{\mathfrak{r}\pi_l(\mathfrak{o})},\quad \forall\mathfrak{r}\in\mathfrak{R}^t, i\in\Theta, \mathfrak{o},\mathfrak{o}'\in\mathfrak{O}^t, \mathfrak{o}\neq\mathfrak{o}', \pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}), \pi_{l'}(\mathfrak{o}')\in\Pi(\mathfrak{o}'), d_l(\mathfrak{o})=o_{l'}(\mathfrak{o}')=i. \tag{12}$$

$$z^t_{\mathfrak{r}i\pi_l(\mathfrak{o})\pi_{l'}(\mathfrak{o}')} \leq x^t_{\mathfrak{r}\pi_{l'}(\mathfrak{o}')},\quad \forall\mathfrak{r}\in\mathfrak{R}^t, i\in\Theta, \mathfrak{o},\mathfrak{o}'\in\mathfrak{O}^t, \mathfrak{o}\neq\mathfrak{o}', \pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}), \pi_{l'}(\mathfrak{o}')\in\Pi(\mathfrak{o}'), d_l(\mathfrak{o})=o_{l'}(\mathfrak{o}')=i. \tag{13}$$

Constraints (11-13) ensure shipment $\mathfrak{r}$ will be transshipped at terminal $i\in\Theta$ between segment $\pi_l(\mathfrak{o})$ and segment $\pi_{l'}(\mathfrak{o}')$ if $x^t_{\mathfrak{r}\pi_l(\mathfrak{o})}=1$ and $x^t_{\mathfrak{r}\pi_{l'}(\mathfrak{o}')}=1$.

- Time constraints:

$$\alpha^{\mathtt{R}}(\mathfrak{r}) \leq \Gamma^{\text{pickup}}_{\mathfrak{r}} \leq \beta^{\mathtt{R}}(\mathfrak{r}),\quad \forall\mathfrak{r}\in\mathfrak{R}^t. \tag{14}$$

Constraints (14) ensure that the pickup time of shipments at their origins must be within their pickup time windows.

$$\alpha^{\mathtt{R}}_l(\mathfrak{o}) \leq D_l(\mathfrak{o}) \leq \beta^{\mathtt{R}}_l(\mathfrak{o}),\quad \forall\mathfrak{o}\in\mathfrak{O}^t, l\in\{1,\ldots,|\Pi(\mathfrak{o})|\}, \tag{15}$$

$$A_l(\mathfrak{o}) = D_l(\mathfrak{o}) + \tau_1(\mathfrak{o}),\quad \forall\mathfrak{o}\in\mathfrak{O}^t, l\in\{1,\ldots,|\Pi(\mathfrak{o})|\}. \tag{16}$$

Constraints (15) ensure that the departure time of time-flexible services at its origin must be within its departure time window. Constraints (16) calculate the arrival time of segment l of service $\mathfrak{o}\in\mathfrak{O}^t$ at its destination node $d_l(\mathfrak{o})$.

$$\Gamma^{\text{pickup}}_{\mathfrak{r}} \leq D_l(\mathfrak{o}) - t^{\mathtt{P}}_{o_{\mathfrak{r}}m_l(\mathfrak{o})} + B(1 - x^t_{\mathfrak{r}\pi_l(\mathfrak{o})}),\quad \forall\mathfrak{r}\in\mathfrak{R}^t, \mathfrak{o}\in\mathfrak{O}^t, \pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}), o_l(\mathfrak{o})=o(\mathfrak{r}), \tag{17}$$

$$\Gamma^{\text{pickup}}_{\mathfrak{r}} \geq D_l(\mathfrak{o}) - t^{\mathtt{P}}_{o_{\mathfrak{r}}m_l(\mathfrak{o})} + B(x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} - 1),\quad \forall\mathfrak{r}\in\mathfrak{R}^t, \mathfrak{o}\in\mathfrak{O}^t, \pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}), o_l(\mathfrak{o})=o(\mathfrak{r}), \tag{18}$$

Let B be a large enough value. Constraints (17-18) ensure that the pickup time of shipments at their origins equals to the departure time of the assigned service segment minus pickup time.

$$\Gamma^{\text{unload}}_{\mathfrak{r}i} \leq A_l(\mathfrak{o}) + B(1 - x^t_{\mathfrak{r}\pi_l(\mathfrak{o})}),\quad \forall\mathfrak{r}\in\mathfrak{R}^t, i\in\Theta, \mathfrak{o}\in\mathfrak{O}^t, \pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}), d_l(\mathfrak{o})=i, \tag{19}$$

$$\Gamma^{\text{unload}}_{\mathfrak{r}i} \geq A_l(\mathfrak{o}) + B(x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} - 1),\quad \forall\mathfrak{r}\in\mathfrak{R}^t, i\in\Theta, \mathfrak{o}\in\mathfrak{O}^t, \pi_l(\mathfrak{o})\in\Pi(\mathfrak{o}), d_l(\mathfrak{o})=i. \tag{20}$$

Constraints (19-20) ensure that the time that shipment $\mathfrak{r}$ will be started unloading at terminal i equals to the arrival time of the assigned service segment.

$$\Gamma_{\mathfrak{r}i}^{\text{store}} \leq A_l(\mathfrak{o}) + t_{im_l(\mathfrak{o})}^{\text{L}} + B(1 - x_{\mathfrak{r}\pi_l(\mathfrak{o})}^t), \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, \mathfrak{o} \in \mathfrak{O}^t, \pi_l(\mathfrak{o}) \in \Pi(\mathfrak{o}), d_l(\mathfrak{o}) = i, \tag{21}$$

$$\Gamma_{\mathfrak{r}i}^{\text{store}} \geq A_l(\mathfrak{o}) + t_{im_l(\mathfrak{o})}^{\text{L}} + B(x_{\mathfrak{r}\pi_l(\mathfrak{o})}^t - 1), \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, \mathfrak{o} \in \mathfrak{O}^t, \pi_l(\mathfrak{o}) \in \Pi(\mathfrak{o}), d_l(\mathfrak{o}) = i. \tag{22}$$

Constraints (21-22) ensure that the time that shipment $\mathfrak{r}$ will be started storage at terminal i equals to the arrival time of the assigned service segment plus unloading time.

$$\Gamma_{\mathfrak{r}i}^{\text{load}} \leq D_l(\mathfrak{o}) - t_{im_l(\mathfrak{o})}^{\text{L}} + B(1 - x_{\mathfrak{r}\pi_l(\mathfrak{o})}^t), \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, \mathfrak{o} \in \mathfrak{O}^t, \pi_l(\mathfrak{o}) \in \Pi(\mathfrak{o}), o_l(\mathfrak{o}) = i, \tag{23}$$

$$\Gamma_{\mathfrak{r}i}^{\text{load}} \geq D_l(\mathfrak{o}^{'}) - t_{im_l(\mathfrak{o})}^{\text{L}} + B(x_{\mathfrak{r}\pi_l(\mathfrak{o})}^t - 1), \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, \mathfrak{o} \in \mathfrak{O}^t, \pi_l(\mathfrak{o}) \in \Pi(\mathfrak{o}), o_l(\mathfrak{o}) = i. \tag{24}$$

Constraints (23-24) ensure that the time that shipment $\mathfrak{r}$ will be started loading at terminal i equals to the departure time of the assigned service segment minus loading time.

$$\Gamma_{\mathfrak{r}i}^{\text{depart}} \leq D_l(\mathfrak{o}) + B(1 - x_{\mathfrak{r}\pi_l(\mathfrak{o})}^t), \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, \mathfrak{o} \in \mathfrak{O}^t, \pi_l(\mathfrak{o}) \in \Pi(\mathfrak{o}), o_l(\mathfrak{o}) = i, \tag{25}$$

$$\Gamma_{\mathfrak{r}i}^{\text{depart}} \geq D_l(\mathfrak{o}) + B(x_{\mathfrak{r}\pi_l(\mathfrak{o})}^t - 1), \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, \mathfrak{o} \in \mathfrak{O}^t, \pi_l(\mathfrak{o}) \in \Pi(\mathfrak{o}), o_l(\mathfrak{o}) = i. \tag{26}$$

Constraints (25-26) ensure that the time that shipment $\mathfrak{r}$ will be departed at terminal i equals to the departure time of the assigned service segment.

$$w_{\mathfrak{r}i} = \Gamma_{\mathfrak{r}i}^{\text{load}} - \Gamma_{\mathfrak{r}i}^{\text{store}} \geq 0, \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta. \tag{27}$$

Constraints (27) calculate the storage time of shipment $\mathfrak{r}$ at terminal i.

$$\Gamma_{\mathfrak{r}}^{\text{delivery}} \leq A_l(\mathfrak{o}) + t_{d_{\mathfrak{r}}m_l(\mathfrak{o})}^{\text{P}} + B(1 - x_{\mathfrak{r}\pi_l(\mathfrak{o})}^t), \quad \forall \mathfrak{r} \in \mathfrak{R}^t, \mathfrak{o} \in \mathfrak{O}^t, \pi_l(\mathfrak{o}) \in \Pi(\mathfrak{o}), d_l(\mathfrak{o}) = d(\mathfrak{r}), \tag{28}$$

$$\Gamma_{\mathfrak{r}}^{\text{delivery}} \geq A_l(\mathfrak{o}) + t_{d_{\mathfrak{r}}m_l(\mathfrak{o})}^{\text{P}} + B(x_{\mathfrak{r}\pi_l(\mathfrak{o})}^t - 1), \quad \forall \mathfrak{r} \in \mathfrak{R}^t, \mathfrak{o} \in \mathfrak{O}^t, \pi_l(\mathfrak{o}) \in \Pi(\mathfrak{o}), d_l(\mathfrak{o}) = d(\mathfrak{r}). \tag{29}$$

Constraints (28-29) ensure that the delivery time of shipments at their destinations equals to the arrival time of the assigned service segment plus delivery time.

$$\alpha^{\text{E}}(\mathfrak{r}) \leq \Gamma_{\mathfrak{r}}^{\text{delivery}} \leq \beta^{\text{L}}(\mathfrak{r}), \quad \forall \mathfrak{r} \in \mathfrak{R}^t. \tag{30}$$

Constraints (30) ensure that the delivery time of shipments must be within their delivery time windows.

- Capacity limitations:

$$\sum_{\mathfrak{r}\in\mathfrak{R}^t} u(\mathfrak{r}) x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} \leq y^t_{\mathfrak{o}} u^t_l(\mathfrak{o}), \quad \forall \mathfrak{o} \in \tilde{\mathfrak{O}}^t, l \in \{1,\ldots,|\Pi(\mathfrak{o})|\}, \tag{31}$$

$$\sum_{\mathfrak{r}\in\mathfrak{R}^t} u(\mathfrak{r}) x^h_{\mathfrak{r}\pi_l(\mathfrak{o})} \leq u^t_l(\mathfrak{o}), \quad \forall \mathfrak{o} \in \dot{\mathfrak{O}}^t, l \in \{1,\ldots,|\Pi(\mathfrak{o})|\}. \tag{32}$$

Constraints (31) ensure that the total volumes of shipments assigned to the l segment of new service offer $\mathfrak{o} \in \tilde{\mathfrak{O}}^t$ cannot exceed its free capacity at time t if offer $\mathfrak{o}$ is accepted by the platform. Constraints (32) ensure that the total volumes of shipments assigned to the l segment of accepted service $\mathfrak{o} \in \dot{\mathfrak{O}}^t$ cannot exceed its free capacity at time t.

$$\sum_{\mathfrak{r}\in\mathfrak{R}^t} u(\mathfrak{r}) \left(g^{\text{unload},p}_{\mathfrak{r}i} + g^{\text{load},p}_{\mathfrak{r}i} \right) \leq u^{\text{L},t,p}_i, \quad \forall i \in \Theta, p \in \{t+1,...,t+T\}. \tag{33}$$

Constraints (33) ensure that the total volumes of shipments assigned to terminal i during time period p for loading and unloading operations cannot exceed its available capacity during that period at decision time t.

$$\theta 1^{\text{unload},p}_{\mathfrak{r}i}, \theta 2^{\text{unload},p}_{\mathfrak{r}i} \in \{0,1\},\ \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{34}$$

$$p - \Gamma^{\text{unload}}_{\mathfrak{r}i} \leq B\theta 1^{\text{unload},p}_{\mathfrak{r}i}, \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{35}$$

$$\Gamma^{\text{unload}}_{\mathfrak{r}i} - p + 1 \leq B(1 - \theta 1^{\text{unload},p}_{\mathfrak{r}i}), \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{36}$$

$$\Gamma^{\text{store}}_{\mathfrak{r}i} + 1 - p \leq B\theta 2^{\text{unload},p}_{\mathfrak{r}i},\ \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{37}$$

$$p - \Gamma^{\text{store}}_{\mathfrak{r}i} \leq B(1 - \theta 2^{\text{unload},p}_{\mathfrak{r}i}), \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{38}$$

$$g^{\text{unload},p}_{\mathfrak{r}i} \geq \theta 1^{\text{unload},p}_{\mathfrak{r}i} + \theta 2^{\text{unload},p}_{\mathfrak{r}i} - 1, \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{39}$$

$$g^{\text{unload},p}_{\mathfrak{r}i} \leq \theta 1^{\text{unload},p}_{\mathfrak{r}i}, \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{40}$$

$$g^{\text{unload},p}_{\mathfrak{r}i} \leq \theta 2^{\text{unload},p}_{\mathfrak{r}i}, \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}. \tag{41}$$

Constraints (34-41) ensure that binary variable $g^{\text{unload},p}_{\mathfrak{r}i} = 1$ if $\Gamma^{\text{unload}}_{\mathfrak{r}i} + 1 \leq p \leq \Gamma^{\text{store}}_{\mathfrak{r}i}$, which indicates the time periods that shipment $\mathfrak{r}$ will be unloaded at terminal i.

$$\theta 1^{\text{load},p}_{\mathfrak{r}i}, \theta 2^{\text{load},p}_{\mathfrak{r}i} \in \{0,1\}, \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{42}$$

$$p - \Gamma^{\text{load}}_{\mathfrak{r}i} \leq B\theta 1^{\text{load},p}_{\mathfrak{r}i}, \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{43}$$

$$\Gamma^{\text{load}}_{\mathfrak{r}i} - p + 1 \leq B(1 - \theta 1^{\text{load},p}_{\mathfrak{r}i}), \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{44}$$

$$\Gamma^{\text{depart}}_{\mathfrak{r}i} + 1 - p \leq B\theta 2^{\text{load},p}_{\mathfrak{r}i}, \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{45}$$

$$p - \Gamma^{\text{depart}}_{\mathfrak{r}i} \leq B(1 - \theta 2^{\text{load},p}_{\mathfrak{r}i}), \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{46}$$

$$g^{\text{load},p}_{\mathfrak{r}i} \geq \theta 1^{\text{load},p}_{\mathfrak{r}i} + \theta 2^{\text{load},p}_{\mathfrak{r}i} - 1 \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{47}$$

$$g^{\text{load},p}_{\mathfrak{r}i} \leq \theta 1^{\text{load},p}_{\mathfrak{r}i}, \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{48}$$

$$g^{\text{load},p}_{\mathfrak{r}i} \leq \theta 2^{\text{load},p}_{\mathfrak{r}i}, \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}. \tag{49}$$

Constraints (42-49) ensure that binary variable $g^{\text{load},p}_{\mathfrak{r}i} = 1$ if $\Gamma^{\text{load}}_{\mathfrak{r}i} + 1 \leq p \leq \Gamma^{\text{depart}}_{\mathfrak{r}i}$, which indicates the time periods that shipment $\mathfrak{r}$ will be loaded at terminal i.

$$\sum_{\mathfrak{r}\in\mathfrak{R}^t} u(\mathfrak{r}) g^{\text{store},p}_{\mathfrak{r}i} \leq u^{\text{W},t,p}_i, \quad \forall i \in \Theta, p \in \{t+1,...,t+T\}. \tag{50}$$

Constraints (50) ensure that the total volumes of shipments assigned to terminal i during time period p for storage cannot exceed its available capacity during that period at decision time t.

$$\theta 1_{\mathfrak{r}i}^{\text{store},p}, \theta 2_{\mathfrak{r}i}^{\text{store},p} \in \{0,1\}, \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{51}$$

$$p - \Gamma_{\mathfrak{r}i}^{\text{store}} \leq B\theta 1_{\mathfrak{r}i}^{\text{store},p}, \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{52}$$

$$\Gamma_{\mathfrak{r}i}^{\text{store}} - p + 1 \leq B(1 - \theta 1_{\mathfrak{r}i}^{\text{store},p}), \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{53}$$

$$\Gamma_{\mathfrak{r}i}^{\text{load}} + 1 - p \leq B\theta 2_{\mathfrak{r}i}^{\text{store},p}, \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{54}$$

$$p - \Gamma_{\mathfrak{r}i}^{\text{load}} \leq B(1 - \theta 2_{\mathfrak{r}i}^{\text{store},p}), \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{55}$$

$$g_{\mathfrak{r}i}^{\text{store},p} \geq \theta 1_{\mathfrak{r}i}^{\text{store},p} + \theta 2_{\mathfrak{r}i}^{\text{store},p} - 1, \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{56}$$

$$g_{\mathfrak{r}i}^{\text{store},p} \leq \theta 1_{\mathfrak{r}i}^{\text{store},p}, \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}, \tag{57}$$

$$g_{\mathfrak{r}i}^{\text{store},p} \leq \theta 2_{\mathfrak{r}i}^{\text{store},p}, \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, p \in \{t+1,...,t+T\}. \tag{58}$$

Constraints (51-58) ensure that binary variable $g_{\mathfrak{r}i}^{\text{store},p} = 1$ if $\Gamma_{\mathfrak{r}i}^{\text{store}} + 1 \leq p \leq \Gamma_{\mathfrak{r}i}^{\text{load}}$, which indicates the time periods that shipment $\mathfrak{r}$ will be stored at terminal i.

- Decision domain:

$$y_{\mathfrak{r}}^t, y_{\mathfrak{o}}^t \in \{0,1\}, \quad \forall \mathfrak{r} \in \tilde{\mathfrak{R}}^t, \mathfrak{o} \in \tilde{\mathfrak{O}}^t, \tag{59}$$

$$\begin{aligned} x_{\mathfrak{r}\pi_l(\mathfrak{o})}^t, z_{\mathfrak{r}i\pi_l(\mathfrak{o})\pi_{l'}(\mathfrak{o}')}^t, g_{\mathfrak{r}i}^{\text{unload},p}, g_{\mathfrak{r}i}^{\text{load},p}, g_{\mathfrak{r}i}^{\text{store},p} \in \{0,1\}, \quad \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta, \mathfrak{o}, \mathfrak{o}' \in \mathfrak{O}^t, \\ \pi_l(\mathfrak{o}) \in \Pi(\mathfrak{o}), \pi_{l'}(\mathfrak{o}') \in \Pi(\mathfrak{o}'), p \in \{t+1,...,t+T\}, \end{aligned} \tag{60}$$

$$D_l(\mathfrak{o}), A_l(\mathfrak{o}) \geq 0, \quad \forall \mathfrak{o} \in \mathfrak{O}^t, l \in \{1,...,|\Pi(\mathfrak{o})|\}, \tag{61}$$

$$\Gamma_{\mathfrak{r}}^{\text{pickup}}, \Gamma_{\mathfrak{r}i}^{\text{unload}}, \Gamma_{\mathfrak{r}i}^{\text{store}}, \Gamma_{\mathfrak{r}i}^{\text{load}}, \Gamma_{\mathfrak{r}i}^{\text{depart}}, \Gamma_{\mathfrak{r}}^{\text{delivery}}, w_{\mathfrak{r}i} \geq 0, \forall \mathfrak{r} \in \mathfrak{R}^t, i \in \Theta. \tag{62}$$

5 Heuristic algorithm

Due to the computational complexity of the MILP model proposed in Section 4, this section proposes a preprocessing-based adaptive large neighborhood search algorithm (P-ALNS) to generate timely but also high quality solutions. The algorithm consists of three steps: the preprocessing of feasible itineraries, the generation of an initial solution, and the improvements by ALNS. Considering that all the offers either with flexible or scheduled services have fixed routes, this paper proposes to generate all the feasible itineraries for all shipment requests before the remove and repair process. In this way, the computation time for constructing feasible itineraries during the repair process can be largely reserved. In the following, the details and pseudocodes are provided.

5.1 Preprocessing of feasible itineraries

The preprocessing of feasible itineraries consists of two steps:

- **The generation of feasible paths.** In this step, the feasible paths consist of service segments from multiple offers between all the nodes are generated. A path is feasible, if it satisfies time and spatial constraints among service segments.
- **The generation of feasible itineraries.** In this step, the feasible itineraries for all the shipments are generated. An itinerary is feasible if the path and the shipment have time and spatial compatibility. We use approximated total cost, including transit cost, transshipment cost, and estimated penalty cost for early and late delivery, to represent the cost of an itinerary.

The pseudocode for preprocessing of feasible itineraries is presented in Algorithm 1.

Algorithm 1 Feasible itinerary generation.

Input: zones $\mathcal{Z}$, terminals Θ, requests $\mathfrak{R}^t$, service offers $\mathfrak{O}^t$, the largest number of service segments in an itinerary $N^{\max}$.
Output: feasible paths $\{P^l_{ij}\}_{i\in\Theta,j\in\Theta,l\in\{1,\dots,N^{\max}\}}$, feasible itineraries $\{I_\mathfrak{r}\}_{\forall\mathfrak{r}\in\mathfrak{R}^t}$, estimated costs $[c_{\mathfrak{r}p}]_{\forall\mathfrak{r}\in\mathfrak{R},p\in I_r}$
Initialize: Let $P \leftarrow \emptyset, I \leftarrow \emptyset, e \leftarrow 2$.

generate feasible paths:
for node $i \in \mathcal{Z}\cup\Theta$, node $j \in \mathcal{Z}\cup\Theta$ **do**
  for service offer $\mathfrak{o} \in \mathfrak{O}^t$ **do**
    for service segment $l \in \{1,\dots,|\Pi(\mathfrak{o})|\}$ **do**
      if origin $o_l(\mathfrak{o}) = i$ and destination $d_l(\mathfrak{o}) = j$ **then**
        feasible path $p \leftarrow [\pi_l(\mathfrak{o})]$
while $e \le N^{\max}$ **do**
  for node $i \in \mathcal{Z}\cup\Theta$, node $j \in \mathcal{Z}\cup\Theta$ **do**
    for service offer $\mathfrak{o} \in \mathfrak{O}^t$ **do**
      for service segment $l \in \{1,\dots,|\Pi(\mathfrak{o})|\}$ **do**
        if origin $o_l(\mathfrak{o}) \ne i$ and destination $d_l(\mathfrak{o}) = j$ **then**
          for feasible path $p \in P^{e-1}_{io_l(\mathfrak{o})}$ **do**
            if earliest arrival time of path $p \le$ latest departure time of service segment $\pi_l(\mathfrak{o})$ **then**
              $P^e_{ij} \leftarrow P^e_{ij} \cup \{[p, \pi_l(\mathfrak{o})]\}$
  $e \leftarrow e+1$
generate feasible itineraries:
for request $\mathfrak{r} \in \mathfrak{R}^t$ **do**
  for $e \in \{1,2,...,N^{\max}\}$ **do**
    for feasible path $p = [\pi_{l_1}(\mathfrak{o}_1),...,\pi_{l_e}(\mathfrak{o}_e)] \in P^e_{o_r d_r}$ **do**
      if departure time window of path p has overlap with pickup time window of request $\mathfrak{r}$ **then**
        $\tilde{c}_{\mathfrak{r}p} \leftarrow$ Transit cost + Transshipment cost+ estimated penalty cost for early and late delivery
        if $\tilde{c}_{\mathfrak{r}p} \le \rho(\mathfrak{r})$ **then**
          $I_\mathfrak{r} \leftarrow I_\mathfrak{r} \cup \{p\}$
  sort $I_\mathfrak{r}$ in descending order based on $\tilde{c}_{\mathfrak{r}p}$

5.2 Initial solution

To generate an initial solution, we use the best insertion idea in which requests are inserted sequentially based on estimated profits. The pseudocode for generating initial solution is presented in Algorithm 2. For each request, if its best itinerary is infeasible in time and capacity constraints, then the second best is checked, until find a feasible itinerary. The pseudocode for feasible schedules and capacity check is presented in Section 5.3.3. For new request $\mathfrak{r} \in \tilde{\mathfrak{R}}^t$, if a feasible itinerary exists, then the request will be accepted; for new offer $\mathfrak{o} \in \tilde{\mathfrak{O}}^t$, if any of its service segments are assigned to shipments, then the offer will be accepted.

Algorithm 2 Initial solution generation.

Input: requests $\mathfrak{R}^t$, service offers $\mathfrak{O}^t$, feasible itineraries $\{I_\mathfrak{r}\}_{\forall \mathfrak{r} \in \mathfrak{R}^t}$, estimated costs $[c_{\mathfrak{r}p}]_{\forall \mathfrak{r} \in \mathfrak{R}, p \in I_\mathfrak{r}}$.
Output: initial itineraries I^{initial}, initial service schedules $[D^{\text{initial}}, A^{\text{initial}}]$, initial shipment schedules Γ^{initial}.
Initialize: let $List \leftarrow \emptyset$.

for $\mathfrak{r} \in \mathfrak{R}^t$ **do**
  estimated profit when best itinerary is selected $\leftarrow \rho(\mathfrak{r}) - \tilde{c}_{\mathfrak{r}p_1}$
$List \leftarrow$ sort requests in descending order based on estimated profits
for $\mathfrak{r} \in List$ **do**
  $index = 1$
  while $index \leq$ the length of feasible itineraries $I_\mathfrak{r}$ **do**
    itinerary $p = I_{\mathfrak{r},index}$
    feasibility$(\mathfrak{r}, p) \leftarrow$ *FEASIBLESCHEDULESANDCAPACITY*
    if feasibility$(\mathfrak{r}, p) = 1$ **then**
      $I_\mathfrak{r}^{\text{initial}} \leftarrow p$
      for service segment $\pi_l(\mathfrak{o}) \in p$ **do**
        service schedules $[D_l^{\text{initial}}(\mathfrak{o}), A_l^{\text{initial}}(\mathfrak{o})] \leftarrow FEASIBLESCHEDULESANDCAPACITY(\mathfrak{r}, p)$
      $\Gamma_\mathfrak{r}^{\text{initial}} \leftarrow FEASIBLESCHEDULESANDCAPACITY(\mathfrak{r}, p)$
    else
      $index = index + 1$
for $\mathfrak{r} \in \tilde{\mathfrak{R}}^t$ **do**
  if $I_\mathfrak{r}^{\text{initial}} \neq \emptyset$ **then**
    $y_\mathfrak{r}^t = 1$
for $\mathfrak{r} \in \mathfrak{R}^t$ **do**
  if $I_\mathfrak{r}^{\text{initial}} \neq \emptyset$ **then**
    for $\pi_l(\mathfrak{o}) \in I_\mathfrak{r}^{\text{initial}}$ **do**
      if $\mathfrak{o} \in \tilde{\mathfrak{O}}^t$ **then**
        $y_\mathfrak{o}^t = 1$

5.3 Adaptive large neighborhood search

In this section, we propose an adaptive large neighborhood search algorithm (ALNS) to improve the solution quality. The ALNS applies several removal and insertion operators to a given solution, as shown in Algorithm 3. The basic idea of ALNS is to search for a better solution at each iteration by removing some requests from the solution and inserting them in a different way. The removal and insertion operators are selected dynamically according to the performance achieved during the search. A weight is associated with each operator and the selection probability of an operator is related to its weight, which is adjusted during the search based on its past successes. The main components of the ALNS are presented in the following subsections.

Algorithm 3 Adaptive large neighborhood search algorithm.

Input: initial solution $\mathbf{x}^{\text{initial}} = [I^{\text{initial}}, D^{\text{initial}}, A^{\text{initial}}, \Gamma^{\text{initial}}]$
Output: best solution $[\mathbf{x}^{\text{best}}]$

```
x^current ← x^initial, x^best ← x^initial
while stopping criterion not met do
    select a remove operator and a repair operator based on roulette-wheel mechanism
    x ← x^current;
    x ← Remove(x);
    x ← Insert(x);
    if total profit(x) > total profit(x^current) then
        x^current ← x
    else
        x^current ← x with probability p = e^((totalprofit(x) − totalprofit(x^current)) / T^temp)
    if total profit(x) > total profit(x^best) then
        x^best ← x
    T^temp ← T^temp.c (c is the cooling rate)
```

5.3.1 Removal operators

The removal stage aims to remove n requests from the current solution and adding them to the removal list $\mathfrak{L}$. For requests in the removal list, their itineraries and schedules are reset to empty. Service segments and terminals associated with these requests are released. At each iteration, n is randomly selected from an interval $[\alpha * |\mathfrak{R}^t|, \beta * |\mathfrak{R}^t|]$.

- *Random removal.* Randomly removes n requests from the current solution, their itineraries and schedules are reset to empty.
- *Worst removal.* Remove n requests with the lowest profits.
- *Related removal.* Randomly select a request to remove, then remove the $n-1$ requests according to distance between requests' origins and destinations, the difference between pickup time windows and delivery time windows, and the difference in volume of the two requests. Note that each component needs to be normalized by dividing the largest value of all requests.

$$\begin{aligned} Relate(I_{\mathfrak{r}_1}, I_{\mathfrak{r}_2}) = \theta_1 \left(dis_{o(\mathfrak{r}_1),o(\mathfrak{r}_2)} + dis_{d(\mathfrak{r}_1),d(\mathfrak{r}_2)}\right) + \theta_2 \left(|\alpha^{\text{R}}(\mathfrak{r}_1) - \alpha^{\text{R}}(\mathfrak{r}_2)| \right. \\ \left. + |\beta^{\text{L}}(\mathfrak{r}_1) - \beta^{\text{L}}(\mathfrak{r}_2)|\right) + \theta_3 \left(|u(\mathfrak{r}_1) - u(\mathfrak{r}_2)|\right) \end{aligned} \tag{63}$$

5.3.2 Insertion operators

All insertion operators iteratively reinsert the removed requests into the solution. They stop when all requests are inserted. Here, we generate the insertion list by randomly sort the removal list.

- *Random insertion.* For each request in the insertion list, a randomly selected itinerary will be inserted.
- *Best insertion.* For each request in the insertion list, inserting the best itinerary, if its infeasible, then insert the second best, until find a feasible insertion or all the itineraries are checked.
- *Regret-2 insertion.* Insert requests based on regret values. Let $\Delta f_{\mathfrak{r}}^2$ be the insertion cost of request $\mathfrak{r} \in \mathfrak{L}$ with the second best itinerary. At each iteration, the operator select the request $\mathfrak{r}^*$ for insertion with the best itinerary such that $\mathfrak{r}^* = argmax_{\mathfrak{r}\in\mathfrak{L}}\,(\Delta f_{\mathfrak{r}}^2 - \Delta f_{\mathfrak{r}}^1)$.
- *Most constrained insertion.* The idea is to insert the request that is most difficult to insert according to distance, time windows, and volume. Note that each component needs to be normalized by dividing the largest value of all requests.

$$Constrain(\mathfrak{r}) = \gamma_1 dis_{o(\mathfrak{r}_1),d(\mathfrak{r}_1)} + \gamma_2\left(\beta^{\mathrm{L}}(\mathfrak{r}_1) - \alpha^{\mathrm{R}}(\mathfrak{r}_1)\right) + \gamma_3 u(\mathfrak{r}_1) \tag{64}$$

5.3.3 Feasibility check and service scheduling

An itinerary is feasible for shipment $\mathfrak{r}$ only if it satisfies time and capacity compatibility:

- *Time compatibility.* The pickup time of shipment $\mathfrak{r}$ at its origin should be within its pickup time window $[\alpha^{\mathtt{R}}(\mathfrak{r}), \beta^{\mathtt{R}}(\mathfrak{r})]$; the delivery time of shipment $\mathfrak{r}$ at its destination should be within its delivery time window $[\alpha^{\mathtt{E}}(\mathfrak{r}), \beta^{\mathtt{L}}(\mathfrak{r})]$; if shipment $\mathfrak{r}$ is transferred at terminal $i \in \Theta(\mathfrak{r})$ between service segment $\pi_l(\mathfrak{o})$ and $\pi_{l'}(\mathfrak{o}')$, the arrival time of service segment $\pi_l(\mathfrak{o})$ plus unloading and loading time should be the earlier than the departure time of service segment $\pi_{l'}(\mathfrak{o}')$. For time-flexible services $\{\mathfrak{o}|\pi_l(\mathfrak{o}) \in I_{\mathfrak{r}}, \text{type}(\mathfrak{o}) = 0\}$, the departure time of service segment $\pi_l(\mathfrak{o})$ must be with its departure time window, and we select the earliest departure times that satisfy shipments' time windows, and changes service segments type to $\text{type}(\pi_l(\mathfrak{o})) = 1$.
- *Capacity compatibility.* For service segment $\pi_l(\mathfrak{o}) \in I_{\mathfrak{r}}$, the volume of request $\mathfrak{r}$ cannot exceed the available capacity of service segment $\pi_l(\mathfrak{o})$; if shipment $\mathfrak{r}$ is transferred at terminal $i \in \Theta(\mathfrak{r})$ during time period p, the volume of request $\mathfrak{r}$ cannot exceed the transshipment capacity within that period; if shipment $\mathfrak{r}$ is stored at terminal $i \in \Theta(\mathfrak{r})$ during time period p, the volume of request $\mathfrak{r}$ cannot exceed the storage capacity within that period.

The pseudocode for feasibility check in time and capacities is presented in Algorithm 4.

Algorithm 4 FEASILESCHEDULESANDCAPACITY.

Input: request $\mathfrak{r}$, itinerary p, terminal Θ.
Output: feasibility$(\mathfrak{r}, p)$, service schedules $[D_l(\mathfrak{o}), A_l(\mathfrak{o})]_{\forall \pi_l(\mathfrak{o}) \in p}$, shipment schedules $\Gamma_{\mathfrak{r}}$.
Initialize: let feasibility$(\mathfrak{r}, p) \leftarrow 1$.

for $\pi_l(\mathfrak{o}) \in p$ **do**
  if $u(\mathfrak{r}) > u_l^t(\mathfrak{o})$ **then**
    feasibility$(\mathfrak{r}, p) \leftarrow 0$
  else
    $u_l^t(\mathfrak{o}) \leftarrow u_l^t(\mathfrak{o}) - u(\mathfrak{r})$
$\pi_{l_1}(\mathfrak{o}_1) \leftarrow$ the first service segment in itinerary p
if the departure time window of $\pi_{l_1}(\mathfrak{o}_1)$ has overlap with the pickup time window of request $\mathfrak{r}$ **then**
  if $\pi_{l_1}(\mathfrak{o}_1)$is time-flexible: type$(\pi_{l_1}(\mathfrak{o}_1)) = 0$ **then**
    select the earliest feasible time to depart: $D_{l_1}(\mathfrak{o}_1) \leftarrow \min\{\alpha^{\mathrm{R}}(\mathfrak{r}), \alpha_{l_1}^{\mathrm{R}}(\mathfrak{o}_1)\}$
    update service segment's arrival time: $A_{l_1}(\mathfrak{o}_1) \leftarrow D_{l_1}(\mathfrak{o}_1) + \tau_{l_1}(\mathfrak{o}_1)$
    update service segment's type: type$(\pi_{l_1}(\mathfrak{o}_1)) \leftarrow 1$
  else
    feasibility$(\mathfrak{r}, p) \leftarrow 0$
for $\pi_{l_i}(\mathfrak{o}_i) \in p : i > 1$ **do**
  if the earliest arrival time of $\pi_{l_{i-1}}(\mathfrak{o}_{i-1}) \leq$ the latest departure time of $\pi_{l_{i-1}}(\mathfrak{o}_{i-1})$ **then**
    if $\pi_{l_{i-1}}(\mathfrak{o}_{i-1})$is time-flexible: type$(\pi_{l_{i-1}}(\mathfrak{o}_{i-1})) = 0$ **then**
      select the earliest feasible time to depart: $D_{l_i}(\mathfrak{o}_i) \leftarrow \min\{A_{l_{i-1}}(\mathfrak{o}_{i-1}) +$ unloading and loadingtime$, \alpha_{l_i}^{\mathrm{R}}(\mathfrak{o}_i)\}$
      update service segment's type: type$(\pi_{l_i}(\mathfrak{o}_i)) \leftarrow 1$
  else
    feasibility$(\mathfrak{r}, p) \leftarrow 0$
if feasibility$(\mathfrak{r}, p) = 1$ **then**
  calculate shipment schedules $\Gamma_{\mathfrak{r}}$ based on the service segments' schedules in the itinerary
  for terminal $i \in \Theta$, $\Gamma_{\mathfrak{r}}^{\text{unload}} + 1 \leq p \leq \Gamma_{\mathfrak{r}}^{\text{store}}$ **do**
    if $u(\mathfrak{r}) > u_i^{\mathrm{L},t,p}$ **then**
      feasibility$(\mathfrak{r}, p) \leftarrow 0$
    else
      $u_i^{\mathrm{L},t,p} \leftarrow u_i^{\mathrm{L},t,p} - u(\mathfrak{r})$
  for terminal $i \in \Theta$, $\Gamma_{\mathfrak{r}}^{\text{store}} + 1 \leq p \leq \Gamma_{\mathfrak{r}}^{\text{load}}$ **do**
    if $u(\mathfrak{r}) > u_i^{\mathrm{W},t,p}$ **then**
      feasibility$(\mathfrak{r}, p) \leftarrow 0$
    else
      $u_i^{\mathrm{W},t,p} \leftarrow u_i^{\mathrm{W},t,p} - u(\mathfrak{r})$
  for terminal $i \in \Theta$, $\Gamma_{\mathfrak{r}}^{\text{load}} + 1 \leq p \leq \Gamma_{\mathfrak{r}}^{\text{depart}}$ **do**
    if $u(\mathfrak{r}) > u_i^{\mathrm{L},t,p}$ **then**
      feasibility$(\mathfrak{r}, p) \leftarrow 0$
    else
      $u_i^{\mathrm{L},t,p} \leftarrow u_i^{\mathrm{L},t,p} - u(\mathfrak{r})$

6 Rolling horizon framework

At any time t, decisions suggested by the optimization model are not all to be implemented. We distinguish between the current implementation and the look-ahead components of the planning horizon, as shown in Figure 6. The acceptance decisions made at time t are implemented, that is, they are not to be changed in the follow up periods, and are transmitted to the appropriate stakeholders and departments of the platform for execution; but the decisions regarding shipment-to-service assignments, service schedules, and shipment itineraries made at time t are changeable. Period $t+1$ thus belongs to the current implementation component of the planning horizon. The following periods, from period $t+2$ to period $t+T$, belong to the look-ahead component. Most decisions of these periods are temporary in nature, they are not to be actually put into practice and executed. Such a procedure is used repeatedly, as time advances and the planning horizon is pushed into the future. This is called the rolling horizon procedure. The pseudocode of the rolling horizon framework is represented in Algorithm 5.

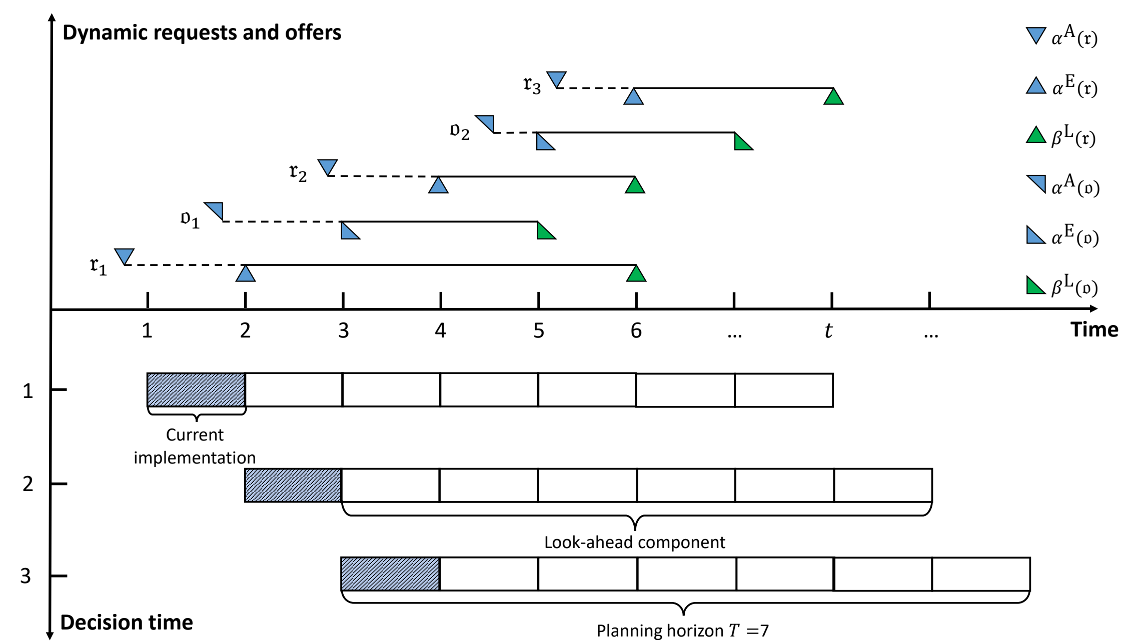


Figure 6: Rolling horizon framework for dynamic shipment-to-service matching.

Based on the decisions made at time t, for accepted requests that will be picked up before the next decision time $t+1$, shipments' itineraries are fixed, the platform thus needs to book the transport, loading, unloading, and storage capacities required for the shipments; for time-flexible services that are assigned to shipments whose itineraries are fixed, their time schedules will also be fixed, the platform thus needs to inform carriers the scheduled departure and arrival times. After implementing the fixed decisions made at time t, the platform will achieves to a new state at time $t+1$, including:

Algorithm 5 Rolling horizon framework.

Input: Terminals Θ; shipment requests $\mathfrak{R}$; service offers $\mathfrak{O}$; planning horizon T.

Output: Acceptance decision $\left[y^t_{\mathfrak{r}}\right]_{\forall \mathfrak{r}\in\tilde{\mathfrak{R}}^t}$ and $\left[y^t_{\mathfrak{o}}\right]_{\forall \mathfrak{o}\in\tilde{\mathfrak{O}}^t}$; assignment decision $\left[x^t_{\mathfrak{r}\pi_l(\mathfrak{o})}\right]_{\forall \mathfrak{r}\in\mathfrak{R}^t,\mathfrak{o}\in\mathfrak{O}^t,\pi_l(\mathfrak{o})\in\Pi(\mathfrak{o})}$; service schedules $\left[D_l(\mathfrak{o})\right], \left[A_l(\mathfrak{o})\right], \forall \mathfrak{o} \in \{\mathfrak{O}^t : \text{type}(\mathfrak{o}) = 0\}, \pi_l(\mathfrak{o}) \in \Pi(\mathfrak{o})$; shipment schedules $\left[\Gamma^{\text{pickup}}_{\mathfrak{r}}\right]$, $\left[\Gamma^{\text{delivery}}_{\mathfrak{r}}\right]$, $\left[\Gamma^{\text{unload}}_{\mathfrak{r}i}\right], \left[\Gamma^{\text{store}}_{\mathfrak{r}i}\right], \left[\Gamma^{\text{load}}_{\mathfrak{r}i}\right], \left[\Gamma^{\text{depart}}_{\mathfrak{r}i}\right], \forall \mathfrak{r}\in\mathfrak{R}^t, i\in\Theta$.

Initialize: Let $\mathfrak{R}^t \leftarrow \emptyset$, $\dot{\mathfrak{R}}^t \leftarrow \emptyset$, $\tilde{\mathfrak{R}}^t \leftarrow \emptyset$, $\mathfrak{O}^t \leftarrow \emptyset$, $\dot{\mathfrak{O}}^t \leftarrow \emptyset$, $\tilde{\mathfrak{O}}^t \leftarrow \emptyset$.

for decision time $t \in \{1, 2, ...\}$ **do**
  update set of accepted requests not yet picked up $\dot{\mathfrak{R}}^t \leftarrow \{\mathfrak{R} : \alpha^{\mathtt{A}}(\mathfrak{r}) \le t-1, \text{statu1}(\mathfrak{r}) = 1, \text{statu2}(\mathfrak{r}) = 0\}$
  update set of new requests $\tilde{\mathfrak{R}}^t \leftarrow \{\mathfrak{r} \in \mathfrak{R}^{\text{spot}} : t-1 < \alpha^{\mathtt{A}}(\mathfrak{r}) \le t, \text{statu1}(\mathfrak{r}) = 0, \text{statu2}(\mathfrak{r}) = 0\}$
  update set of active requests $\mathfrak{R}^t \leftarrow \dot{\mathfrak{R}}^t \cup \tilde{\mathfrak{R}}^t$
  update set of accepted offers $\dot{\mathfrak{O}}^t \leftarrow \{\mathfrak{o} \in \mathfrak{O} : \alpha^{\mathtt{A}}(\mathfrak{o}) \le t-1, \text{statu3}(\mathfrak{o}) = 1\}$
  update set of new offers $\tilde{\mathfrak{O}}^t \leftarrow \{\mathfrak{o} \in \mathfrak{O}^{\text{spot}} : t-1 < \alpha^{\mathtt{A}}(\mathfrak{o}) \le t, \text{statu3}(\mathfrak{o}) = 0\}$
  update set of active offers $\mathfrak{O}^t \leftarrow \dot{\mathfrak{O}}^t \cup \tilde{\mathfrak{O}}^t$
  generate acceptance decisions, assignment decisions, service schedules, shipment itineraries $\leftarrow$ using the P-ALNS algorithm proposed in Section 5
  for request $\mathfrak{r} \in \tilde{\mathfrak{R}}^t$ **do**
    if $y^t_{\mathfrak{r}} = 1$ **then**
      update decision status $\text{statu1}(\mathfrak{r}) \leftarrow 1$
      inform shippers that request $\mathfrak{r}$ is accepted
  for request $\mathfrak{r} \in \mathfrak{R}^t$ **do**
    if $\text{statu1}(\mathfrak{r}) = 1$ and $\Gamma^{\text{pickup}}_{\mathfrak{r}} \le t+1$ **then**
      update operation status $\text{statu2}(\mathfrak{r}) \leftarrow 1$
      inform shippers the shipment's itinerary
  for offer $\mathfrak{o} \in \tilde{\mathfrak{O}}^t$ **do**
    if $y^t_{\mathfrak{o}} = 1$ **then**
      update decision status $\text{statu3}(\mathfrak{o}) \leftarrow 1$
      inform carriers that offer $\mathfrak{o}$ is accepted
  for offer $\mathfrak{o} \in \mathfrak{O}^t$, **do**
    for $\pi_l(\mathfrak{o}) \in \Pi(\mathfrak{o})$ **do**
      update capacity $u^{t+1}_l(\mathfrak{o}) = u^t_l(\mathfrak{o}) - \sum_{\mathfrak{r}\in\{\mathfrak{R}^t:\text{statu1}(\mathfrak{r})=1,\text{statu2}(\mathfrak{r})=1\}} u(\mathfrak{r}) x^t_{\mathfrak{r}\pi_l(\mathfrak{o})}$
    inform carriers the booked transport capacity
    if $\text{type}(\mathfrak{o}) = 0$ and $\text{statu3}(\mathfrak{o}) = 1$ **then**
      for $\pi_l(\mathfrak{o}) \in \Pi(\mathfrak{o})$ **do**
        if $\sum_{\mathfrak{r}\in\{\mathfrak{R}^t:\text{statu1}(\mathfrak{r})=1,\text{statu2}(\mathfrak{r})=1\}} x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} \ge 1$ **then**
          update $\text{type}(\mathfrak{o}) \leftarrow 1$
          inform carriers the service schedules
  for terminal $i \in \Theta, p \in \{t+1, ..., t+T\}$ **do**
    update loading and unloading capacity $u^{\text{L},t+1,p}_i = u^{\text{L},t,p}_i - \sum_{\{\mathfrak{R}^t:\text{statu1}(\mathfrak{r})=1,\text{statu2}(\mathfrak{r})=1\}} u(\mathfrak{r}) \left(g^{\text{unload},p}_{\mathfrak{r}i} + g^{\text{load},p}_{\mathfrak{r}i}\right)$
    update storage capacity $u^{\text{W},t+1,p}_i = u^{\text{W},t,p}_i - \sum_{\mathfrak{r}\in\{\mathfrak{R}^t:\text{statu1}(\mathfrak{r})=1,\text{statu2}(\mathfrak{r})=1\}} u(\mathfrak{r}) g^{\text{store},p}_{\mathfrak{r}i}$

- Status updates:
  - $\text{statu1}(\mathfrak{r}) \leftarrow 1$ if $y^t_{\mathfrak{r}} = 1$, $\forall \mathfrak{r} \in \tilde{\mathfrak{R}}^t$;
  - $\text{statu2}(\mathfrak{r}) \leftarrow 1$ if $\Gamma^{\text{pickup}}_{\mathfrak{r}} \le t+1, \forall \mathfrak{r} \in \mathfrak{R}^t$;
  - $\text{statu3}(\mathfrak{o}) \leftarrow 1$ if $y^t_{\mathfrak{o}} = 1$, $\forall \mathfrak{o} \in \tilde{\mathfrak{O}}^t$.
- Service type updates:
  - For accepted time-flexible service $\mathfrak{o} \in \{\mathfrak{O}^t : \text{type}(\mathfrak{o}) = 0, \text{statu3}(\mathfrak{o}) = 1\}$, the time schedules will be fixed if any of the assigned shipments will be picked up before the next decision time, i.e., if $\sum_{\mathfrak{r}\in\{\mathfrak{R}^t:\text{statu1}(\mathfrak{r})=1,\text{statu2}(\mathfrak{r})=1\}} x^t_{\mathfrak{r}\pi_l(\mathfrak{o})} \ge 1$, $\forall l \in \{1, \ldots, |\Pi(\mathfrak{o})|\}$, update $\text{type}(\mathfrak{o}) \leftarrow 1$.

- Capacity updates:

$$u_i^{\mathrm{L},t+1,p} = u_i^{\mathrm{L},t,p} - \sum_{\mathfrak{r} \in \{\mathfrak{R}^t : \mathrm{statu1}(\mathfrak{r})=1, \mathrm{statu2}(\mathfrak{r})=1\}} u(\mathfrak{r}) \left(g_{\mathfrak{r}i}^{\mathrm{unload},p} + g_{\mathfrak{r}i}^{\mathrm{load},p} \right), \quad \forall i \in \Theta, \\ p \in \{t+2, ..., t+T\}, \tag{65}$$

$$u_i^{\mathrm{L},t+1,t+1+T} = u_i^{\mathrm{L}}, \quad \forall i \in \Theta, \tag{66}$$

$$u_i^{\mathrm{W},t+1,p} = u_i^{\mathrm{W},t,p} - \sum_{\mathfrak{r} \in \{\mathfrak{R}^t : \mathrm{statu1}(\mathfrak{r})=1, \mathrm{statu2}(\mathfrak{r})=1\}} u(\mathfrak{r}) g_{\mathfrak{r}i}^{\mathrm{store},p}, \quad \forall i \in \Theta, \\ p \in \{t+1, ..., t+T\}, \tag{67}$$

$$u_i^{\mathrm{W},t+1,t+1+T} = u_i^{\mathrm{W}}, \quad \forall i \in \Theta, \tag{68}$$

$$u_l^{t+1}(\mathfrak{o}) = u_l^t(\mathfrak{o}) - \sum_{\mathfrak{r} \in \{\mathfrak{R}^t : \mathrm{statu1}(\mathfrak{r})=1, \mathrm{statu2}(\mathfrak{r})=1\}} u(\mathfrak{r}) x_{\mathfrak{r}\pi_l(\mathfrak{o})}^t, \quad \forall \mathfrak{o} \in \mathfrak{O}^t, \\ l \in \{1, \dots, |\Pi(\mathfrak{o})|\}. \tag{69}$$

Equations (65,67) indicate that the loading and unloading, and storage capacity at terminal i during time period p at the next decision time $t+1$ equals the loading and unloading, and storage capacity at terminal i during time period p at the time t minus the booked loading and unloading, and storage capacity at time t, $p \in \{t+1, ..., t+T\}$, respectively. Equations (66,68) indicate that the loading and unloading, and storage capacity at terminal i during time period $t+1+T$ at decision time $t+1$ equals the maximum loading and unloading, and storage capacity at terminal i, respectively. Equations (69) indicate that the transport capacity of segment l of service $\mathfrak{o}$ at the next decision time $t+1$ equals the capacity of segment l of service $\mathfrak{o}$ at the time t minus the booked capacity at time t.

7 Numerical experiments

In this section, we use an interurban multimodal network in China to evaluate the performance of the proposed approaches. All approaches are implemented in MATLAB, and all experiments are executed on 3.70 GHz Intel Xeon processors with 32 GB of RAM. The optimization problems are solved with CPLEX 12.6.3. The topology of the network is shown in Figure 7, which includes 10 urban areas, 14 high-speed railway stations, 4 river/maritime ports, and 7 airports. Let instance $I - n_1 - n_2 - n_3 - n_4$ represent the instance with n_1 zones, n_2 terminals, n_3 shipment requests, and n_4 service offers.

7.1 Parameter tuning

To tune the algorithm parameters under the P-ALNS, we vary the values of the simulation length, removal fraction rate, σ_1, σ_2, σ_3, θ_1, θ_2, θ_3, γ_1, γ_2, and γ_3 for a given instance

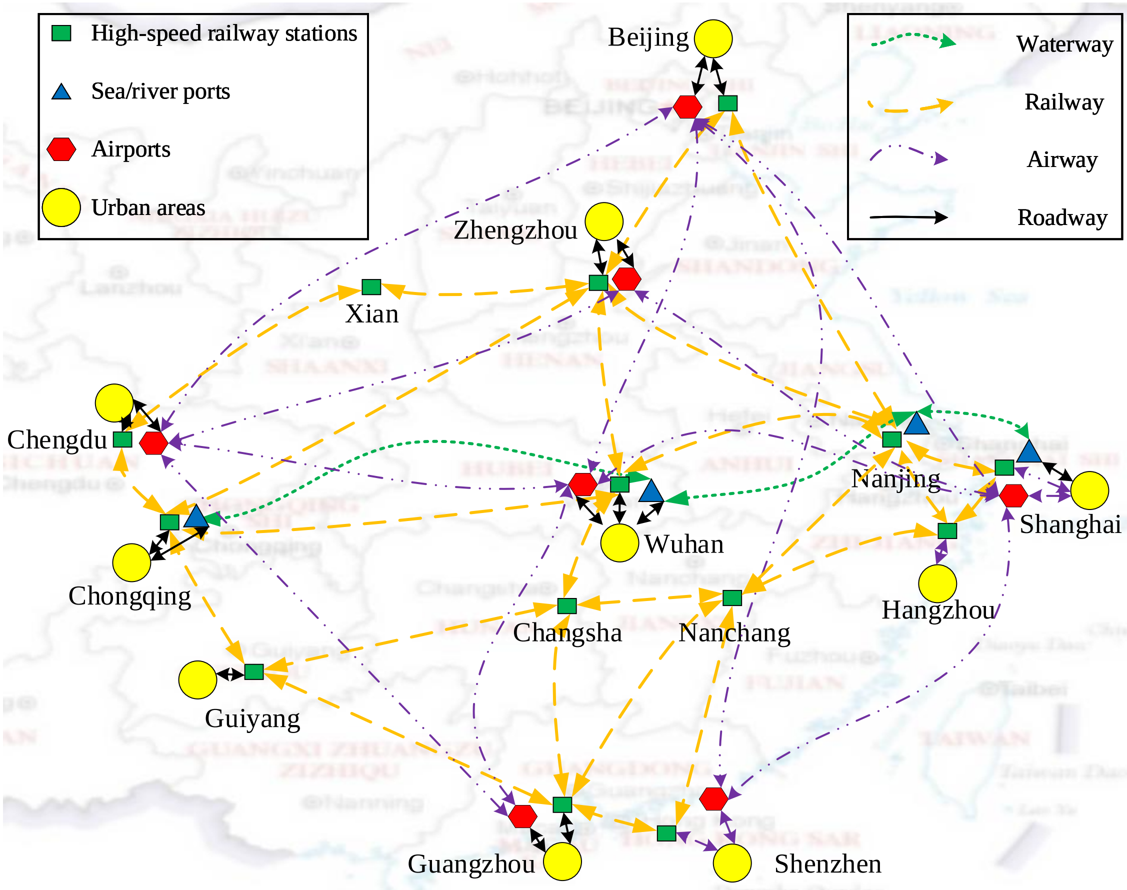


Figure 7: The topology of an interurban multimodal network in China.

$I - 10 - 10 - 150 - 50$. For each case, we run 10 times to obtain the average and best values. Table 2 shows that the larger the simulation length, the better solution quality the P-ALNS algorithm and the heavier the computation time. Increasing the removal fraction rate from 1% to 40%, the solution quality increases dramatically. When further increase the fraction rate to 60%, the performance of P-ALNS becomes worse. Also, the higher the removal fraction rate, the longer the CPU. Another interest finding is that the P-ALNS can find the best solution when the score for worse but new solutions σ_3 is higher than the score for the best solutions σ_1 and the score for a better and new solution σ_2. It turns out to be an effective way for diversifying the search. The P-ALNS performs the best when $\theta_1 > \theta_3 > \theta_2$, which means that in related removal process, the requests with similar distances are more likely to select the same service segments. Interestingly, the P-ALNS performs the best when $\gamma_2 > \gamma_3 > \gamma_1$, which means that in most constrained insertion process, the requests with larger distances are not that important. Instead, requests with tighter time windows become more difficult to insert, and therefore must be considered first.

The algorithm parameters in the following experiments are designed concerning a trade-off between solution quality and CPU time, as shown in Table 3.

Table 2: Sensitivity analysis of algorithm parameters.

ID	Simulation length	Removal fraction	σ_1	σ_2	σ_3	θ_1	θ_2	θ_3	γ_1	γ_2	γ_3	P-ALNS$_{best}$	P-ALNS$_{average}$	CPU
1	1000	20%	33	9	13	9	3	2	9	3	2	19292	18954	15
2	3000	20%	33	9	13	9	3	2	9	3	2	19787	19411	46
3	5000	20%	33	9	13	9	3	2	9	3	2	20037	19700	74
4	**7000**	20%	33	9	13	9	3	2	9	3	2	**20065**	**19919**	110
5	5000	1%	33	9	13	9	3	2	9	3	2	18837	18220	7
6	5000	5%	33	9	13	9	3	2	9	3	2	19881	19341	23
7	5000	10%	33	9	13	9	3	2	9	3	2	19802	19530	44
8	5000	20%	33	9	13	9	3	2	9	3	2	20037	19700	74
9	5000	**40%**	33	9	13	9	3	2	9	3	2	**20085**	**19750**	151
10	5000	60%	33	9	13	9	3	2	9	3	2	19652	19215	173
11	5000	20%	9	13	33	9	3	2	9	3	2	20110	19676	74
12	5000	20%	9	33	13	9	3	2	9	3	2	20121	**19789**	68
13	5000	20%	**13**	**9**	**33**	9	3	2	9	3	2	**20381**	19737	79
14	5000	20%	13	33	9	9	3	2	9	3	2	19981	19783	67
15	5000	20%	33	9	13	9	3	2	9	3	2	20037	19700	74
16	5000	20%	33	13	9	9	3	2	9	3	2	20041	19784	70
17	5000	20%	33	9	13	2	3	9	9	3	2	19739	19417	82
18	5000	20%	33	9	13	2	9	3	9	3	2	19656	19423	72
19	5000	20%	33	9	13	3	2	9	9	3	2	19778	19304	71
20	5000	20%	33	9	13	3	9	2	9	3	2	19916	19445	74
21	5000	20%	33	9	13	**9**	**2**	**3**	9	3	2	**20070**	**19847**	73
22	5000	20%	33	9	13	9	3	2	9	3	2	20037	19700	74
23	5000	20%	33	9	13	9	3	2	2	3	9	19983	19672	75
24	5000	20%	33	9	13	9	3	2	**2**	**9**	**3**	**20300**	**19796**	72
25	5000	20%	33	9	13	9	3	2	3	2	9	20059	19715	75
26	5000	20%	33	9	13	9	3	2	3	9	2	20176	19778	73
27	5000	20%	33	9	13	9	3	2	9	2	3	20241	19628	73
28	5000	20%	33	9	13	9	3	2	9	3	2	20037	19700	74

7.2 Comparison results between CPLEX and P-ALNS

To evaluate the performance of the P-ALNS algorithm, we compare it with the lower bounds, which represent the best feasible objective values, found by the CPLEX solver under 14 instances. Each instance is executed 10 times under the P-ALNS algorithm. For clarity, the abbreviations of the performance indicators along with their definition are provided in Table 4. Table 5 shows that the P-ALNS outperforms the CPLEX solver in finding superior feasible solutions in 10 out of 13 instances. For example, in instance $I-4-6-45-15$, the CPLEX solver takes 1199 seconds, while the P-ALNS achieves near-optimal solutions with an average gap of just 0.47% in 34 seconds. In instance $I-10-10-240-80$, the P-ALNS improves solution quality by an average of 55.38%. Additionally, in instance $I-10-10-270-90$, the CPLEX solver encounters an out of memory error. Moreover, for instances with 10 terminals, the P-ALNS consistently finds better solutions than CPLEX in a significantly shorter computation time.

Table 3: Default settings.

Parameters	Value
Number of runs per instance	10
Simulation length per run	5000
Number of route-wheel iterations	100
Reaction factor	0.1
Score for new best solution	13
Score for better solution	9
Score for worse but new solution	33
Start temperature control parameter	0.05
Cooling rate	0.9998
Lower limit of removal fraction	10%
Upper limit of removal fraction	40%
First Shaw parameter of distance	9
Second Shaw parameter of time	2
Third Shaw parameter of volume	3
First MCI parameter of distance	2
Second MCI parameter of time	9
Third MCI parameter of volume	3
Initial weights for removal operators	1,1,1
Initial weights for repair operators	1,1,1,1
CPLEX time limitation in seconds	3600

Table 4: Abbreviation of performance indicators and definition.

Abbreviations	Definition
UB_{MILP}	The upper bound of the MILP model obtained by CPLEX in a preset running time
LB_{MILP}	The best feasible objective value found by CPLEX solver in a preset running time
P-ALNS$_{best}$	The best feasible objective value obtained by the P-ALNS under default settings
P-ALNS$_{average}$	The average feasible objective value obtained by the P-ALNS under default settings
IMP_{best}	The improvement of P-ALNS$_{best}$ over LB_{MILP}: $\frac{P-ALNS_{best}-LB_{MILP}}{LB_{MILP}}$
$IMP_{average}$	The improvement of P-ALNS$_{average}$ over LB_{MILP}: $\frac{P-ALNS_{average}-LB_{MILP}}{LB_{MILP}}$
CPU_{MILP}	CPU time for solving the MILP model by CPLEX, unit: seconds
CPU_{P-ALNS}	The average computing duration of the P-ALNS algorithm, unit: seconds

7.3 Comparison results between FCFS and RHA

To investigate the effectiveness of the Rolling Horizon Approach (RHA) in handling dynamic shipment requests and dynamic service offers, we compare it to a First-Come-First-Serve (FCFS) approach. In the FCFS method, shipment-to-service assignments, shipments' itineraries, and service schedules cannot be reoptimized. We vary the number of spot requests and offers under instance $I-4-6-50-50$. Table 6 indicates that RHA outperforms FCFS in 9 out of 10 cases regarding total profits and the number of accepted requests and offers. This is because under the FCFS method, the platform allocates service capacities to shipment requests based on their arrival order. In contrast, RHA allows for the reassignment of service capacities to late-arriving, high-valued shipments,

Table 5: Comparison results between CPLEX and P-ALNS.

Instances	UB_{MILP}	LB_{MILP}	P-ALNS$_{best}$	P-ALNS$_{average}$	IMP_{best}	$IMP_{average}$	CPU_{MILP}	CPU_{P-ALNS}
I-4-6-15-5	3113	3113	3113	3113	0.00%	0.00%	19	5
I-4-6-30-10	5164	5164	5164	5164	0.00%	0.00%	136	11
I-4-6-45-15	8850	8850	8818	8808	-0.36%	-0.47%	1199	34
I-4-6-60-20	12079	11075	11130	11049	0.50%	-0.23%	3600	74
I-4-6-75-25	17742	15692	15974	15799	1.80%	0.68%	3600	167
I-4-6-90-30	20880	18832	18508	18270	-1.72%	-2.98%	3600	204
I-4-6-105-35	23784	19465	20776	20484	6.73%	5.23%	3600	274
I-4-6-120-40	31824	21271	26923	26510	26.57%	24.63%	3600	377
I-10-10-120-40	20803	17795	18450	18199	3.68%	2.27%	3600	69
I-10-10-150-50	24434	19617	19991	19815	1.90%	1.01%	3600	99
I-10-10-180-60	31923	22088	26022	25787	17.81%	16.75%	3600	136
I-10-10-210-70	37590	23764	29022	28754	22.12%	21.00%	3600	178
I-10-10-240-80	42066	20099	31708	31230	57.76%	55.38%	3600	222
I-10-10-270-90	Out of memory		33039	31990	-	-	3600	231
Average					**10.52%**	**9.48%**		

Table 6: Comparison results between FCFS and RHA.

Cases	$\lvert\mathfrak{R}^{\text{contract}}\rvert$	$\lvert\mathfrak{R}^{\text{spot}}\rvert$	$\lvert\mathfrak{O}^{\text{contract}}\rvert$	$\lvert\mathfrak{O}^{\text{spot}}\rvert$	FCFS			RHA		
					Total profit	Number of accepted requests	Number of accepted offers	Total profit	Number of accepted requests	Number of accepted offers
1	0	50	50	0	5687	18	50	11976	48	50
2	10	40	50	0	5432	19	50	12218	48	50
3	20	30	50	0	5796	20	50	12366	48	50
4	30	20	50	0	8958	31	50	12474	49	50
5	40	10	50	0	10418	41	50	12326	49	50
6	50	0	50	0	12327	50	50	12245	50	50
7	30	20	40	10	9175	31	40	12515	48	40
8	30	20	30	20	9487	32	30	12679	48	30
9	30	20	20	30	9596	33	20	11816	42	28
10	30	20	10	40	8134	36	14	11153	34	21

optimizing resource allocation dynamically. In case 6, all shipment requests and service offers are contracted, and their information are known. Since no new requests or offers arrive in this case, reoptimization under RHA is unnecessary. Compare cases 1 to 6, we can observe that the higher the ratio of contracted request, the better the performance of FCFS, which benefits from 'global' optimization. Interestingly, under RHA, total profit increases from cases 1 to 4, then declines from cases 4 to 6. This decline can be attributed to the necessity of accepting non-profitable contracted requests in cases 5 and 6. When comparing cases 4, 7, 8, we see that both FCFS and RHA benefit from the ability to reject non-profitable service offers. This underscores the importance of incorporating decision-making flexibility to enhance profitability. Finally, when comparing cases 9 and 10, it becomes clear that some profitable spot service offers may be rejected because they arrive before the spot requests that could be matched to them. This highlights the need for predictive mechanisms to temporarily hold or prioritize service offers that could be matched to future high-value shipment requests.

Table 7: Detailed information of time-scheduled service offers.

Offer ID	Type	Route	Offer fixed cost	Segment origin	Segment destination	Segment mode	Segment capacity	Segment departure time	Segment arrival time	Segment travel time	Segment travel cost	Segment distance
1	1	Chengdu-Chongqing-Wuhan-Shanghai	100	Chengdu	Chongqing	HSR	150	Mon 05:30	Mon 07:30	2	2	504
				Chongqing	Wuhan	HSR	150	Mon 08:00	Mon 11:30	3.5	4	876
				Wuhan	Shanghai	HSR	150	Mon 12:00	Mon 15:30	3.5	4	807
2	1	Shenzhen-Guangzhou-Wuhan-Beijing	100	Shenzhen	Guangzhou	HSR	150	Mon 06:30	Mon 07:00	0.5	1	147
				Guangzhou	Wuhan	HSR	150	Mon 07:30	Mon 11:30	4	4.5	920
				Wuhan	Beijing	HSR	150	Mon 12:00	Mon 16:00	4	4.5	1013
3	1	Chongqing-Wuhan-Shanghai	10	Chongqing	Wuhan	Barge	150	Mon 05:00	Wed 02:30	45.5	0.25	1274
				Wuhan	Shanghai	Barge	150	Wed 04:30	Thu 20:30	40	0.25	1125
4	1	Shenzhen-Shanghai	120	Shenzhen	Shanghai	Airplane	150	Mon 08:00	Mon 10:00	2	30	1600
5	1	Chengdu-Beijing	120	Chengdu	Beijing	Airplane	150	Mon 08:00	Mon 10:00	2	30	1503

7.4 Sensitivity analysis of scenario parameters

In this section, we use instance $I-7-13-12-22$ to perform a sensitivity analysis of scenario parameters. Detailed information on time-scheduled and time-flexible service offers are presented in Table 7 and Table 8, respectively. Based on the real operations of various transport modalities, we set the speeds as follows: high-speed rail (HSR) at 250 km/h, airplane at 800 km/h, truck at 80 km/h, barge at 28 km/h, intelligent vehicles at 60 km/h, and drones at 98 km/h, respectively. The travel time for each segment is calculated using the Euclidean distance. We designate HSR, airplane, and barge as long-haul transport modalities that connect consolidation and transshipment terminals, while intelligent vehicles and drones are allocated for feeder connections between service zones and consolidation terminals. The loading/unloading costs for HSR, airplane, truck, barge, intelligent vehicles, and drones are set as 0.3, 0.3, 0.3, 0.6, 0.3, 0.1 per unit, respectively. The loading/unloading times of HSR, airplane, truck, barge, intelligent vehicles, and drones are established at 0.25, 0.5, 0.5, 1, 0.25, 0.1 hours, respectively. The storage cost at terminals is set to 0.01 per unit per hour. Both the maximum storage and handling capacities at terminals are set as 150 units. Detailed information of shipment requests is presented in Table 9. For all shipment requests, we have established differentiated fare classes. Requests with shorter lead times (1, 2, 4, 5, 7, 8, 10, 11) feature higher fares and early/later delivery penalties, while requests with longer lead times (3, 6, 9) have lower fares and penalties.

Table 8: Detailed information of time-flexible service offers.

Offer ID	Type	Route	Offer fixed cost	Segment origin	Segment destination	Segment mode	Segment capacity	Segment time	departure window	Segment travel time	Segment travel cost	Segment distance
6	0	Guangzhou-Guangzhou	20	Guangzhou service zone	Guangzhou railway station	Intelligent vehicle	150	Mon 00:00	Mon 08:00	0.5	1	30
7	0	Guangzhou-Shenzhen	20	Guangzhou service zone	Shenzhen airport	Truck	150	Mon 00:00	Mon 08:00	2	4	160
8	0	Shenzhen-Shenzhen	20	Shenzhen service zone	Shenzhen railway station	Drone	150	Mon 00:00	Mon 08:00	0.2	0.8	20
9	0	Shenzhen-Shenzhen	20	Shenzhen service zone	Shenzhen airport	Drone	150	Mon 00:00	Mon 08:00	0.2	0.8	20
10	0	Chongqing-Chongqing	20	Chongqing service zone	Chongqing railway station	Intelligent vehicle	150	Mon 00:00	Mon 08:00	0.5	1	30
11	0	Chongqing-Chongqing	20	Chongqing service zone	Chongqing riverport	Intelligent vehicle	150	Mon 00:00	Mon 08:00	1	2	60
12	0	Chongqing-Chengdu	20	Chongqing service zone	Chengdu airport	Truck	150	Mon 00:00	Mon 08:00	7	14	560
13	0	Chengdu-Chengdu	20	Chengdu service zone	Chengdu railway station	Intelligent vehicle	150	Mon 00:00	Mon 08:00	0.5	1	30
14	0	Chengdu-Chengdu	20	Chengdu service zone	Chengdu airport	Intelligent vehicles	150	Mon 00:00	Mon 08:00	1	2	60
15	0	Chengdu-Chongqing	20	Chengdu service zone	Chongqing river port	Truck	150	Mon 00:00	Mon 08:00	7	14	560
16	0	Beijing-Beijing	20	Beijing railway station	Beijing service zone	Intelligent vehicle	150	Mon 10:00	Mon 24:00	0.5	1	30
17	0	Beijing-Beijing	20	Beijing airport	Beijing service zone	Intelligent vehicle	150	Mon 10:00	Mon 24:00	0.5	1	30
18	0	Shanghai-Shanghai	20	Shanghai railway station	Shanghai service zone	Drone	150	Mon 10:00	Mon 24:00	0.2	0.8	20
19	0	Shanghai-Shanghai	20	Shanghai airport	Shanghai service zone	Drone	150	Mon 10:00	Mon 24:00	0.2	0.8	20
20	0	Shanghai-Shanghai	20	Shanghai seaport	Shanghai service zone	Intelligent vehicle	150	Mon 10:00	Mon 24:00	1	2	60
21	0	Shanghai-Shanghai	20	Shanghai seaport	Shanghai service zone	Intelligent vehicle	150	Thu 10:00	Thu 24:00	1	2	60
22	0	Wuhan-Wuhan	20	Wuhan railway station	Wuhan river port	Intelligent vehicle	150	Mon 24:00	Wed 04:30	1	2	60

Table 9: Detailed information of shipment requests.

ID	Origin	Destination	Volume	Release time window		Early delivery time window		Target delivery time window		Late delivery time window		Fare	Early delivery penalty	Late delivery penalty
1	Guangzhou	Beijing	10	Mon 00:00	Mon 08:00	Mon 08:00	Mon 12:00	Mon 12:00	Mon 20:00	Mon 20:00	Mon 24:00	400	1	4
2	Guangzhou	Shanghai	10	Mon 00:00	Mon 08:00	Mon 08:00	Mon 12:00	Mon 12:00	Mon 20:00	Mon 20:00	Mon 24:00	450	1	4.5
3	Guangzhou	Shanghai	15	Mon 00:00	Mon 08:00	Mon 08:00	Tue 24:00	Wed 00:00	Thu 24:00	Fri 00:00	Fri 24:00	300	1	0.3
4	Shenzhen	Beijing	10	Mon 00:00	Mon 08:00	Mon 08:00	Mon 12:00	Mon 12:00	Mon 20:00	Mon 20:00	Mon 24:00	450	1	4.5
5	Shenzhen	Shanghai	10	Mon 00:00	Mon 08:00	Mon 08:00	Mon 12:00	Mon 12:00	Mon 20:00	Mon 20:00	Mon 24:00	500	1	5
6	Shenzhen	Shanghai	15	Mon 00:00	Mon 08:00	Mon 08:00	Tue 24:00	Wed 00:00	Thu 24:00	Fri 00:00	Fri 24:00	350	1	0.35
7	Chengdu	Beijing	10	Mon 00:00	Mon 08:00	Mon 08:00	Mon 12:00	Mon 12:00	Mon 20:00	Mon 20:00	Mon 24:00	500	1	5
8	Chendu	Shanghai	10	Mon 00:00	Mon 08:00	Mon 08:00	Mon 12:00	Mon 12:00	Mon 20:00	Mon 20:00	Mon 24:00	450	1	4.5
9	Chendu	Shanghai	15	Mon 00:00	Mon 08:00	Mon 08:00	Tue 24:00	Wed 00:00	Thu 24:00	Fri 00:00	Fri 24:00	300	1	0.3
10	Chongqing	Beijing	10	Mon 00:00	Mon 08:00	Mon 08:00	Mon 12:00	Mon 12:00	Mon 20:00	Mon 20:00	Mon 24:00	500	1	5
11	Chongqing	Shanghai	10	Mon 00:00	Mon 08:00	Mon 08:00	Mon 12:00	Mon 12:00	Mon 20:00	Mon 20:00	Mon 24:00	400	1	4
12	Chongqing	Shanghai	15	Mon 00:00	Mon 08:00	Mon 08:00	Tue 24:00	Wed 00:00	Thu 24:00	Fri 00:00	Fri 24:00	250	1	0.25

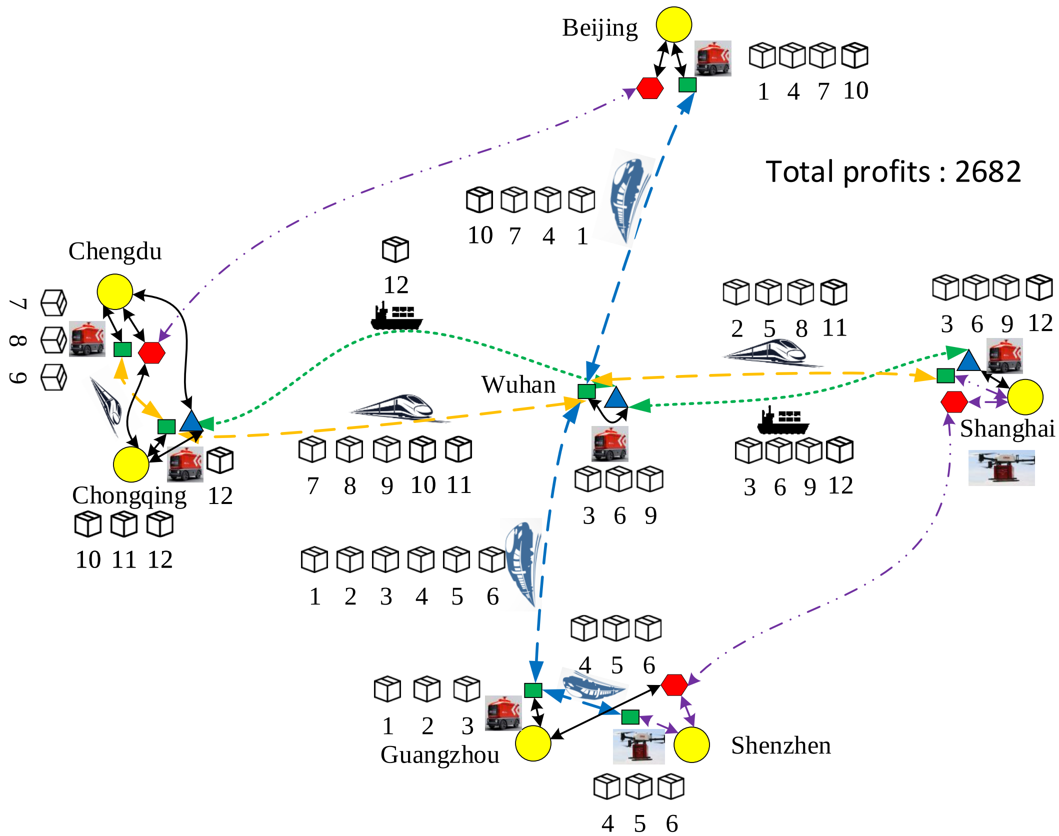


Figure 8: Shipment itineraries under the benchmark scenario.

Figure 8 illustrates shipment itineraries under the benchmark scenario. Shipments 1, 2, 3 will be transported by intelligent vehicles from Guangzhou service zone to Guangzhou railway station. From there, they will take a high-speed rail (HSR) to Wuhan railway station. Shipment 2 will then transfer to another HSR, heading to Shanghai railway station, while shipment 3 will switch to a barge service from Wuhan port to Shanghai port. Shipments 4, 5, 6 will be transported by drones from Shenzhen service zone to Shenzhen railway station. After this, they will be transported via HSR from Shenzhen railway station to Wuhan railway station. Shipment 5 will switch to another HSR service, while shipment 6 will transfer to a barge service. Shipments 7, 8, 9 will be transported by intelligent vehicles from Chengdu service zone to Chengdu railway station, where they will then transfer to a HSR bound for Wuhan railway station. Shipment 7 will continue on another HSR from Wuhan railway station to Beijing railway station, and shipment 9 will be transferred to a barge service from Wuhan port to Shanghai port. Lastly, shipments 10, 11, 12 will be initially transported from Chongqing service zone to Chongqing railway station. They will then board a HSR service from Chongqing railway station to Wuhan railway station. Afterward, shipment 10 will transfer to another HSR service from Wuhan railway station to Beijing railway station, while shipment 12 will switch to a barge service from Wuhan port to Shanghai port.

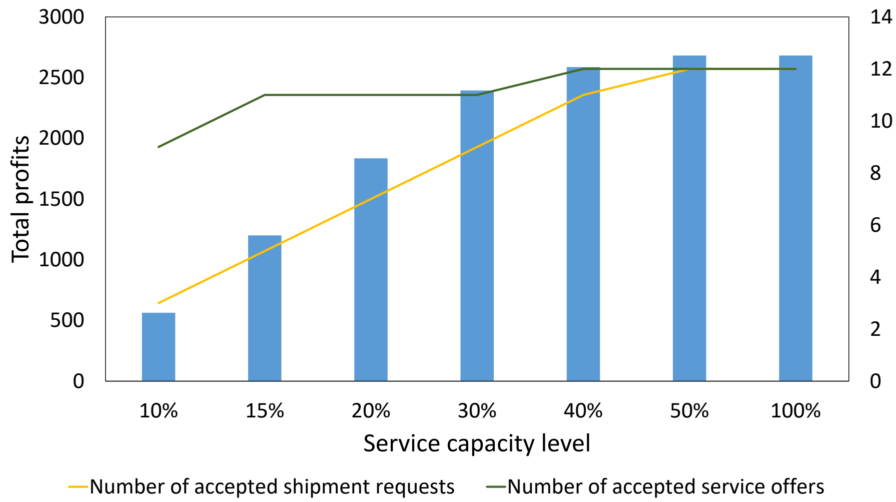


Figure 9: Impact of service capacity limitations.

7.4.1 Impact of service and terminal capacity limitations

In the benchmark scenario, the capacity of all services and terminals is set at 150 units, which exceeds the total volumes of all shipments. To evaluate the impact of service capacity limitations, we reduce the service capacity to 10%. Figure 9 shows that both total profits and the number of accepted shipment requests and service offers decline as the service capacity level decreases. At a service capacity level of 30%, each service segment's capacity is reduced to 45 units, allowing a maximum of 4 shipments per segment. Consequently, the original transport plan under the benchmark scenario becomes infeasible at this capacity level. As a result, shipments 3, 6, 9, which have lower fares and profits, are rejected by the platform since there are no profitable itineraries available to replace them. The experiment highlights the critical role of service capacity in maintaining profitability and operational feasibility. When service capacity is significantly reduced, the system becomes constrained, leading to a decline in both total profits and the number of accepted shipment requests. This is particularly evident for lower-fare shipments, which are more likely to be rejected when capacity is limited, as they do not provide sufficient profitability to justify their inclusion in the transport plan.

To evaluate the impact of terminal capacity limitations, we reduce the terminal capacity to 10%. Figure 9 illustrates that the total profits decreases as the terminal capacity decreases. At a terminal capacity level of 20%, each terminal can only handle and storage 30 units at each time period. Shipments 1, 6, and 9 are thus rejected due to terminal handling capacity limitations. The experiment demonstrates that terminal capacity constraints directly impact operational efficiency and profitability. When terminal capacity is reduced, the system's ability to handle and store shipments is significantly limited, leading to the rejection of certain shipments and a decline in total profits. This underscores the importance of terminal capacity as a critical bottleneck in the logistics network.

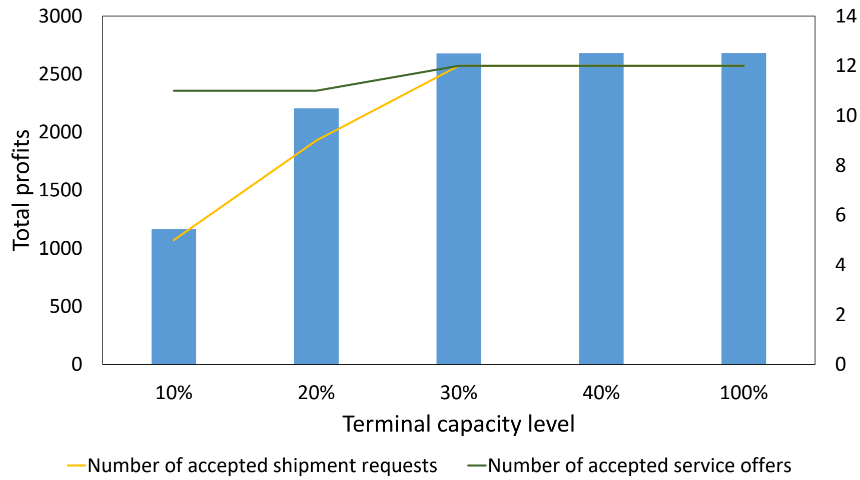


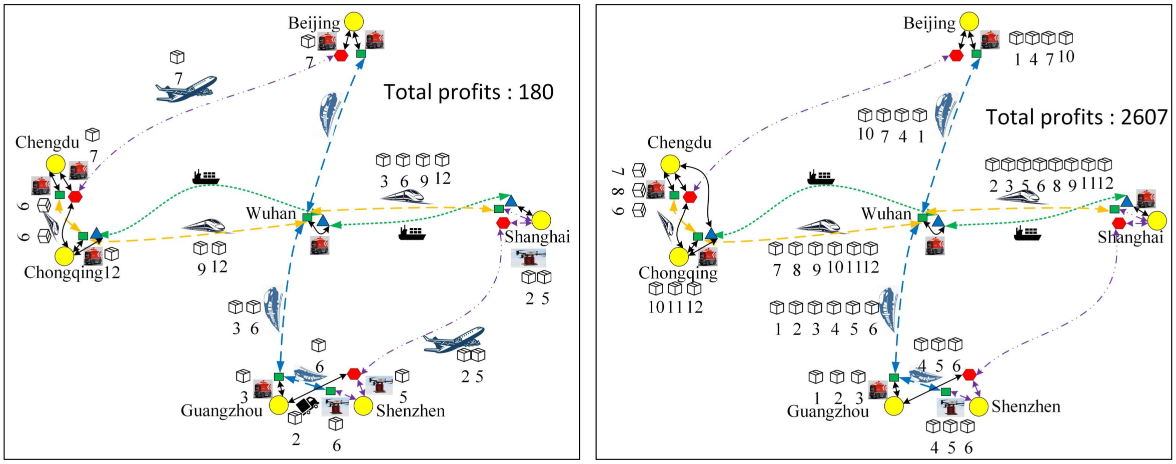
Figure 10: Impact of terminal capacity limitations.

7.4.2 Impact of shipment time windows

To evaluate the impact of shipment time windows, we adjusted the length of the delivery time windows to 50% and 75% of the benchmark scenario. When shipments' time windows reduces to 75% of the benchmark scenario, the latest delivery time of shipments 3, 6, 9, and 12 changes to Thursday 18:00. Barge service 3 becomes infeasible for these shipments since its arrival time is Thursday 20:30. As a result, these shipments' itineraries from Wuhan to Shanghai switch from barge service 3 to HSR service 1. This change results in total profits decreasing from 2682 to 2607. When we further decrease shipments' time windows to 50% of the benchmark scenario, the latest delivery time of shipments 1, 2, 4, 5, 7, 8, 10, 11 changes to Monday 12:00. In this scenario, only airplane services 4 and 5 are feasible for these shipments. The platform rejects non-profitable requests 1, 4, 8, 10, and 11, and assigns shipments 2, 5, 7 to the airplane services. As a result, total profits drop to 180. The experiment reveals that tighter shipment time windows significantly impact the feasibility of transportation modes and overall profitability. When delivery time windows are reduced, certain services become infeasible due to misaligned time schedules, forcing a shift to more expensive alternatives (e.g., high-speed rail or airplane services). This not only increases operational costs but also leads to the rejection of less profitable shipments, further reducing total profits.

7.4.3 Impact of differentiated fare classes

To assess the impact of differentiated fare classes, we design three fare categories for shipment request 2 from Guangzhou to Shanghai:



(a) Time window = 50%*benchmark scenario
(b) Time window = 75%*benchmark scenario

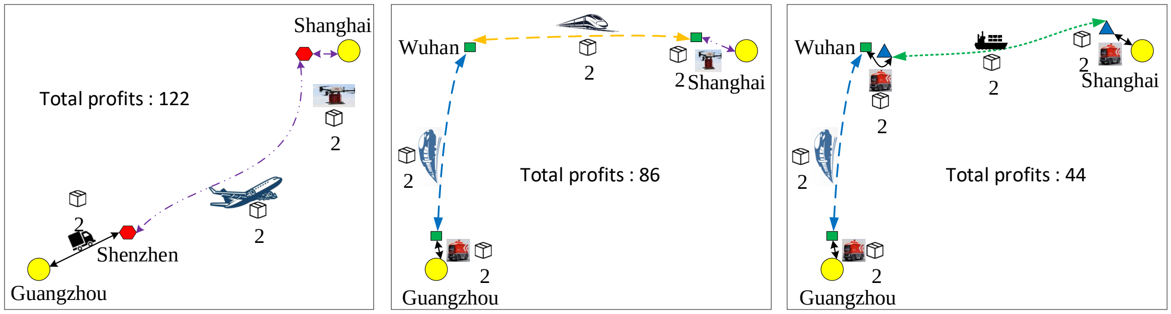
Figure 11: Impact of shipment time windows.

- High fare class: this option guarantees delivery before Monday 12:00, with a fare of 650 and a delay penalty of 6.5 per unit per hour;
- Medium fare class: this option ensures delivery before Monday 24:00, with a fare of 450 and a delay penalty of 4.5 per unit per hour;
- Low fare class: this option provides delivery before Friday 24:00, with a fare of 350 and a delay penalty of 0.35 per unit per hour.

Figure 12 illustrates the transport modes used under each fare class scenario. Under the high fare class, the shipment is transported by airplane for long-haul transportation. In the medium fare class scenario, the shipment is moved using two high-speed rail lines. For the low fare class, the shipment utilizes a combination of HSR and barge services for the long-haul transportation. The experiment highlights the importance of aligning transportation modes with fare classes and customer preferences. High fare classes, typically associated with high-value products and strict time windows, justify the use of faster but more expensive transportation modes like airplanes. In contrast, low fare classes, often linked to low-value products and flexible time windows, allow for the use of cost-effective but slower multimodal options, such as a combination of high-speed rail and barge services. This demonstrates that fare class segmentation enables the optimization of transportation costs while meeting diverse customer needs.

7.4.4 Impact of resource sharing among different carriers

To evaluate the benefits of resource sharing among different carriers, we design a scenario involving three carriers that operate in long-haul transportation: a HSR carrier, a barge carrier, and an air carrier. The HSR carrier operates HSR services 1 and 2 and receives



(a) High fare class (b) Medium fare class (c) Low fare class

Figure 12: Impact of differentiated fare classes.

Table 10: Impact of resource sharing among different carriers.

Carriers	Requests	Without resource sharing	With resource sharing
HSR	1	$Guangzhou \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Beijing$	$Guangzhou \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Beijing$
	3	$Guangzhou \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Shanghai$	$Guangzhou \xrightarrow{HSR} Wuhan \xrightarrow{barge} Shanghai$
	4	$Shenzhen \xrightarrow{HSR} Guangzhou \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Beijing$	$Shenzhen \xrightarrow{HSR} Guangzhou \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Beijing$
	6	$Shenzhen \xrightarrow{HSR} Guangzhou \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Shanghai$	$Shenzhen \xrightarrow{HSR} Guangzhou \xrightarrow{HSR} Wuhan \xrightarrow{barge} Shanghai$
	8	$Chengdu \xrightarrow{HSR} Chongqing \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Shanghai$	$Chengdu \xrightarrow{HSR} Chongqing \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Shanghai$
	11	$Chongqing \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Shanghai$	$Chongqing \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Shanghai$
Barge	9	reject	$Chengdu \xrightarrow{HSR} Chongqing \xrightarrow{HSR} Wuhan \xrightarrow{barge} Shanghai$
	12	$Chongqing \xrightarrow{barge} Wuhan \xrightarrow{barge} Shanghai$	$Chongqing \xrightarrow{barge} Wuhan \xrightarrow{barge} Shanghai$
Air	2	$Guangzhou \xrightarrow{truck} Shenzhen \xrightarrow{air} Shanghai$	$Guangzhou \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Shanghai$
	5	$Shenzhen \xrightarrow{air} Shanghai$	$Shenzhen \xrightarrow{HSR} Guangzhou \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Shanghai$
	7	reject	$Chengdu \xrightarrow{HSR} Chongqing \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Beijing$
	10	reject	$Chongqing \xrightarrow{HSR} Wuhan \xrightarrow{HSR} Beijing$
Total profits		1262	2682

shipment requests 1, 3, 4, 6, 8, and 11. The barge carrier operates barge service 3 and receives shipment requests 9, and 12. The air carrier operates air services 4 and 5 and receives shipment requests 2, 5, 7, and 10. Table 10 indicates that without resource sharing, shipment requests 7, 9, 10 are rejected. Furthermore, shipments 3 and 6 switch from barge service to HSR service, while shipments 2 and 5 switch from HSR service to air service, which incurs higher costs. Consequently, total profits decrease from 2682 to 1262. The experiment underscores the significant benefits of resource sharing among carriers in enhancing operational efficiency and profitability. Without resource sharing, carriers are constrained by their individual capacities and service offerings, leading to the rejection of certain shipments and suboptimal routing decisions (e.g., switching to more expensive transportation modes). This results in a substantial decline in total profits. Resource sharing, however, enables carriers to leverage synergies, optimize capacity utilization, and provide more flexible and cost-effective transportation solutions, ultimately supporting synchromodality in interurban transportation.

8 Conclusions and future research

This paper investigated a dynamic shipment-to-service matching problem within a synchromodal platform for interurban transportation. We formulated a mixed-integer linear programming model that integrates the acceptance decisions of shipment requests and service offers, shipment-to-service assignments, shipment itineraries and service time schedules. Given the computational complexity, we designed a preprocessing-based adaptive large neighborhood search algorithm (P-ALNS) to solve the optimization problem. To accommodate dynamic shipment requests and service offers, we developed a rolling horizon approach (RHA) that controls the implementation and re-optimization of decisions. We validated these approaches using an interurban multimodal network in China. The experimental results demonstrate that the P-ALNS outperforms the CPLEX solver in large-scale instances. Furthermore, the RHA surpasses the first-come-first-serve approach in all scenarios except when there are no dynamic requests and offers. Additionally, we conducted a sensitivity analysis to assess the effects of service and terminal capacity limitations, shipment time windows, differentiated fare classes, and resource sharing among various carriers. The analysis provides valuable managerial insights based on the results obtained:

- **Balance Capacity and Demand**: Managers should align service capacity with demand to optimize profitability and efficiency, avoiding costs from overcapacity and revenue loss from undercapacity.
- **Optimize Terminal Capacity**: Enhancing terminal utilization through infrastructure investment, efficient storage, and demand forecasting is critical. This helps prevent lost revenue from capacity constraints.
- **Improve Transportation Schedules**: To address tight time windows of shipment requests, managers should focus on optimizing transportation schedules and improving service flexibility.
- **Tailor Service Offerings**: Managers should develop differentiated services for various fare classes. High fare shippers should receive fast and reliable options, while low fare shippers benefit from cost-effective multimodal solutions.
- **Promote Collaboration**: Managers should actively pursue collaboration and resource-sharing agreements with other carriers to improve system-wide efficiency and profitability. By pooling resources and coordinating operations, carriers can reduce inefficiencies, accommodate more shipments, and offer competitive transportation options. Investing in platforms or technologies that facilitate real-time resource sharing and coordination among carriers can further enhance the benefits of synchromodality.

While this study provides valuable insights into the dynamic shipment-to-service matching problem within a synchromodal platform, several avenues for future research remain unexplored. First, in interurban transportation, many factors, such as weather condition, traffic congestion, terminal congestion, demand and capacity fluctuation, and dynamic pricing, can significantly impact the feasibility and efficiency of transport plans. Future research could focus on the integration of advanced predictive analytics and machine learning techniques to enhance the responsiveness and resilience of the rolling horizon approach in such uncertain environments. Second, the current model could be extended to incorporate first-mile and last-mile logistics, including intelligent vehicle and drone routing. This extension would address the critical challenge of synchronizing operations at consolidation terminals, ensuring seamless transitions between long-haul transportation and local delivery. Exploring collaboration between intelligent vehicles and drones, could further optimize the efficiency of the entire logistics chain.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grants 72301203 and 62173263. While working on the project, the second author held the UQAM Chair on Intelligent Logistics and Transportation Systems Planning, and was Adjunct Professor, Department of Computer Science and Operations Research, Université de Montréal, while the fifth held the Canada Research Chair in Stochastic Optimization of Transport and Logistics Systems. The authors gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Council of Canada (NSERC) through its Collaborative Research and Development, Discovery, Acceleration, and Engage grant programs, and the support of Fonds de recherche du Québec through their CIRRELT infrastructure grants.

References

Archetti, C., Peirano, L., and Speranza, M. G. (2022). Optimization in multimodal freight transportation problems: A survey. *European Journal of Operational Research*, 299(1):1–20.

Arslan, A. M., Agatz, N., Kroon, L., and Zuidwijk, R. (2019). Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science*, 53(1):222–235.

Behrendt, A., Savelsbergh, M., and Wang, H. (2024). Task assignment, pricing, and capacity planning for a hybrid fleet of centralized and decentralized couriers. *Transportation Research Part C: Emerging Technologies*, 160:104533.

Berbeglia, G., Cordeau, J.-F., and Laporte, G. (2010). Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202(1):8–15.

Ferrucci, F. and Bock, S. (2014). Real-time control of express pickup and delivery processes in a dynamic environment. *Transportation Research Part B: Methodological*, 63:1–14.

Filom, S. and Razavi, S. (2025). A learning-based robust optimization framework for synchromodal freight transportation under uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 195:103967.

Ghilas, V., Cordeau, J.-F., Demir, E., and Woensel, T. V. (2018). Branch-and-price for the pickup and delivery problem with time windows and scheduled lines. *Transportation Science*, 52(5):1191–1210.

Ghilas, V., Demir, E., and Van Woensel, T. (2016a). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines. *Computers & Operations Research*, 72:12–30.

Ghilas, V., Demir, E., and Woensel, T. V. (2016b). A scenario-based planning for the pickup and delivery problem with time windows, scheduled lines and stochastic demands. *Transportation Research Part B: Methodological*, 91:34–51.

Giusti, R., Manerba, D., Bruno, G., and Tadei, R. (2019). Synchromodal logistics: An overview of critical success factors, enabling technologies, and open research issues. *Transportation Research Part E: Logistics and Transportation Review*, 129:92–110.

Guo, W., Atasoy, B., van Blokland, W. B., and Negenborn, R. R. (2020). A dynamic shipment matching problem in hinterland synchromodal transportation. *Decision Support Systems*, page 113289.

Guo, W., Atasoy, B., van Blokland, W. B., and Negenborn, R. R. (2021). Global synchromodal transport with dynamic and stochastic shipment matching. *Transportation Research Part E: Logistics and Transportation Review*, 152:102404.

Labarthe, O., Ahmadi, G., Klibi, W., Deschamps, J.-C., and Montreuil, B. (2024). A sustainable on-demand urban delivery service enabled by synchromodality and synergy in passenger and freight mobility. *Transportation Research Part C: Emerging Technologies*, 161:104544.

Larsen, R. B., Guo, W., and Atasoy, B. (2023). A real-time synchromodal framework with co-planning for routing of containers and vehicles. *Transportation Research Part C: Emerging Technologies*, 157:104412.

Li, L., Negenborn, R. R., and Schutter, B. D. (2015). Intermodal freight transport planning – a receding horizon control approach. *Transportation Research Part C: Emerging Technologies*, 60:77–95.

Lyu, Z. and Yu, A. J. (2023). The pickup and delivery problem with transshipments: Critical review of two existing models and a new formulation. *European Journal of Operational Research*, 305(1):260–270.

Madani, B., Ndiaye, M., and Salhi, S. (2024). Hybrid truck-drone delivery system with multi-visits and multi-launch and retrieval locations: Mathematical model and adaptive variable neighborhood search with neighborhood categorization. *European Journal of Operational Research*, 316(1):100–125.

Miller, J., Nie, Y. M., and Liu, X. (2020). Hyperpath truck routing in an online freight exchange platform. *Transportation Science*, 54(6):1676–1696.

Mitrović-Minić, S. and Laporte, G. (2006). The pickup and delivery problem with time windows and transshipment. *INFOR: Information Systems and Operational Research*, 44(3):217–227.

Mousavi, K., Bodur, M., Cevik, M., and Roorda, M. J. (2024). Approximate dynamic programming for pickup and delivery problem with crowd-shipping. *Transportation Research Part B: Methodological*, 187:103027.

Park, A., Chen, R., Cho, S., and Zhao, Y. (2023). The determinants of online matching platforms for freight services. *Transportation Research Part E: Logistics and Transportation Review*, 179:103284.

Qu, W., Rezaei, J., Maknoon, Y., and Tavasszy, L. (2019). Hinterland freight transportation replanning model under the framework of synchromodality. *Transportation Research Part E: Logistics and Transportation Review*, 131:308–328.

Qu, Y. and Bard, J. F. (2012). A grasp with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers & Operations Research*, 39(10):2439–2456.

Rais, A., Alvelos, F., and Carvalho, M. (2014). New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research*, 235(3):530–539.

Rivera, A. E. P. and Mes, M. R. K. (2022). Anticipatory scheduling of synchromodal transport using approximate dynamic programming. *Annals of Operations Research*.

Sakti, S., Zhang, L., and Thompson, R. G. (2023). Synchronization in synchromodality. *Transportation Research Part E: Logistics and Transportation Review*, 179:103321.

SteadieSeifi, M., Dellaert, N., Nuijten, W., van Woensel, T., and Raoufi, R. (2014). Multimodal freight transportation planning: A literature review. *European Journal of Operational Research*, 233(1):1–15.

Stoia, S., Laganà, D., and Ohlmann, J. W. (2025). Dynamic pickup-and-delivery for collaborative platforms with time-dependent travel and crowdshipping. *European Journal of Operational Research*, 322(1):70–84.

Su, E., Qin, H., Li, J., and Pan, K. (2023). An exact algorithm for the pickup and delivery problem with crowdsourced bids and transshipment. *Transportation Research Part B: Methodological*, 177:102831.

Ulmer, M. W., Thomas, B. W., Campbell, A. M., and Woyak, N. (2021). The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transportation Science*, 55(1):75–100.

van Riessen, B., Negenborn, R. R., and Dekker, R. (2016). Real-time container transport planning with decision trees based on offline obtained optimal solutions. *Decision Support Systems*, 89:1–16.

Wolfinger, D. and Salazar-González, J.-J. (2021). The pickup and delivery problem with split loads and transshipments: A branch-and-cut solution approach. *European Journal of Operational Research*, 289(2):470–484.

Wu, Y., Huang, J., and Chen, X. (2024). The information value of logistics platforms in a freight matching market. *European Journal of Operational Research*, 312(1):227–239.

Yang, C., Bian, J., Guo, X., and Jiang, W. (2024). Logistics outsourcing strategy with online freight platforms. *Omega*, 125:103042.

Zhang, H., Luo, K., Xu, Y., Xu, Y., and Tong, W. (2022). Online crowdsourced truck delivery using historical information. *European Journal of Operational Research*, 301(2):486–501.

Zhang, Y., Negenborn, R. R., and Atasoy, B. (2023). Synchromodal freight transport re-planning under service time uncertainty: An online model-assisted reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 156:104355.

Zhang, Y., Tan, X., Gan, M., Liu, X., and Atasoy, B. (2025). Operational synchromodal transport planning methodologies: Review and roadmap. *Transportation Research Part E: Logistics and Transportation Review*, 194:103915.

Zhen, L., Gao, J., Zhang, N., Li, H., and Fan, T. (2023). A decision model for high-speed railway express systems. *Computers & Industrial Engineering*, 177:109087.

Zhen, L., He, X., Zhang, N., Yang, Z., and Ren, Y. (2024). Optimizing high-speed railway express system under uncertainty. *Computers & Operations Research*, 165:106565.