Transportation Research Part E 214 (2026) 105051

Contents lists available at ScienceDirect

Transportation Research Part E

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

Model-assisted multi-agent reinforcement learning for collaborative synchromodal transport re-planning under service time uncertainty

Xiaoyu Luo [a], Yimeng Zhang [a,b,c,*], Mi Gan [a,b,c], Xiaobo Liu [a,b,c], Bilge Atasoy [d]

[a] School of Transportation & Logistics, Southwest Jiaotong University, Chengdu, China
[b] Intelligent Comprehensive Transportation Key Laboratory of Sichuan Province, Southwest Jiaotong University, Chengdu, China
[c] National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Southwest Jiaotong University, Chengdu, China
[d] Department of Maritime and Transport Technology, Delft University of Technology, Delft, The Netherlands

ARTICLE INFO

Keywords:
Synchromodal transport
Service time uncertainty
Multi-agent reinforcement learning
Collaborative planning
Intermodal transport

ABSTRACT

Synchromodal transport operations frequently require rapid re-planning in response to uncertain terminal service times, yet such decisions are typically made by multiple carriers with heterogeneous objectives and decentralized control. This study investigates collaborative re-planning in synchromodal transport networks under service-time uncertainty from a multi-carrier perspective. We formulate the problem as a decentralized decision-making process in which carriers must coordinate re-planning actions while preserving individual rationality. To address this challenge, we propose a model-assisted multi-agent reinforcement learning (MARL) framework that combines heuristic plan generation with learning-based carrier coordination. An adaptive large neighborhood search (ALNS) heuristic is used to generate feasible candidate re-planning options, while carriers are modeled as learning agents that decide whether to accept or reject these options through a consensus mechanism. Rather than directly optimizing a centralized model, carriers learn acceptance policies based on local observations and realized outcomes, without requiring prior knowledge of service-time distributions. An event-triggered and regret-based learning scheme is introduced to align global delay mitigation with carrier-specific cost preferences under uncertainty. Extensive computational experiments on a realistic synchromodal transport network demonstrate that the proposed framework significantly improves re-planning performance compared with benchmark strategies. The results show that decentralized, heterogeneous multi-agent coordination can outperform centralized and homogeneous approaches, particularly in scenarios with high uncertainty and avoidable delays. Mechanism and ablation analyses further reveal the structural importance of multi-agent decomposition, carrier heterogeneity, and the temporal allocation of decision preferences along the transport chain. Additional analyses examine sensitivity to key reward hyperparameters and negotiation round limits, robustness under alternative disruption distributions and network-graph perturbations, and controlled extension settings with more carriers and transshipments. The proposed approach provides a practical decision-support methodology for collaborative synchromodal transport re-planning under uncertainty.

* Corresponding author at: Southwest Jiaotong University, No. 999, Xi'an Road, Pidu District, Chengdu, Sichuan, PR China.
E-mail address: yimengzhang@swjtu.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.tre.2026.105051
Received 29 January 2026; Received in revised form 23 June 2026; Accepted 24 June 2026
1366-5545/© 2026 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

1. Introduction

Freight transport systems are increasingly exposed to time uncertainty caused by unexpected disruptions, such as terminal congestion, equipment failures, and weather-related events, which demand greater flexibility and resilience in transport planning (Zhang et al., 2025, 2023; Guo et al., 2024, 2021). In this context, synchromodal transport has gained prominence as an operational paradigm that dynamically coordinates multiple transport modes (e.g., road, rail, and inland waterways) through continuous information updates and on-the-fly re-planning (Zhang et al., 2022a,c). By enabling timely mode switching and smoother inter-terminal transfers, synchromodal operations can reduce total logistics cost, improve robustness to disruptions, and better utilize network capacity (Zhang et al., 2025).

A fundamental barrier to realizing these benefits is service-time uncertainty at terminals. In synchromodal settings, service time comprises the operational time consumed by pickup, delivery, and transshipment activities at terminals–typically including queuing, loading/unloading, inspections, and other handling processes. These times are frequently disrupted by unanticipated events such as equipment maintenance and breakdowns, adverse weather, labor and gate congestion, and yard-level operational disturbances (Zhang et al., 2023). As a result, even a well-designed plan can become infeasible once terminal service times deviate from expectations, propagating delays across tightly coupled schedules. Therefore, effective synchromodal operations require decision strategies that adapt to terminal-specific service-time realizations in real time, rather than relying solely on static planning assumptions.

The challenge is compounded when synchromodal plans span multiple carriers (Guo et al., 2024). Indeed, this shift towards multi-carrier collaboration strongly aligns with the overarching vision of the Physical Internet (PI), which advocates open and shared logistics networks (Pan et al., 2025). PI research further emphasizes that realizing such openness requires not only operational flexibility but also enabling governance arrangements and interoperable digital information infrastructures (e.g., track-and-trace capabilities supported by open interface platforms), especially around key hubs such as ports and terminals (Fahim et al., 2021b,a). However, translating this vision into day-to-day operational decisions is practically complex: although centralized optimization can, in principle, produce a system-optimal solution, it presumes a single decision authority and overlooks the strategic behavior of self-interested participants. In practice, synchromodal networks are operated by carriers with heterogeneous objectives and operational constraints. For example, a barge operator may emphasize cost efficiency and tolerate longer lead times, whereas a trucking operator may prioritize reliability and tight delivery windows even at a higher cost. A globally cost-minimizing centralized plan can shift risk and delay disproportionately onto one party, violating individual rationality and undermining willingness to collaborate. Moreover, classical centralized approaches–such as stochastic programming and robust optimization–often require strong distributional assumptions and incur substantial computational burden, which limits their suitability for real-time, multi-actor synchromodal decision-making.

To address these challenges, Multi-Agent Reinforcement Learning (MARL) provides a natural and promising paradigm by modeling each carrier as an autonomous decision-making agent that learns adaptive re-planning strategies through interaction with the environment and other agents (Ren et al., 2022). MARL is particularly attractive for synchromodal transport, as it enables decentralized decision-making while capturing strategic interdependencies among self-interested carriers. However, directly applying MARL in this context is far from trivial. The problem is characterized by a combinatorial action space, arising from route-mode-schedule choices, which leads to severe scalability issues commonly referred to as the curse of dimensionality. In addition, effective coordination among heterogeneous agents is difficult to achieve in the absence of a central coordinator, especially when agents pursue different operational objectives and possess only partial system information (Gosavi, 2009; Liu et al., 2025).

To overcome these obstacles, this study proposes a hybrid learning optimization framework that tightly integrates reinforcement learning with operations research-based planning methods. Building on the synchromodal transport model developed by Zhang et al. (2023), we extend the original single-agent setting to a multi-agent cooperative re-planning framework. Within this framework, an Adaptive Large Neighborhood Search (ALNS) algorithm plays a central role: it generates high-quality transport plans, provides structured and informative state representations, checks solution feasibility for reward evaluation, and supplies candidate plan combinations that guide agents' decision spaces. By embedding ALNS within the MARL loop, the proposed approach significantly reduces the effective dimensionality of both the state and action spaces, thereby improving learning efficiency and convergence stability. This model-assisted MARL architecture leverages the complementary strengths of optimization and learning–optimization ensures feasibility and structural guidance, while learning enables adaptability under uncertainty.

Fig. 1 illustrates the re-planning mechanism under service time uncertainty. Unexpected events at terminals induce disruptions in service times, causing delays across water, rail, and road operations. The duration and propagation of such disruptions are not known in advance, requiring carriers to react dynamically. In the illustrated scenario, two transport requests, r_1 and r_2, are initially assigned to carriers c_1 and c_2 and planned to move from terminal A to destinations E and F, respectively. When unexpected events occur at terminals A, B, and D, the original plan becomes infeasible, and both requests would incur delivery delays if no corrective action is taken. Through decentralized re-planning, each carrier adjusts its routing and mode choices based on its learned policy and local assessment of system conditions. Specifically, the transport mode for request r_1 between terminals B and E is switched from rail to barge, while the route for request r_2 is reconfigured from A-D-F to A-C-F. These coordinated yet decentralized adjustments eliminate delays and restore on-time delivery for both requests.

By adopting a MARL-based framework, the proposed approach enables multiple carriers to collaboratively and adaptively respond to service time uncertainty, achieving real-time re-planning without centralized control. The framework supports decentralized autonomy while preserving system-level performance through structured coordination induced by optimization-assisted learning.

The main contributions of this paper are threefold: (a) we formulate a multi-carrier synchromodal transport re-planning problem that explicitly accounts for terminal service time uncertainty; (b) we develop a collaborative, decentralized decision-making framework that enables carriers to coordinate re-planning actions without a central authority; and (c) we propose a model-assisted

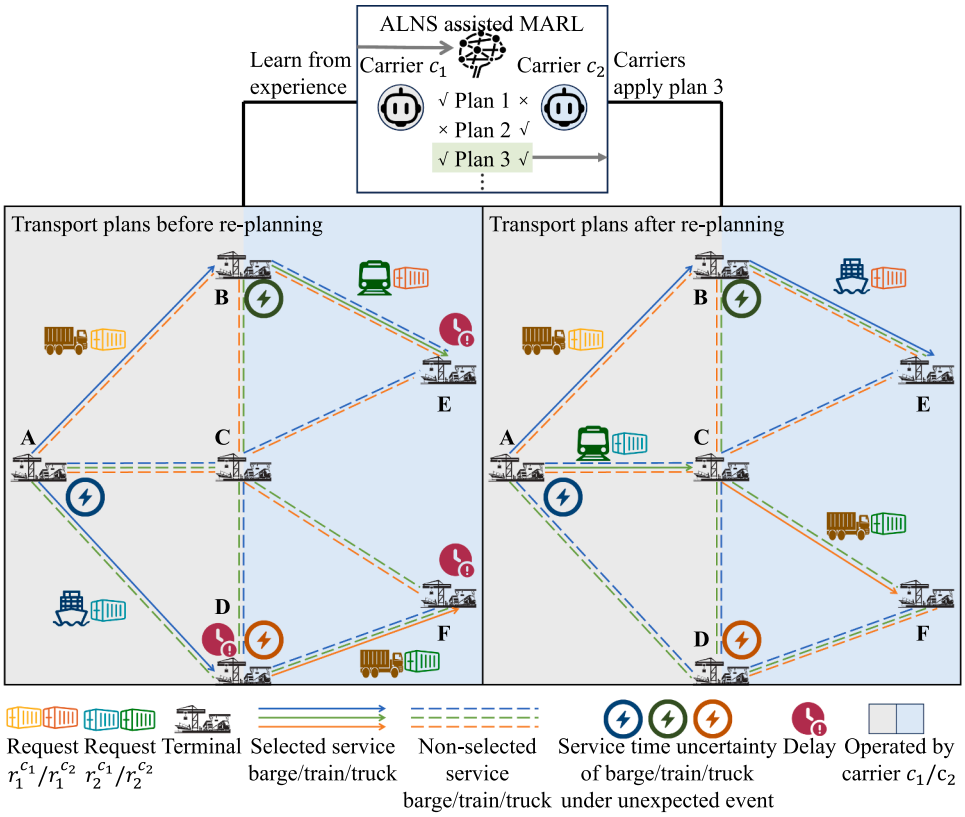


Fig. 1. MARL for synchromodal transport planning under service time uncertainty.

MARL methodology that integrates ALNS-based optimization with reinforcement learning to achieve scalable, feasible, and adaptive decision-making under uncertainty.

The remainder of this paper is organized as follows. Section 2 reviews related work on synchromodal transport planning under uncertainty, multi-carrier cooperation, and learning-based decision methods. Section 3 describes the problem setting. Section 4 formulates the multi-carrier synchromodal re-planning problem under service time uncertainty. Section 5 presents the proposed ALNS-assisted MARL framework, including the re-planning process, the ALNS planning module, the decentralized multi-agent decision-making process, and the learning algorithm. Section 6 reports the computational experiments, including the experimental setup, training dynamics, comparative performance evaluation, and ablation studies. Finally, Section 7 concludes the paper and outlines directions for future research.

2. Literature review

We first review existing studies on synchromodal transport planning under uncertainty, with an emphasis on how disruptions and stochastic service conditions are modeled and managed. We then examine the literature on multi-carrier cooperation in synchromodal systems, focusing on coordination mechanisms and decision structures. Finally, we discuss learning-based approaches for vehicle routing and transport planning under uncertainty, thereby positioning the methodological contribution of this study within the broader literature.

2.1. Synchromodal transport planning under uncertainty

A substantial body of literature has examined synchromodal transport planning under various sources of operational uncertainty. Early studies primarily addressed travel time variability and demand uncertainty, typically through stochastic optimization and rolling-horizon frameworks. For example, Demir et al. (2016) incorporated travel time uncertainty into a stochastic programming model to design reliable and environmentally sustainable intermodal service networks, while Li et al. (2015) employed a receding-horizon control approach to periodically re-optimize intermodal flows under dynamically evolving traffic conditions.

More recent research has shifted attention toward real-time re-planning mechanisms that respond explicitly to operational disruptions. Hrušovský et al. (2021) proposed a simulation-optimization framework that triggers re-planning upon disruption detection

and selects from predefined response strategies, whereas Yee et al. (2021) formulated synchromodal routing as a Markov decision process to derive adaptive routing policies for individual shipments under stochastic travel times. Although these approaches enhance operational robustness, they are often constrained to small-scale problem instances or rely on simplified state representations, which limit their scalability and applicability to complex synchromodal networks.

To further improve adaptability, several studies have explored data-driven and learning-assisted optimization approaches. Filom and Razavi (2025) combined machine learning with robust optimization to quantify travel time uncertainty and support rolling-horizon planning, while Zhang et al. (2023) developed a model-assisted reinforcement learning framework that integrates ALNS to address terminal service time uncertainty without requiring prior distributional assumptions. Despite these advances, the majority of existing methods continue to assume a centralized decision maker with full control over routing and mode-selection decisions. This assumption substantially limits their applicability in realistic synchromodal systems, which are characterized by decentralized control, multiple carriers, and heterogeneous operational objectives.

2.2. Synchromodal transport planning under multi-carrier cooperation

Recognizing the decentralized nature of real-world synchromodal systems, a growing body of research has examined transport planning problems involving multiple carriers. More broadly, synchromodal transport has been widely regarded as an important operational paradigm supporting the vision of the "Physical Internet" (PI) (Ambra et al., 2019; Pan et al., 2025). The PI advocates an open, hyperconnected logistics network in which freight flows can dynamically utilize shared infrastructure and resources across multiple logistics service providers (Ambra et al., 2019). In this vision, ports and terminals act as key hubs where freight is transferred and (re)consolidated, and cross-actor interoperability and track-and-trace capabilities become essential enablers for real-time coordination (Fahim et al., 2021b,a). Within such an open logistics environment, synchromodality provides the operational capability to dynamically switch transport modes and routes in response to real-time conditions, thereby enabling flexible and efficient utilization of network capacity (Lemmens et al., 2019). Consequently, collaboration among multiple carriers becomes a fundamental mechanism for realizing the PI vision, as it allows capacity pooling and coordinated decision-making across otherwise independent logistics actors (Pan et al., 2025).

Some research investigates multi-carrier collaboration through simulation-based approaches. For instance, Alaei et al. (2024) employed an agent-based simulation framework to analyze collaborative and competitive behaviors among logistics service providers governed by predefined interaction rules. Such approaches provide valuable insights into emergent system dynamics, but typically rely on fixed behavioral rules and do not support adaptive policy optimization or learning-based decision-making.

Another stream of research focuses on decentralized or coordinated optimization mechanisms to enable collaboration among carriers. Among these, the auction-based mechanism is among the most widely studied approaches for multi-carrier coordination. For instance, Los et al. (2025) developed a large-scale decentralized multi-agent system based on iterative auctions to enable collaborative vehicle routing. Zhang et al. (2022b) proposed an auction-based collaborative planning framework for intermodal transport, in which a coordinator allocates transport requests based on carriers' bids while considering eco-label preferences. Yuan et al. (2023) highlighted the inherent complexity of transportation service procurement auctions, proposing a two-stage model to align carriers' bidding strategies with their actual truck routing.

While auction-based coordination is highly effective for task allocation and capacity exchange among carriers, its core logic differs from the coordination problem considered in this study. Auction mechanisms typically rely on explicit bidding and winner determination to allocate requests or service bundles, and are therefore well suited to settings in which carriers can evaluate alternatives ex ante through relatively stable pricing rules. By contrast, the present problem is an event-triggered re-planning problem under unknown disruption duration, where the key challenge is not how to price requests, but how multiple carriers can rapidly assess and agree on feasible recovery plans generated under time pressure. For this reason, we adopt a consensus-based mechanism in which ALNS first produces feasible candidate plans, and carriers then make distributed *Accept/Reject* decisions based on local observations and heterogeneous preferences. This design avoids the need for explicit bid design and auction clearing while better matching the real-time and uncertainty-driven nature of collaborative synchromodal re-planning.

Beyond this, a line of research relies on mathematical decomposition or coordination through optimization frameworks. For example, Guo et al. (2024) proposed a coordinated synchromodal planning framework using augmented Lagrangian relaxation, in which incentives are employed to align local carrier decisions with global objectives under travel time uncertainty. While such approaches provide strong theoretical guarantees and coordination efficiency, they typically assume limited adaptability during execution.

Although these approaches effectively address coordination and scalability, they predominantly rely on static or rule-based interaction mechanisms. Carrier strategies are typically determined ex ante and remain fixed during execution, even as operational conditions evolve. Consequently, existing multi-carrier coordination models lack the capability to adapt carrier decision policies dynamically in response to unfolding uncertainty, particularly in the presence of stochastic and terminal, specific service time disruptions. This limitation highlights the need for learning-enabled, decentralized decision frameworks that can support continuous policy adaptation in multi-carrier synchromodal environments.

2.3. Learning-based approaches for vehicle routing problems

Learning-based methods, particularly reinforcement learning (RL), have attracted increasing attention for addressing dynamic routing and sequential decision-making problems under uncertainty. By learning decision policies directly through interaction with

the environment, these approaches circumvent explicit uncertainty modeling and demonstrate strong adaptability in non-stationary and large-scale settings.

Within the VRP literature, early RL-based studies primarily focused on centralized formulations, where routing decisions for multiple vehicles are generated by a single decision-maker. Representative examples include the multi-vehicle routing problem with soft time windows studied by Zhang et al. (2020) and Ren et al. (2022), where multi-agent reinforcement learning architectures are employed to construct vehicle routes offline under centralized control. Similarly, Pan and Liu (2023) proposed a deep RL framework for dynamic and uncertain VRPs, modeling routing decisions as a partially observable Markov decision process to accommodate stochastic demand evolution.

More recent studies have extended learning-based VRP approaches toward decentralized and collaborative settings. Mak et al. (2023) investigated collaborative vehicle routing through a deep multi-agent reinforcement learning framework that jointly addresses route allocation and profit-sharing among carriers, emphasizing fairness and individual rationality. In large-scale contexts, Los et al. (2025) combined multi-agent systems with auction-based mechanisms to enable decentralized coordination among hundreds of carriers in dynamic pickup-and-delivery problems. Related decentralized MARL formulations for VRPs under real-time disturbances and uncertain travel conditions have also been explored by Singh et al. (2023).

In the context of synchromodal and intermodal transport, learning-based approaches remain comparatively limited. Guo et al. (2022) proposed a Q-learning-based framework for global shipment matching under stochastic travel times, demonstrating improved adaptability over stochastic optimization methods. Building on this direction, Zhang et al. (2023) integrated online reinforcement learning with heuristic optimization to enable real-time re-planning under terminal service time uncertainty without relying on prior distributional assumptions.

Despite these advances, most existing learning-based routing studies remain confined to single-carrier or centrally controlled settings, where a unified decision-maker optimizes system-wide objectives. Applications of decentralized learning in multi-carrier synchromodal systems are still scarce. In particular, the joint consideration of heterogeneous carriers, decentralized decision-making, and terminal-level service time uncertainty within a unified learning framework has not yet been adequately addressed. This gap directly motivates the multi-agent, model-assisted reinforcement learning approach proposed in this study.

2.4. Summary

The existing literature on synchromodal transport planning has made substantial progress in addressing operational uncertainty, coordination mechanisms, and adaptive decision-making. As summarized in Table 1, early and representative studies primarily focus on centralized synchromodal transport planning (STP) under travel time or demand uncertainty, using stochastic optimization, robust optimization, rolling-horizon control, or Markov decision processes. While these approaches provide valuable insights into system-level optimization and robustness, they generally assume a single decision-maker with full authority and rely on either prior distributional information or simplified uncertainty representations.

More recent work has incorporated real-time re-planning and learning-assisted optimization, enabling adaptive responses to disruptions without requiring explicit probabilistic assumptions. In particular, model-assisted reinforcement learning approaches have demonstrated strong potential for handling terminal service time uncertainty and improving real-time adaptability. However, as indicated in Table 1, these methods largely remain centralized, with learning and re-planning decisions executed from a system-wide perspective.

Parallel to this stream, a growing body of research has investigated multi-carrier and decentralized decision-making in synchromodal systems. Agent-based simulations, auction-based multi-agent systems, and coordination mechanisms based on augmented Lagrangian relaxation or hierarchical frameworks have been proposed to improve scalability and preserve carrier autonomy. These studies successfully relax centralized control assumptions and enable collaboration among heterogeneous carriers. Nevertheless, carrier strategies in these models are typically static or rule-based, determined ex ante and fixed during execution, even when operational conditions deviate from expectations.

From a learning perspective, reinforcement learning has been widely applied to vehicle routing and dynamic transport problems, demonstrating strong adaptability under uncertainty. Yet, as reflected in Table 1, most learning-based routing studies consider single-carrier or centrally controlled environments, and only a limited number address decentralized decision structures. Critically, the joint consideration of decentralized learning, heterogeneous carriers, real-time re-planning, and terminal-level service time uncertainty remains largely unexplored in the existing literature.

In summary, current studies tend to address uncertainty, decentralization, and learning in isolation. There is a clear gap in methods that simultaneously (i) preserve decentralized carrier autonomy, (ii) enable adaptive policy learning rather than fixed coordination rules, (iii) handle terminal-specific service time uncertainty in real time, and (iv) maintain computational tractability in large-scale synchromodal networks. This gap motivates the ALNS-assisted multi-agent reinforcement learning framework proposed in this study.

3. Problem description

The notation is shown in Table 2. The core challenge of this study lies in establishing a dynamic collaboration mechanism among carriers that enables agents to perform adaptive learning and decision-making in an environment with uncertain service times, thereby enabling dynamic transport re-planning. We define a transport network $G = (N, A)$ with multiple transport modes $w \in W$, in which multiple carriers transport container requests through the network. The network consists of a set of terminals N that provide services such as pickup, transshipment, and delivery. Each carrier $c \in C$ manages a regional subnetwork $G_c \subseteq G$ and operates a subset of

Table 1
Comparison between the proposed model and existing approaches in the literature.

	Problem characteristics					Methodologies				
Article	Mode	Problem	Vehicle routing	Uncertainty	Event location	Approach	Learning	Re-planning	Required prior information	Decision Structure
Synchromodal transport										
Zhang et al. (2023)	road, railway, inland waterway	STP	✓	travel time	arcs	model-assisted RL	✓, online	real-time	none	centralized
Filom and Razavi (2025)	road, railway, inland waterway	STP	✓	travel time	arcs	ML, RO	✓, offline	periodical	historical data	centralized
Demir et al. (2016)	road, railway, inland waterway	STP	✓	travel/service time, demand	arcs	SO	–	–	none	centralized
Alaei et al. (2024)	road, railway, inland waterway	STP	✓	disruption	network	ABS	–	real-time	distribution	decentralized
Hrušovský et al. (2021)	road, railway, inland waterway	ITP	✓	travel/service time	arcs/terminals	SO	–	real-time	none	centralized
Guo et al. (2021)	road, railway, inland waterway	STP	–	travel time, demand	arcs	SO	–	periodical	none	centralized
Yee et al. (2021)	road, railway, inland waterway	STP	–	travel time	arcs	MDP	–	real-time	distribution	centralized
Guo et al. (2022)	road, railway, inland waterway	STP	–	travel time	arcs	RL	✓, offline	periodical	distribution	centralized
Akyüz et al. (2023)	road, railway, inland waterway	ITP	–	disruption	terminals	CG	–	real-time	none	centralized
Zhang et al. (2022b)	road, railway, inland waterway	ITP	✓	–	–	Auction, ALNS	–	–	none	coordinated
Yuan et al. (2023)	road, railway	ITP	✓	demand	–	Auction, HA	–	–	distribution	decentralized
Li et al. (2015)	road, railway, inland waterway	STP	–	–	–	RHC	–	periodical	–	centralized
Guo et al. (2024)	road, railway, inland waterway, maritime	STP	–	travel time	arcs/terminal	ALR, HA, RHF, BS	–	real-time	distribution	coordinated
Vehicle routing problems										
Los et al. (2025)	road	CDPDP	✓	demand	–	Auction-based MAS	–	real-time	none	decentralized
Mak et al. (2023)	road	CVR	✓	–	–	MARL	✓, offline	–	–	coordinated
Zhang et al. (2020)	road	MVRPSTW	✓	–	–	MARL	✓, offline	–	–	centralized
Ren et al. (2022)	road	MVRPSTW	✓	–	–	MARL	✓, offline	–	–	centralized
Singh et al. (2023)	road	VRP	✓	travel time	arcs	MARL	✓, online	real-time	historical data	distributed
Pan and Liu (2023)	road	VRP	✓	demand	–	DRL	✓, online	real-time	historical data	centralized
This research	road, railway, inland waterway	STP	✓	service time	terminals	model-assisted MARL	✓, online	real-time	none	coordinated

–: it means that the relevant item is not mentioned in the article.

Problem Variants:

STP: Synchromodal Transport Planning; ITP: Intermodal Transport Planning; VRP: Vehicle Routing Problem; CVR: Collaborative Vehicle Routing; CDPDP: Collaborative Dynamic Pickup and Delivery Problem; MVRPSTW: Multi-Vehicle Routing Problem with Soft Time Windows.

Methodologies:

RL: Reinforcement Learning; DRL: Deep RL; MARL: Multi-Agent RL; ML: Machine Learning; MDP: Markov Decision Process; SO: Stochastic Optimization; RO: Robust Optimization; CG: Column Generation; ALR: Augmented Lagrangian Relaxation; ABS: Agent-Based Simulation; MAS: Multi-Agent System; RHC/RHF: Receding/Rolling Horizon Control/Framework; HA: Heuristic Algorithm; BS: Beam Search.

Table 2
Notation.

Sets:	
W	Transport modes, indexed by w. $W_c \subseteq W$: transport modes operated by carrier c.
R	Requests, indexed by r. $R_{ue} \subseteq R$: requests affected by unexpected event ue.
C	Carriers, indexed by c.
U	Unexpected events, indexed by ue.
N	Terminals, indexed by i and j. $P/D/T \subseteq N$: pickup, delivery, and transshipment terminals. $O \subseteq N$: depots.
K	Vehicles, indexed by k. $K_{ue} \subseteq K$: vehicles affected by unexpected event ue. $K_c \subseteq K$: vehicles operated by carrier c.
A	Arcs in the transport network. $(i,j) \in A$: arc from terminal i to j. $A_p/A_d \subseteq A$: pickup and delivery arcs.
Parameters:	
u_k	Vehicle k's capacity (TEU).
q_r	Request r's quantity (TEU).
τ_{ij}^k	Travel time (hours) of vehicle k on arc (i,j).
$[a_{p(r)}, b_{p(r)}]$	Request r's pickup time window.
$[a_{d(r)}, b_{d(r)}]$	Request r's delivery time window.
d_{ij}^k	Distance (km) of vehicle k between terminals i and j.
e_k	Vehicle k's CO_2 emission rate (kg per container per km).
c_k^n	Cost (€) parameters of vehicle k: $c_k^1/c_k^{1'}$ transport cost per hour/km per container, c_k^2 handling cost per container, c_k^3 carbon tax coefficient per ton, c_k^4 waiting cost per hour.
c_r^{delay}	Request r's delay penalty per container per hour.
c_r^{storage}	Request r's storage cost per container per hour.
t_{ue}^-/t_{ue}^+	Unexpected event ue's start / end time.
Δt_{ue}	Unexpected event ue's duration time.
Variables:	
y_{ij}^{kr}	1 if request r is serviced by vehicle k traverses arc (i,j), 0 otherwise.
s_{ir}^{kl}	1 if request r transferred from vehicle k to l ($l \neq k$) at terminal i, 0 otherwise.
$t_i^{kr}/t_i^{-kr}/t_i^{+kr}$	Timestamps denoting the arrival, service start, and service completion for request r on vehicle k at terminal i.
t_{ki}^{wait}	Waiting duration of vehicle k at terminal i.
t_r^{delay}	Delay time of request r at its delivery terminal.

transport modes $W_c \subseteq W$ with a fleet of transport vehicles $K_c \subseteq K$, and is responsible for the routing and scheduling decisions of its vehicles. The origin and destination depots for a specific vehicle $k \in K_c$ are represented by terminals $o(k) \in N$ and $o'(k) \in N$, respectively.

Shippers submit transport requests $r \in R$, each with a designated pickup terminal $p(r)$ and delivery terminal $d(r)$. Request r is subject to pickup and delivery windows, defined as $[a_{p(r)}, b_{p(r)}]$ and $[a_{d(r)}, b_{d(r)}]$ at the respective terminals $p(r)$ and $d(r)$. Request r can be served by a set of vehicles K_r, potentially involving transshipment. We distinguish between single-carrier operations (where all $k \in K_r$ belong to a specific carrier c) and collaborative multi-carrier operations. The latter represents the cases when K_r involves vehicles from different carriers, i.e., $|\{c(k) \mid k \in K_r\}| \geq 2$, where $c(k)$ denotes the carrier of vehicle k, that means the request r involves at least two carriers.

During transport execution, unexpected events $ue \in U$ may occur at terminals, affecting service operations over a time interval from the start time t_{ue}^- to the end time t_{ue}^+. When designing the initial transport plan, the occurrence time and duration of unexpected events are unknown, which introduces uncertainty in terminal service times. If a vehicle k encounters an unexpected event ue at terminal i and is unable to transport request r as originally planned, then request r is defined as an *affected request*. Continuing to execute the original transport plan under such conditions may cause the delivery time of request r to exceed the latest allowable delivery time $b_{d(r)}$, thereby incurring a delay penalty.

To avoid delivery delays, it is necessary to re-plan the transport plan when unexpected events occur. This study addresses the following research question: how can multiple agents learn to collaboratively re-plan affected transport requests in real time so as to avoid delivery delays under uncertain service times in a synchromodal transport context?

To illustrate how multiple carriers collaboratively handle service time uncertainty, Fig. 2 presents an example involving two carriers, c_1 and c_2. A transport request r must be shipped from terminal A to terminal E through a network comprising barges, trains, trucks, and a transshipment terminal B. In the initial transport plan (*Plan0*), request r is scheduled to be transported first by *Barge1* operated by carrier c_1, and subsequently by *Train2* operated by carrier c_2. However, due to unexpected events, the service times of *Barge1* and *Train2* become unreliable, resulting in an infeasible schedule that violates the specified time window constraints. Once such a disruption occurs, the two carriers must jointly generate and evaluate alternative feasible transport plans (*Plan1*, *Plan2*, *Plan3*, …) based on their available vehicles and operational constraints. Examples of such alternative plans include:

- Plan1: carrier c_1 selects *Barge2* for the first leg, and the shipment is then transferred to *Truck2* operated by carrier c_2;
- Plan2: carrier c_1 uses *Truck1* for the first leg, and the shipment is subsequently transferred to *Train1* operated by carrier c_2;
- Plan3: carrier c_1 assigns *Truck1*, and the shipment is handed over to *Barge3* operated by carrier c_2;

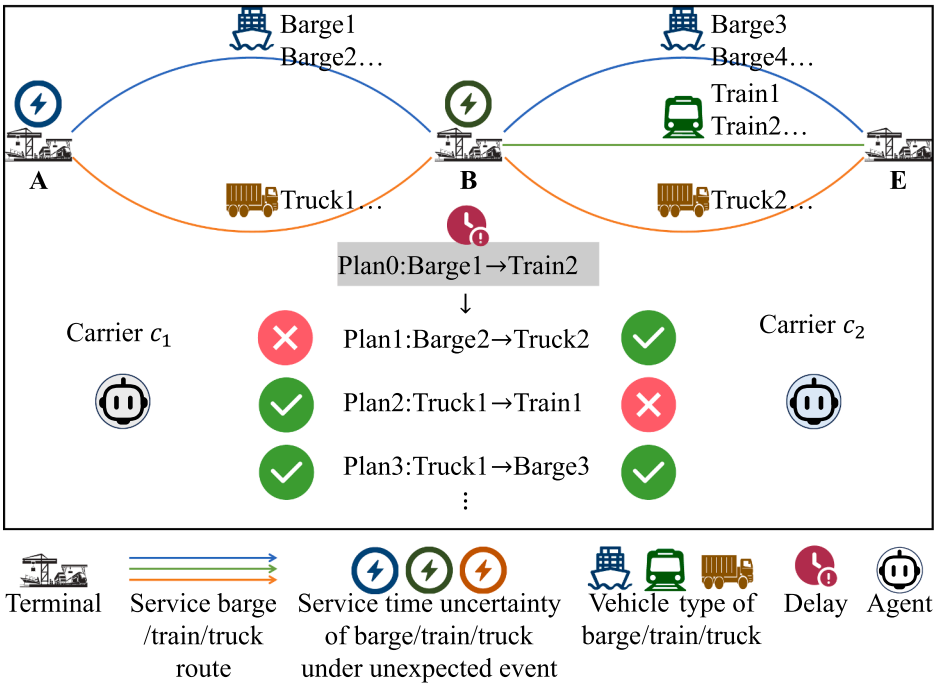


Fig. 2. An illustrative example of the cooperative decision-making process between two carriers.

Each candidate plan is evaluated independently by the two carriers. In Fig. 2, the green and red markers indicate whether a carrier considers a plan feasible or infeasible, respectively. In this illustrative example, for *Plan1*, carrier c_1 rejects the plan because *Barge2* may be affected by an unexpected event occurring at terminal A. For *Plan2*, carrier c_2 rejects the plan since *Train1* is likely to be influenced by the disruption at terminal B. In contrast, for *Plan3*, both carriers c_1 and c_2 consider the plan feasible and therefore accept it. It should be noted that this example is intentionally simplified. In realistic operational settings, each carrier evaluates candidate plans according to its own heterogeneous decision criteria, including operational costs, time-window compliance, resource availability, scheduling preferences, and other internal constraints. As a result, carriers may employ distinct objective functions when assessing the feasibility and desirability of a plan. A candidate plan is accepted only when all involved carriers confirm its feasibility.

The following assumptions are adopted in this study to facilitate model formulation and analysis. First, the physical transport network and terminal infrastructure are assumed to be fixed during the planning horizon, while operational disruptions only affect terminal service times. This assumption simplifies the setting and allows us to focus on terminal service-time uncertainty under a fixed topology. Second, unexpected events occur exogenously and independently of carrier decisions; their occurrence times are unknown at the planning stage but can be detected once they occur. Third, carriers are willing to participate in collaborative re-planning and truthfully evaluate the feasibility of candidate plans, although their feasibility assessments may differ due to heterogeneous constraints. This assumption reflects a cooperative platform or contractual setting in which carriers share only disruption-response information, including affected requests, current route and schedule information, disrupted terminals and modes, and time-window constraints, while carriers retain private cost and preference information. The platform checks shared hard constraints across involved carriers, whereas carrier-specific resource feasibility is locally evaluated and assumed to be reported truthfully; any rejection acts as a veto, after which the process moves to another candidate or falls back to waiting. Strategic misreporting, bargaining, side payments, and coalition formation are beyond the scope of this study.

4. Mathematical model

This section presents a centralized system-level optimization model in Section 4.1, which provides a normative benchmark and motivates the subsequent multi-carrier decomposition in Section 4.2.

4.1. Centralized system-level optimization model

Under a centralized decision-making setting, the objective of the proposed synchromodal transport planning model is to minimize the total system cost. To this end, the total cost function, extended from the model proposed in Zhang et al. (2022a), is formulated

as the sum of six key components: transit cost (F_1), transfer cost (F_2), carbon cost (F_3), waiting cost (F_4), storage cost (F_5), and delay penalty (F_6), as shown in Eqs. (1)–(7).

$$\min F = F_1 + F_2 + F_3 + F_4 + F_5 + F_6 \tag{1}$$

$$F_1 = \sum_{k\in K}\sum_{(i,j)\in A}\sum_{r\in R}(c_k^1\tau_{ij}^k + c_k^{1'}d_{ij}^k)q_r y_{ij}^{kr} \tag{2}$$

$$F_2 = \sum_{k,l\in K,k\neq l}\sum_{r\in R}\sum_{i\in T}(c_k^2 + c_l^2)q_r s_{ir}^{kl} + \sum_{k\in K}\sum_{(i,j)\in A_p}\sum_{r\in R} c_k^2 q_r y_{ij}^{kr} + \sum_{l\in K}\sum_{(i,j)\in A_d}\sum_{r\in R} c_l^2 q_r y_{ij}^{lr} \tag{3}$$

$$F_3 = \sum_{k\in K}\sum_{(i,j)\in A}\sum_{r\in R} c_k^3 e_k q_r d_{ij}^k y_{ij}^{kr} \tag{4}$$

$$F_4 = \sum_{k\in K}\sum_{i\in N} c_k^4 t_{ki}^{\text{wait}} \tag{5}$$

$$F_5 = \sum_{k,l\in K,k\neq l}\sum_{r\in R}\sum_{i\in T} c_r^{\text{storage}} q_r s_{ir}^{kl}(t_i^{-lr} - t_i^{+kr}) + \sum_{k\in K}\sum_{(i,j)\in A_p}\sum_{r\in R} c_r^{\text{storage}} q_r y_{ij}^{kr}(t_i^{-kr} - a_{p(r)}) \tag{6}$$

$$F_6 = \sum_{r\in R} c_r^{\text{delay}} q_r t_r^{\text{delay}} \tag{7}$$

The constraint set of the synchromodal transport planning model follows Zhang et al. (2023). Different from that model, all vehicles in this study transport on predefined routes, and specific vehicle routing decisions are not considered. This assumption reduces model complexity and supports efficient re-planning in multi-carrier collaborative settings.

Although the centralized model provides a normative benchmark, it is not solved directly by the learning agents. Instead, it serves three purposes in the proposed framework: (i) feasibility validation of candidate re-planning solutions, (ii) cost evaluation during ALNS-based plan generation, and (iii) oracle benchmarking for regret-based reward shaping.

4.2. Cost decomposition for multi-carrier heterogeneity

While the objective function (1) minimizes the total generalized cost of the system from a centralized perspective, the transport network is, in practice, operated by a set of carriers $C = \{c_1, \dots, c_n\}$. Each carrier $c \in C$ manages a specific subset of vehicles $K_c \subseteq K$ and seeks to minimize its own operational expenditures rather than the system-wide cost.

To formalize this heterogeneity, the global objective F is decomposed into individual cost functions F_c for each carrier c. The individual cost F_c consists of the direct operational costs incurred by vehicles belonging to carrier c (i.e., transit, carbon, and waiting costs), as well as an allocated share of transfer costs, storage costs, and delay penalties, as shown in Eq. (8):

$$F_c = \sum_{k\in K_c}\big(F_1(k) + F_3(k) + F_4(k)\big) + \sum_{k\in K_c}\alpha_c\big(F_2(k) + F_5(k)\big) + \beta_c F_6, \tag{8}$$

where $F_1(k)$, $F_3(k)$, and $F_4(k)$ denote the transit, carbon, and waiting costs generated by vehicle k, respectively, while $F_2(k)$ and $F_5(k)$ denote the transfer and storage costs associated with vehicle k, respectively. The coefficient α_c specifies the cost allocation rule for transfer and storage costs at transshipment nodes (e.g., costs borne by the receiving carrier or shared among carriers). The term F_6 represents the global delay penalty. Since delay is a service-level metric affecting the entire logistics chain, the coefficient β_c denotes the proportion of the delay penalty allocated to carrier c, reflecting shared responsibility for on-time delivery.

It is important to note that minimizing the global objective F does not necessarily imply minimizing the individual costs F_c for all $c \in C$. A centralized optimal plan may require a specific carrier to incur higher operational costs (e.g., longer waiting times or detours) in order to reduce the overall system delay, thereby creating a conflict of interest. This discrepancy between *global social welfare* (Eq. (1)) and *individual rationality* (Eq. (8)) motivates the need for a multi-agent decision-making framework, as developed in Section 5.

5. Solution framework: ALNS-assisted multi-agent reinforcement learning

Applying RL to synchromodal transport routing poses significant challenges due to the vast and high-dimensional state space, particularly in complex networks involving multiple transport modes and synchronized transshipment operations (Guo et al., 2022). In contrast, the ALNS algorithm, a well-established metaheuristic, exhibits strong search capability and robustness in vehicle routing problems by iteratively constructing and improving feasible solutions through a “destroy-and-repair” mechanism.

To combine the strengths of learning and optimization, Zhang et al. (2023) propose a hybrid framework that integrates RL with ALNS, where RL is used to adaptively select ALNS operators. This integration alleviates, to some extent, the difficulty of policy learning in large state spaces and enhances solution quality under uncertainty. However, the framework is built on a single-agent assumption that treats the intermodal system as a centralized decision-making entity. As a result, it neglects key characteristics of real-world synchromodal systems, including heterogeneous local information, carrier behavior, and the cooperative or competitive interactions among multiple carriers. This centralized design limits the model’s ability to capture realistic multi-party decision-making processes and information asymmetry, and may therefore lead to solutions that deviate from actual operational mechanisms.

These limitations become more pronounced under service-time uncertainty. In practical synchromodal operations, stochastic delays at transshipment terminals are perceived heterogeneously by carriers due to differences in local schedules, resource commitments,

and contractual obligations. In such settings, centralized decision-making not only faces scalability challenges but also fails to capture decentralized information structures and asynchronous carrier responses.

To address service-time uncertainty in multi-carrier synchromodal routing, this study proposes a hybrid framework that integrates MARL with ALNS, building upon the model-assisted learning paradigm introduced in Zhang et al. (2023, 2022a). In the proposed framework, ALNS is responsible for generating candidate transshipment plans, verifying their feasibility, and providing structured cost and feasibility feedback to guide the policy learning and updating process of MARL agents. By leveraging the heuristic guidance of ALNS, the MARL agents can focus on a reduced and decision-relevant action space, thereby maintaining real-time adaptability while significantly improving learning efficiency.

Furthermore, by exploiting the distributed nature of MARL, the proposed framework explicitly captures the cooperative and competitive dynamics among multiple carriers. This enables cross-agent coordination in dynamic re-planning and resource allocation without centralized control. Consequently, the proposed approach demonstrates enhanced adaptability and robustness in large-scale, multi-disturbance, and multi-carrier synchromodal environments, offering a novel technical pathway for intelligent transport decision-making under service-time uncertainty.

5.1. Event-triggered re-planning framework

We develop an event-triggered re-planning framework that accommodates multiple decision-making strategies. These include rule-based heuristics, namely (a) *Waiting* and (b) *Average Duration*, as well as a learning-based approach, referred to as (c) *Model-Assisted MARL*. In addition, a fourth strategy, (d) *Oracle*, is incorporated as a full-information reference: during training, it provides hindsight labels for reward shaping, and during evaluation, it provides an ex post benchmark. The *Oracle* strategy assumes full knowledge of all unexpected events and is therefore not implementable in practice, and it is never used as an input to the deployed MARL policy.

When an unexpected event ue occurs before the scheduled start time of the service ($t_{ue}^{-} < t_i^{-kr}$), the service cannot be started until the event has been resolved. This requirement is enforced by Constraints (9):

$$t_i^{-kr} \geq t_{ue}^{+} \qquad \forall i \in N, \forall k \in K_{ue}, \forall r \in R. \tag{9}$$

Since the event end time t_{ue}^{+} referenced in Constraints (9) is initially unknown, the operations of vehicle k at terminal i for the request become uncertain. Without appropriate action, this uncertainty risks incurring excessive waiting time t_{ki}^{wait}, thereby leading to delivery delay t_r^{delay} at the destination $d(r)$.

Adopting an event-triggered strategy, the framework functions in a dual-phase manner: (i) activation of the *re-planning phase* when an unexpected event occurs at t_{ue}^{-}, followed by (ii) the *learning phase*, which is activated after the event ends at t_{ue}^{+}. To address the research question formulated in Section 3, agents are required to take appropriate actions during the re-planning phase to mitigate potential delivery delays.

Fig. 3 illustrates the overall structure of the proposed event-triggered re-planning framework, highlighting the interaction between unexpected events, re-planning strategies, and the subsequent online learning process.

In the *Waiting* strategy, all vehicles simply wait for the duration of the unexpected event, mimicking traditional operational practices. Once the event ends, affected requests are re-planned only if delays have occurred and re-planning is feasible. The *Average Duration* strategy collects the durations of unexpected events online, then assumes that the duration of the current event is equal to the historical average, which is continuously updated as new observations become available. This strategy reflects experience-based decision-making by carriers, albeit in a simplified manner compared with learning-based approaches. The *Oracle* strategy assumes full knowledge of all unexpected events, including their occurrence times and durations, and is used exclusively to generate oracle solutions that serve as optimal references during MARL training.

The proposed *Model-assisted MARL* strategy integrates real-time re-planning and learning in an event-triggered multi-agent framework. During the *re-planning phase*, when an unexpected event occurs at t_{ue}^{-}, the state is recorded, and ALNS is invoked to generate a set of feasible re-planning candidates. Based on these candidates, MARL agents make cooperative decisions and execute a selected plan under uncertainty about the event duration. Since the consequences of these decisions cannot be fully observed until the event ends, learning is deferred to time t_{ue}^{+}. Once the event finishes and its duration becomes known, the realized outcomes are evaluated, and the corresponding rewards are computed to update the agents' policies in the learning phase.

5.1.1. Re-planning when the unexpected event occurs

The *re-planning phase* is triggered at t_{ue}^{-} when an unexpected event ue affects the service of a specific transport mode w_{ue} at terminal i. As illustrated in Fig. 3, this phase is activated immediately after the event occurrence, where different re-planning strategies can be adopted. Since the duration of the event is initially unknown (except for the *Oracle* strategy used for benchmarking during learning), the system must determine immediate actions to mitigate potential delays. The re-planning process consists of two sequential steps.

Step 1: Identification of affected requests. The first step determines the impact of the disruption. The system identifies the set of potentially affected requests, denoted by R_{ue}. A request r is considered affected if it is currently served by, or scheduled to be served by, a vehicle $k \in K_{ue}$ that passes through terminal i, and its scheduled start time of the service t_i^{-kr} is later than the event start time t_{ue}^{-}. Formally, the affected request set is defined as

$$R_{ue} = \left\{ r \in R \mid \exists k \in K_{ue},\ i \in \text{route}(k) \text{ such that } t_i^{-kr} > t_{ue}^{-} \right\}. \tag{10}$$

If $R_{ue} = \emptyset$, no immediate action is required. Otherwise, the process proceeds to the strategy execution step.

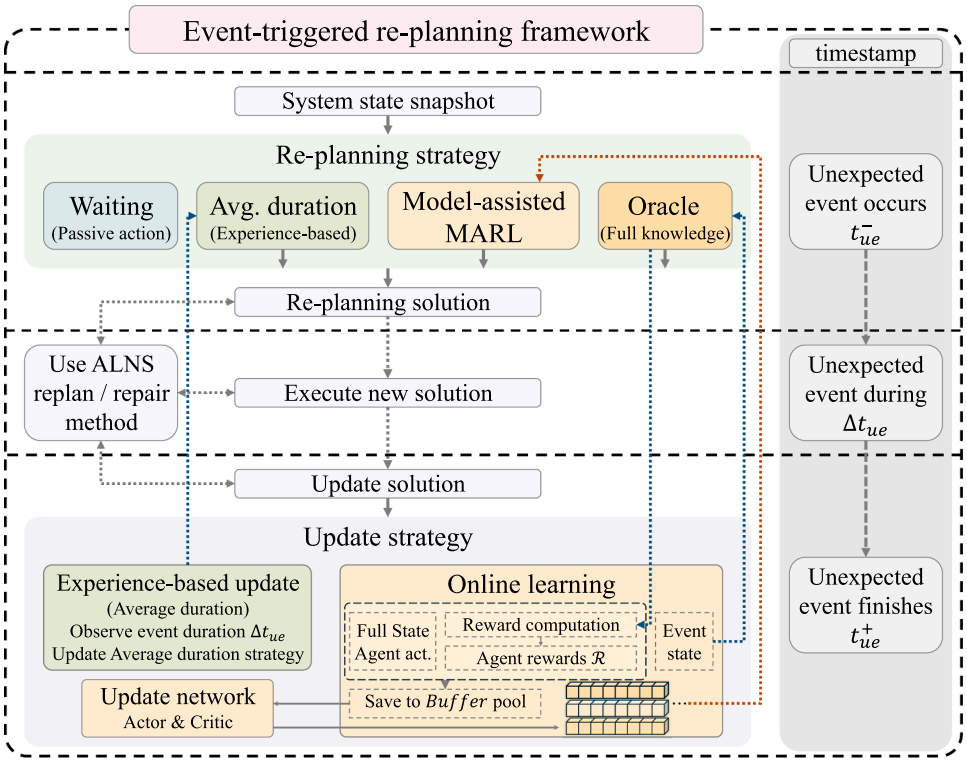


Fig. 3. Event-triggered re-planning framework.

Step 2: Strategy execution. Depending on the selected configuration, the framework executes one of the following strategies to handle the affected request set R_{ue}:

1. Strategy (a): *Waiting*. As outlined in Algorithm 1, this strategy adopts a passive response during the event. No re-planning is performed at time t_{ue}^{-}. Vehicles and requests wait at the affected terminal. After the event ends at time t_{ue}^{+}, re-planning is performed only if delays occur, treating the problem as a deterministic repair task using `ALNS repair method`, as described in Algorithm 7.
2. Strategy (b): *Average Duration*. As shown in Algorithm 2, this strategy estimates the event duration $\bar{d}$ based on historical records $\mathcal{H}_{ue}$. A predicted disruption scenario $\bar{ue}$ is constructed, and the set of potentially delayed requests $R_{\bar{ue}}$ is identified under this assumption. The procedure `ALNS replan method` is then invoked, as described in Algorithm 8, to identify alternative routing plans that minimize the expected cost and delay. This strategy represents experience-based decision-making using statistical information.
3. Strategy (c): *Model-assisted MARL*. As presented in Algorithm 3, this strategy employs a *Propose-and-Review* mechanism. Instead of performing a single optimization step, the ALNS operator generates a set of feasible candidate plans $\mathcal{P}_r$ for each affected request r. The multi-agent system then enters a cooperative decision loop:
   (a) Observation: Each agent i collects its local observation $o_{i,p}$ associated with a candidate plan $p \in \mathcal{P}_r$.
   (b) Cooperation: Based on their policy networks, agents generate actions $a_{i,p} \in \{0, 1\}$ indicating acceptance or rejection of plan p.
   (c) Consensus: If all agents accept, i.e., $a_{i,p} = 1\ \forall i \in \mathcal{N}$, the plan p is executed. Otherwise, the process continues with the next candidate plan until a consensus is reached or the maximum number of cooperation rounds T_r is exceeded.

   To maintain computational scalability, the review loop is invoked only for carriers involved in the candidate plan, so the online carrier-review burden scales with the realized affected subset rather than the full carrier population. For request r, let $\mathcal{P}_r$ denote the candidate-plan set and let $\mathcal{C}_r \subseteq \mathcal{C}$ (equivalently $\mathcal{N}_r \subseteq \mathcal{N}$) denote the participating carriers (agents). Then the *Propose-and-Review* burden has a worst-case upper bound of $O(|\mathcal{P}_r| \cdot |\mathcal{N}_r|)$ if all candidate plans are reviewed by all involved carriers.

   In actual execution, candidate plans are reviewed sequentially and the loop terminates once consensus is reached; hence, the realized number of carrier-plan reviews can be lower than this upper bound. This term should therefore be interpreted as the decentralized review burden after ALNS has generated and screened feasible candidate plans, rather than as the full end-to-end online computational effort. The latter additionally includes affected-request identification, candidate generation, and feasibility checking/candidate screening, whose cost depends on the search budget, the size of the affected request set, and the feasible insertion/search space.

4. Strategy (d): *Oracle*. As described in Algorithm 4, this strategy assumes full knowledge of the true event duration Δt_{ue} at time t_{ue}^{-}. The procedure `ALNS replan method` is applied to compute the ex-post optimal solution. This strategy has two offline roles: during MARL training, it generates hindsight reference values for regret-based reward computation (see Section 5.3.5); in the experiments, it is also reported as an ex post benchmark. It is not available during online deployment.

Algorithm 1: Waiting strategy.

Input: ue, S; **Output:** S; // S represents the solution
if *event ue occurs* **then**
  Maintain current schedules, requests wait at the affected terminal;
end
if *event ue finishes* **then**
  Obtain the affected request set R_{ue} and event duration Δt_{ue};
  $S \leftarrow$ Update the schedules of R_{ue} by adding the event duration Δt_{ue}, and check schedule feasibility;
  Identify $R_{repair} \subseteq R_{ue}$: requests that are infeasible or delayed but still re-plannable;
  if $R_{repair} \neq \emptyset$ **then**
    $S \leftarrow$ using Algorithm 7: ALNS repair method(S, R_{repair});
  end
end

Algorithm 2: Average duration strategy.

Input: ue, S, $\mathcal{H}_{ue}$; **Output:** S; // $S/\mathcal{H}_{ue}$ represents the solution/historical unexpected event log
if *event ue occurs* **then**
  Calculate average duration $\bar{d} \leftarrow \frac{1}{|\mathcal{H}_{ue}|} \sum_{ue \in \mathcal{H}_{ue}} \Delta t_{ue}$;
  Create predicted scenario $\bar{ue}$ assuming event ue lasts for $\bar{d}$;
  $R_{\bar{ue}} \leftarrow$ get will affected request set under scenario $\bar{ue}$;
  if $R_{\bar{ue}} \neq \emptyset$ **then**
    $S \leftarrow$ using Algorithm 8: ALNS replan method(S, $R_{\bar{ue}}$, $\bar{ue}$);
  end
end
if *event ue finishes* **then**
  Obtain the affected request set R_{ue} and event duration Δt_{ue};
  $S \leftarrow$ Update the schedules of R_{ue} by adding the event duration Δt_{ue}, and check schedule feasibility;
  Identify $R_{repair} \subseteq R_{ue}$: requests that are infeasible or delayed but still re-plannable;
  if $R_{repair} \neq \emptyset$ **then**
    $S \leftarrow$ using Algorithm 7: ALNS repair method(S, R_{repair});
  end
  $\mathcal{H}_{ue} \leftarrow \mathcal{H}_{ue} \cup ue$;
end

Compared with auction-based coordination, the proposed *Propose-and-Review* mechanism in Strategy (c) differs in some aspects. Auction-based methods are primarily designed for task or capacity allocation through explicit bidding, and typically rely on predefined bidding strategies together with a winner-determination or clearing procedure. By contrast, the coordination problem considered in this study is an event-triggered re-planning problem under unknown disruption duration, where the key challenge is not how to price requests, but how multiple carriers can rapidly assess and agree on feasible recovery plans under time pressure. For this reason, our framework adopts a Propose-and-Review consensus mechanism: ALNS first generates feasible candidate plans, and the involved carriers then evaluate these plans through distributed *Accept/Reject* decisions based on local observations and heterogeneous preferences. This design avoids the need for explicit bid construction and auction clearing, while making the coordination process more compatible with real-time synchromodal re-planning under service time uncertainty.

5.1.2. Learning when the unexpected event finishes

The learning phase is triggered at time t_{ue}^{+}, when the true duration Δt_{ue} of the unexpected event is revealed. A hindsight evaluation is then conducted by invoking the strategy (d), which yields the ex post optimal solution under the realized disruption. This solution provides a delayed but deterministic benchmark for assessing the quality of the agents' re-planning actions.

Based on *Oracle*, under strategy (c), individual rewards are computed by quantifying the deviation between the *agent-specific outcomes* of the executed plan (including shared delay penalties and private operational costs) and the theoretical baselines, using the

Algorithm 3: MARL cooperation strategy.

Input: ue, S, C, T_r; **Output:** S; // S/T_r represents the solution/maximum cooperation round
if *event ue occurs* **then**
  Identify the request set $\hat{R}_{ue}$ potentially affected by event ue;
  for $r \in \hat{R}_{ue}$ **do**
    $\mathcal{P}_r \leftarrow$ generate the set of feasible alternative transport plans for r using Algorithm 6;
    repeat
      for $p \in \mathcal{P}_r$ **do**
        $p^* \leftarrow p$; $\mathcal{S}_p \leftarrow$ get global state by ALNS;
        $\mathcal{O}_p \leftarrow \{ o_{i,p} \mid i \in C \}$; // observations
        $\mathcal{A}_p \leftarrow \{ a_{i,p} \mid a_{i,p} = \pi_i(o_{i,p}),\ i \in C \}$; // actions
        if $a_{i,p} = 1, \forall a_{i,p} \in \mathcal{A}_p$ **then**
          break; // reach consensus
        end
      end
      $t_r \leftarrow t_r + 1$;
    until $t_r \geq T_r$ ***or*** *reach consensus*;
    if *reach consensus* **then**
      $S \leftarrow$ apply p^* for r;
    else
      Maintain current plan for r (waiting);
    end
  end
end
if *event ue finishes* **then**
  Update each agent i's policy using the approach in Section 5.4;
  Obtain the affected request set R_{ue} and event duration Δt_{ue};
  $S \leftarrow$ Update the schedules of R_{ue} by adding the event duration Δt_{ue}, and check schedule feasibility;
  Identify $R_{repair} \subseteq R_{ue}$: requests that are infeasible or delayed but still re-plannable;
  if $R_{repair} \neq \emptyset$ **then**
    $S \leftarrow$ using Algorithm 7: ALNS repair method(S, R_{repair});
  end
end

reward function defined in Eq. (14). The resulting interaction tuple is then stored in the multi-agent replay buffer to support policy updates. Details of the learning and update procedure are presented in Section 5.4. If no consensus is achieved, the affected request falls back to the waiting strategy.

This delayed decision evaluation pattern follows the event-triggered learning paradigm illustrated in Fig. 3, where policy updates are postponed until the event duration becomes fully observable. In addition, under strategy (b), the historical event log is updated with the newly observed event duration, thereby refining the historical estimate used for future duration predictions.

5.2. The ALNS approach

Within the proposed re-planning framework, as illustrated in Algorithm 5, ALNS serves as the operational engine for exploring the solution space. Given the NP-hard nature of the synchromodal transport planning model defined in Section 4, ALNS is adopted for its ability to efficiently handle complex constraints and to escape local optima through its adaptive destroy-and-repair mechanism based on removal and insertion operators. In the proposed framework, ALNS is used both to generate initial feasible solutions and to perform re-planning when disruptions occur.

To define the search space for the ALNS operator in the synchromodal transport re-planning framework, we categorize re-planning scenarios into two types based on the transport structure:

(a) Direct transport without transshipment: the request is served by a single vehicle and thus by a single carrier from origin to destination. Since this scenario does not involve multi-carrier collaboration and has been extensively studied in previous work (Zhang et al., 2023), it is excluded from the scope of the present multi-agent study.
(b) Collaborative transport with transshipment: the request involves one or more transfers at designated terminals, requiring coordination and cooperation among multiple carriers.

Algorithm 4: Oracle strategy.

Input: ue, S; **Output:** S; // S represents the solution
if *event ue occurs* **then**
  $R_{ue} \leftarrow$ get affected request set under real unexpected event ue;
  if $R_{ue} \neq \emptyset$ **then**
    $S \leftarrow$ using Algorithm 8: ALNS replan method(S, R_{ue}, ue);
  end
end
if *event ue finishes* **then**
  Obtain the affected request set R_{ue} and event duration Δt_{ue};
  $S \leftarrow$ Update the schedules of R_{ue} by adding the event duration Δt_{ue}, and check schedule feasibility;
  Identify $R_{repair} \subseteq R_{ue}$: requests that are infeasible or delayed but still re-plannable;
  if $R_{repair} \neq \emptyset$ **then**
    $S \leftarrow$ using Algorithm 7: ALNS repair method(S, R_{repair});
  end
end

Algorithm 5: Adaptive large neighborhood search.

Input: $I_{\max}$, T_0, η, α; **Output:** S_{best}; // $I_{\max}/T_0/\alpha/\eta/S_{best}$ represents the max iterations/initial temperature/cooling rate/reaction factor/best solution
$S_{curr} \leftarrow$ construct initial solution;
$S_{best} \leftarrow S_{curr}$;
$T \leftarrow T_0$;
Initialize weights $\mathcal{W}^-$ for removal operators Ω^- and $\mathcal{W}^+$ for insertion operators Ω^+;
Initialize historical scores for operators;
Initialize reward parameters $\sigma_1, \sigma_2, \sigma_3, \sigma_4$;
for $k = 1$ **to** $I_{\max}$ **do**
  Select removal operator $\omega^- \in \Omega^-$ using $\mathcal{W}^-$;
  Select insertion operator $\omega^+ \in \Omega^+$ using $\mathcal{W}^+$;
  $S_{temp} \leftarrow S_{curr}$;
  $S_{destroyed}, R_{removed} \leftarrow \omega^-(S_{temp})$;
  $S_{new} \leftarrow \omega^+(S_{destroyed}, R_{removed})$;
  Calculate cost $C(S_{new})$;
  if $C(S_{new}) < C(S_{best})$ **then**
    $S_{best} \leftarrow S_{new}$;
    $S_{curr} \leftarrow S_{new}$;
    $\pi \leftarrow \sigma_1$; // score for new global best solution
  else if $C(S_{new}) < C(S_{curr})$ **then**
    $S_{curr} \leftarrow S_{new}$;
    $\pi \leftarrow \sigma_2$; // score for better solution
  else if $e^{-(C(S_{new})-C(S_{curr}))/T} > random(0,1)$ **then**
    $S_{curr} \leftarrow S_{new}$;
    $\pi \leftarrow \sigma_3$; // score for accepted solution by Metropolis criterion
  else
    $\pi \leftarrow \sigma_4$; // score for rejected solution
  end
  Update historical score of ω^- and ω^+ using π;
  if *the update count reaches the threshold* **then**
    Update weights $\mathcal{W}^-$ and $\mathcal{W}^+$ based on historical scores and η;
  end
  $T \leftarrow \max(T_{\min}, T \times \alpha)$; // update temperature
end

5.2.1. Generation of candidate plans

To define a valid and tractable search space for the ALNS operator, we first categorize re-planning instances based on the spatiotemporal relationship between the request status and the location of the unexpected event. This classification determines the effective *degrees of freedom (DoF)* available for re-planning:

(a) Full flexibility (DoF: Path and Mode): When the request is located at the origin or at a transshipment hub upstream of the disrupted terminal, ALNS is allowed to explore both alternative routing paths (i.e., switching transshipment hubs) and alternative transport modes.
(b) Partial flexibility (DoF: Mode only): When the request is already at the disrupted terminal (e.g., waiting for departure), the physical routing path is fixed by the current cargo location. In this case, ALNS is restricted to searching for alternative transport modes (e.g., switching from barge to truck) or adopting waiting strategies.
(c) Restricted flexibility (DoF: None): When the disruption occurs at the destination terminal while the request is already in transit, the current route cannot be modified.

Guided by these topological constraints, the ALNS operator identifies the *reschedulable segment* of each route according to the following operational rules:

(a) Identification of the immutable past: Route segments that have already been completed or are currently being executed are considered *fixed* and cannot be altered.
(b) Re-planning of the future: Removal and insertion operators are applied only to the unexecuted route segments downstream of the request's current location.
(c) Continuity of stakeholders: Although executed route segments are physically immutable, the carriers that performed them remain *active participants* in the collaborative decision process. Since the final payoff, particularly the global delay penalty, is coupled across the entire logistics chain, these carriers retain their participation rights to protect their vested interests.

Based on this logic, ALNS is employed to iteratively generate a candidate plan set $\mathcal{P}_r$ for evaluation by the MARL agents. As outlined in Algorithm 6, the procedure maintains a temporary *candidate pool* $\mathcal{P}_r$, to which each newly identified feasible route is added. To construct a concise yet representative candidate set $\mathcal{P}_r$, a pruning mechanism based on Pareto dominance is applied. However, due to service time uncertainty, different pruning rules are adopted for *safe paths* and *risky paths*.

Deterministic dominance in safe paths. For candidate plans that successfully bypass the disrupted terminal or switch to an unaffected transport mode, travel times and costs are deterministic. A plan $\mathbf{x}$ is considered dominated by plan $\mathbf{y}$ and is removed from the candidate pool if and only if both plans share the same topological structure (i.e., identical sequences of terminals and modes) and satisfy the following strict dominance condition:

$$\forall i \in \mathcal{N}^{\wedge},\ (C^i_{\mathbf{x}} \geq C^i_{\mathbf{y}}) \wedge (D_{\mathbf{x}} \geq D_{\mathbf{y}}) \quad \wedge \quad \exists j,\ (C^j_{\mathbf{x}} > C^j_{\mathbf{y}}) \vee (D_{\mathbf{x}} > D_{\mathbf{y}}). \tag{11}$$

In this case, plan $\mathbf{x}$ is strictly inferior, as plan $\mathbf{y}$ yields a uniformly better payoff structure.

Uncertainty preservation in risky paths. For routes that traverse the disrupted terminal using the affected transport mode, the realized delay depends on the unknown duration of the unexpected event. Even if a plan $\mathbf{x}$ appears inferior to plan $\mathbf{y}$ under an estimated average event duration, it may still offer latent advantages, such as better alignment with the eventual recovery time window. To avoid premature elimination of potentially valuable options, dominance pruning is suspended for these risky paths, and they are retained in the candidate pool. This allows MARL agents to evaluate the associated risk-reward trade-offs based on their learned policies.

This candidate generation and pruning mechanism ensures that MARL agents are presented with a diverse set of high-quality alternatives, spanning time-optimal, cost-optimal, and risk-aware plans, rather than being restricted to a single deterministic solution. This design reflects a key modeling principle underlying the proposed *model-assisted MARL* framework. Specifically, ALNS is responsible for enforcing structural feasibility and providing high-quality candidate solutions that establish a reliable performance baseline, while MARL focuses on learning strategic decision-making under service time uncertainty by evaluating the risk-reward trade-offs among feasible alternatives. By decoupling structural optimization from policy learning, the proposed framework avoids premature elimination of potentially valuable solutions under uncertainty and enables learning agents to make adaptive, experience-driven choices in complex synchromodal environments. While effective for feasibility preservation and fast online deployment, this separation leaves open the possibility of future extensions in which learning agents partially modify or construct recovery plans directly.

5.2.2. Operational methods: repair and re-planning

Beyond generating candidate pools for MARL agents, the ALNS operator operates in two distinct modes to support baseline strategies and feasibility maintenance within the proposed framework (see Section 5.1).

Algorithm 6: Candidate plan generation.

Input: S, r, ue, $I_{\max}$; **Output:** $\mathcal{P}_r$; // $S/I_{\max}/\mathcal{P}_r$ represents the solution/ max iterations/candidate plan set for request r

Initialize temporary pool $\mathcal{P}_r \leftarrow \emptyset$;
Add current plan of r to $\mathcal{P}_r$;
for $k = 1$ **to** $I_{\max}$ **do**
  Select removal operator ω^- and insertion operator ω^+;
  $S' \leftarrow \omega^+(\omega^-(S, r))$;
  if S' *is feasible* **then**
    Extract new route **x** for r from S';
    Determine if **x** is *Safe* or *Risky*;
    if **x** *is Safe* **then**
      $is\_dominated \leftarrow$ False;
      $\mathcal{D}_r(\mathbf{x}) \leftarrow \emptyset$; // set of plans dominated by **x**
      for **y** $\in \mathcal{P}_r$ **do**
        if **y** *is Safe* ***and*** *shares same topology as* **x** **then**
          if **y** *dominates* **x** *(satisfies Eq. (11))* **then**
            $is\_dominated \leftarrow$ True;
            break;
          else if **x** *dominates* **y** **then**
            $\mathcal{D}_r(\mathbf{x}) \leftarrow \mathcal{D}_r(\mathbf{x}) \cup \{\mathbf{y}\}$;
          end
        end
      end
      if ***not*** $is\_dominated$ **then**
        $\mathcal{P}_r \leftarrow \mathcal{P}_r \setminus \mathcal{D}_r(\mathbf{x})$;
        $\mathcal{P}_r \leftarrow \mathcal{P}_r \cup \{\mathbf{x}\}$;
      end
    else
      $\mathcal{P}_r \leftarrow \mathcal{P}_r \cup \{\mathbf{x}\}$;
    end
  end
end
return $\mathcal{P}_r$;

Reactive repair method for the waiting strategy. This method corresponds to Algorithm 7 and is activated when the resolution of an unexpected event reveals that the original transport plan has become infeasible, for example, due to time-window violations caused by realized delays. The primary objective of this mode is *feasibility restoration* rather than global optimization.

The operator iterates over the set of disrupted requests R_{repair}. For each request, a removal operator is applied to detach it from the infeasible route, followed immediately by a repair attempt using a *greedy insertion operator*. This operator searches for the first feasible insertion position that satisfies all hard constraints while incurring minimal incremental cost. If a feasible insertion is found, the solution is locally updated; otherwise, the removal is rolled back, and the request is marked as temporarily unresolved.

Proactive re-planning method for the Average Duration and Oracle strategies. This method corresponds to Algorithm 8 and serves as the decision engine for deterministic strategies. Its objective is to identify a single best transport plan under a specific assumption on the event duration, either predicted (*Average Duration* strategy) or known ex post (*Oracle* strategy).

For each affected request r, the operator first evaluates the baseline metrics $(C_{\text{wait}}, D_{\text{wait}})$ associated with retaining the current plan. It then generates a set of alternative candidate plans $\mathcal{P}_r$ using the core ALNS search procedure. In contrast to the MARL-based approach, which delegates plan selection to learning agents, this mode applies a strict *lexicographic greedy selection* rule to determine the optimal plan p^*, as defined in Eq. (12):

$$p^* = \arg \min_{p \in \mathcal{P}_r \cup \{\text{current}\}} \{D_p, C_p\}. \tag{12}$$

Specifically, a candidate plan p replaces the current plan if it strictly reduces the delay ($D_p < D^*$), or if it maintains the same minimal delay while reducing cost ($D_p = D^* \wedge C_p < C^*$). This selection rule ensures that the chosen plan is strictly Pareto-improving with respect to the system's primary objective of delivery timeliness.

Algorithm 7: ALNS repair method.

Input: S, R_{repair}; **Output:** S; // S/R_{repair} represents the solution/requests requiring repair
for $r \in R_{repair}$ **do**
  Remove r from its current route in S;
  Find best insertion position for r in S;
  if *feasible insertion found* **then**
    $S \leftarrow$ insert r into the best position;
  else
    Recall removal operation;
  end
end

Algorithm 8: ALNS replan method.

Input: S, R_{ue}, ue; **Output:** S; // S/R_{ue} represents the solution/affected requests by unexpected event ue
for $r \in R_{ue}$ **do**
  Calculate cost C_{wait} and delay D_{wait} for keeping r in current schedule under ue, get current plan p^{curr};
  $\mathcal{P}_r \leftarrow$ generate set of feasible alternative transport plan for r using Algorithm 6;
  $p^* \leftarrow p^{\text{curr}}$, $C^* \leftarrow C_{wait}$, $D^* \leftarrow D_{wait}$;
  for $p \in \mathcal{P}_r$ **do**
    Calculate cost C_p and delay D_p for assigning r to p under ue;
    if $D_p < D^*$ ***or*** $(D_p = D^*$ ***and*** $C_p < C^*)$ **then**
      $p^* \leftarrow p$, $C^* \leftarrow C_p$, $D^* \leftarrow D_p$;
    end
  end
  if $p^* \neq p^{curr}$ **then**
    $S \leftarrow$ apply p^* for r;
  end
end

5.3. Multi-agent decision-making process

This subsection formalizes the decentralized decision-making process underlying the proposed collaborative re-planning framework. It first introduces a decentralized Markov game formulation to characterize agent interactions under partial observability in Section 5.3.1, followed by the specification of agents, state and observation structures, action mechanisms, and the associated reward design in Sections 5.3.2–5.3.5.

5.3.1. Decentralized Markov game formulation

Given the heterogeneous objectives and the potential for non-cooperative behavior among carriers (i.e., each carrier may keep internal operational details private and act autonomously), the problem is formulated as a *partially observable decentralized Markov game (PO-DMG)* (Littman, 1994), defined by the tuple $\mathcal{G} = \langle \mathcal{C}, \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$.

In this study, the term "non-cooperative" is used primarily in an information-structural sense: carriers may not fully share local costs, fleet status, schedules, or feasibility constraints, so coordination must be learned and executed under partial observability with local observations. Accordingly, the adopted binary *Accept/Reject* decision protocol (Section 5.3.4) is not intended to fully capture the strategic richness of non-cooperative bargaining, but rather to represent a lightweight collaboration mechanism over ALNS-generated feasible candidate plans.

Here, $\mathcal{C} = \{1, \ldots, |C|\}$ denotes the set of agents, each representing a carrier. $\mathcal{S}$ is the global state space of the transport network. $\mathcal{O} = \{\mathcal{O}_1, \ldots, \mathcal{O}_{|C|}\}$ represents the set of local observation spaces, where $\mathcal{O}_c$ corresponds to the information available to agent c. $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_{|C|}\}$ denotes the joint action space, with $\mathcal{A}_c$ being the action space of agent c.

The state transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ captures the transport system dynamics and the realization of unexpected events. In the proposed framework, $\mathcal{T}$ is implicitly governed by the heuristic search process of the ALNS operator, which determines the subsequent candidate plan state based on the joint feedback provided by the agents.

The reward structure is defined by $\mathcal{R} = \{R_1, \ldots, R_{|C|}\}$, where R_c specifies the individual reward function of agent c. Finally, $\gamma \in [0, 1)$ is the discount factor that balances immediate and future rewards.

5.3.2. Agents

The set C consists of heterogeneous carriers operating different transport modes. Agents are assumed to be *self-interested but cooperative*: each agent $c \in C$ seeks to maximize its own cumulative reward, which captures a trade-off between private operational costs, such as vehicle usage and waiting costs, and shared service-level performance.

5.3.3. State and observation space

We next define the global state, local observations, global context, and the structured candidate set that together form the information basis for decentralized decision-making.

Global state $\mathcal{S}$. The global state $s_t \in \mathcal{S}$ contains complete information about the transport network at time t, including the status of all transport requests, the realized characteristics of unexpected events, and the full attribute sets of all carriers. This information is accessible only to the centralized critic during the training phase, following the centralized training and decentralized execution (CTDE) paradigm, in order to facilitate stable and efficient learning.

Local observation $\mathcal{O}_i$. Due to information asymmetry and privacy considerations, agent i does not observe the global state directly but instead receives a partial observation $o_{i,t}$. To address the challenge posed by a dynamic decision topology, where the number of feasible re-planning alternatives may vary over time, the observation is structured into two components: a fixed-length global context and a variable-length set of candidate plans. Formally,

$$o_{i,t} = \langle \mathbf{c}_t, \mathcal{X}_t \rangle.$$

Global context $\mathbf{c}_t$. The global context $\mathbf{c}_t$ is a fixed-length feature vector that encodes key attributes of the currently affected request, such as its origin, destination, and time-window constraints, together with publicly observable event information, including the disruption location and the affected transport mode.

Candidate set $\mathcal{X}_t$. The candidate set $\mathcal{X}_t$ represents the collection of feasible re-planning options provided as the decision input to agent i at decision step t. The underlying feasible routes are generated by the ALNS operator for the disrupted request r and constitute a request-level candidate plan set $\mathcal{P}_r$, as defined in Section 5.2.1. This set remains fixed throughout the collaborative decision process for the corresponding request.

To ensure compatibility with neural network architectures, the variable-sized set $\mathcal{P}_r$ is transformed into a fixed-length representation $\mathcal{X}_t$ using standard padding and masking techniques. Formally,

$$\mathcal{X}_t = \{\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \dots, \mathbf{x}_{t,M}\},$$

where M denotes a predefined maximum number of candidate slots. Each element $\mathbf{x}_{t,m}$ corresponds either to a feasible candidate plan in $\mathcal{P}_r$ or to a padded null element, which is masked out during the decision-making process.

Each feasible candidate plan $\mathbf{x}_{t,m}$ is encoded as a feature vector from the perspective of agent i, comprising two components: *shared features*, such as the estimated normalized total cost and feasibility indicators, and *private features*, including the agent-specific cost share, resource compatibility, and role encoding.

5.3.4. Action space

The decision-making process proceeds iteratively. For a proposed target candidate plan $\mathbf{x}_{t,m} \in \mathcal{X}_t$, each agent i selects a binary action $a_{i,t} \in \{0, 1\}$, where $a_{i,t} = 1$ indicates *acceptance* and $a_{i,t} = 0$ indicates *rejection*.

A re-planning proposal is adopted only if a consensus is achieved, i.e., $\forall i \in \mathcal{N},\ a_{i,t} = 1$. If the proposal is rejected, the environment, through the ALNS operator, either presents the next candidate plan or terminates the episode once the maximum number of negotiation rounds is reached.

This binary action space is intentionally aligned with the proposed *Propose-and-Review* consensus mechanism: ALNS proposes feasible candidate plans, and each involved carrier either accepts or rejects a proposal based on its private utility and local feasibility assessment. As a result, strategic effects arising from heterogeneous objectives and private information are captured through decentralized evaluation and veto power, whereas richer strategic behaviors, such as explicit price bids, side payments, coalition formation, or strategic misreporting, are not modeled.

While the binary *Accept/Reject* action design keeps the decision process tractable and guarantees feasibility through the ALNS-generated candidate set, it also limits the agents' proactive planning capability. In particular, agents cannot directly modify candidate plans or construct new recovery paths beyond the proposed alternatives. This design therefore frames MARL as a coordination layer over heuristic plan generation rather than a fully constructive planner. Extending the action space toward partial plan modification or constructive recovery generation is a promising direction for future research.

5.3.5. Reward function

To address the challenges of conflicting interests and coordination in multi-carrier collaboration, we propose a *regret-based hybrid reward mechanism*. The mechanism is designed to achieve two complementary objectives: (1) guiding collective behavior toward the Pareto-optimal frontier with respect to global delay and system cost, and (2) preserving the individual rationality of heterogeneous agents by ensuring local profitability.

For each agent i, let C^i_{wait} denote the cost incurred under the conservative *waiting strategy*. The parameters w^i_d and w^i_c represent agent i's internal preference weights for delay reduction and cost saving, respectively, while $\alpha_i \in [0,1]$ controls the relative emphasis placed on global delay regret versus local cost regret. The tolerance threshold τ_i specifies the level of sub-optimality that agent i is willing to accept before rejecting a proposed plan. The term σ^i_c denotes the cost normalization factor for agent i, and a common delay normalization factor σ_d is used across all agents in the computation of delay regret.

Several global parameters further shape the behavioral incentives of the mechanism. The parameter w_r penalizes prolonged decision-making processes, measured by the number of negotiation rounds. The parameter w_{rej} controls the reward magnitude associated with a *valid rejection*, while w_{irr} penalizes *irrational acceptance*, defined as accepting a plan that is unfavorable according to the agent's own regret evaluation.

When evaluating a candidate plan $\mathbf{x}$ for agent i, let $D_{\mathbf{x}}$ denote the resulting global delay and $C^i_{\mathbf{x}}$ the corresponding local cost incurred by agent i. Let D^* denote the theoretical minimum achievable delay and C^i_* the theoretical minimum achievable cost for agent i within the current feasible search space, as determined by *Oracle*. These theoretical optima are computed offline using full trajectory information and are not accessible to agents during decision-making. They are used exclusively for reward shaping during the training phase. For training, these references are computed per simulated instance in a hindsight manner by using the *Oracle* strategy in Section 5.1.1 together with the deterministic ALNS re-planning method in Section 5.2.2. Concretely, after the event duration is realized in the training simulator, the deterministic full-information procedure evaluates the current plan and the feasible ALNS-generated candidate plans under that realized duration. They are used only to compute regret-based rewards after the realized event duration becomes known. During online execution, agents do not observe D^* or C^i_* and make decisions only from their local observations and ALNS-generated candidates. The experimental Oracle benchmark follows the same full-information procedure defined in Section 5.1.1, whereas the Oracle values here serve as delayed reward-shaping references inside the training simulator. In other words, the training use is label generation for reward computation, while the experimental benchmark use is an ex post comparator reported alongside *MARL*, *Waiting*, and *Average Duration*; neither use gives the deployed MARL policy access to future event information.

Let $\delta_d(\mathbf{x}) = \max(0, D_{\mathbf{x}} - D^*)$ denote the *global delay regret*, which measures the deviation of the delay incurred by plan $\mathbf{x}$, denoted by $D_{\mathbf{x}}$, from the theoretical minimum delay D^*. Similarly, let $\delta^i_c(\mathbf{x}) = \max(0, C^i_{\mathbf{x}} - C^i_*)$ denote the *local cost regret* of agent i, representing the deviation of the local cost $C^i_{\mathbf{x}}$ from the corresponding theoretical minimum C^i_*.

To quantify the degree of dissatisfaction experienced by agent i when evaluating a candidate plan $\mathbf{x}$, we define a *composite regret score* $\Psi_i(\mathbf{x})$ as follows:

$$\Psi_i(\mathbf{x}) = \alpha_i \frac{\delta_d(\mathbf{x})}{\sigma_d} + (1-\alpha_i)\frac{\max(0, C^i_{\mathbf{x}} - C^i_{\text{wait}})}{\sigma^i_c}, \tag{13}$$

where the first term captures regret associated with global delay relative to the theoretical optimum, and the second term captures regret associated with local cost relative to the conservative waiting strategy.

Based on the composite regret score, the reward function $r_{i,t}$ is defined in a piecewise manner, as shown in Eq. (14), to distinguish among three coordination outcomes during the collaborative decision process.

$$r_{i,t} = \begin{cases} \underbrace{w^i_d \cdot \left(1 - \frac{\delta_d(\mathbf{x})}{\sigma_d}\right)}_{\text{Global Alignment}} + \underbrace{w^i_c \cdot \left(1 - \frac{\delta^i_c(\mathbf{x})}{\sigma^i_c}\right)}_{\text{Local Utility}} - \underbrace{w_r \cdot t}_{\text{Efficiency}} & \text{if } \forall j \in \mathcal{N}, a_{j,t} = 1 \\ \underbrace{w_{rej} \cdot \left(\Psi_i(\mathbf{x}) - \tau_i\right)}_{\text{Valid Rejection}} - \underbrace{w_r \cdot t}_{\text{Efficiency}} & \text{if } a_{i,t} = 0 \\ \underbrace{w_{irr} \cdot \left(\tau_i - \Psi_i(\mathbf{x})\right)}_{\text{Irrationality Penalty}} - \underbrace{w_r \cdot t}_{\text{Efficiency}} & \text{if } a_{i,t} = 1 \exists j \neq i : a_{j,t} = 0 \end{cases} \tag{14}$$

The rationale underlying each reward component is summarized as follows.

Consensus reward (global alignment and local utility). When all agents reach consensus on a candidate plan $\mathbf{x}$, agent i receives a reward composed of two parts. The *global alignment* component incentivizes the reduction of delay regret. As $\delta_d(\mathbf{x}) \to 0$, this term approaches its maximum value w^i_d, indicating that the collective decision achieves the theoretically optimal delay performance. The *local utility* component reflects agent i's individual cost consideration. In this case, the theoretical minimum cost C^i_* is used as the reference baseline for computing the local cost regret $\delta^i_c(\mathbf{x})$, ensuring that consensus decisions remain economically acceptable for each participating carrier.

Valid rejection bonus. If agent i rejects a proposed plan, i.e., chooses the *Reject* action ($a_{i,t} = 0$), the reward depends on the quality of the rejected plan. When the composite regret score satisfies $\Psi_i(\mathbf{x}) > \tau_i$, the rejection is considered rational, and the agent receives a positive reward proportional to the regret magnitude. This mechanism encourages agents to actively filter out inferior proposals while compensating for the additional negotiation time required to search for better alternatives.

Irrationality penalty. In the event of a failed consensus where agent i chooses the *Accept* action ($a_{i,t} = 1$) but the proposal is rejected by other agents, the rationality of agent i's action is evaluated. If the candidate plan is in fact unfavorable to agent i, i.e., $\Psi_i(\mathbf{x}) > \tau_i$, a negative reward is imposed. This penalty discourages blind acceptance and free-riding behavior, forcing agents to adhere to their own rationality constraints even when attempting to facilitate cooperation.

Efficiency penalty. To discourage excessive negotiation and promote timely convergence, a linear time penalty $w_r \cdot t$ is applied at the system level, where t denotes the current negotiation round. This term balances solution quality against decision efficiency.

To capture carrier heterogeneity, parameters such as w_d^i, w_c^i, σ_c^i, and τ_i are agent-specific. For example, a time-sensitive truck carrier is characterized by a higher w_d^i, whereas a cost-sensitive barge operator typically exhibits larger values of w_c^i and σ_c^i, reflecting its stronger emphasis on cost efficiency.

5.4. Learning algorithm: MADDPG

To solve the formulated PO-DMG, we employ the Multi-agent Deep Deterministic Policy Gradient (MADDPG) algorithm (Lowe et al., 2017), as summarized in Algorithm 9. MADDPG is adopted for its effectiveness in multi-agent settings and its ability to stabilize learning under the *CTDE* paradigm.

5.4.1. CTDE framework

In the proposed framework, each carrier agent i is equipped with two neural networks: an *actor network* $\mu_i(o_i; \theta_i^\mu)$, which maps local observations to actions, and a *critic network* $Q_i(\mathbf{s}, a_1, \dots, a_N; \theta_i^Q)$, which evaluates the joint action value using global information. Here, $\mathbf{s}$ denotes the global state features accessible during training.

Under the *CTDE* paradigm, the critic networks are allowed to access the observations and actions of all agents during training in order to mitigate the non-stationarity induced by concurrent policy updates. During execution (deployment), only the actor networks are used, thereby ensuring fully decentralized decision-making.

Standard MADDPG assumes differentiable action spaces in order to compute policy gradients, whereas the proposed re-planning problem involves discrete binary actions (*Accept/Reject*). To bridge this gap, we employ the *Gumbel–Softmax estimator* to obtain a differentiable approximation of discrete action sampling.

Instead of producing a hard discrete action, the actor network outputs a continuous probability distribution over the action space. Let $\mathbf{h}_i$ denote the logits produced by the actor of agent i. An action sample is generated using

$$\mathbf{a}_i = \mathrm{softmax}\left(\frac{\mathbf{h}_i + \mathbf{g}}{\tau_{\mathrm{temp}}}\right), \tag{15}$$

where $\mathbf{g}$ is a vector of independent samples drawn from the Gumbel$(0, 1)$ distribution, and τ_{temp} is the temperature parameter controlling the degree of smoothness in the approximation. As τ_{temp} decreases, the sampled action distribution becomes increasingly peaked and approaches a one-hot representation.

This continuous relaxation enables gradients to be backpropagated from the critic to the actor during training, while allowing discrete actions to be recovered during execution.

5.4.2. Network updates

The networks are trained using a replay buffer D that stores transition tuples of the form $(\mathbf{s}, \mathbf{a}, \mathbf{r}, \mathbf{s}')$, where $\mathbf{s}$ and $\mathbf{s}'$ denote the current and next global states, respectively.

Critic update. Each critic network is updated by minimizing the Bellman temporal-difference (TD) error:

$$\mathcal{L}(\theta_i^Q) = \mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{r},\mathbf{s}')\sim D}\left[\left(y_i - Q_i(\mathbf{s}, a_1, \dots, a_N)\right)^2\right]. \tag{16}$$

The target value y_i is computed using target networks as

$$y_i = r_i + \gamma\, Q_i'\left(\mathbf{s}', a_1', \dots, a_N'\right)\Big|_{a_j' = \mu_j'(o_j')}. \tag{17}$$

Actor update. Each actor network is updated using the deterministic policy gradient:

$$\nabla_{\theta_i^\mu} J \approx \mathbb{E}_{\mathbf{s}\sim D}\left[\nabla_{\theta_i^\mu} \mu_i(o_i) \nabla_{a_i} Q_i(\mathbf{s}, a_1, \dots, a_N)\Big|_{a_i = \mu_i(o_i)}\right]. \tag{18}$$

Target network update. To stabilize training, target networks are maintained for both actor and critic networks. Their parameters are updated via soft updates:

$$\theta_i' \leftarrow \tau\theta_i + (1 - \tau)\theta_i', \tag{19}$$

where $\tau \in (0, 1)$ is the update rate.

Algorithm 9: MADDPG for collaborative re-planning.

Input: Batch size B, max episodes M, max steps T; **Output:** Trained policies $\mu_1, \dots, \mu_{|C|}$;
Initialize Replay Buffer D
Initialize Actor networks μ_i and Critic networks Q_i with weights $\theta_i^{\mu}, \theta_i^{Q}$ for each agent $i \in C$
Initialize Target networks μ_i', Q_i' with weights $\theta_i^{\mu'} \leftarrow \theta_i^{\mu}, \theta_i^{Q'} \leftarrow \theta_i^{Q}$
for *episode* $= 1$ **to** M **do**
  Receive initial global state $\mathbf{s}_1$ from ALNS
  for *step* $t = 1$ **to** T **do**
    for *each agent* $i \in C$ **do**
      Receive local observation $o_{i,t}$
      Select action $a_{i,t} = \text{Gumbel-Softmax}(\mu_i(o_{i,t}; \theta_i^{\mu}), \tau_{temp})$
    end
    Execute joint action $\mathbf{a}_t = (a_{1,t}, \dots, a_{|C|,t})$
    Observe reward $\mathbf{r}_t$ and new global state $\mathbf{s}_{t+1}$
    Store transition $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{r}_t, \mathbf{s}_{t+1})$ in D
    $\mathbf{s}_t \leftarrow \mathbf{s}_{t+1}$
    for *each agent* $i \in C$ **do**
      Sample minibatch of B samples $(\mathbf{s}^j, \mathbf{a}^j, \mathbf{r}^j, \mathbf{s'}^j)$ from D
      Set $\mathbf{a'}^j = (\dots, \text{Gumbel-Softmax}(\mu'_k(o'^j_k)), \dots)$
      Set $y^j = r_i^j + \gamma Q_i'(\mathbf{s'}^j, \mathbf{a'}^j)|_{\theta_i^{Q'}}$
      Update Q_i by minimizing loss: $L(\theta_i^{Q}) = \frac{1}{B}\sum_j (y^j - Q_i(\mathbf{s}^j, \mathbf{a}^j))^2$
      Update μ_i using the sampled policy gradient: $\nabla_{\theta_i^{\mu}} J \approx \frac{1}{B}\sum_j \nabla_{\theta_i^{\mu}} \mu_i(o_i^j) \nabla_{a_i} Q_i(\mathbf{s}^j, \mathbf{a}^j)|_{a_i = \mu_i(o_i^j)}$
    end
    Update target parameters: $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$
    if *consensus reached* ***or*** *candidate pool exhausted* **then**
      break loop
    end
  end
end

6. Computational experiments

This section presents a computational assessment of the proposed *MARL* framework for synchromodal transport re-planning under service time uncertainty. The experiments are designed to examine three key aspects: (i) the learning stability and convergence behavior of the framework, (ii) its effectiveness and robustness relative to benchmark strategies across different disruption scenarios and problem scales, and (iii) the structural necessity and behavioral mechanisms underlying the proposed multi-agent architecture.

To this end, we first describe the experimental setup in Section 6.1, including the simulation environment, agent configuration, and scenario generation process. Section 6.2 then examines the training dynamics and convergence behavior of the MARL agents. Subsequently, Section 6.3 presents a comprehensive comparative analysis to evaluate the robustness and scalability of the proposed framework against multiple benchmarks. Section 6.4 further reports supplementary robustness tests under distributional shifts and network-graph perturbations. Section 6.5 presents a series of ablation studies aimed at isolating the contributions of architectural decomposition, agent heterogeneity, and preference allocation mechanisms. Section 6.6 examines the sensitivity of the proposed reward mechanism to hyperparameter perturbations and the impact of negotiation round limits on coordination efficiency. Finally, Section 6.7 assesses the scalability of the framework through controlled extensions involving more carriers and transshipments.

6.1. Experimental setup

The experimental setup is designed to reflect a realistic synchromodal transport environment while enabling a controlled evaluation of collaborative re-planning under uncertainty. The following subsections describe the simulation environment, agent configuration, scenario generation process, and the classification of problem instances according to their improvement potential.

6.1.1. Simulation environment

The experimental evaluation is conducted on a realistic synchromodal transport network inspired by the European Gateway Services (EGS) network in the Rhine-Alpine corridor (EGS, 2021). To focus on collaborative dynamics and re-planning efficiency, we construct a representative sub-network comprising six key terminals: the deep-sea terminal *Rotterdam* and five inland terminals (*Venlo, Duisburg, Neuss, Dortmund, and Nuremberg*), as shown in Fig. 4. The network provides a total of 52 transport services, including 17 barge services, 25 train services, and 10 truck services, based on publicly available information from the EGS website. Barge and

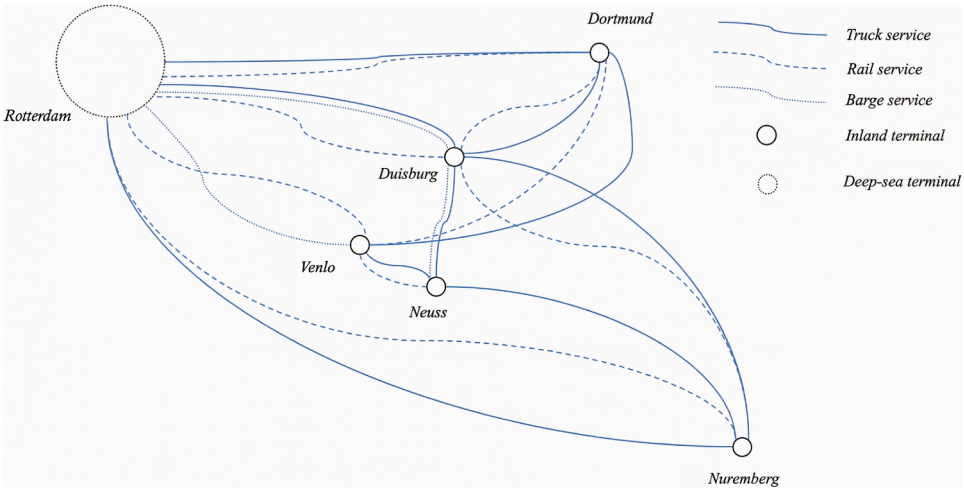


Fig. 4. Transport network of EGS, source: Zhang et al. (2022c).

Table 3
Heterogeneous preference profiles and parameters for agents.

Parameter	Symbol	Agent 1	Agent 2
Primary Mode	–	Barge / Truck	Train / Truck
Service Scope	–	Port → Hubs	Hubs → Hinterland terminals
Delay Weight	w_d^i	10.0	**20.0**
Cost Weight	w_c^i	**2.0**	1.0
Preference Factor	α_i	0.3	0.7
Cost Normalization (€)	σ_c^i	2000	4000
Delay Normalization (h)	σ_d	10.0	10.0
Rejection Bonus	w_{rej}	2.0	2.0
Irrationality Penalty	w_{irr}	2.0	2.0
Tolerance Threshold	τ_i	0.2	0.2

train services operate according to fixed schedules derived from real-world operations. In contrast, truck services are modeled as a flexible fleet with unlimited capacity, offering direct connections between any pair of terminals and serving as a reliable backup transport option.

6.1.2. Agent configuration

To simulate a realistic multi-actor logistics chain with structural dependencies and potentially conflicting interests, the transport network services are partitioned between two distinct agents. This division of operational responsibilities creates an inherent need for collaboration, as requests destined for deep hinterland terminals (e.g., Nuremberg) necessarily require transshipment and coordinated decision-making between the two agents.

The agents are configured with distinct operational roles and preference profiles as follows:

Agent 1 (Seaport-hinterland operator). This carrier manages the primary transport legs originating from the deep-sea terminal, operating connections from *Rotterdam* to intermediate hubs (*Venlo*, *Duisburg*, and *Neuss*). Its primary transport modes are *barge* and *truck*. Reflecting the characteristics of bulk transport in the early stages of the supply chain, Agent 1 is modeled as *cost-sensitive*. Its objective is to minimize operational costs and it exhibits a relatively higher tolerance for delay in order to utilize lower-cost transport modes such as barges.

Agent 2 (Inland distribution operator). This carrier manages the downstream distribution legs within the hinterland, operating connections from intermediate hubs (*Venlo* and *Duisburg*) to inland terminals (*Neuss*, *Dortmund*, and *Nuremberg*). Its primary transport modes are *train* and *truck*. Given its focus on regional distribution and final delivery, Agent 2 is modeled as *time-sensitive*. It prioritizes service reliability and strict adherence to delivery deadlines, and is willing to incur higher operational costs to avoid delay penalties.

The specific parameter settings of the regret-based reward function, defined in Section 5.3, are summarized in Table 3. These parameters quantitatively encode the heterogeneous behavioral profiles described above. In particular, the cost normalization factor σ_c^i differs across agents to reflect the distinct operational cost scales associated with their respective transport modes.

Table 4
Parameter settings for unexpected event scenarios.

Scenario Class	Mean Duration (μ)	Std. Dev (σ)
Type I (Minor)	5 h	{1, 10, 20}
Type II (Moderate)	20 h	{1, 10, 20}
Type III (Severe)	40 h	{1, 10, 20}
Type IV (Extreme)	80 h	{1, 10, 20}

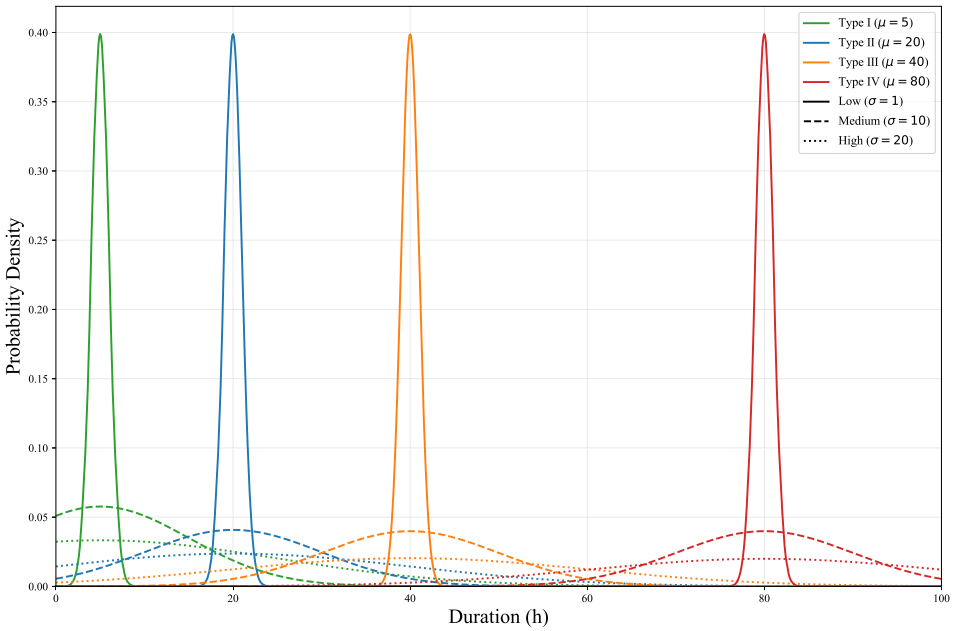


Fig. 5. Scenarios comparison.

6.1.3. Scenario generation

To assess the scalability and robustness of the proposed approach, we create experimental instances with different levels of demand, namely $|R| \in \{5, 20, 50, 100\}$ shipment requests. Request origins and destinations are randomly selected from the set of deep-sea and inland terminals. The volume of the container q_r of each request is evenly distributed in the interval $[10, 30]$ TEUs. The earliest pickup time $a_{p(r)}$ is sampled uniformly from $[1, 120]$ h. The latest delivery time is defined as $b_{d(r)} = a_{p(r)} + LD_r$, where the lead time LD_r is independently drawn from $\{24, 48, 72\}$ h with probabilities $\{0.15, 0.6, 0.25\}$, respectively, following the experimental settings in Zhang et al. (2023).

Unexpected events (e.g., terminal congestion or equipment failures) are simulated using a stochastic generation mechanism. Each unexpected event ue is characterized by a tuple $\langle l_{ue}, w_{ue}, t^{-}_{ue}, \Delta t_{ue} \rangle$, representing the event location, affected transport mode, start time, and duration, respectively. Among these elements, the event duration Δt_{ue} constitutes the primary source of uncertainty and is modeled using a *truncated normal distribution*:

$$\Delta t_{ue} \sim \mathcal{N}_{\text{trunc}}(\mu, \sigma^2; a = 0, b = +\infty), \tag{20}$$

where μ denotes the expected severity of the disruption and σ represents the level of uncertainty. A full factorial experimental design is employed with four levels of severity ($\mu \in \{5, 20, 40, 80\}$ h) and three levels of predictability ($\sigma \in \{1, 10, 20\}$), resulting in 12 distinct disruption scenario classes, as summarized in Table 4. Fig. 5 further illustrates the probability distributions of event durations under different combinations of severity (μ) and uncertainty (σ), providing an intuitive comparison of disruption intensity and variability across scenario classes. For each scenario, the event location l_{ue} and the affected transport mode w_{ue} are randomly sampled from the set of available services.

In the experiment, the size of the review loop is small and relatively stable. For each affected request, the ALNS-generated candidate set typically contains about 6–7 plans. This candidate-set size is mainly determined by the available transport-service alternatives in the network and the generation rules defined in Section 5.2.1, rather than by the disruption duration. In terms of affected-carrier counts, the baseline collaborative setting involves two agents, while the controlled scalability extensions involve at most four agents in the overlapping-responsibility setting and at most three agents in the multi-layer transshipment setting. Since the review loop is invoked only for carriers involved in the candidate plan, the realized number of carriers can be smaller than these configuration-level upper bounds.

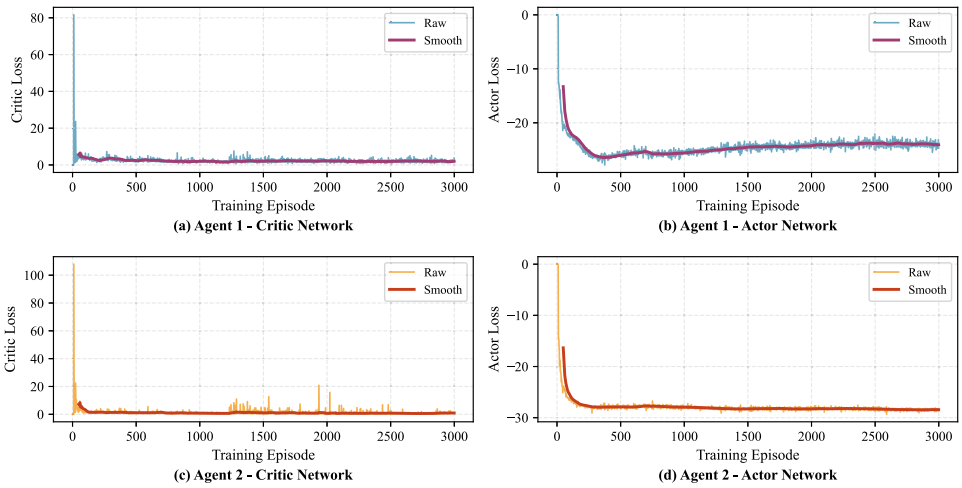


Fig. 6. Training dynamics of the MADDPG agents. (a) and (c) illustrate the rapid decline of critic loss, indicating that the value function approximation stabilizes. (b) and (d) show the actor loss converging to a stable value, reflecting the policy settling into an equilibrium.

6.1.4. *Characterization of improvement potential*

A granular performance analysis requires distinguishing between trivial instances and those that genuinely require intelligent intervention. To this end, we categorize problem instances by comparing the passive outcome under the waiting strategy (D_{wait}, C_{wait}) with the theoretical optimum (D^*, C^*).

Based on this comparison, instances are classified into five scenario types. *Scenario 1 (Wait optimal)* and *Scenario 5 (Worst-case optimal)* correspond to boundary cases in which the status quo cannot be improved through re-planning and therefore serve as reference conditions. In *Scenario 1*, disruptions are either negligible or do not affect downstream feasibility, such that the passive *Waiting* strategy can remain globally optimal. In contrast, *Scenario 5* represents irrecoverable disruption cases where the disruption impact is unavoidable and fully propagated, and any re-planning attempt can only incur additional cost or delay without improving feasibility. *Scenario 2 (Cost reducible without delay)* captures situations where disruptions do not induce delivery delays, yet re-planning can yield economic benefits, thereby testing agents' sensitivity to operational cost reductions. Crucially, *Scenario 3 (Avoidable delay)* represents recoverable disruptions in which the waiting strategy leads to delays ($D_{wait} > 0$), while an alternative plan exists that can eliminate them ($D^* < D_{wait}$). This category serves as the primary indicator of the system's resilience and adaptive decision-making capability. Finally, *Scenario 4 (Cost reducible with delay)* encompasses severe disruptions for which delays cannot be entirely eliminated but can be partially mitigated through re-planning.

Implementation details of the procedure are provided in Appendix A.

6.2. *Training dynamics and convergence*

We evaluate the learning stability and convergence behavior of the proposed *MARL* framework over 3000 training episodes. To facilitate a focused analysis of the learning dynamics, the experiments in this subsection are conducted under a representative medium-scale setting with $|R| = 20$ shipment requests per episode. Unexpected events are generated according to a truncated normal distribution with parameters $\mu = 20$ h and $\sigma = 10$ h. The performance metrics reported below are aggregated over 1000 independent realizations of unexpected events in order to ensure statistical reliability and to mitigate the influence of stochastic variability.

6.2.1. *Algorithmic stability*

Fig. 6 illustrates the training loss trajectories of both the critic and actor networks.

As observed, the *critic loss* (Fig. 6(a) and (c)) decreases rapidly during the initial training phase (episodes 0-500) and subsequently stabilizes at a low level. It is worth noting that, due to the inherent stochasticity in event durations, the critic loss does not converge to zero but instead approaches the residual variance induced by environmental uncertainty.

In parallel, the *actor loss* (Fig. 6(b) and (d)) stabilizes after approximately 1,000 training episodes, indicating convergence of the policy updates. This behavior confirms that the Gumbel–Softmax estimator successfully enables gradient backpropagation for the discrete *Accept/Reject* decision process.

While loss trajectories reflect optimization stability, the evolution of cumulative episode rewards provides a more direct assessment of policy quality. Owing to stochastic disruptions and exploration noise, reward signals exhibit natural variance; nevertheless, a clear and consistent upward trend is observed throughout training. Detailed reward convergence curves for both agents are reported in Appendix B.

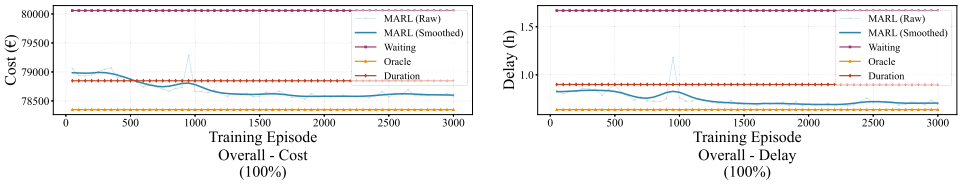


Fig. 7. Evolution of global cost and delay during training (overall average). The MARL agent (blue curve) starts with random exploration and gradually converges towards the oracle bound (orange line), significantly outperforming the waiting baseline (purple line) and duration benchmark (red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

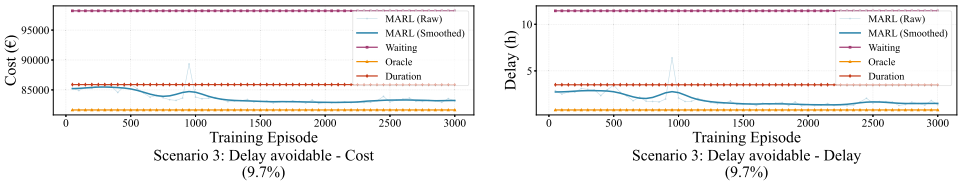


Fig. 8. Performance details in scenario 3 (avoidable delay). the *MARL* framework demonstrates robust performance, achieving a balanced delay–cost trade-off that consistently outperforms the heuristic baseline (duration) and approaches the theoretical lower bound defined by the oracle.

6.2.2. Convergence of efficiency

Fig. 7 illustrates the evolution of the system-level performance metrics, total generalized cost and total delay, averaged across all scenarios.

In the figure, the horizontal lines correspond to strategies (*Waiting, Duration*, and *Oracle*), whereas the fluctuating curve represents the learning trajectory of the *MARL* agents.

Delay reduction (right panel): The average delivery delay achieved by the *MARL* framework decreases substantially from the initial exploration phase and converges to approximately 0.68 h. This result markedly outperforms the *Waiting* baseline (1.67 h) and the *Duration* benchmark (0.90 h) and closely approaches the theoretical lower bound defined by the *Oracle* (0.64 h).

Cost optimization (left panel): At the same time, the total generalized cost stabilizes at around €78,560. Although this value is slightly higher than the *Oracle* bound, it remains highly competitive, indicating that the agents have learned to effectively reduce delay without incurring excessive operational costs.

While these aggregated results demonstrate strong overall performance, averages across all scenarios may obscure the framework's effectiveness in critical situations. To more clearly reveal the agents' decision-making intelligence, we next focus on *Scenario 3 (Avoidable Delay)*, a challenging setting in which proactive re-planning is essential to prevent significant delivery delays.

As illustrated in Fig. 8, Scenario 3 represents a setting in which delivery delays are avoidable through proactive re-planning, albeit at the expense of increased operational cost. The *Waiting* strategy results in the highest total generalized cost among all benchmarks. Although it avoids proactive re-planning actions, the excessive delivery delay of 11.44 h leads to substantial delay penalties, which dominate the overall cost. The *MARL* framework agents learn to resolve this trade-off in a rational and economically consistent manner. As shown in the right panel of Fig. 8, the delay under the *MARL* framework decreases rapidly during training and converges to a low and stable level. Although this delay is slightly higher than the theoretical lower bound achieved by the *Oracle* (0.81 h), it substantially outperforms the heuristic *Duration* benchmark (3.51 h) and avoids the severe delays associated with *Waiting*.

In the left panel of Fig. 8, the cost under *MARL* strategy stabilizes at a level significantly lower than the *Waiting* and *Duration* strategies, while remaining within a small margin of the *Oracle* strategy. This behavior indicates that the agents deliberately accept a moderate increase in operational cost, such as switching to faster transport modes, in order to achieve a disproportionately larger reduction in delay penalties.

Importantly, the learned policy does not blindly favor the fastest transport option. Instead, it reflects a balanced delay–cost trade-off, in which marginal increases in transport cost are justified by substantial reductions in delay-related penalties, resulting in a Pareto-efficient outcome.

This behavior validates the effectiveness of the proposed regret-based hybrid reward mechanism, which successfully aligns agents' local objectives with the global system goal. By explicitly internalizing both delay regret and cost regret, the reward design incentivizes agents to deviate from the "cheap but slow" waiting option when delay costs dominate re-planning expenses.

Scenario 3 represents the most challenging decision-making case, as it requires resolving conflicting cost-delay trade-offs under uncertainty. For completeness, the convergence behavior and comparative performance results for the remaining scenarios are provided in Appendix C.

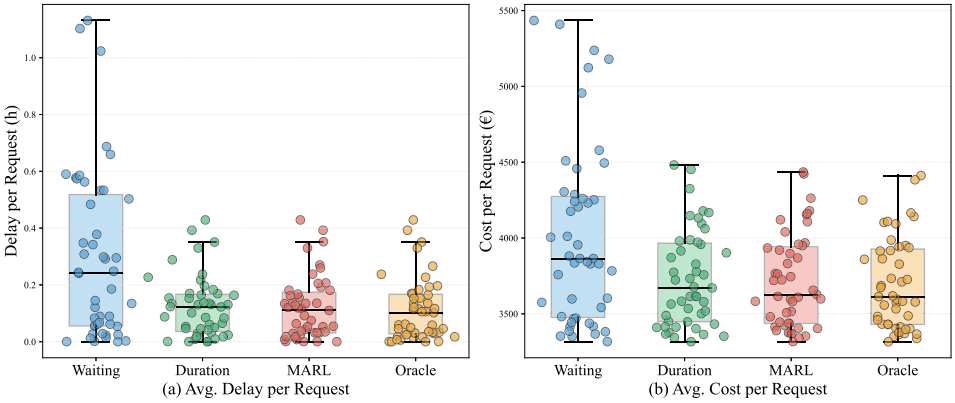


Fig. 9. Overall performance comparison normalized by order size across all test instances.

6.3. Comparative performance evaluation

6.3.1. Robustness and scalability analysis

To assess the generalization capability of the proposed framework across different problem scales ($|R| \in \{5, 20, 50, 100\}$) and disruption intensities (μ, σ), we analyze the distributions of total cost and delay normalized by the number of requests. This normalization removes scale-induced bias and enables a fair comparison of algorithmic stability across scenarios. Fig. 9 presents the comparative results using boxplots, where the dispersion reflects the variability and reliability of each strategy.

As illustrated in Fig. 9(a), in terms of delay-handling performance, the *Waiting* strategy exhibits the largest interquartile range and a substantial number of extreme outliers, confirming that passive waiting is inadequate for coping with stochastic disruptions. The *Duration* strategy, which serves as a strong heuristic baseline, substantially reduces delays compared to *Waiting*. Nevertheless, its delay distribution remains relatively dispersed, particularly in the upper quartiles, indicating vulnerability to high-variance disruption realizations.

In contrast, the proposed *MARL* framework demonstrates markedly improved stability. Its delay distribution is more compact than that of the *Duration* strategy and closely aligns with the *Oracle*, which represents the theoretical lower bound under perfect information. Although *MARL* cannot outperform the *Oracle*, the strong overlap between their distributions indicates that the multi-agent system effectively learns to mitigate tail risks, delivering consistently low delays even in challenging scenarios.

Fig. 9(b) further highlights the economic advantages of the proposed approach. Both the *Duration* strategy and *MARL* significantly outperform the *Waiting* strategy in terms of normalized cost. Importantly, *MARL* achieves lower normalized costs than the *Duration* strategy. A key observation is that while the *Duration* strategy can reduce delays, it frequently incurs a disproportionately high marginal cost per request, as reflected by its higher median and extended upper whiskers in the cost distribution.

Notably, the shorter upper whisker of the *MARL* cost distribution indicates a reduced likelihood of selecting solutions that are strictly time-optimal but excessively expensive. By explicitly balancing the trade-off between delay reduction and operational cost, the *MARL* framework attains a performance profile that closely approaches the global optimum (*Oracle*) while maintaining robustness and scalability across increasing problem sizes.

6.3.2. Mechanism analysis of decision-making in critical scenarios

To investigate the internal decision-making logic of the proposed framework, we focus on *Scenario 3 (Avoidable Delay)*, which represents the most challenging class of instances in which disruptions are theoretically recoverable through re-planning. This scenario therefore provides a stringent test of whether intelligent intervention can effectively resolve cost-delay trade-offs. We examine the behavioral divergence between strategies by jointly analyzing delay responsiveness (Fig. 10) and economic efficiency (Fig. 11).

Fig. 10 reports the average delivery delay under varying disruption parameters. A clear performance divergence emerges as a function of both disruption intensity (μ) and uncertainty level (σ). The results confirm the necessity of active intervention in these settings. The passive *Waiting* strategy consistently produces severe delays that scale approximately linearly with the number of requests, demonstrating that the induced disruptions are non-trivial and cannot be absorbed through inaction. In contrast, both the heuristic *Duration* strategy and the proposed *MARL* framework substantially mitigate delays, reducing them from the passive baseline to levels approaching the *Oracle* theoretical lower bound.

Complementing the delay analysis, Fig. 11 presents the corresponding total generalized costs across the same disruption configurations. These results highlight the economic infeasibility of the passive *Waiting* strategy. Across all tested instances, its accumulated costs increase sharply, driven primarily by heavy penalties associated with late deliveries. This observation reinforces the premise that in *Scenario 3*, inaction leads to prohibitive economic losses and that proactive re-planning is essential.

While both active re-planning strategies yield substantial cost reductions relative to the passive baseline, a clear hierarchy in economic efficiency emerges. The *MARL* framework consistently achieves lower costs than the heuristic *Duration* strategy and more closely approximates the *Oracle* lower bound. Although the *Duration* strategy remains competitive by significantly improving upon

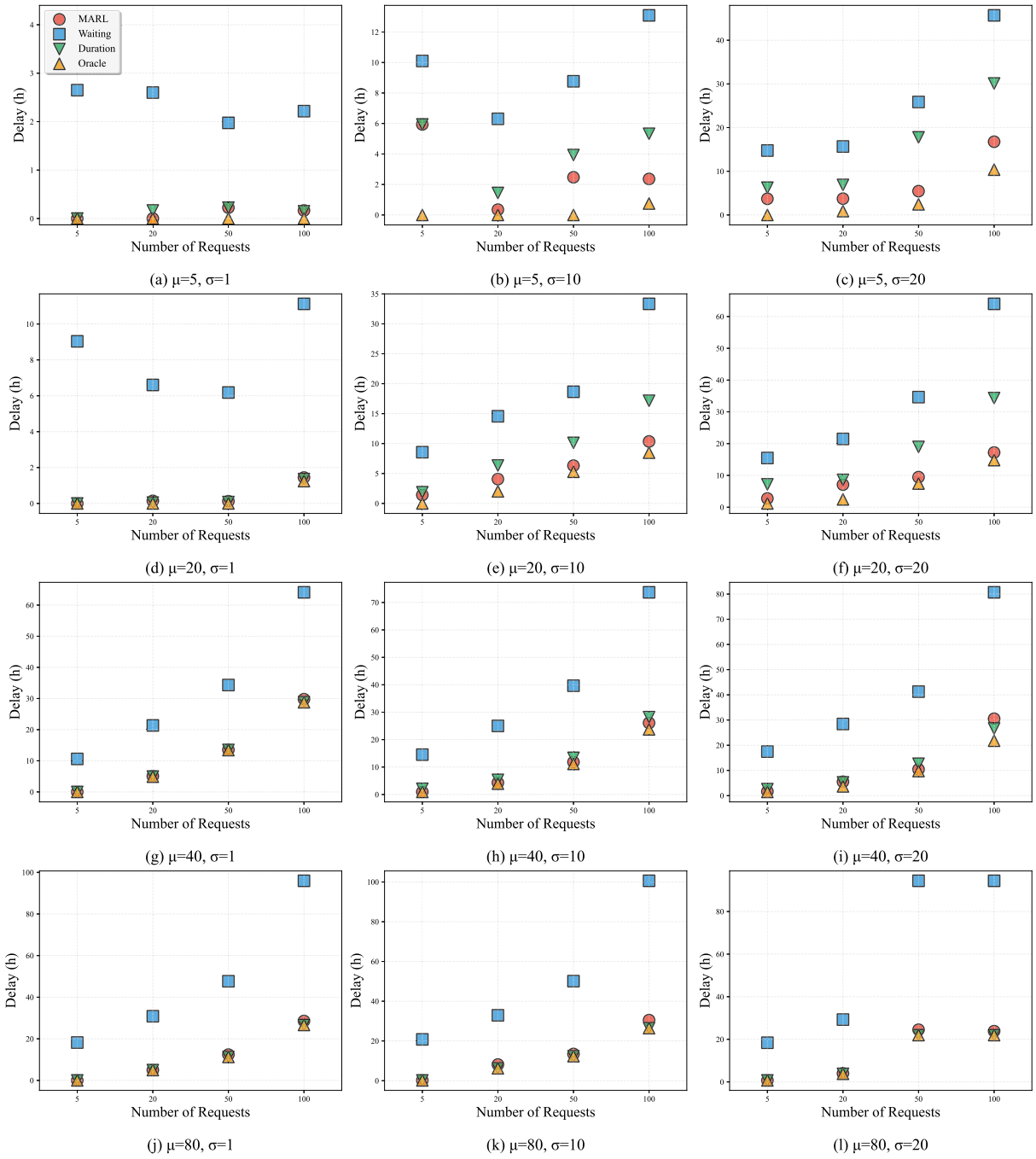


Fig. 10. Detailed delay analysis in scenario 3 (avoidable delay). The breakdown by disruption mean (μ, rows) and uncertainty (σ, columns).

waiting, it systematically incurs a marginal yet non-negligible cost premium. This gap indicates that *MARL* learns a more refined balance between delay recovery and operational expenditure, explicitly accounting for economic sensitivity that is often overlooked by time-centric heuristics.

Nevertheless, the advantage of *MARL* over the heuristic is not uniform across all conditions. A more granular examination reveals that the relative efficacy of the strategies is strongly state-dependent, governed by the interplay between disruption intensity (μ) and uncertainty (σ). Specifically, the performance regimes can be distinguished using the coefficient of variation $C_v = \sigma/\mu$, which serves as a proxy for the signal-to-noise ratio of the disruption process. For example, in the instance shown in Fig. 10(c) ($\mu = 5, \sigma = 20$), the coefficient of variation is high ($C_v = 4.0$), indicating that stochastic variability dominates the mean disruption. In this high-uncertainty

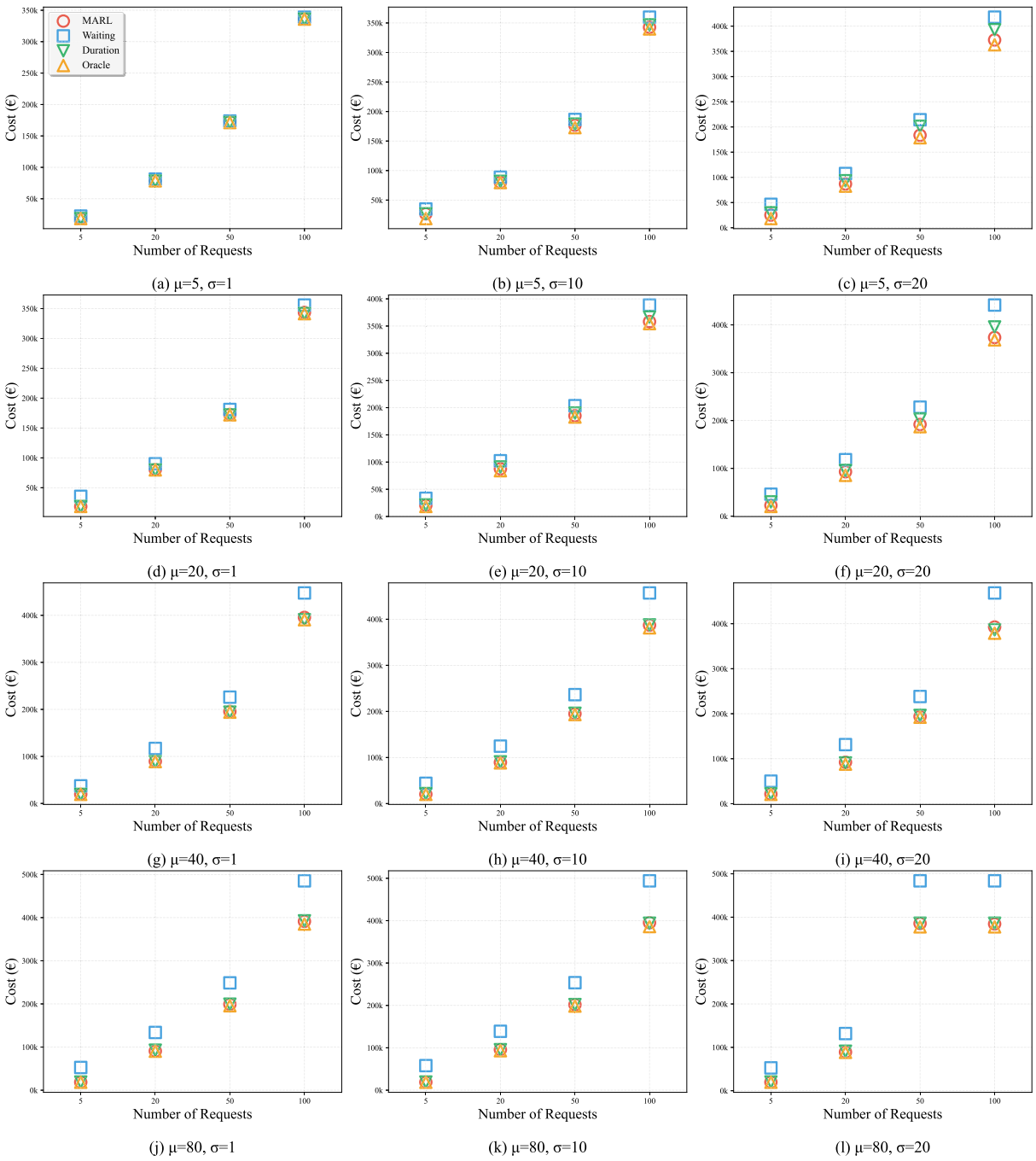


Fig. 11. Detailed cost analysis in scenario 3. Compared to the delay analysis in Fig. 10, MARL achieves lower operational costs than the duration strategy.

setting, the *Duration* strategy based on static expectations fails to distinguish minor recoverable fluctuations from critical tail risks, leading to substantial performance degradation. In contrast, *MARL* exhibits strong adaptive robustness and significantly outperforms the *Duration* strategy. By learning from interaction, the agent implicitly captures the shape of the delay distribution and selects plans that are more resilient to tail risks, demonstrating clear advantages in ambiguous and highly uncertain environments.

Importantly, a joint examination with the cost results in Fig. 11(c) shows that *MARL* superiority in this regime is not achieved through a cost-delay trade-off. Unlike typical cases where delay reduction comes at higher cost, *MARL* outperforms the *Duration* strategy on both delay and cost dimensions. The *Duration* strategy, misled by high variance, often selects routes that are both expensive

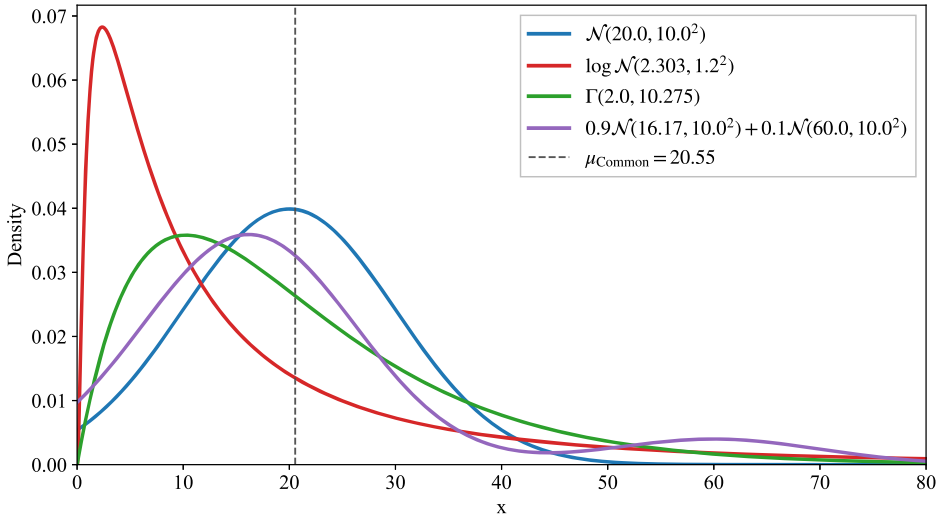


Fig. 12. Comparison of service-time disruption distributions.

and ineffective in recovering time, resulting in inefficient resource allocation. *MARL* avoids such choices, maintaining lower costs while delivering better service reliability.

By contrast, when the mean disruption becomes dominant, as in Fig. 10(l) ($\mu = 80, \sigma = 20$), the performance gap narrows and the *Duration* strategy occasionally matches or slightly outperforms *MARL*. This convergence can be attributed to a low coefficient of variation ($C_v = 0.25$), where the disruption is severe but predictable. In such signal-dominant scenarios, the need for re-planning is obvious, and most reasonable strategies trigger major adjustments. Differences then arise mainly from solution selection rather than decision quality. The *Duration* strategy's slight delay advantage stems from its deterministic, time-greedy execution, whereas *MARL* retains a small degree of exploration to preserve policy robustness, occasionally selecting near-optimal rather than strictly optimal routes.

From a cost perspective, however, *MARL* maintains a consistent modest advantage even in these extreme scenarios, as shown in Fig. 11(l). While the savings are smaller than in high-uncertainty cases, they indicate that *MARL* performs finer-grained optimization. Even under severe disruptions, the agent identifies subtle cost differences among recovery options that time-focused heuristics tend to overlook.

6.4. Additional robustness analyses

Beyond the main robustness and scalability analysis, we report two supplementary robustness tests that relax key modeling assumptions. First, we examine robustness under distributional shift by evaluating the trained policies under alternative nonnegative service-time disruption distributions with different tail behaviors. Second, we test adaptability to network-graph perturbations by temporarily disabling randomly selected transport service options to emulate specific service unavailability. Unless otherwise stated, these experiments follow the baseline setting with $|R| = 20$ and $(\mu, \sigma) = (20, 10)$.

6.4.1. Robustness under distributional shift

The main experiments in this paper adopt a truncated normal distribution to generate service-time disruptions. To assess whether the proposed framework depends critically on this assumption, we conduct supplementary experiments under distributional shift. In particular, we take the truncated normal (TN) setting with $(\mu, \sigma) = (20, 10)$ as the baseline reference case, and construct alternative nonnegative distributions including Mixture Normal (MN), Lognormal (LN), and Gamma (GA). These alternatives differ from the baseline mainly in skewness and tail behavior.

Because truncation changes the actual mean of the baseline distribution, the distributional shift is calibrated against the realized mean of the left-truncated normal baseline rather than its nominal mean. This realized mean is approximately 20.55. Fig. 12 illustrates the four disruption distributions considered in this section.

To examine robustness, we compare *MARL* trained under different disruption distributions and evaluated under all four test distributions. Table 5 reports the total cost and delay. The first observation is that the policy trained under the baseline truncated normal distribution generalizes reasonably well under distributional shift. When tested on MN, LN, and GA disruptions, it continues to outperform both heuristic baselines, *Waiting* and *Duration*, in terms of both total cost and delay. This indicates that the effectiveness of the proposed framework does not rely critically on the truncated normal assumption.

At the same time, the performance degradation is not uniform across test environments. The largest deterioration is consistently observed under the LN test distribution, for which all trained policies incur substantially higher delay and cost. For example, the TN-

Table 5
Performance under alternative training and test disruption distributions.

Method	Total Cost (€)				Delay (h)			
	TN	MN	LN	GA	TN	MN	LN	GA
TN	95,059	96,572	102,495	96,580	0.959	1.991	7.439	1.872
MN	95,183	96,457	101,615	96,509	1.161	1.960	6.331	1.928
LN	95,202	96,497	105,153	96,584	1.146	1.965	9.026	1.978
GA	94,987	96,406	101,618	96,245	1.045	1.891	6.658	1.796
Waiting	96,237	98,681	108,950	99,085	1.961	3.513	12.106	3.729
Duration	95,070	97,486	107,868	97,389	1.056	2.655	11.348	2.550
Oracle	94,612	95,404	95,149	95,389	0.822	1.336	1.477	1.285

Table 6
Performance under original and perturbed network conditions.

Metric	Strategy	Orig.	D1	D2	D3	D4	D5	D6	D7	D8	Mean
Total Cost (€)	MARL	84,922	98,488	85,659	87,957	86,372	92,520	85,670	85,345	85,538	88,052
	Waiting	100,987	103,451	100,987	97,925	100,987	100,959	100,987	100,987	101,621	100,988
	Duration	87,539	98,718	87,539	87,705	87,539	93,646	87,538	87,539	88,016	89,531
	Oracle	81,956	96,229	81,956	83,217	83,039	89,320	81,957	82,208	82,814	84,744
Delay (h)	MARL	2.614	2.960	2.857	3.264	3.147	3.277	2.765	2.789	2.556	2.914
	Waiting	12.300	5.450	12.300	10.267	12.300	7.337	12.300	12.300	12.302	10.762
	Duration	4.188	3.099	4.188	3.725	4.188	4.023	4.188	4.188	4.091	3.986
	Oracle	0.905	1.740	0.905	0.906	1.486	1.667	0.905	1.024	0.905	1.160

Note: Orig. denotes the original network condition, and D1–D8 denote the eight feasible service-option removal draws. For each metric–strategy pair, each entry in Orig. or D1–D8 is the average result over the Scenario 3 evaluation cases under the corresponding network condition. The Mean column is the arithmetic average across Orig. and D1–D8.

trained policy increases from 95,059 € and 0.959 h under TN testing to 102,495 € and 7.439 h under LN testing. This suggests that heavier-tailed disruptions create a substantially more challenging recovery environment, mainly because rare but severe service-time delays are more likely to propagate through downstream synchronization constraints and amplify coordination difficulty.

A second observation is that the baseline TN-trained policy is not uniformly the best MARL model under all shifted test distributions. In several cases, the policies trained under MN or GA perform slightly better. For instance, GA training yields the lowest total cost under TN and GA testing, while MN training performs best under LN testing. This implies that the main value of the TN-based training environment lies not in universal dominance, but in stable and competitive generalization across a range of distributional shifts.

By contrast, direct training under the LN distribution does not lead to stronger performance. In fact, the LN-trained policy performs worst among the MARL variants in several settings, including the matched LN–LN case. This pattern suggests that the heaviest-tailed disruption environment is also the most difficult one for policy learning, and that direct exposure to highly irregular disruptions does not automatically improve decision quality. Instead, moderately skewed training environments such as MN or GA may provide a better balance between representativeness and learnability.

6.4.2. Robustness to network-graph perturbations

In practice, individual services on certain arcs may become temporarily unavailable, which effectively perturbs the feasible network graph. To assess the adaptability of the proposed framework to such changes, we conduct a repeated perturbation-draw experiment. A perturbation draw is defined as the removal of one service-option group, i.e., all services associated with one transport mode on one terminal pair. The perturbed network is fixed within a draw and used for the whole evaluation run. We generated eight feasible perturbation draws; these draws exhaust the feasible barge/train service-option removals retained after feasibility screening. The MARL policy used in this test is trained under the original network and then evaluated without re-training in the perturbed-network environments. Within each draw, all compared strategies use the same perturbed network and realized event set. For the event-seed stability check, this eight-draw evaluation is repeated for ten independently generated event-seed groups, and each reported seed-level value is the average across the feasible perturbation draws under that event group. We focus on Scenario 3, where delays can be avoided through rerouting or mode switching under the original network.

Table 6 reports performance under the original network and the eight feasible service-option removal draws, so it is not based on a single perturbation draw. Across these nine network conditions, *MARL* obtains a mean total cost of 88,052 € and a mean delay of 2.914 h. It remains below both practical baselines: compared with *Waiting*, the mean cost and delay are lower by 12.81% and 72.92%, respectively; compared with *Duration*, they are lower by 1.65% and 26.90%, respectively.

Table 7 reports the complementary event-seed stability summary, which separates event-sampling variability from network-condition variability. Across event seeds, *MARL* obtains a mean cost of 88,444 € and a mean delay of 2.952 h. The corresponding values for the practical baselines are 100,988 € and 10.570 h for *Waiting*, and 89,780 € and 3.961 h for *Duration*. Thus, compared

Table 7
Statistical stability across independently generated event-seed groups.

Metric	Strategy	1	2	3	4	5	6	7	8	9	10	Mean	Std.
Total Cost (€)	MARL	88,822	86,096	87,741	87,761	90,211	87,485	90,279	88,259	88,130	89,651	88,444	1317
	Waiting	101,251	98,955	100,884	98,385	104,987	100,193	101,880	100,673	101,768	100,907	100,988	1799
	Duration	92,148	88,110	88,833	89,076	93,011	88,856	92,366	88,600	87,610	89,189	89,780	1950
	Oracle	86,476	83,645	83,981	86,058	86,136	84,408	87,075	84,072	83,439	85,633	85,092	1322
Delay (h)	MARL	3.425	1.929	2.293	2.934	3.946	2.389	3.979	2.626	2.613	3.382	2.952	0.705
	Waiting	11.917	9.118	10.390	9.556	12.477	10.007	11.356	9.739	10.622	10.514	10.570	1.067
	Duration	5.503	3.135	3.208	3.985	5.680	3.271	5.398	3.108	2.712	3.610	3.961	1.132
	Oracle	1.972	0.509	0.521	1.959	1.657	0.844	2.174	0.515	0.379	1.392	1.192	0.713

Note: Columns 1–10 denote independently generated event-seed groups. For each metric–strategy pair, each numbered entry is obtained by first evaluating the corresponding event-seed group under the eight feasible perturbation draws and then averaging the eight draw-level results. Mean and Std. are the arithmetic mean and sample standard deviation across the ten numbered entries.

with *Waiting*, *MARL* reduces mean cost and delay by 12.42% and 72.07%, respectively; compared with *Duration*, the reductions are 1.49% and 25.47%, respectively.

Taken together, the two summaries show that the perturbation result is not driven by a single favorable service-option removal or a single event sample. Across the nine network conditions in Table 6 and the ten event-seed groups in Table 7, *MARL* performance remains better than the practical baselines, especially in delay reduction. The event-seed standard deviations in Table 7 indicate that delay is affected by the realized disruption events, but the mean delay advantage of *MARL* remains clear. Therefore, these results support the effectiveness of the trained policy under the tested moderate service-option removal regime.

6.5. Ablation study of multi-agent architectures and coordination mechanism

To validate the structural design of the proposed framework, we conduct a comprehensive ablation study comparing three primary architectural configurations: (1) *Centralized Single-Agent*, (2) *Homogeneous MARL*, and (3) the proposed *Heterogeneous MARL*. Table 8 summarizes the performance of these architectures under the most challenging setting, *Scenario 3 (Avoidable Delay)*. In addition, a supplementary analysis on agent preference design is reported in Section 6.5.3.

To ensure a fair comparison, all architectures interact with the same ALNS-based candidate plan generator. They differ only in their decision-making scope and reward formulation derived from Eq. (14).

Centralized single-agent. A single decision maker observes the full global state and directly selects a candidate plan. This architecture is trained using the PPO algorithm. The reward is defined as a scalarized global objective, equivalent to the consensus branch of Eq. (14), but computed using the *global total* cost and delay of the network. As a result, the multi-agent negotiation and coordination mechanisms are entirely removed.

Homogeneous MARL. This configuration employs the same MADDPG algorithm and network architecture as the proposed method, but assumes fully aligned agent interests. Both agents share identical preference weights ($w_d^1 = w_d^2$, $w_c^1 = w_c^2$) and receive identical rewards. Accordingly, the local utility terms $\delta_c^i(\mathbf{x})$ in Eq. (14) are replaced by the *global* total cost, transforming the problem into a fully cooperative game in which agents jointly optimize system-wide performance.

6.5.1. Necessity of multi-agent decomposition

We first address a fundamental architectural question: is a multi-agent decomposition necessary, or can a centralized learner achieve comparable performance? The empirical results in Table 8 provide a clear answer. The *Centralized Single-Agent* architecture consistently underperforms all multi-agent variants across the evaluated scenarios. This gap is particularly pronounced in the critical *Scenario 3 (Avoidable Delay)*, where the centralized agent incurs a global delay of 2.655 h, nearly twice that of the proposed *Heterogeneous MARL* framework (1.310 h), along with a higher total cost.

This performance degradation can be attributed to two structural limitations inherent to centralized architectures in this domain:

1. Curse of dimensionality. In a centralized setting, a single policy must map the entire joint state space to a joint action space. As the synchromodal network involves multiple carriers, transport modes, and disruption realizations, the state-action space grows combinatorially. This complexity severely hampers efficient exploration, causing the centralized learner to converge prematurely to suboptimal local solutions rather than discovering effective recovery strategies.

2. Objective masking and gradient conflict. More fundamentally, the centralized approach relies on a scalarized reward function that aggregates time and cost objectives. This formulation is prone to gradient dominance: cost-related signals are typically dense and continuous, whereas delay penalties are often sparse or threshold-driven. As a result, optimization tends to favor cost minimization, masking the critical learning signal required to avoid severe delays. This imbalance prevents the centralized learner from reliably internalizing delay-sensitive decision rules.

In contrast, the proposed multi-agent architecture alleviates these issues through explicit problem decomposition. By partitioning the global decision process into agent-specific subproblems, each agent learns a value function aligned with its operational role

Table 8
Comparison of architectures: global performance vs. individual payoffs.

Scenario (Ratio)	Architecture	Global Metrics		Individual Payoffs (Cost)	
		Delay (h)	Total Cost (€)	Carrier 1 (€)	Carrier 2 (€)
Overall (1.00)	Heterogeneous	**0.686**	78,569	41,544	**37,025**
	Homogeneous	0.705	**78,552**	**41,022**	37,530
	Single Agent	0.817	78,812	41,451	37,361
Scenario 1 (0.82)	Heterogeneous	–	76,603	39,136	**37,467**
	Homogeneous	–	**76,557**	38,913	37,644
	Single Agent	–	76,661	**38,858**	37,793
Scenario 2 (0.04)	Heterogeneous	–	**79,674**	45,373	**34,301**
	Homogeneous	–	79,916	**44,011**	35,905
	Single Agent	–	80,115	44,432	35,683
Scenario 3 (0.10)	**Heterogeneous**	**1.310**	**82,878**	47,502	**35,376**
	Homogeneous	1.503	82,956	**46,782**	36,174
	Single Agent	2.655	84,635	47,213	37,422
Scenario 4 (0.004)	Heterogeneous	15.475	**107,329**	55,155	**52,174**
	Homogeneous	15.475	107,998	**54,182**	53,816
	Single Agent	15.475	108,931	57,129	51,802
Scenario 5 (0.04)	Heterogeneous	12.746	105,113	53,428	51,685
	Homogeneous	12.746	105,113	53,428	51,685
	Single Agent	12.746	105,113	53,428	51,685

(e.g., upstream versus downstream transport). This decomposition reduces the effective learning complexity and mitigates gradient interference, enabling the system to jointly achieve cost efficiency and timely delivery.

6.5.2. *The role of heterogeneity in multi-agent coordination*

We now compare the two multi-agent architectures to clarify when heterogeneity is beneficial. As shown in Table 8, their relative performance depends strongly on the disruption scenario.

In low-complexity environments (e.g., *Scenario 1*), the *Homogeneous* configuration slightly outperforms the proposed method. In these cases, disruptions are minor, and the optimal action is often to keep the original plan. Because all agents receive the same global reward, the *Homogeneous* setting enables quick agreement on a *Waiting* decision, avoiding unnecessary re-planning costs.

The situation changes fundamentally in *Scenario 3*, which involves avoidable but severe delays. Here, the proposed *Heterogeneous* architecture clearly outperforms the *Homogeneous* one, reducing global delay from 1.503 h to 1.310 h while also lowering total cost. This result shows that allowing agents to pursue different objectives can improve not only individual outcomes but also overall system performance.

The reason lies in how different reward designs treat local failures. In the *Homogeneous* setting, agents optimize a single global reward. Under this formulation, poor performance in one transport segment can be compensated by improvements elsewhere. As a result, locally risky decisions may be accepted as long as the average system performance looks reasonable.

In contrast, the *Heterogeneous* setting assigns each agent a private reward aligned with its own operational responsibility. The time-sensitive agent (Agent 2) rejects any candidate plan that causes excessive delay in its segment, regardless of potential upstream cost savings. Similarly, the cost-sensitive agent (Agent 1) rejects plans that are too expensive. This explicit disagreement prevents local problems from being ignored and forces the system to search for solutions that satisfy both agents simultaneously.

Importantly, this mechanism does not lead to a zero-sum outcome. As shown in Table 8, Agent 2 achieves a lower private cost in the *Heterogeneous* setting than in the Homogeneous one. This indicates that strict local constraints help the system avoid unreliable plans and guide it toward solutions that are both faster and more cost-effective overall.

6.5.3. *Preference sequencing in multi-stage transport decisions*

We examine the impact of the temporal allocation of agent preferences along the sequential transport chain. Unexpected events are stochastic in location, timing, and affected mode. In many cases, the upstream transport leg has already been completed when a disruption occurs, leaving only downstream decisions adjustable. This creates an inherent asymmetry in where uncertainty becomes operationally binding. As the shipment progresses, uncertainty about terminal service times is gradually resolved; however, delay risk tends to accumulate because downstream stages face a tighter remaining delivery window and fewer recovery options. Despite this, under the consensus mechanism, both agents continue to participate in decision-making based on their private utilities.

To study this effect, we compare the proposed configuration with a reversed setup, where Agent 1 is Time-Sensitive and Agent 2 is Cost-Sensitive. As reported in Table 9, the reversed configuration leads to both higher global delay and higher total cost compared to the proposed setting.

In the reversed setup, the upstream time-sensitive agent (Agent 1) indeed incurs a slightly lower private cost (€46,780 versus €47,502). By prioritizing delay avoidance, it reduces delay-related penalties within its own transport action. However, this local

Table 9
Impact of agent preference in scenario 3.

Configuration	Global Performance		Agent Private Costs (€)	
	Delay (h)	Total Cost (€)	Ag.1 (Leg 1)	Ag.2 (Leg 2)
Proposed (Cost → Time)	**1.310**	**82,878**	47,502	**35,376**
Reverse (Time → Cost)	1.798	83,369	**46,780**	36,589
Homogeneous (Baseline)	1.503	82,956	46,782	36,174

Table 10
Sensitivity analysis of reward parameters.

Metric	Change	α_1	τ_1	w_c^1	w_d^1	α_2	τ_2	w_c^2	w_d^2
Total Cost (€)	−20%	79,155	79,182	79,125	79,137	79,114	79,218	79,164	79,177
	−10%	79,173	79,132	79,096	79,086	79,135	79,107	79,128	79,097
	+10%	79,097	79,140	79,134	79,125	79,120	79,148	79,159	79,119
	+20%	79,118	79,155	79,160	79,170	79,120	79,115	79,159	79,163
Delay (h)	−20%	1.054	1.033	1.028	1.045	1.039	1.058	1.044	1.043
	−10%	1.058	1.026	1.021	1.018	1.043	1.019	1.018	1.025
	+10%	1.029	1.053	1.046	1.036	1.021	1.053	1.047	1.015
	+20%	1.036	1.046	1.035	1.044	1.018	1.034	1.048	1.044

cost advantage does not translate into better global outcomes, since the transport structure itself implies that most uncertainty is inherently handled at the downstream stage.

The main problem appears at the downstream stage. In the reversed configuration, Agent 2 is cost-sensitive and incurs a higher private cost (€36,589 versus €35,376). Because the cost component, excluding the effect of disruption duration, is deterministic, this agent evaluates re-planning options mainly based on the expected disruption duration and tends to ignore variability and tail risk. As a result, it selects solutions that appear economical on average but lead to a higher global delay (1.798 h versus 1.310 h) and a higher total cost.

In contrast, the proposed Cost-then-Time configuration places the time-sensitive agent at the final transport leg. This agent directly faces the consequences of delay and therefore reacts strongly to duration uncertainty. By rejecting fragile plans and enforcing sufficient safety margins, it effectively protects the system against worst-case scenarios, yielding the lowest delay and total cost among the compared configurations.

6.6. *Sensitivity and operational analysis of the coordination mechanism*

This subsection further examines the robustness and operational characteristics of the proposed coordination mechanism. We first analyze the sensitivity of the regret-based hybrid reward to moderate perturbations in agent-specific hyperparameters. We then study how different negotiation round limits influence fallback behavior, system efficiency, and computational overhead, in order to clarify the practical trade-offs of the consensus process.

6.6.1. *Sensitivity analysis of regret-based reward hyperparameters*

To examine whether the performance of the regret-based hybrid reward depends on narrowly tuned, agent-specific hyperparameters, we conduct a sensitivity analysis based on the nominal parameter settings reported in Table 3. Specifically, we independently perturb the reward-related parameters, including the preference factor α_i, tolerance threshold τ_i, cost weight w_c^i, and delay weight w_d^i, for both agents by ±10% and ±20%, while keeping all other parameters fixed. The analysis is conducted under the same representative medium-scale setting as in Section 6.2, $|R| = 20$ shipment requests and truncated-normal unexpected events with $\mu = 20$ h and $\sigma = 10$ h.

Table 10 shows that the proposed method remains stable under moderate parameter perturbations. The total cost varies only from 79,086 to 79,218 € around the nominal baseline of 79,133 €, corresponding to a maximum deviation of less than 0.11%. Similarly, the delay remains within 1.015–1.058 h around the baseline value of 1.040 h. For reference, the *Waiting* and *Duration* benchmarks yield 80,226 € and 79,447 € in total cost, with delays of 1.793 h and 1.217 h, respectively, while the *Oracle* achieves 78,850 € and 0.915 h.

Fig. 13 further confirms the robustness of the proposed framework from a multi-metric perspective. For both agents, perturbing α, τ, w_c, and w_d by up to ±20% leads only to small deviations from the nominal baseline in total cost and total delay, while fallback rate and total reward also remain within a narrow range. The trends across different perturbation levels are generally smooth, and no parameter exhibits abrupt degradation or unstable switching behavior. This indicates that the learned coordination policy is not overly sensitive to moderate changes in any single reward component. The results suggest that the performance gains of the proposed framework are not driven by narrowly tuned hyperparameter choices, but instead arise from a stable coordination mechanism that preserves its main behavior under moderate agent-specific perturbations.

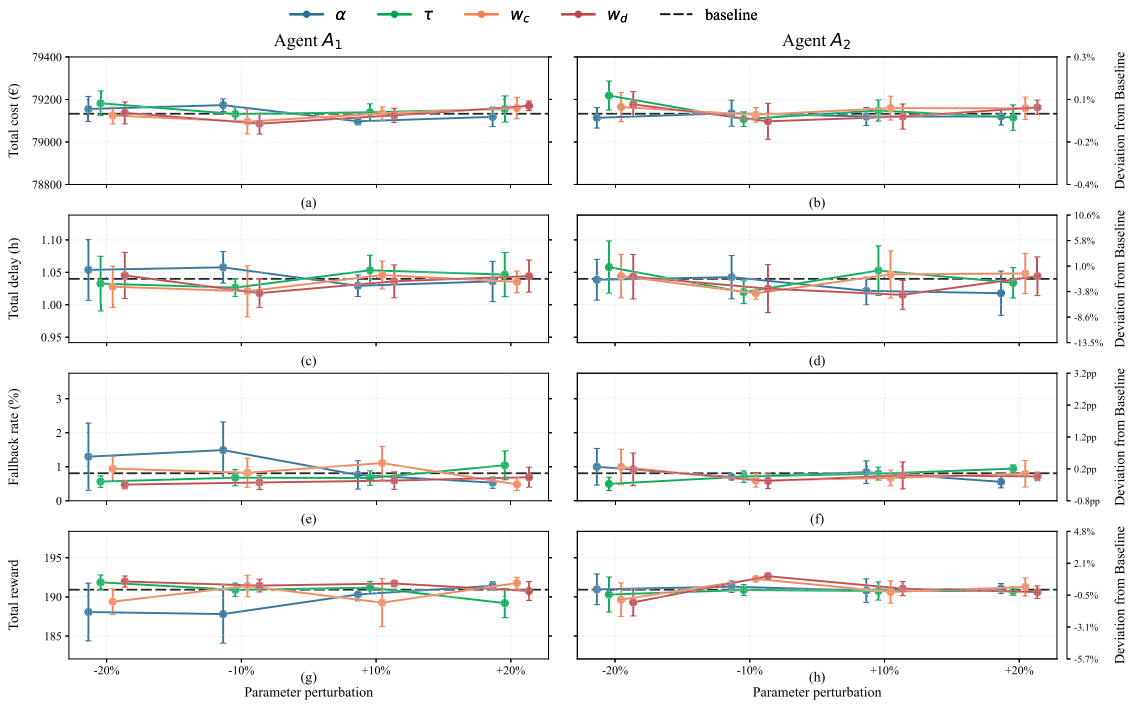


Fig. 13. Sensitivity of the proposed framework to ±10% and ±20% perturbations in reward hyperparameters for both agents. The error bars indicate the min–max range, and the dashed line denotes the nominal baseline.

6.6.2. *Impact of negotiation round limits on fallback behavior and efficiency*

To better understand the operational behavior of the consensus mechanism, we evaluate how different negotiation round limits affect fallback frequency, system efficiency, and computational overhead. Specifically, the round limit is set to 1, 3, 5, and 10, and performance is tested under $|R| = 20$ shipment requests, different disruption scenarios, and unexpected-event distributions.

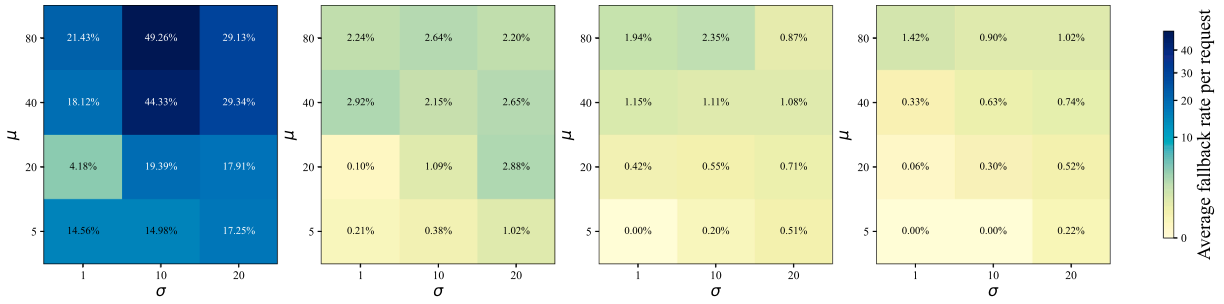
As shown in Fig. 14, fallback occurs frequently when the round limit is set to 1, especially under severe disruption conditions. In several challenging settings, the fallback rate exceeds 20%, and reaches 49.26% in the worst case. Once the round limit is increased to 3, the fallback rate drops sharply to only a few percent in most cases, while limits of 5 and 10 reduce it even further. This suggests that many fallback events are caused by overly early termination of the negotiation process rather than by a fundamental inability of the agents to reach an agreement.

The reduction in fallback is accompanied by noticeable improvements in both total cost and delay, especially in high-disruption settings, as shown in Figs. 15-16. The improvement is most evident in challenging environments, while the differences among round limits become much smaller in relatively mild settings. For instance, when $\mu = 80$ and $\sigma = 10$, increasing the round limit from 1 to 3 reduces fallback from 49.26% to 2.64%, lowers total cost from 94,133 € to 88,569 €, and decreases delay from 8.2462 h to 4.9499 h. Further increasing the limit to 5 or 10 yields additional gains, but these gains are clearly smaller, indicating diminishing returns.

Fig. 17 shows that the computational overhead remains limited across all tested round limits, although the runtime is slightly higher for a round limit of 10 than for moderate settings such as 3 and 5 in several cases. This can be explained by the fact that a larger negotiation budget enables the agents to learn to resolve difficult requests through longer negotiation rather than converging too early on a suboptimal coordination outcome. In deployment, most requests are still settled within the first few exchanges, but a small number of hard cases continue for more rounds under the round-10 policy, which increases the average runtime. In contrast, a very small round limit, such as 1, restricts the agents' ability to learn effective negotiation patterns and leads to more frequent fallback, which also incurs additional processing cost. Therefore, moderate round limits such as 3 and 5 provide most of the coordination benefit without the extra runtime associated with unnecessarily long negotiations.

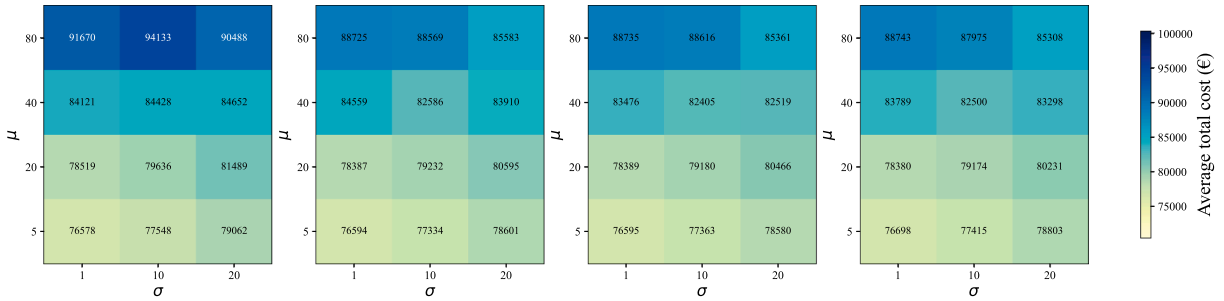
6.7. *Scalability analysis with an extended multi-layer, multi-agent setting*

Synchromodal systems often involve more than two carriers and multiple transshipments. We therefore study two controlled extension scenarios under the same ALNS-assisted MARL framework and negotiation protocol: (i) a *carrier-scaling* setting with more carriers and overlapping responsibilities on the original network; and (ii) a *network-scaling* setting with an additional transshipment layer and more sequentially connected agents.



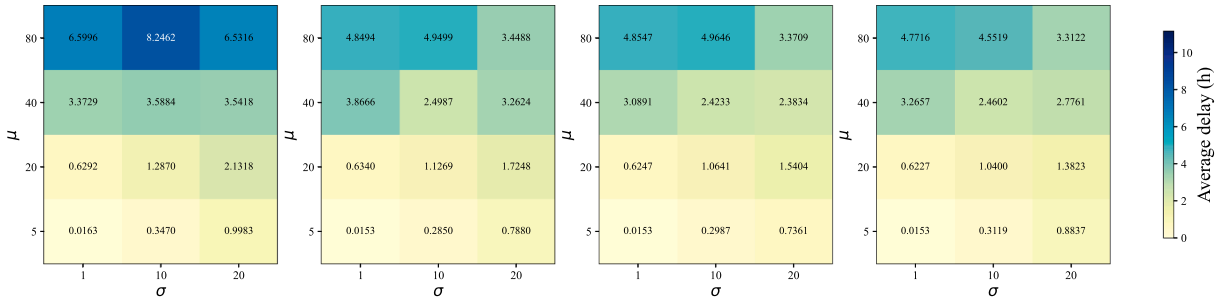
(a) Round limit = 1 (b) Round limit = 3 (c) Round limit = 5 (d) Round limit = 10

Fig. 14. Fallback rate under different negotiation round limits and unexpected-event distributions.



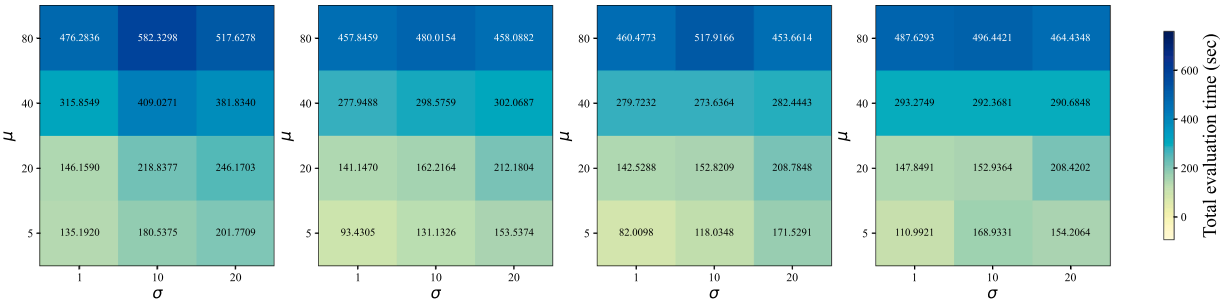
(a) Round limit = 1 (b) Round limit = 3 (c) Round limit = 5 (d) Round limit = 10

Fig. 15. Average total cost under different negotiation round limits and unexpected-event distributions.



(a) Round limit = 1 (b) Round limit = 3 (c) Round limit = 5 (d) Round limit = 10

Fig. 16. Average delay under different negotiation round limits and unexpected-event distributions.



(a) Round limit = 1 (b) Round limit = 3 (c) Round limit = 5 (d) Round limit = 10

Fig. 17. Total evaluation time under different negotiation round limits and unexpected-event distributions.

Table 11
Performance comparison in overlapping carrier responsibilities setting under scenario 3.

(μ, σ)	Total Cost (€)				Delay (h)			
	MARL	Waiting	Duration	Oracle	MARL	Waiting	Duration	Oracle
(5, 1)	79,449	81,360	78,245	78,488	0.000	2.600	0.167	0.000
(5, 10)	83,672	88,833	81,775	79,610	1.832	6.306	1.445	0.000
(5, 20)	90,104	107,664	93,274	82,558	4.570	15.676	6.881	0.876
(20, 1)	81,687	90,194	79,796	79,824	0.040	6.604	0.050	0.000
(20, 10)	87,589	102,454	91,026	84,117	3.970	14.561	6.336	2.017
(20, 20)	92,787	118,194	95,658	85,423	4.960	21.478	8.588	2.442
(40, 1)	90,189	116,997	88,973	88,890	5.159	21.350	4.942	4.893
(40, 10)	90,901	124,658	90,605	88,322	4.683	25.003	5.337	3.943
(40, 20)	90,160	131,376	90,667	87,936	4.562	28.439	5.272	3.662
(80, 1)	91,557	133,856	92,810	90,322	5.213	30.850	4.959	4.959
(80, 10)	95,276	138,937	95,534	93,010	6.476	32.915	6.240	6.240
(80, 20)	90,489	131,797	91,045	88,430	4.394	29.278	3.802	3.802
Average	88,655	113,860	89,117	85,578	3.822	19.588	4.502	2.736

6.7.1. *Multi-carrier scaling with overlapping network*

To examine applicability in denser organizational settings, we extend the baseline two-agent configuration to a four-agent configuration on the same transport network by splitting both the upstream and downstream segments into two carriers each. Relative to the baseline setting in Section 6.1, this introduces overlapping responsibilities within each segment while keeping the physical network, disruption generation, and demand level unchanged. The experiment is conducted with $|R| = 20$ shipment requests, and carriers within the same segment share identical preference parameters in Table 3.

For comparability, we keep the ALNS-assisted candidate generation, reward design, and consensus mechanism unchanged. As discussed in Section 5.1.1, each negotiation is triggered only for carriers involved in the corresponding candidate plan, which helps keep the realized communication burden localized even when more carriers are present.

Since Scenario 3 is the most coordination-intensive setting in the study, it is selected here for detailed analysis. Table 11 provides initial evidence that the proposed framework remains effective in the tested four-agent overlapping-responsibility setting. Compared with the original two-agent setting, this extension does not change the physical network structure, but it substantially increases the density of decentralized interactions. Coordination is no longer limited to a single upstream-downstream interface. Instead, multiple carriers within the same segment may respond to the same disruption and compete for similar re-planning opportunities, which makes the negotiation process more complex.

Despite this added complexity, *MARL* consistently outperforms *Waiting* in all 12 disruption distributions, with average reductions of 25,205 € in total cost and 15.766 h in total delay. Relative to *Duration*, *MARL* also performs better on average, reducing total cost by 462 € and total delay by 0.68 h.

To further study policy transferability, we additionally evaluate on the same instances using the two-agent policy trained in the baseline setting. This transferred policy yields an average total cost of 87,505 € and an average delay of 3.797 h, versus 88,655 € and 3.822 h in the dedicated experiment of this subsection. The delay gap is very small (0.025 h), while the main difference appears in cost (about 1,150 €, 1.31%). This pattern is consistent with the view that, when delay remains controllable, additional overlapping responsibilities primarily reshape economic coordination: because rejecting a non-best offer can still imply losing the shipment to a competing carrier (and thus little or no realized payoff), agents tend to accept feasible requests more aggressively, which preserves timeliness but can modestly weaken cost control.

These results suggest that the proposed *MARL* is not restricted to the original two-agent proof-of-concept setting under this controlled extension. When more carriers operate on overlapping service segments, the learned coordination policy can still capture decentralized interaction effects and maintain competitive overall performance.

6.7.2. *Scaling network complexity through multi-layer transshipment*

To examine whether the proposed framework remains applicable when the transport process involves deeper interdependencies, we reconfigure the goods-flow structure of the baseline setting so that shipments to the final inland destination must pass through two sequential transshipments instead of one. Specifically, the network is reorganized into four layers. Shipments originate from *Rotterdam* in Layer 1, move to intermediate terminals in Layer 2 (*Venlo*, *Dortmund*, and *Duisburg*), are then transferred to second-stage hubs in Layer 3 (*Neuss* and *Duisburg*), and finally reach *Nuremberg* in Layer 4. Under this routing structure, shipments to the final destination require two transshipment operations, which increases the propagation of disruption effects along the transport chain.

The transport service is partitioned among three sequentially connected agents. Agent 1 is responsible for transport from Layer 1 to Layer 2, Agent 2 from Layer 2 to Layer 3, and Agent 3 from Layer 3 to Layer 4. To preserve comparability with the baseline setting, the preference configurations of Agent 1 and 3 follow those of the original upstream and downstream carriers, respectively. Agent 2, which operates in the intermediate layer, is assigned the mean preference parameters of the original two agents, representing a more balanced trade-off between cost efficiency and delay sensitivity. All other components of the framework are kept unchanged as in Section 6.7.1.

Table 12
Performance comparison in multi-layer transshipment setting under scenario 3.

(μ, σ)	Total Cost (€)				Total Delay (h)			
	MARL	Waiting	Duration	Oracle	MARL	Waiting	Duration	Oracle
(5, 1)	111,258	111,549	111,467	111,467	0.000	1.667	0.000	0.000
(5, 10)	116,425	128,003	124,111	115,780	0.647	9.565	5.859	0.588
(5, 20)	120,247	144,348	130,783	117,959	4.094	19.020	10.188	2.037
(20, 1)	119,153	124,333	118,831	118,829	2.563	8.375	2.625	2.563
(20, 10)	120,575	138,817	123,416	118,413	3.681	17.716	6.407	2.263
(20, 20)	132,812	157,650	137,258	126,533	9.524	25.600	12.791	6.323
(40, 1)	130,705	166,187	129,666	129,497	7.208	31.355	7.226	7.167
(40, 10)	139,179	176,931	137,981	135,290	11.307	36.681	11.516	9.980
(40, 20)	142,416	181,432	142,539	139,575	12.801	36.835	12.806	11.456
(80, 1)	155,231	194,078	155,990	150,318	19.656	43.560	16.859	16.859
(80, 10)	152,042	195,691	151,532	145,650	19.024	48.495	15.405	15.405
(80, 20)	150,174	195,516	151,290	145,536	17.857	47.583	15.113	15.113
Average	132,518	159,544	134,572	129,571	9.030	27.204	9.733	7.480

Table 12 confirms that enforcing multi-layer transshipments substantially increases disruption propagation and reduces downstream temporal slack, making recovery outcomes more path-dependent on earlier decisions. Compared with the baseline single-transshipment configuration, the transport chain becomes more tightly coupled because each shipment must satisfy constraints of two sequential transshipments rather than one. As a result, local recovery decisions are more likely to generate downstream effects, and feasible re-planning requires stronger consistency across service segments.

Despite this increase in structural complexity, *MARL* remains effective relative to the passive *Waiting* strategy. Across the 12 disruption distributions in Scenario 3, *MARL* reduces both total cost and delay in every case, with average savings of 27,026 € and 18.174 h, respectively. Relative to *Duration*, *MARL* also performs better on average, reducing total cost by 2,054 € and delay by 0.703 h. This result provides initial evidence that coordinated adaptive re-planning remains useful in the tested three-agent, two-transshipment setting, where disruption impacts propagate across multiple layers of the transport chain.

A key implication of this experiment is that the proposed framework remains applicable when network depth increases in this controlled extension. In the multi-layer setting, successful recovery depends not only on restoring local feasibility at one transshipment terminal, but also on preserving compatibility with subsequent downstream connections. The maintained performance of *MARL* suggests that the learned policy can accommodate stronger sequential dependence and deeper inter-agent coupling within the tested setting, without changing the core learning or negotiation architecture.

6.7.3. Cross-setting discussion on scalability, complexity, and learning implications

While two scenarios emphasize different sources of scalability pressure, they also reveal several common implications for the applicability of the proposed framework beyond the baseline setting. These two experiments provide initial evidence that the proposed ALNS-assisted MARL framework is not limited to the original two-agent proof-of-concept setting, but can remain operational in the controlled extensions considered here, where either the number of interacting carriers or the depth of the transport chain increases.

The two extensions reveal a common dependent performance pattern relative to the *Duration* heuristic consistent with the interpretation in Section 6.3.2. In several high-severity settings, especially when disruption is large and the delay objective becomes dominant, a deterministic time-greedy rule can become highly competitive and may even approach the *Oracle* benchmark. This does not necessarily indicate a failure of *MARL* scalability. Rather, it reflects a structural shift in the decision environment: when recovery success is governed primarily by immediate temporal feasibility, a narrowly delay-oriented rule may capture much of the attainable benefit. By contrast, the *MARL* optimizes a broader coordination objective that jointly accounts for cost, delay, and decentralized interaction effects, and therefore tends to prioritize balanced and robust average performance across heterogeneous disruption regimes. The relative weakness of *MARL* in some high-μ cases should thus be interpreted as a regime-specific trade-off rather than as evidence that the learning-based framework does not scale.

As discussed in Section 5.1.1, negotiation is triggered only for carriers involved in the current candidate plan, so communication scales with the realized affected subset (review complexity $O(|\mathcal{P}_r| \cdot |\mathcal{N}_r|)$). Consequently, in larger networks, the dominant runtime pressure is expected to come more from candidate generation and feasibility checking than from negotiation alone, especially when additional transshipment layers deepen temporal dependencies. From the learning perspective, CTDE may introduce a scalability bottleneck at the centralized critic: with more agents, the critic must process a higher-dimensional joint input, and credit assignment across agents becomes more difficult, which can make training less stable. However, in our current experiments (four agents in an overlapping setting and three agents in a multi-transshipment setting), this effect does not yet appear to be the bottleneck.

From an organizational point of view, the results further suggest that scaling decentralized coordination does not only depend on adding more carriers, but also on how responsibilities overlap and how strongly operational decisions propagate across the transport chain. Overlapping service coverage increases the potential for local competition and strategic interaction, whereas additional transshipment layers increase the potential for downstream ripple effects and path dependence. These two dimensions of complexity are qualitatively different, but both are relevant in real synchromodal platforms. The fact that the same framework remains effective in both extensions provides preliminary support for its applicability under different forms of controlled system growth.

7. Conclusions and future research

This study investigates collaborative re-planning in synchromodal transport networks under service-time uncertainty from a multi-carrier perspective. To address decentralized decision-making and heterogeneous carrier objectives, we propose a model-assisted multi-agent reinforcement learning (MARL) framework in which each carrier learns acceptance policies over candidate re-planning plans generated by an ALNS-based heuristic. The proposed approach enables adaptive coordination while explicitly respecting individual rationality and does not require prior knowledge of service-time distributions.

The key contribution of this work is the development of a *multi-carrier collaborative re-planning methodology under service-time uncertainty* based on *MARL*. Specifically, we formulate the decentralized re-planning process as a partially observable decentralized Markov game and propose an ALNS-assisted MARL framework in which carriers act as autonomous agents that coordinate through an acceptance-based consensus mechanism. The framework operationalizes a principled division of labor: ALNS is responsible for generating structurally feasible, high-quality candidate plans and providing tractable feasibility/cost evaluation, whereas MARL learns *strategic* carrier-level acceptance policies that explicitly account for heterogeneous objectives and individual rationality under uncertainty. To enable learning despite delayed uncertainty resolution, we further introduce an event-triggered re-planning and hindsight learning scheme, together with a regret-based hybrid reward that aligns global timeliness with local objective. Beyond algorithmic performance, our ablation and mechanism analyses provide actionable design insights for decentralized synchromodal operations: (a) multi-agent decomposition is structurally necessary for avoiding dimensionality and objective masking in centralized learning; (b) carrier heterogeneity can improve global outcomes via an adversarial-filter effect that prevents locally unacceptable plans from being compensated in an aggregated objective; and (c) the temporal placement of cost- versus time-sensitive preferences along a sequential transport chain critically determines robustness to stochastic disruption durations.

Additional robustness analyses, together with sensitivity and controlled scalability tests, provide further evidence on robustness and applicability within the experimental scope of this study. The learned policy remains stable under moderate reward-parameter perturbations and continues to perform well under distributional shifts and moderate network-graph perturbations. The controlled scalability extensions provide initial evidence of operational feasibility as organizational interaction becomes denser or the transport chain deepens, but larger-scale validation is still needed before making platform-scale scalability claims.

In practice, the proposed framework can be deployed as a decision-support layer that augments existing planning systems rather than replacing them. For logistics service providers and carriers, the MARL agents can be used as policy advisors that recommend whether to accept or reject re-planning options generated by heuristic planners, while allowing human dispatchers to retain final control. Critically, carrier-specific preferences such as cost sensitivity or delivery strictness can be explicitly encoded. For terminal operators and synchromodal platform managers, the framework enables structured information sharing on disruptions (e.g., affected locations and modes) without requiring disclosure of private cost data, and can be used to test coordination protocols and identify failure modes of decentralized decision-making before live deployment.

The results of this study provide several actionable insights for decision makers involved in synchromodal freight operations, particularly in environments characterized by decentralized control and uncertain terminal service times. Beyond algorithmic performance, the findings reveal how coordination structures, preference design, and decision timing fundamentally shape operational resilience and cost efficiency. Key managerial insights are summarized below:

(a) Decentralized coordination can outperform centralized control under uncertainty. A central managerial takeaway is that forcing a single decision-maker to optimize system-wide plans under uncertainty can be counterproductive. The empirical results show that centralized learning consistently underperforms multi-agent coordination due to dimensionality and objective masking. From a managerial perspective, this implies that delegating decision authority to carriers while coordinating through structured interaction rules can yield better outcomes than imposing centralized re-planning directives.

(b) Heterogeneous objectives are not an obstacle but a coordination asset. Contrary to the common belief that aligned objectives are necessary for collaboration, the study demonstrates that properly structured heterogeneity improves global performance. Cost-sensitive and time-sensitive carriers act as mutual filters, preventing locally unacceptable plans from being compensated by aggregate averages. This suggests that diversity in operational priorities, when embedded in an appropriate coordination mechanism, can enhance system-level robustness rather than undermine it.

(c) Robust re-planning requires sensitivity to uncertainty variance, not only expected delay. The scenario analysis highlights that heuristic strategies based on expected disruption duration fail in high-uncertainty regimes. Managers relying on average-based planning rules are exposed to tail risks that significantly degrade performance. Learning-based approaches that internalize the distributional shape of disruptions–rather than just their mean–are better suited for volatile environments such as congested ports or weather-sensitive inland terminals.

(d) The timing of decision authority along the transport chain matters. The sequencing experiments reveal that placing time-sensitive decision authority at the final transport stage is critical for robustness. This reflects an execution asymmetry: with a shipment closer to delivery, the remaining time buffer becomes smaller and there are fewer recovery options, so delay risk is most critical near the final stage. Practically, this suggests stage-dependent decision rules, which can improve robustness. Such sequencing can be operationalized through contracts and operating procedures that specify handover deadlines, delay-accountability allocation, and escalation triggers.

(e) Negotiation budgets are a practical design parameter for real-time coordination. Very small round limits can cause frequent fallback under severe disruptions, whereas moderate limits deliver most benefits with limited additional runtime; larger limits show diminishing returns.

Although the proposed framework demonstrates strong performance in collaborative re-planning under service-time uncertainty, several limitations arise from deliberate modeling and methodological choices:

(a) The learning agents do not directly control routing or transshipment decisions; instead, their actions are limited to accepting or rejecting candidate plans generated by a heuristic procedure. This acceptance-based design improves tractability and feasibility in a combinatorial transport setting, but it also implies that the quality of coordination ultimately depends on the expressiveness of the candidate plan set. If critical recovery options are not generated by the heuristic, they cannot be selected by the agents. Moreover, representing inter-carrier coordination as a binary *Accept/Reject* decision provides a tractable reduced-form model of agreement, but does not explicitly capture richer bargaining spaces that may arise in partially competitive markets.
(b) Uncertainty is handled implicitly through interaction and regret-based feedback rather than explicit probabilistic inference. While this allows the framework to remain agnostic to service-time distributions and robust to misspecification, it prevents agents from forming forward-looking beliefs or exploiting early signals about disruption evolution. As a result, the framework prioritizes reactive robustness over anticipatory planning.
(c) The coordination mechanism assumes a cooperative training environment with aligned incentives enforced through reward design. Although agents exhibit heterogeneous preferences and decentralized execution, strategic behaviors such as misreporting, opportunistic blocking, or coalition formation are not modeled.

    In competitive settings, misreporting or strategic blocking may increase negotiation time, trigger fallback, or cause plan execution failures, which is not captured in this study. This limits the direct applicability of the framework to fully competitive or market-driven settings.
(d) The scalability extensions provide initial evidence beyond the two-agent setting, but the interaction structure may become substantially more complex in larger networks with many carriers, overlapping networks, denser service options, or more transshipments. Larger-scale validation with explicit runtime profiling is therefore needed to assess CTDE critic scalability, candidate-generation costs, and feasibility-checking overhead.

These limitations open several avenues for future research:

(a) An important direction is to co-learn proposal generation and coordination, so agents can modify or construct recovery plans beyond evaluating ALNS candidates. Hybrid designs that combine learning-based generation with feasibility-preserving repair can enlarge the decision space while maintaining tractability.
(b) Another important extension concerns uncertainty representation. Incorporating belief learning, distributional reinforcement learning, or risk-sensitive objectives could enable agents to distinguish between high-variance and high-severity disruptions more explicitly, supporting anticipatory re-planning and improved tail-risk management. In addition, grounding the disruption model in empirical terminal delay data and testing alternative heavy-tailed delay patterns would further strengthen external validity and clarify how tail behavior affects learning stability and convergence.
(c) Future work may also investigate coordination under weaker cooperation assumptions. Introducing limited information sharing, asymmetric observability, or incentive-incompatible objectives would allow the study of strategic behavior among carriers and the design of learning-based mechanisms that remain stable under partial competition.
(d) Another practical direction is to evaluate and extend the framework to time-varying transport graphs by incorporating terminal and arc availability into observations and action masks, and training agents to generalize across topological perturbations.
(e) Finally, while the extended experiments suggest applicability beyond the baseline two-agent setting, future work should further examine scalability in larger synchromodal networks with more carriers, requests, denser services, and multiple transshipment layers. This includes explicit runtime profiling of key ALNS and CTDE components and the development of scalable solution strategies validated with real operational data.

CRediT authorship contribution statement

Xiaoyu Luo: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization; **Yimeng Zhang:** Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization; **Mi Gan:** Writing – review & editing, Supervision, Funding acquisition; **Xiaobo Liu:** Writing – review & editing, Supervision, Funding acquisition; **Bilge Atasoy:** Writing – review & editing, Visualization, Conceptualization.

Declaration of competing interest

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

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 52402521), Sichuan Science and Technology Program (No. 2025NSFSC1969), National Natural Science Foundation of China (No. U2568219), and National Key R&D Program of China (No. 2025YFB2606500, 2025YFB2606502).

Data availability

Data will be made available on request.

Appendix A. Implementation details

The proposed framework is implemented in Python 3.11. The ALNS operator logic is custom-built, incorporating a roulette-wheel selection mechanism to adaptively choose among destroy and repair operators. The MARL agents are implemented using PyTorch. The training process runs for 3000 episodes on a standard desktop CPU and GPU. We employ the Gumbel–Softmax estimator to handle discrete actions in the MADDPG algorithm, enabling end-to-end differentiable training.

Table A.1 details the specific parameters used for training the MARL agents and configuring the ALNS heuristic. These values match the default configuration in our codebase and were selected based on preliminary tuning to ensure stable convergence.

Both actor and critic networks are implemented as multi-layer perceptrons (MLPs), enabled by the fixed-length observation representation described in Section 5.3. The actor network takes the local observation o_i as input and consists of three fully connected hidden layers, each with 256 units and ReLU activation. The output layer produces logits corresponding to the *Accept* and *Reject* actions, followed by a Gumbel–Softmax operation to enable differentiable sampling during training. The critic network adopts a centralized architecture and shares the same multilayer perceptron structure as the actor network, consisting of three fully connected hidden layers with 256 units each and ReLU activations, followed by a linear output layer that produces a scalar estimate of the joint action-value function.

Table A.1
Detailed hyperparameters for learning and heuristic algorithms.

MADDPG (MARL)		ALNS (Heuristic)	
Parameter	Value	Parameter	Value
Actor Learning Rate	5×10^{-4}	Max Iterations	1000
Critic Learning Rate	1×10^{-3}	Initial Temperature	Autoset
Optimizer	Adam	Cooling Rate	0.99
Discount Factor (γ)	0.99	Operators Selection	Roulette Wheel
Batch Size	256	Remove Op.	Random, Greedy, Related
Replay Buffer Size	1×10^{5}	Insert Op.	Greedy, Regret, Random
Target Update (τ)	0.01	Segment Length	10
Gumbel Temp	1.0	Regret Level	2

Appendix B. Supplementary training metrics

While Section 6.2.1 focuses on the convergence of Loss and Global Cost to validate the stability of the optimization process, the evolution of the accumulated episode reward provides a direct measure of the agents' policy quality from a game-theoretic perspective. Fig. B.1 illustrates the moving average of episode rewards for Agent 1 and Agent 2 over 3000 training episodes. The reward curves of Agent 1 and Agent 2 are highly correlated, suggesting that despite their heterogeneous preferences, they successfully found a cooperative equilibrium. Unlike the Loss curves, the Reward curves exhibit noticeable variance even in the stable phase. This is expected and attributable to two factors: (a) the intrinsic stochasticity of the unexpected events (varying severity μ and uncertainty σ per episode), and (b) the penalty terms in the regret-based reward function, which are sensitive to the optimality gap of the generated candidates.

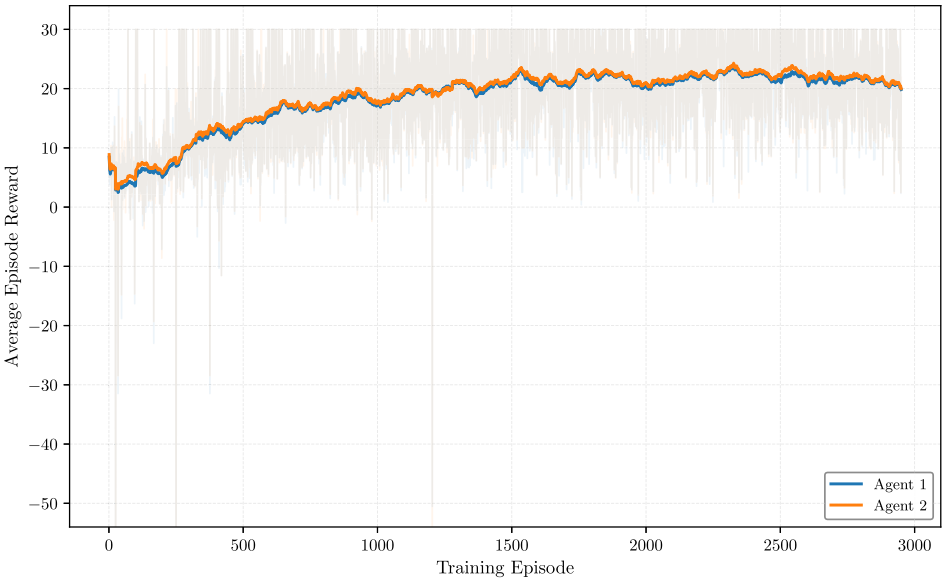


Fig. B.1. Evolution of average episode reward during training. The solid lines represent the smoothed average, while the shaded areas indicate the variance across raw episodes. Both agents show a consistent upward trend, stabilizing around episode 1500.

Appendix C. Detailed performance across all scenarios

Section 6.2.2 focuses on the *Avoidable Delay* scenario, in which the proposed MARL framework exhibits its maximum performance. For completeness, this appendix examines the remaining scenario types to provide a transparent and comprehensive assessment of the framework's behavioral characteristics relative to the benchmark strategies.

Fig. C.1 illustrates the evolution of cost and delay for the overall average as well as for representative sub-cases across different scenario categories.

Scenario 1: Wait optimal (82.0% of cases). In the majority of cases, disruptions are minor and the passive *Waiting* strategy constitutes the theoretical global optimum. The MARL framework converges to a performance level comparable to the *Duration* strategy, which incurs a slightly higher cost than the *Waiting* baseline. As the MARL framework follows a proactive re-planning mechanism similar to the *Duration* strategy, a marginal additional cost is observed relative to the purely passive *Waiting* approach. The limited magnitude of this gap indicates that unnecessary re-planning interventions are largely avoided in such scenarios.

Scenario 2: Cost reducible without delay (4.0% of cases). In this scenario type, delays are absent while alternative routes with lower operational costs are available. The MARL framework outperforms the *Waiting* strategy and converges to a cost level comparable to the *Duration* strategy. This behavior demonstrates the framework's capability to exploit cost-saving opportunities when service reliability is not at risk. The observed cost oscillations reflect the inherent stochasticity of MARL training under near-indifference conditions, rather than a lack of convergence.

Scenario 4: Cost reducible with delay (0.4% of cases). Although rare, these scenarios are operationally critical, as delays cannot be avoided while cost mitigation remains feasible. The MARL framework substantially outperforms both the *Waiting* and *Duration* strategies and converges to the *Oracle* benchmark cost level. This result highlights the framework's ability to identify cost-efficient re-planning alternatives under adverse conditions, where rule-based heuristics may remain locked into standard routing patterns.

Scenario 5: Worst-case optimal scenarios (3.9% of cases). These cases correspond to force majeure situations in which both delay and cost are largely fixed, and the *Waiting* strategy is theoretically optimal. The MARL framework exhibits a slightly higher cost compared to the *Waiting* baseline, which can be interpreted as an exploration cost in a decision space with limited improvement potential. Despite this marginal cost increase, the magnitude of this gap is negligible in absolute terms, remaining below 0.1% of the total cost.

In summary, the proposed MARL framework demonstrates a robust and well-balanced performance across different scenario types. It avoids unnecessary interventions in dominant scenarios, effectively exploits cost-reduction opportunities when available, and maintains stable behavior even in cases where optimization potential is structurally limited.

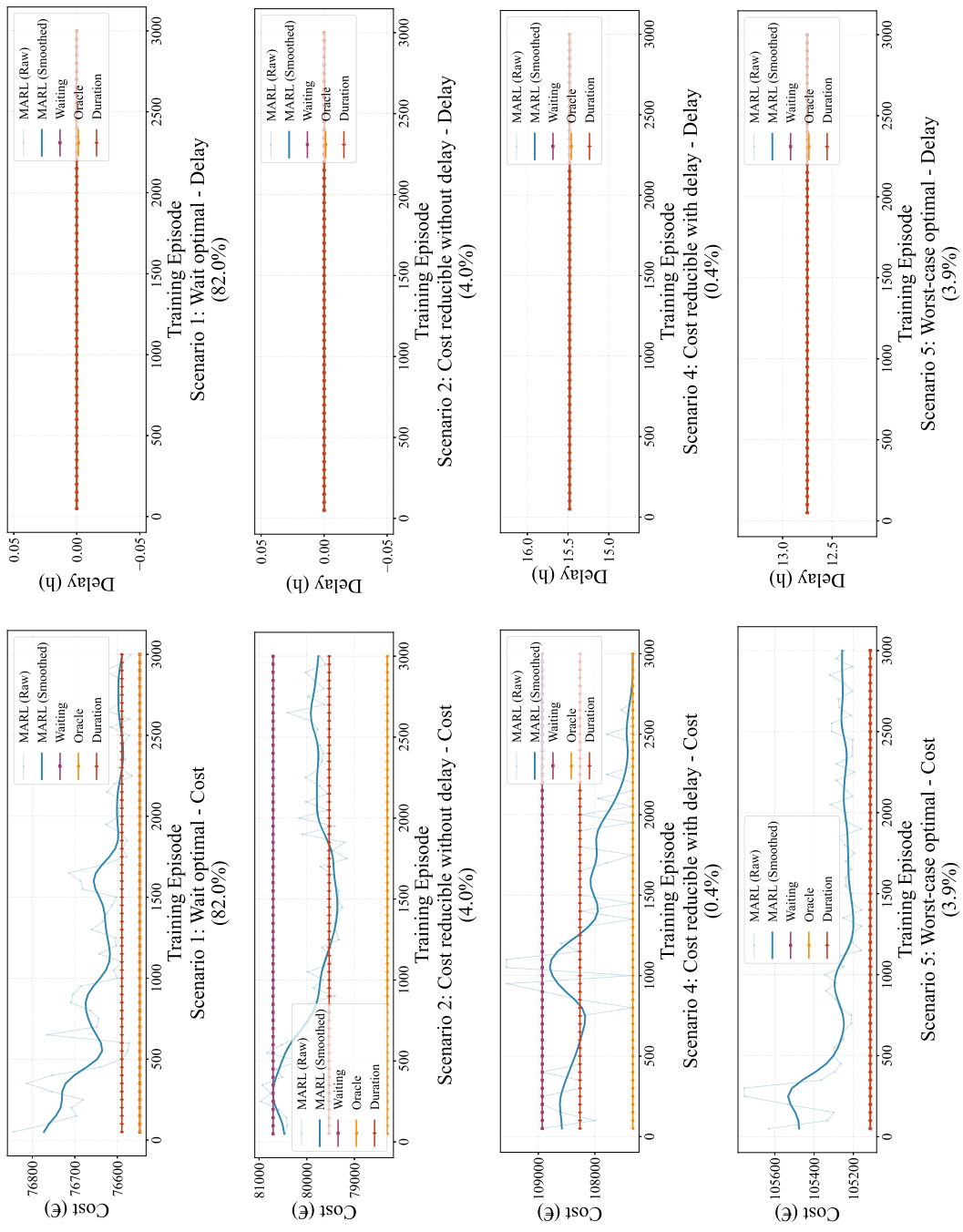


Fig. C.1. Detailed convergence performance across different scenario types. scenario 3 (avoidable delay), which represents the primary performance case, is discussed in detail in Section 6.2.2. The MARL framework (blue) is compared with static baseline strategies: waiting (purple), Oracle (orange), and Duration-based re-planning (red). Note that the y-axis scales differ across scenarios.(For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

References

Akyüz, M.H., Dekker, R., Sharif Azadeh, S., 2023. Partial and complete replanning of an intermodal logistic system under disruptions. Transp. Res. E Logist. Transp. Rev. 169, 102968. https://doi.org/10.1016/j.tre.2022.102968

Alaei, S., Durán-Micco, J., Macharis, C., 2024. Synchromodal transport re-planning: an agent-based simulation approach. Eur. Transp. Res. Rev. 16 (1), 1. https://doi.org/10.1186/s12544-023-00624-y

Ambra, T., Caris, A., Macharis, C., 2019. Towards freight transport system unification: reviewing and combining the advancements in the physical internet and synchromodal transport research. Int. J. Prod. Res. 57 (6), 1606–1623. https://doi.org/10.1080/00207543.2018.1494392

Demir, E., Burgholzer, W., Hrušovský, M., Arıkan, E., Jammernegg, W., Woensel, T.V., 2016. A green intermodal service network design problem with travel time uncertainty. Transp. Res. B Methodol. 93, 789–807. https://doi.org/10.1016/j.trb.2015.09.007

EGS, 2021. Transport services of EGS. https://www.europeangatewayservices.com/en/services/intermodal-services. [Online; accessed 22-February-2022].

Fahim, P. B.M., An, R., Rezaei, J., Pang, Y., Montreuil, B., Tavasszy, L., 2021a. An information architecture to enable track-and-trace capability in physical internet ports. Comput. Ind. 129, 103443. https://doi.org/10.1016/j.compind.2021.103443

Fahim, P. B.M., Martinez De Ubago Alvarez De Sotomayor, M., Rezaei, J., Van Binsbergen, A., Nijdam, M., Tavasszy, L., 2021b. On the evolution of maritime ports towards the physical internet. Futures 134, 102834. https://doi.org/10.1016/j.futures.2021.102834

Filom, S., Razavi, S., 2025. A learning-based robust optimization framework for synchromodal freight transportation under uncertainty. Transp. Res. E Logist. Transp. Rev. 195, 103967. https://doi.org/10.1016/j.tre.2025.103967

Gosavi, A., 2009. Reinforcement learning: a tutorial survey and recent advances. INFORMS J. Comput. 21 (2), 178–192.

Guo, W., Atasoy, B., Negenborn, R.R., 2022. Global synchromodal shipment matching problem with dynamic and stochastic travel times: a reinforcement learning approach. Ann. Oper. Res. 350 (1), 63–94. https://doi.org/10.1007/s10479-021-04489-z

Guo, W., Atasoy, B., Van Blokland, W.B., Negenborn, R.R., 2021. Global synchromodal transport with dynamic and stochastic shipment matching. Transp. Res. E Logist. Transp. Rev. 152, 102404. https://doi.org/10.1016/j.tre.2021.102404

Guo, W., Zhang, Y., Li, W., Negenborn, R.R., Atasoy, B., 2024. Augmented Lagrangian relaxation-based coordinated approach for global synchromodal transport planning with multiple operators. Transp. Res. E Logist. Transp. Rev. 185, 103535. https://doi.org/10.1016/j.tre.2024.103535

Hrušovský, M., Demir, E., Jammernegg, W., Van Woensel, T., 2021. Real-time disruption management approach for intermodal freight transportation. J. Clean. Prod. 280, 124826. https://doi.org/10.1016/j.jclepro.2020.124826

Lemmens, N., Gijsbrechts, J., Boute, R., 2019. Synchromodality in the physical internet – dual sourcing and real-time switching between transport modes. Eur. Transp. Res. Rev. 11 (1), 19. https://doi.org/10.1186/s12544-019-0357-5

Li, L., Negenborn, R.R., De Schutter, B., 2015. Intermodal freight transport planning – a receding horizon control approach. Transp. Res. C Emerg. Technol. 60, 77–95. https://doi.org/10.1016/j.trc.2015.08.002

Littman, M.L., 1994. Markov games as a framework for multi-agent reinforcement learning. In: Machine Learning Proceedings 1994. Elsevier, pp. 157–163. https://doi.org/10.1016/B978-1-55860-335-6.50027-1

Liu, M., Demir, E., Jian, L., 2025. Reinforcement learning for the vehicle routing problem: methodologies, applications, and research outlook. Arab. J. Sci. Eng. . https://doi.org/10.1007/s13369-025-10744-3

Los, J., Schulte, F., Gansterer, M., Hartl, R.F., Spaan, M. T.J., Negenborn, R.R., 2025. Large-scale collaborative vehicle routing. Ann. Oper. Res. 350 (1), 201–233. https://doi.org/10.1007/s10479-021-04504-3

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I., 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA, p. 6382-6393.

Mak, S., Xu, L., Pearce, T., Ostroumov, M., Brintrup, A., 2023. Fair collaborative vehicle routing: a deep multi-agent reinforcement learning approach. Transp. Res. C Emerg. Technol. 157, 104376. https://doi.org/10.1016/j.trc.2023.104376

Pan, S., Ballot, E., Huang, G.Q., 2025. Operations with physical internet. In: Reference Module in Social Sciences. Elsevier, p. B9780443289934001712. https://doi.org/10.1016/B978-0-443-28993-4.00171-2

Pan, W., Liu, S.Q., 2023. Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. Appl. Intell. 53 (1), 405–422. https://doi.org/10.1007/s10489-022-03456-w

Ren, L., Fan, X., Cui, J., Shen, Z., Lv, Y., Xiong, G., 2022. A multi-agent reinforcement learning method with route recorders for vehicle routing in supply chain management. IEEE Trans. Intell. Transp. Syst. 23 (9), 16410–16420. https://doi.org/10.1109/TITS.2022.3150151

Singh, J., Dhurandher, S.K., Woungang, I., Ngatched, T. M.N., 2023. Multi-agent reinforcement learning based approach for vehicle routing problem. In: Ngatched Nkouatchah, T.M., Woungang, I., Tapamo, J.-R., Viriri, S. (Eds.), Pan-African Artificial Intelligence and Smart Systems. Springer Nature Switzerland, Cham. Vol. 459, pp. 411–422. https://doi.org/10.1007/978-3-031-25271-6_26

Yee, H., Gijsbrechts, J., Boute, R., 2021. Synchromodal transportation planning using travel time information. Comput. Ind. 125, 103367. https://doi.org/10.1016/j.compind.2020.103367

Yuan, Y., Zhou, J., Li, H., Li, X., 2023. A two-stage optimization model for road-rail transshipment procurement and truckload synergetic routing. Adv. Eng. Inform. 56, 101956. https://doi.org/10.1016/j.aei.2023.101956

Zhang, K., He, F., Zhang, Z., Lin, X., Li, M., 2020. Multi-vehicle routing problems with soft time windows: a multi-agent reinforcement learning approach. Transp. Res. C Emerg. Technol. 121, 102861. https://doi.org/10.1016/j.trc.2020.102861

Zhang, Y., Guo, W., Negenborn, R.R., Atasoy, B., 2022a. Synchromodal transport planning with flexible services: mathematical model and heuristic algorithm. Transp. Res. C Emerg. Technol. 140, 103711. https://doi.org/10.1016/j.trc.2022.103711

Zhang, Y., Heinold, A., Meisel, F., Negenborn, R.R., Atasoy, B., 2022b. Collaborative planning for intermodal transport with eco-label preferences. Transp. Res. D Transp. Environ. 112, 103470. https://doi.org/10.1016/j.trd.2022.103470

Zhang, Y., Li, X., Van Hassel, E., Negenborn, R.R., Atasoy, B., 2022c. Synchromodal transport planning considering heterogeneous and vague preferences of shippers. Transp. Res. E Logist. Transp. Rev. 164, 102827. https://doi.org/10.1016/j.tre.2022.102827

Zhang, Y., Negenborn, R.R., Atasoy, B., 2023. Synchromodal freight transport re-planning under service time uncertainty: an online model-assisted reinforcement learning. Transp. Res. C Emerg. Technol. 156, 104355. https://doi.org/10.1016/j.trc.2023.104355

Zhang, Y., Tan, X., Gan, M., Liu, X., Atasoy, B., 2025. Operational synchromodal transport planning methodologies: review and roadmap. Transp. Res. E Logist. Transp. Rev. 194, 103915. https://doi.org/10.1016/j.tre.2024.103915