

Article

Solver-Independent MAPPO for Disruption-Aware Last-Mile Routing via Candidate Parcel-Locker Consolidation Points

Mohamed-Ali Ejjanfi , Kadim Lahcen Nadime * and Jamal Benhra

LARILE, ENSEM, Hassan II University of Casablanca, Casablanca 20230, Morocco;
ejjanfimed@gmail.com (M.-A.E.); jamal.benhra@ensem.ac.ma (J.B.)

* Correspondence: nadime.kadim@gmail.com

Abstract

Urban last-mile routes must adapt when demand and network conditions change after dispatch. This study proposes a solver-independent multi-agent proximal policy optimization (MAPPO) framework that combines a shared fleet policy, dynamically ranked candidate parcel-locker consolidation points, disruption-aware observations, fleet-level value learning, and structural feasibility masking. Evaluation on a Los Angeles road-network simulation used chronological data splits, ten random seeds, matched routing baselines, and compound-disruption stress tests. MAPPO served 98.4% of demand versus 96.6% for static OR-Tools and reduced failed packages by 52.2%; under severe compound disruption, service increased from 89.4% to 93.0%. Higher-service adaptive methods reached 99.6–99.7%, with summed online computation times of 0.111–0.503 s per episode compared with 0.029 s for MAPPO. The integrated framework therefore improves disruption recovery over static planning while exposing a clear quality–computation trade-off against adaptive search.

Keywords: multi-agent reinforcement learning; MAPPO; dynamic vehicle routing; parcel lockers; last-mile logistics; disruption resilience; feasibility masking; solver-independent execution

1. Introduction

Urban last-mile routes combine assignment, sequencing, capacity, and depot-return decisions on networks that can change after dispatch [1,2]. Demand revisions, road closures, risk, and weather can invalidate an otherwise efficient static plan, making service continuity depend on how quickly a controller uses revealed information [3,4]. Consolidation through shared delivery points adds coupled fleet decisions because serving one aggregate-demand node changes the remaining demand and useful actions of every vehicle [5–7]. This study represents those shared nodes as analytically derived candidate parcel-locker consolidation points and evaluates operational routing rather than facility siting.

Adaptive optimization can repair or rebuild routes after disruption. Ant colony optimization (ACO), adaptive large neighborhood search (ALNS), and complete reoptimization devote online search to the revealed instance and can deliver strong route quality [8–10]. Reinforcement learning (RL) instead amortizes computation through a trained state-to-action policy [11–13]. Its practical position therefore depends on service quality, execution time, state representation, constraint treatment, and the authority granted to each comparator [14–16].



Academic Editor: Garzia Fabio

Received: 24 July 2026

Revised: 7 September 2026

Accepted: 13 September 2026

Published: 21 September 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the

[Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

Multi-agent reinforcement learning (MARL) provides a cooperative formulation for interdependent vehicles. Multi-agent proximal policy optimization (MAPPO) combines a shared actor with centralized training and decentralized execution (CTDE), allowing fleet-level value estimation during learning and vehicle-level actions during operation [17–19]. State-dependent feasibility masking complements this policy by removing already served, capacity-incompatible, closed-arc, and invalid-depot actions before sampling [20]. MAPPO, CTDE, parameter sharing, and masking are established components; the contribution here is their routing-specific integration with dynamic candidate observations and compound-disruption information.

Existing studies address dynamic demand, safe electric-vehicle routing, forecast-informed control, candidate delivery options, and cooperative fleets [21–25]. The underexplored intersection is a common candidate-consolidation environment that combines forecast demand, risk, closures, weather, explicit feasibility, cooperative policy execution, and comparison with constructive, metaheuristic, repair, static, and complete-reoptimization baselines. The framework developed here fills that intersection and quantifies its service–computation trade-off.

The evaluation uses a Los Angeles road-network simulation, a saved Open Source Routing Machine (OSRM) matrix, chronological training–validation–test splits, ten random seeds, and compound-disruption stress tests. Greedy nearest, ACO, ALNS repair, static OR-Tools, limited OR-Tools repair, Full OR-Tools dynamic reoptimization, and MAPPO operate on matched cases. The analysis addresses three research questions:

- RQ1.** To what extent does the proposed MAPPO policy improve service continuity relative to static and limited-repair routing under last-mile disruptions?
- RQ2.** What service-quality and computational-runtime trade-off does MAPPO provide relative to ACO, ALNS, and Full OR-Tools dynamic reoptimization?
- RQ3.** How do feasibility masking, demand forecasts, risk information, road-closure information, extended observations, centralized training, and the environmental reward affect the reported performance?

The contribution is a problem-specific decision framework that (i) rebuilds a fixed-dimensional vehicle view from a changing global consolidation-point state; (ii) combines a shared actor with fleet-level value learning; (iii) refreshes forecast, risk, closure, weather, and scenario features at each decision epoch; and (iv) enforces immediate admissibility before solver-independent action selection. Its empirical contribution is a matched comparison across seven routing methods, together with ablation, sensitivity, robustness, and dependence-aware statistical analyses. This positioning centers the contribution on the integrated disruption-aware framework and its evaluation.

2. Related Work

2.1. Dynamic and Disruption-Aware Vehicle Routing

Dynamic routing is defined by information revealed during execution. Pillac et al. [3] classify dynamism and information quality, while Soeffker et al. [4] distinguish decision and information models. Optimization approaches range from local repair to anticipatory or complete replanning [26]; ALNS is especially relevant because destroy–repair operators can revise disrupted route segments [9,27,28]. Population-based search has likewise improved logistics-routing outcomes in adjacent settings [29].

RL moves repeated solution effort into offline training. Safe, partially observed, forecast-informed, and reoptimization-comparison studies demonstrate state-conditioned control under uncertain requests, travel, and energy [21–23,30]. PPO has also been compared with ACO for dynamic last-mile consolidation routing [31]. These results

motivate the present matched comparison of policy inference, route repair, and complete dynamic reoptimization.

2.2. Parcel Lockers and Alternative Delivery Points

Research on parcel lockers and alternative delivery points spans network design, accessibility, capacity, customer choice, and vehicle routing [5–7,32]. Location studies show that coverage and performance depend on spatial density and stochastic parcel flows [33–36], while mobile and multitrip models integrate delivery options with routing and recourse [24,37–39].

Operational evidence also links consolidation to courier time, although effects vary by outcome and setting [40]. Environmental effects depend on vehicle efficiency and recipient collection travel [41–44]. These studies distinguish physical locker design and operation from the analytical candidate consolidation points used here.

2.3. Reinforcement Learning for Routing and Combinatorial Optimization

Learned combinatorial optimization includes constructive policies, improvement policies, and learning-guided search [11–13]. Reviews identify generalization, constraints, scaling, and fair benchmarking as central concerns [14–16]. In the present comparison, MAPPO is a sequential policy, ALNS is an improvement mechanism, and OR-Tools performs solver-based search.

2.4. Multi-Agent Reinforcement Learning and MAPPO

CTDE uses fleet-level information during training while retaining local policies for execution, and parameter sharing provides a common rule for homogeneous agents [17,45]. MAPPO applies PPO's clipped update to cooperative trajectories; its performance depends on implementation and critic design [18,19]. MARL routing studies cover multi-depot routing, collaborative gain sharing, and electric-vehicle coordination [25,46,47], supporting a cooperative fleet representation while leaving the present compound-disruption setting open.

2.5. Constraint Handling and Feasibility Masking

Invalid-action masking renormalizes a discrete policy over the currently feasible subset and is widely used for routing capacity and visitation constraints [12,13,20]. Here, it enforces immediate served-status, capacity, closure, and depot logic. Hybrid methods show how learned decisions can also be combined with stronger optimization-based feasibility and improvement mechanisms [14,48].

2.6. Research Gap and Study Positioning

Table 1 compares 20 related studies by demand uncertainty (DU), road disruption (RD), parcel lockers or alternative delivery points (PLs), multiple agents (MAs), network basis, method, and comparator coverage. A circle denotes partial or adjacent coverage.

The synthesis identifies a specific gap: existing streams separately emphasize locker design, dynamic replanning, learned routing, or cooperative fleets, while rarely combining analytical candidate consolidation points, compound road disruptions, explicit feasibility, and a broad adaptive-baseline comparison. The present framework unifies these elements to evaluate service recovery and the quality–computation trade-off of a frozen cooperative policy.

Table 1. Synthesis of directly relevant dynamic-routing, parcel-locker, RL, and MARL studies. ● = explicit coverage; ○ = partial or adjacent coverage; – = not a principal feature.

Reference	Problem and Decision Mode	DU	RD	PL	MA	Real Network	Method/Constraint Treatment	Optimization Baselines	Main Difference from This Study
Pillac et al. [3]	Dynamic VRP review	●	○	–	–	○	Taxonomy of dynamism and information quality	Exact, heuristic, and metaheuristic families	Review; no locker-specific MARL evaluation
Soeffker et al. [4]	Stochastic dynamic VRP review	●	○	–	–	○	Prescriptive-analytics taxonomy	Reoptimization and approximate dynamic programming	Review; no common locker/disruption test bed
Zhang et al. [26]	Dynamic VRP with stochastic requests	●	–	–	–	○	Multi-stage anticipatory optimization	Myopic and anticipatory optimization	Request uncertainty rather than compound road disruption
Basso et al. [21]	Dynamic stochastic EV routing	●	○	–	–	●	Safe RL with energy/reliability constraints	Reactive and anticipatory policies	Charging and request uncertainty; no lockers or MARL
Kullman et al. [22]	Dynamic electric ride-hailing	●	–	–	–	●	Deep RL with learned value approximations	Reoptimization and dual bounds	Ride-hailing assignment/repositioning, not parcel routing
Pan and Liu [30]	Dynamic and uncertain VRP	●	○	–	–	○	Deep RL under partial observation	Conventional routing comparators	Single-agent controller; no consolidation points
Zhou et al. [23]	Dynamic VRP with stochastic pickups	●	–	–	–	●	Forecast-informed RL; historical base policy	Nearest-neighbor heuristic	Single courier and demand arrivals; narrow baseline set
Grabenschweiger et al. [32]	Routing with heterogeneous locker boxes	–	–	●	–	–	Mathematical model and meta-heuristic; packing constraints	Literature instances and exact small cases	Static locker capacity and packing; no disruptions
Tilk et al. [6]	VRP with delivery options	–	–	●	–	○	Exact route-based optimization; service-level and capacity constraints	Exact algorithmic comparisons	Static planning; no learned execution
Lyu and Teo [33]	Locker network location and coverage	○	–	●	–	●	Facility-location model with customer choice	Scenario/model comparisons	Strategic location, not operational routing
Raviv [35]	Service-point location and capacity	●	–	●	–	○	Stochastic capacity model and facility location	Deterministic and stochastic formulations	Network design and occupancy rather than disruption routing
Dell’Amico et al. [7]	Pickup and delivery with lockers	–	–	●	–	–	MILP and branch-and-cut; time windows and locker choices	Exact formulation variants	Static exact routing; no forecasts or policy inference
Liu et al. [38]	Mobile-locker location-routing	○	○	●	–	○	Hybrid Q-learning with global/local search	Gurobi and genetic algorithm	Hybrid optimizer and mobile-locker stopovers; not pure MARL execution

Table 1. Cont.

Reference	Problem and Decision Mode	DU	RD	PL	MA	Real Network	Method/Constraint Treatment	Optimization Baselines	Main Difference from This Study
Janinhoff et al. [24]	Multitrip parcel routing with options	○	–	●	–	●	Data-driven mathematical optimization	Operational scenario comparisons	Tactical/operational delivery options without compound disruptions
Wang et al. [39]	Mobile-locker routing with uncertain demand	●	–	●	–	○	Stochastic model, recourse, tailored tabu search	No-recourse and algorithm variants	Recourse optimization; no multi-agent learned policy
Nazari et al. [11]	Static capacitated VRP	–	–	–	–	–	Attention-based constructive RL; capacity masking	Heuristics and solver-derived references	Canonical synthetic CVRP; no disruptions
Da Costa et al. [13]	Learned improvement for TSP/VRP	–	–	–	–	–	RL-guided 2-opt; explicit capacity checks	LKH and learned methods	Static improvement policy; no locker or road features
Arishi and Krishnan [46]	Multi-depot VRP	–	–	–	●	–	Multi-agent deep RL	Heuristic/learning comparisons	Multi-agent routing without compound disruption state
Mak et al. [47]	Fair collaborative vehicle routing	○	–	–	●	○	Coalitional bargaining with deep MARL	Shapley-value-related comparisons	Gain sharing rather than vehicle-to-consolidation-point routing
Wang et al. [25]	EV routing and energy scheduling	●	○	–	●	○	MARL under energy-system uncertainty	Scheduling and energy baselines	Energy coordination, not consolidation-point service continuity

3. Problem Description and Formulation

3.1. Operational Setting

The operational system contains one depot, analytically derived candidate parcel-locker consolidation points, and a homogeneous vehicle fleet. Parcel demand is aggregated at these candidate consolidation points. Vehicles select feasible points sequentially and may return to the depot to reload. Service is atomic: the full remaining demand at a selected point must fit the acting vehicle's remaining capacity. Demand, risk, closures, weather, and scenario information evolve during an episode. The shared objective prioritizes service while penalizing failures, distance, disruption exposure, infeasibility, and operational energy. Throughout, “candidate consolidation point” denotes an analytical routing node, while “parcel locker” denotes physical infrastructure.

Let $G = (V, A)$ be a directed graph. Node $0 \in V$ is the depot and $L = V \setminus \{0\}$ is the set of candidate consolidation points. The vehicle set is $K = \{1, \dots, m\}$. Point $i \in L$ has snapshot demand $q_i \geq 0$ —initial demand for static planning and revealed remaining demand for adaptive replanning—and vehicle $k \in K$ has capacity Q_k . The trip index $\mathcal{H}_k = \{1, \dots, \bar{h}_k\}$ permits repeated depot-to-depot trips. Arc $(i, j) \in A$ has saved OSRM distance d_{ij} , travel time τ_{ij} , normalized risk $r_{ij,t}$, and closure state $c_{ij,t} \in \{0, 1\}$; w_t and z_t encode weather and scenario information.

For an optimization snapshot at decision time t , let $x_{ij}^{kh} = 1$ if vehicle k traverses arc (i, j) on trip h , let $y_i^{kh} = 1$ if that trip serves candidate consolidation point i , let $y_i = 1$ if point i is served by any trip, and let $\delta_{kh} = 1$ if trip h is activated. All four variables are binary. Let $\bar{d}$ and $\bar{\tau}$ be the 95th-percentile distance and travel-time values fitted on the training set. The implemented revealed arc cost is

$$\ell_{ij,t} = 0.45 \frac{d_{ij}}{\bar{d}} + 0.45 \frac{\tau_{ij}}{\bar{\tau}} + 0.10 r_{ij,t}. \quad (1)$$

The deterministic comparators use the following lexicographic service-first objective:

$$\text{lexmin}_{x,y,\delta} \left(\sum_{i \in L} q_i (1 - y_i), \sum_{k \in K} \sum_{h \in \mathcal{H}_k} \sum_{(i,j) \in A} \ell_{ij,t} x_{ij}^{kh} \right), \quad (2)$$

implemented in two passes: the first minimizes unmet packages; the second fixes that optimum and minimizes normalized routing cost. Thus, no unspecified big- M service penalty is used. The service, routing, and per-trip capacity constraints are

$$\sum_{k \in K} \sum_{h \in \mathcal{H}_k} y_i^{kh} = y_i, \quad \forall i \in L, \quad (3)$$

$$\sum_{j \in V} x_{ij}^{kh} = \sum_{j \in V} x_{ji}^{kh} = y_i^{kh}, \quad \forall i \in L, k \in K, h \in \mathcal{H}_k, \quad (4)$$

$$\sum_{i \in L} q_i y_i^{kh} \leq Q_k \delta_{kh}, \quad \forall k \in K, h \in \mathcal{H}_k, \quad (5)$$

$$\sum_{j \in L} x_{0j}^{kh} = \sum_{i \in L} x_{i0}^{kh} = \delta_{kh}, \quad \forall k \in K, h \in \mathcal{H}_k, \quad (6)$$

$$x_{ij}^{kh} \leq 1 - c_{ij,t}, \quad \forall (i, j) \in A, k \in K, h \in \mathcal{H}_k. \quad (7)$$

Standard subtour-elimination constraints are imposed within each activated trip. For example, order variables π_i^{kh} can satisfy

$$y_i^{kh} \leq \pi_i^{kh} \leq |L| y_i^{kh}, \quad \forall i \in L, k \in K, h \in \mathcal{H}_k, \quad (8)$$

$$\pi_i^{kh} - \pi_j^{kh} + |L| x_{ij}^{kh} \leq |L| - 1 + |L| (2 - y_i^{kh} - y_j^{kh}), \quad \forall i \neq j \in L, k \in K, h \in \mathcal{H}_k. \quad (9)$$

Because y_i may equal zero, the formulation permits unmet demand and defines failed packages as $\sum_{i \in L} q_i(1 - y_i)$. Capacity is enforced separately on every depot-to-depot trip, so a vehicle can reload after returning to the depot without exceeding Q_k on any trip. The static OR-Tools baseline solves this planning snapshot before disruption; its realized service may subsequently fall when the retained plan encounters changed conditions. Limited repair modifies a restricted part of the incumbent plan after the state changes, whereas Full OR-Tools dynamic reoptimization rebuilds the remaining plan using revealed disruption information.

3.2. Assumptions and Scope

Table 2 summarizes the modeling boundary.

Table 2. Modeling assumptions and their implications.

Assumption	Operational Implication
Candidate consolidation points are fixed cluster representatives.	The model evaluates routing through analytical service nodes; installed-site selection and facility configuration belong to deployment planning.
Demand is aggregated and served atomically.	Per-trip vehicle capacity is enforced, and a depot return restores capacity for the next trip.
Vehicles are homogeneous and cooperative.	Agents share a team reward; heterogeneous or competing-carrier objectives require an extended game.
Road costs use a saved directed OSRM matrix.	All methods receive identical road-based distance and travel-time inputs.
Disruptions are calibrated simulation stress tests.	Results measure response in the constructed environment; local operational transfer requires field calibration.
The electric outcome is a routing-level proxy.	Energy and tailpipe CO ₂ are reported under the stated fleet assumption; lifecycle accounting requires charging, grid, production, infrastructure, and recipient-travel data.

3.3. Cooperative Multi-Agent Formulation

The routing process is represented as a cooperative partially observable Markov game

$$\mathcal{M} = \langle K, \mathcal{S}, \{\mathcal{O}^k\}_{k \in K}, \{\mathcal{A}^k\}_{k \in K}, P, R, \gamma \rangle. \quad (10)$$

The global state $s_t \in \mathcal{S}$ contains fleet and environment information; vehicle k receives a local observation $o_t^k \in \mathcal{O}^k$; the joint action is $\mathbf{a}_t = (a_t^1, \dots, a_t^m)$; P is the simulated state-transition mechanism; R is the shared reward; and γ is the discount factor. Centralized training uses the global state for value estimation. Decentralized execution means that the actor for vehicle k receives o_t^k and its feasibility mask, not the full critic input.

The team objective is

$$J(\theta) = \mathbb{E}_{\pi_\theta} \left[\sum_{t=0}^{T-1} \gamma^t R_t \right], \quad (11)$$

where the expectation is induced by the shared actor policy and the simulated disruption process. Parameter sharing is appropriate because the vehicles have the same decision role. Heterogeneous behavior can still arise because their positions, remaining capacities, candidate sets, and histories differ.

3.4. State and Observation Spaces

The global critic state is constructed from all vehicle positions and remaining capacities, the remaining and forecast demand vectors, served status, risk values, the closure mask, weather and scenario indicators, service progress, and fleet configuration:

$$s_t = [p_t^1, \dots, p_t^m, Q_t^1, \dots, Q_t^m, \mathbf{q}_t, \hat{\mathbf{q}}_t, \mathbf{v}_t, \mathbf{r}_t, \mathbf{c}_t, w_t, z_t, S_t, \kappa]. \quad (12)$$

Here, p_t^k and Q_t^k denote vehicle k 's current node and remaining capacity; $\mathbf{q}_t$ and $\hat{\mathbf{q}}_t$ denote remaining and forecast demand; $\mathbf{v}_t$ denotes served status; S_t denotes cumulative service; and κ denotes fleet configuration.

Let $M = 10$ denote the maximum number of ranked candidate nodes exposed to each vehicle at a decision step. The local observation uses the vehicle-specific variables and this candidate layer:

$$o_t^k = [p_t^k, Q_t^k, S_t, w_t, z_t, \kappa, \{\phi_{i,t}^k : i \in \mathcal{C}_{t,k}^{(M)}\}], \quad (13)$$

where $\mathcal{C}_{t,k}^{(M)}$ is vehicle k 's ranked set of up to M candidate nodes and $\phi_{i,t}^k$ contains node demand, forecast demand, distance or time from the vehicle, risk, closure status, and served status. Section 5.4 defines the ranking and mask relationship.

3.5. Action Space and Feasibility Conditions

At decision time t , vehicle k chooses a candidate consolidation point or the depot:

$$\mathcal{A}_t^k = \mathcal{C}_{t,k}^{(M)} \cup \{0\}. \quad (14)$$

Here, 0 represents the depot. Thus, routing actions are restricted to the current top- M candidate set and the depot; nodes outside this set cannot be selected at that decision step. The binary feasibility mask $m_t^k(a)$ is one if action a is currently admissible and zero otherwise. A candidate action is removed when the node is already fully served, its remaining demand is incompatible with the remaining vehicle capacity under the implemented service rule, or the connecting arc is closed; the depot action is masked when it violates the episode's depot logic. The masked policy is

$$\pi_\theta(a | o_t^k, m_t^k) = \frac{\exp(f_\theta(o_t^k)_a) m_t^k(a)}{\sum_{a' \in \mathcal{A}_t^k} \exp(f_\theta(o_t^k)_{a'}) m_t^k(a')}. \quad (15)$$

The environment retains at least a safe depot action whenever depot return is required, or no delivery action remains admissible. This rule prevents an empty feasible set.

3.6. Reward and Evaluation Objectives

The team reward is service first:

$$R_t = r_{\text{del}} \Delta y_t - \lambda_f F_t - \lambda_s \max(0, 0.95 - S_t) - \lambda_d \Delta \ell_t - \lambda_r X_t - \lambda_b B_t - \lambda_q C_t - \lambda_e E_t, \quad (16)$$

where Δy_t is newly delivered demand, F_t is failed demand, S_t is service rate, $\Delta \ell_t$ is traveled distance, X_t is risk exposure, B_t is blocked-edge exposure, C_t is a capacity-violation term, and E_t is an energy proxy. Delivered and failed demand are divided by episode demand; distance and energy are divided by their 95th-percentile training values; and risk, blocked-edge exposure, and capacity violations are scaled to $[0, 1]$ with training-set bounds. Under the full masked policy, C_t is zero by construction; B_t can be nonzero only when a closure becomes active after a movement has been committed. Both terms remain active

as defensive checks and make the no-mask ablation finite rather than silently discarding invalid transitions. The selected coefficient vector is given in Table 3.

Table 3. Reward coefficients and validation selection.

Component	Coefficient	Selection Role
Newly delivered demand r_{del}	1.00	Primary positive term
Failed demand λ_f	2.00	Service-failure penalty
Service deficit λ_s	5.00	Penalizes cumulative service below 0.95
Distance λ_d	0.05	Secondary route-efficiency term
Risk exposure λ_r	0.10	Secondary disruption-exposure term
Blocked movement λ_b	1.00	Strong penalty for blocked-edge attempts
Capacity violation λ_q	2.00	Strong infeasibility penalty
Energy proxy λ_e	0.02	Secondary operational-energy term

The coefficients were selected on the eight validation days using multiplicative candidates $\{0.5, 1, 2\}$ around the displayed vector. Configurations were ranked lexicographically by service, failed packages, recovery time, and energy, so secondary terms could not compensate for lower service. The test split was used once after the vector and policy checkpoint were fixed.

Performance is evaluated through service rate, failed packages, recovery time, runtime, distance, blocked-edge traversals, risk exposure, tailpipe CO₂, and energy. The main cross-method results available for all baselines are service rate, failed packages, recovery time, and runtime. Energy is reported for the selected electric proxy, while lifecycle environmental comparisons are not made.

4. Data and Simulation Environment

4.1. Study Environment

The Los Angeles simulation combines geocoded delivery stops, a directed road-network cost matrix, and scenario features. All methods receive the same candidate consolidation points, demand, fleet setting, and scenario state for each matched case.

4.2. Operational Fidelity and Transfer Boundary

Data-grounded elements comprise spatial demand, chronological daily variation, and directed OSRM costs; constructed elements comprise the candidate consolidation points and calibrated disruption generator. Table 4 states the corresponding transfer requirements and keeps the empirical interpretation at the level of disruption-aware routing through analytical consolidation points.

Table 4. Operational-fidelity elements and transfer requirements.

Dimension	Represented Element	Requirement for Field Transfer
Urban network	Directed OSRM road distance and travel time	Live traffic, access restrictions, service and dwell times
Demand	Geocoded stops, daily variation, chronological split	Carrier manifests, cancellations, time windows, recipient behavior
Fleet operation	Multiple vehicles, per-trip capacity, depot reloads	Shift rules, heterogeneous vehicles, driver constraints, charging or fueling
Disruption response	Revealed demand, risk, closures, weather, and scenario state	Locally calibrated frequencies, correlations, detection delay, and forecast error
Consolidation points	Cluster-derived analytical service nodes	Installed sites, compartment mix, occupancy, overflow, and customer acceptance

4.3. Demand and Candidate Consolidation-Point Construction

Candidate consolidation points are obtained from the 24 training days. Stop coordinates are projected to California Albers coordinates (EPSG:3310); *k*-means++ with 50 initializations selects the lowest within-cluster sum of squares for the reference $n = 40$ clusters. Each centroid is snapped to the nearest observed training stop, with the stop identifier breaking ties. Validation and test stops are assigned to the nearest fixed centroid and daily demand is aggregated accordingly. Density tests repeat the pipeline at $n = 30$ and $n = 50$. These analytical points remain distinct from installed parcel-locker infrastructure.

For candidate i on day d , the one-day-ahead demand feature is an exponentially weighted mean,

$$\hat{q}_{i,d} = 0.35q_{i,d-1} + 0.65\hat{q}_{i,d-1}, \quad (17)$$

initialized by the mean of the first seven training days. Model selection and scaling use the 24 training days only. Validation and test forecasts are generated with a rolling origin: on day d , only demand observed before d is available, and no future day enters clustering, scaling, feature construction, or model selection. This protocol follows leakage-aware forecast evaluation practice [49]. On the validation split, the forecast produced an MAE of 7.84 packages per candidate-day, an RMSE of 11.62, and a WAPE of 14.7%; the corresponding held-out test values were 8.11, 12.05, and 15.2%.

4.4. Road Network and OSRM Matrix

Road distance d_{ij} and travel time τ_{ij} are read from a saved OSRM Table API matrix. Caching the matrix ensures that every method uses the same arc costs and that the final evaluation does not silently fall back to straight-line distance. The directed formulation permits asymmetric travel times or distances when the road network produces them. Closure scenarios act on the eligible arc set or its cost representation rather than replacing the saved base matrix with Euclidean geometry.

4.5. Risk, Weather, and Road-Closure Information

Road risk is derived from the City of Los Angeles Traffic Collision Data portal [50]. Only collision records dated before the evaluation period are used. A Gaussian kernel with a 500 m bandwidth converts geocoded collisions into a spatial intensity at each candidate consolidation point; arc risk is the mean of the origin and destination intensities. Values are winsorized at the 1st and 99th percentiles of the training data and min–max normalized with those fixed training bounds. Missing coordinates receive the training median before normalization. Weather applies the scenario multiplier in Table 5; it does not change the fitted spatial risk surface. Road-closure variables remain binary and enter both the policy state and feasibility mask. Thus, risk is a graded exposure feature, whereas closure is an immediate admissibility condition.

Table 5. Compound-disruption generator and calibration levels.

Level	Demand Multiplier	Risk Multiplier	Closed Arcs	Travel-Time Multiplier	Closure Duration
Mild	1.00	1.00	1%	1.00	60 min
Moderate	1.15	1.25	3%	1.10	180 min
Severe	1.35	1.60	8%	1.25	Episode remainder

4.6. Training, Validation, and Test Division

The data are divided into 24 training days, 8 validation days, and 12 held-out test days. The division is chronological and nonoverlapping at the day level. Ten random seeds are used. The frozen selected policy is evaluated on the held-out days and scenarios; training

does not continue during test execution. The balanced main test contains seven methods, seven predeclared scenarios, ten seeds, and twelve days, for $7 \times 7 \times 10 \times 12 = 5880$ method–scenario–seed–day records. Paired tests match observations by seed, day, and scenario.

4.7. Disruption Scenarios

The disruption generator uses a shared episode-level severity draw so that demand, risk, travel-time degradation, and closures are correlated rather than sampled independently. Conditional on the severity level, node demand is drawn from a Poisson distribution with mean equal to forecast demand times the demand multiplier and truncated at the full-trip vehicle capacity to preserve the atomic service rule. Seventy percent of closed arcs are sampled from the highest risk quartile and the remainder from all eligible arcs. A connectivity check rejects a draw that disconnects every depot path to a still-serviceable candidate. The disruption is activated and revealed to every adaptive method at simulation minute 120; closure duration is measured from that epoch.

The seven held-out scenarios are mild reference, demand variation, risk elevation, road closures, adverse weather, moderate compound, and severe compound. The four single-factor scenarios apply only the corresponding moderate multiplier from Table 5; the moderate and severe compound scenarios apply the complete row, and the mild reference applies the mild row. This construction yields the seven-scenario balanced count reported above. The mild and moderate levels span the central training-day variation. The severe level combines the upper demand and risk tails with an adverse-weather travel-time factor and the largest closure fraction that preserved at least one depot-connected feasible component in calibration. It is therefore a controlled stress test rather than an estimate of historical event frequency. Matched methods receive identical sampled demand, multipliers, closures, reveal time, and random seed.

4.8. Fleet and Environmental Assumptions

The reference fleet contains four vehicles with $Q_k = 300$ package-equivalents per depot-to-depot trip; capacity tests use 240 and 360. The selected electric proxy reports operational energy in kWh and zero tailpipe CO₂ by assumption. Its boundary is routing-level operation; lifecycle accounting requires battery, charging, grid, production, infrastructure, and recipient-travel data.

4.9. Temporal and Event-Order Model

Each episode covers 2160 simulated minutes, and the disruption reveal at minute 120 leaves a 2040 min recovery window. Vehicles operate asynchronously. The simulator advances to the next vehicle-arrival, service-completion, depot-reload, disruption, or demand-realization event; ties between vehicles are resolved by vehicle identifier. At a shared timestamp, exogenous demand, weather, risk, and closure updates are applied first, completed travel and service events second, and new routing decisions third. Travel advances the acting vehicle's clock by the saved OSRM travel time multiplied by the active scenario factor. Candidate service requires 5 min, and a depot return requires 15 min to reload to Q_k . An episode ends when all demand is served, and all vehicles have returned to the depot, or when minute 2160 is reached. This event order ensures that every adaptive method observes the same revealed state before acting.

Table 6 summarizes the experimental settings.

Table 6. Experimental settings used in the evaluation.

Item	Value
Study environment	Los Angeles urban last-mile simulation
Train/validation/test days	24/8/12
Random seeds	10
Final test records	5880 rows: 7 methods $\times$ 7 scenarios $\times$ 10 seeds $\times$ 12 days
Training episodes per seed	1500
Road costs	Saved OSRM Table API matrix
Candidate consolidation points	40 reference candidates; 30 and 50 in density tests
Dynamic information	Demand forecasts, road risk, closures, weather, and scenario indicators
Severe stress test	Severe compound-disruption scenario; 120 runs per reported method
Selected fleet	4 vehicles; 300 package-equivalents per depot-to-depot trip
Episode clock	2160 min horizon; disruption revealed at minute 120; asynchronous event advance
Service/depot reload	5 min per candidate/15 min per reload
Fleet interpretation	Electric routing proxy; energy in kWh and zero tailpipe CO ₂ by assumption

5. Proposed MAPPO-Based Routing Approach

5.1. Methodological Positioning

The methodological contribution is a routing-specific framework that maps a changing global consolidation-point problem into aligned vehicle candidate blocks, uses a shared actor for cooperative actions, supplies fleet-level information to the training critic, and applies structural admissibility before action selection. Forecast, risk, closure, weather, and scenario features are refreshed at each decision epoch. MAPPO and CTDE provide the optimization and training architecture for this integrated decision interface; the contribution is the complete problem-specific composition and its comparative evaluation.

5.2. Centralized Training and Decentralized Execution

The proposed solver-independent MAPPO policy uses a shared actor and a centralized value function. During training, trajectories from all vehicles are collected under the current actor. The critic receives s_t and estimates the expected team return, allowing advantage estimates to incorporate fleet-wide progress. During execution, vehicle k receives o_t^k and m_t^k and samples or selects its next node from the masked actor. The critic is not required to choose an operational action.

Figure 1 summarizes the information flow. Solid arrows identify execution-time data. Dashed arrows identify training-only value information. The separation between the local actor input and global critic state is the operational meaning of CTDE in this implementation.

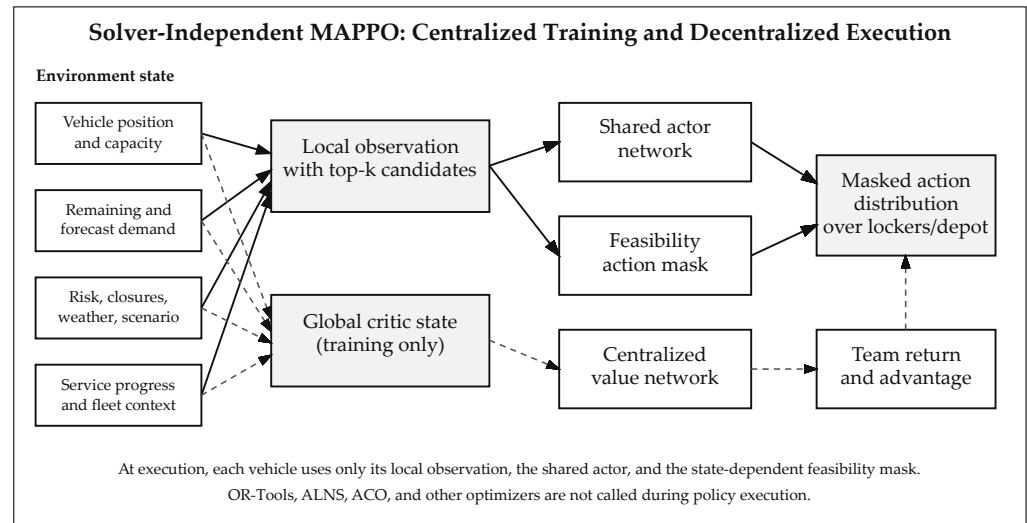


Figure 1. Architecture of the proposed solver-independent MAPPO policy. Local top- k observations and feasibility masks generate vehicle actions. Fleet-wide state and team returns are used by the centralized critic during training only. OR-Tools, ALNS, ACO, and other optimization procedures are external comparators and are not called during policy execution.

5.3. Actor and Critic Architecture

The actor standardizes continuous inputs with training-set means and standard deviations, applies two fully connected 128-unit layers with tanh activation, and maps the final representation to $M + 1$ logits for the candidate slots and depot. The critic applies the same two-layer 128-unit tanh architecture to the global state and returns one scalar value. Linear layers use orthogonal initialization with zero biases. Actor parameters are shared across the four homogeneous vehicles. Sharing reduces model count and allows experience collected by any vehicle to update the common decision rule. Vehicle identity is expressed through local state rather than a distinct policy network.

5.4. Observation Construction and Top- k Candidates

At each decision point, candidate features are updated from remaining demand, the demand forecast, road cost from the vehicle's current position, risk, closure status, and served status. For every not-yet-served candidate, demand, forecast, travel time, and risk are scaled to $[0, 1]$ with training-set bounds. Vehicle k ranks candidate i by

$$g_{i,t}^k = 0.45\tilde{q}_{i,t} + 0.20\tilde{\hat{q}}_{i,t} - 0.25\tilde{\tau}_{p_i^k} - 0.10\tilde{r}_{p_i^k,t}. \quad (18)$$

Candidates are sorted in descending score order, with the candidate identifier as a deterministic tie-breaker, and the first ten are placed in the extended observation layer. Ranking occurs before the capacity and closure mask is applied: served nodes are excluded from the pool, while capacity-incompatible and closed actions may enter the top ten but receive zero probability through the mask. Padding and an associated feasibility value preserve a fixed actor input when fewer than ten delivery actions remain. Fleet and scenario context is concatenated with this candidate block.

The implemented value $M = 10$ bounds both actor input and the current decision set. A node outside the top ten becomes selectable when it enters the ranked layer. Matched retraining at $M \in \{5, 10, 15, 20\}$ evaluates the coverage–performance–runtime trade-off (Section 7.6).

5.5. Action Masking

Before the softmax calculation, the environment constructs m_t^k . Logits for invalid actions receive zero probability through Equation (15). The mask is recomputed after each state transition. It eliminates served nodes, capacity-incompatible moves, closed arcs, and invalid depot choices before sampling. This is externally imposed structural admissibility, not learned constraint satisfaction. Removing the mask leaves the same discrete action representation but removes its feasibility layer; the resulting ablation measures the architecture's dependence on explicit constraint processing rather than the policy's ability to learn constraints.

5.6. Reward Function

Equation (16) prioritizes newly delivered demand and penalizes failure and sub-threshold service before secondary movement, risk, closure, capacity, and environmental terms. The shaping is cooperative: every vehicle receives the same scalar team reward. This reduces direct competition for high-demand nodes but creates a credit-assignment problem because an individual action's contribution is filtered through later fleet behavior. The centralized critic provides a fleet-level value estimate, and its measured service effect is reported in the ablation.

5.7. MAPPO Training Procedure

For policy parameters θ , the shared actor is optimized from per-agent samples. For vehicle k , define the importance ratio

$$\rho_t^k(\theta) = \frac{\pi_\theta(a_t^k | o_t^k, m_t^k)}{\pi_{\theta_{\text{old}}}(a_t^k | o_t^k, m_t^k)}. \quad (19)$$

The clipped surrogate objective is

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_{t,k} \left[\min \left(\rho_t^k(\theta) \hat{A}_t^k, \text{clip}(\rho_t^k(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t^k \right) \right], \quad (20)$$

where $\hat{A}_t^k$ is the team-return advantage attached to vehicle k 's sampled action and is computed with generalized advantage estimation [51]. With temporal-difference residual

$$\delta_t = R_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t), \quad (21)$$

the estimate is

$$\hat{A}_t = \sum_{l=0}^{T-t-1} (\gamma \lambda)^l \delta_{t+l}. \quad (22)$$

The per-agent training sample uses the corresponding team advantage, $\hat{A}_t^k = \hat{A}_t$. The bootstrapped critic target is explicitly $\hat{R}_t = \hat{A}_t + V_{\phi_{\text{old}}}(s_t)$ for nonterminal steps, with the terminal value set to zero. The critic minimizes

$$L^V(\phi) = \mathbb{E}_t \left[(V_\phi(s_t) - \hat{R}_t)^2 \right], \quad (23)$$

and the implemented loss combines the negative clipped objective, value loss, and entropy regularization.

The training and held-out evaluation procedure is summarized in Algorithm 1.

Algorithm 1. MAPPO training and held-out evaluation.

1. Initialize the Los Angeles simulation from demand, candidate consolidation points, risk, scenario, fleet, and saved OSRM inputs.
2. Initialize the shared actor and centralized critic.
3. For each of the 1500 training episodes per seed, sample a training day and disruption scenario.
4. For every active vehicle, build o_t^k , construct m_t^k , select a masked action, and apply the joint action to the environment.
5. Update delivered packages, remaining demand, distance, risk and closure exposure, energy, service, and recovery status; then compute the shared reward.
6. Store 512 rollout steps, estimate advantages, and update actor and critic for 12 PPO epochs.
7. Use the validation split to select the saved policy; freeze it before evaluation.
8. Evaluate on the 12 held-out days and ten seeds, and compare the resulting records with all external baselines.

5.8. Solver-Independent Policy Execution

Solver-independent execution begins after training and validation. A decision consists of observation construction, mask construction, and an actor forward pass. Candidate consolidation points, forecasts, and the saved road matrix are preprocessing inputs; action selection itself is completed by the frozen policy without an OR-Tools model, ACO population, or ALNS destroy–repair loop.

6. Baselines and Experimental Protocol**6.1. Common Information and Decision Authority**

Every method receives the same consolidation-point coordinates, daily demand instance, vehicle capacities, depot, and saved directed OSRM cost matrix for a matched seed–day–scenario case. Static OR-Tools receives the pre-disruption snapshot and retains its plan. Greedy, ACO, ALNS, limited repair, Full OR-Tools dynamic reoptimization, and MAPPO operate on the same realized case, but their permitted adaptation differs by design. Limited repair modifies a restricted part of the incumbent plan; ALNS applies repeated destroy–repair moves; Full OR-Tools dynamic reoptimization can rebuild the complete unexecuted plan; and MAPPO selects one action from its current candidate set. These authority differences are part of the comparator definitions and must be considered together with runtime.

The timing boundary begins after the matched case and saved road matrix have been loaded and ends when the method returns the action or revised route used by the simulator. Per-episode runtime is the sum of every online call made during the complete 2160 min episode. For MAPPO, it includes every observation construction, mask construction, and actor inference; for search methods, it includes every construction, repair, or reoptimization call through its stopping condition. Shared file I/O, candidate clustering, matrix loading, and offline MAPPO training are excluded. Timing uses Python’s high-resolution monotonic clock after 25 warm-up episodes. The returned decision units are not identical—MAPPO returns one next action, whereas complete reoptimization can return an entire remaining plan—so runtime is interpreted as observed episode-level online computation under each method’s stated authority, not as equal computational work per call.

All methods were executed in Python 3.11.7 on Windows 11 using one CPU thread on an AMD Ryzen 7 5800H workstation with 32 GB RAM. OR-Tools 9.10 and PyTorch 2.2.1 were used; MAPPO inference was CPU-only. Baseline settings were chosen on the eight validation days with three tuning seeds. Each stochastic baseline received 30 candidate configurations. Selection was lexicographic—service, failed packages, recovery, then runtime—and the selected configuration was frozen before the held-out test.

Every adaptive search call had a 0.50 s ceiling, while iteration, stagnation, or solution limits could terminate it earlier.

6.2. Greedy Nearest

Greedy nearest repeatedly selects the closest feasible unserved candidate consolidation point from the current vehicle position. It returns to the depot when required by capacity or completion. The method provides a low-computation constructive reference and does not perform look-ahead search.

6.3. Ant Colony Optimization

ACO constructs solutions probabilistically. The selected configuration uses 20 ants, at most 50 iterations, and a ten-iteration stagnation stop. The parameters are $\alpha = 1$, $\beta = 2$, evaporation $\rho = 0.10$, and exploitation probability $q_0 = 0.90$. Initial pheromone is $1/(nL_{NN})$, where L_{NN} is the greedy feasible-route cost; the global-best solution deposits pheromone, and 2-opt is applied after construction. For ant h at node i , the transition probability to feasible node j is

$$p_{ij}^h(t) = \frac{[\xi_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^h} [\xi_{il}(t)]^\alpha [\eta_{il}]^\beta}, \quad (24)$$

where ξ is pheromone, η is heuristic desirability, and $\mathcal{N}_i^h$ is the feasible neighborhood. Pheromone is updated by

$$\xi_{ij}(t+1) = (1-\rho)\xi_{ij}(t) + \sum_h \Delta \xi_{ij}^h(t). \quad (25)$$

The tuning grid used 10, 20, or 40 ants; 25, 50, or 100 iterations; $\beta \in \{1, 2, 3\}$; $\rho \in \{0.10, 0.20, 0.40\}$; and $q_0 \in \{0.80, 0.90\}$. Random configurations were sampled without replacement and evaluated under the common 0.50 s ceiling. ACO is an external population-based comparator and is not used to repair MAPPO actions.

6.4. Adaptive Large Neighborhood Search

ALNS repair starts from the disrupted incumbent and uses four destroy operators—random, worst-cost, related-node, and contiguous-route removal—with greedy, regret-2, and regret-3 insertion. Each iteration removes 10–30% of currently unexecuted candidates. Operator weights start at one and are updated every 50 iterations with a reaction factor of 0.10 and scores of 33, 9, and 3 for a new global best, an improving accepted solution, and an accepted non-improving solution. Simulated annealing accepts a deterioration Δ with probability $\exp(-\Delta/T)$; T_0 gives 0.50 acceptance for a 5% deterioration and cools by 0.995 per iteration. Search stops at 500 iterations, 75 stagnant iterations, or 0.50 s. The tuning budget varied the removal range, reaction factor, cooling, and the iteration/stagnation limits. The label “ALNS repair only” distinguishes this route repair from complete replanning [9,10].

6.5. Static OR-Tools

Static OR-Tools computes a capacitated route plan before disruption with parallel cheapest insertion followed by guided local search, a 0.50 s ceiling, and a 200-solution limit. It receives no revealed disruption state and makes no post-dispatch call. It represents optimized pre-dispatch planning without adaptation. OR-Tools exposes these strategies and explicit time and solution limits for reproducible routing search [52].

6.6. Limited OR-Tools Repair

Limited OR-Tools repair freezes completed movements, the next committed stop of each active vehicle, and all unaffected route segments. It may reassign or resequence only candidates incident to a newly closed arc or with a demand deviation above 15%. The repair uses parallel cheapest insertion, guided local search, a 0.05 s ceiling, and a 50-solution limit. It provides an intermediate comparator between static execution and unrestricted replanning.

6.7. Full OR-Tools Dynamic Reoptimization

Full OR-Tools dynamic reoptimization is triggered when the disruption is revealed and whenever a subsequent closure or demand realization invalidates the active plan. Completed movements are fixed, but every unexecuted candidate may be reassigned and resequenced. The incumbent is supplied as a warm start, followed by parallel cheapest insertion when needed and guided local search; the call uses a 0.50 s ceiling, a 200-solution limit, one thread, and the matched scenario seed. It is the strongest dynamic OR-Tools comparator because it can revise the complete unexecuted plan.

Table 7 summarizes the external-baseline settings and the MAPPO execution protocol.

Table 7. Reproducibility settings for the external baselines.

Method	Selected Implementation and Stopping Rule
Greedy nearest	Closest feasible unserved candidate; deterministic; stop when all feasible demand is served, or the horizon is reached.
ACO	20 ants; 50 iterations; stagnation 10; $\alpha = 1$, $\beta = 2$, $\rho = 0.10$, $q_0 = 0.90$; global-best update; 2-opt; 0.50 s ceiling.
ALNS repair	Four destroy and three repair operators; 10–30% removal; reaction 0.10; simulated annealing; cooling 0.995; 500 iterations; stagnation 75; 0.50 s ceiling.
Static OR-Tools	Parallel cheapest insertion; guided local search; 200-solution/0.50 s limit; pre-dispatch call only.
Limited OR-Tools repair	Completed and committed route portions fixed; affected candidates only; parallel cheapest insertion; guided local search; 50-solution/0.05 s limit.
Full OR-Tools dynamic reoptimization	Completed movements fixed; all unexecuted decisions open; incumbent warm start; parallel cheapest insertion; guided local search; 200-solution/0.50 s limit.
MAPPO	Frozen shared actor; CPU inference; observation, mask, and forward pass included; no test-time optimization.

6.8. Hyperparameters and Training Protocol

Table 8 lists the selected MAPPO settings. Each seed is trained for 1500 episodes using the 24-day training set, assessed on eight validation days, and frozen before the 12-day test. The baseline methods are executed externally on the same test instances.

Table 8. Selected MAPPO hyperparameters.

Parameter	Value
Actor learning rate	0.001
Critic learning rate	0.0005
Discount factor γ	0.99
GAE parameter λ	0.90
PPO clipping parameter ϵ	0.30
Entropy coefficient	0.01
PPO epochs per update	12
Rollout steps	512
Hidden representation size	128
Top- k observation	10
Training episodes per seed	1500

6.9. Evaluation Metrics

Service rate is the proportion of demand served, and failed packages count unmet demand. The recovery clock starts when disruption is revealed at minute 120 and stops at the first subsequent epoch with at least 95% cumulative service. This threshold represents near-complete restoration while reserving the final 5% for exception handling. Cases below the threshold at minute 2160 are right-censored at 2040 min and reported as *not recovered* (N/R). Descriptive means use recovered episodes with the N/R share; inference uses restricted mean time to recovery so every episode remains represented [53]. Runtime is summed online computation per episode under the common timing boundary.

6.10. Statistical Analysis

MAPPO is paired with static OR-Tools and Full OR-Tools dynamic reoptimization using the same seed, day, and scenario. Outcomes are first averaged across matched scenarios within each seed–day cell. Because the same 12 held-out days are crossed with all 10 seeds, dependence-aware intervals use 10,000 two-way pigeonhole bootstrap replicates that resample seed indices and day indices independently while retaining every matched method and scenario in the selected cells [54]. Exact Wilcoxon tests are applied to the ten seed-aggregated mean differences and Holm-adjusted across the six primary method–metric contrasts; the twelve day-aggregated differences are reported as a crossed-factor directional diagnostic. Recovery contrasts use restricted mean time to recovery, so N/R episodes remain in the analysis. Record-level results are retained only as a precision check. The sign convention is MAPPO minus comparator. Positive service differences and negative failed-package or recovery differences favor MAPPO.

7. Results

7.1. Overall Method Comparison

Table 9 reports the aggregate comparison. MAPPO served 98.4% of demand versus 96.6% for static OR-Tools, reduced failed packages from 180.461 to 86.328, shortened conditional mean recovery from 1055.710 to 862.487 min, and reduced N/R from 18.3% to 9.2%. ACO, ALNS repair, and Full OR-Tools dynamic reoptimization achieved the highest service at 99.6–99.7%, establishing the quality side of the trade-off.

Table 9. Overall method comparison.

Method	Service	Failed Packages	Recovery (min)	N/R	Runtime (s)
ACO	0.996	20.735	814.749	2.5%	0.11100
ALNS repair only	0.997	13.944	703.143	1.7%	0.50300
Greedy nearest	0.981	95.798	878.715	10.8%	0.00046
Full OR-Tools dynamic reoptimization	0.997	13.944	696.373	1.7%	0.30200
OR-Tools limited repair	0.980	98.181	899.753	11.7%	0.00200
OR-Tools static	0.966	180.461	1055.710	18.3%	0.00300
Proposed MAPPO	0.984	86.328	862.487	9.2%	0.02900

Note: Recovery is the mean among recovered episodes; N/R is the share right-censored at the episode horizon. Runtime is the sum of all online calls per episode.

7.2. Severe Compound-Disruption Results

Under severe compound disruption, MAPPO served 93.0% versus 89.4% for static OR-Tools, reduced failed packages from 622.200 to 397.260, shortened conditional recovery from 1756.610 to 1062.830 min, and reduced N/R from 31.7% to 22.5% (Table 10). ACO and the two strongest adaptive methods reached 98.8–99.0% service, preserving the same quality ordering (Figure 2).

Table 10. Severe compound-disruption scenario comparison.

Method	Service	Failed Packages	Recovery (min)	N/R	Runs
ACO	0.988	61.355	1179.490	5.0%	120
ALNS repair only	0.990	48.825	964.023	3.3%	120
Full OR-Tools dynamic reoptimization	0.990	48.825	994.386	3.3%	120
OR-Tools static	0.894	622.200	1756.610	31.7%	120
Proposed MAPPO	0.930	397.260	1062.830	22.5%	120

Note: Recovery is conditional on recovery; N/R reports right-censoring at 2040 min after disruption reveal.

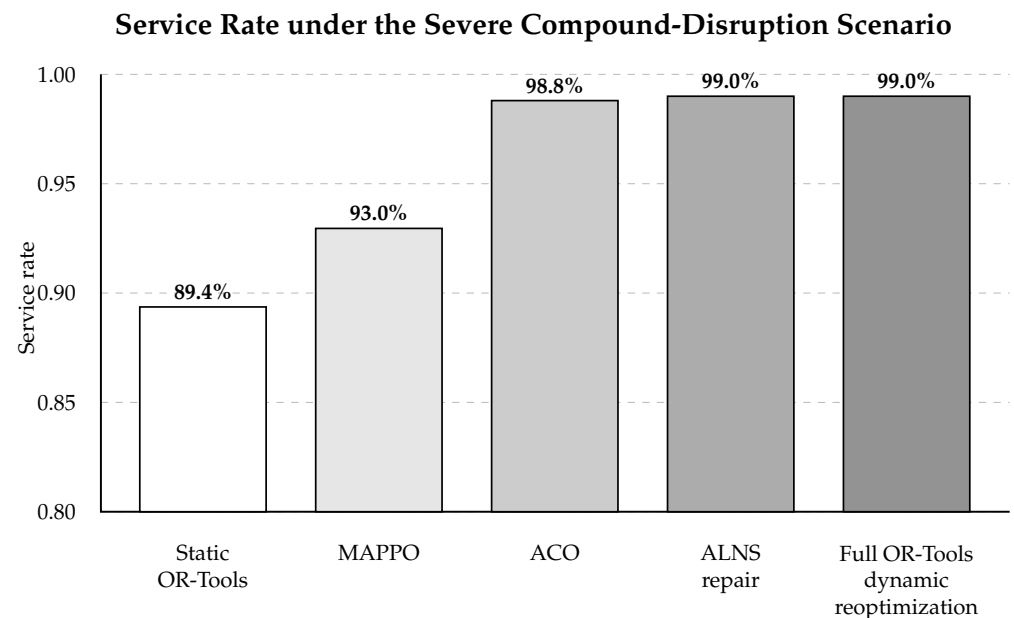


Figure 2. Service rate under the severe compound-disruption scenario. MAPPO improves service relative to static OR-Tools, while ACO, ALNS repair, and Full OR-Tools dynamic reoptimization obtain the highest service. All labels report service as percentages rounded to one decimal place.

7.3. Scenario-Specific Heterogeneity

MAPPO exceeded static OR-Tools across demand, risk, closure, weather, and moderate compound scenarios, with the largest service differences under closures and compound disruption (Table 11).

Table 11. Scenario-specific performance of MAPPO and static OR-Tools.

Scenario Family	Method	Service	Failed Packages	Recovery (min)
Demand variation	Proposed MAPPO	0.991	47.120	789.230
	OR-Tools static	0.982	97.400	947.150
Risk elevation	Proposed MAPPO	0.989	58.870	809.640
	OR-Tools static	0.976	127.230	992.880
Road closures	Proposed MAPPO	0.978	119.540	948.210
	OR-Tools static	0.950	265.770	1181.220
Adverse weather	Proposed MAPPO	0.987	67.180	835.460
	OR-Tools static	0.972	148.620	1028.430
Moderate compound	Proposed MAPPO	0.975	138.930	929.895
	OR-Tools static	0.949	263.285	1128.870

7.4. Service-Quality and Runtime Trade-Off

MAPPO required 0.029 s of summed online computation per episode, compared with 0.111 s for ACO, 0.302 s for Full OR-Tools dynamic reoptimization, and 0.503 s for ALNS repair (Figure 3). All remained subsecond on the test hardware. The absolute differences therefore represent computational headroom for repeated, parallel, or tightly

timed decisions rather than a universal operational delay; method selection also depends on the additional service and replanning authority delivered by adaptive search.

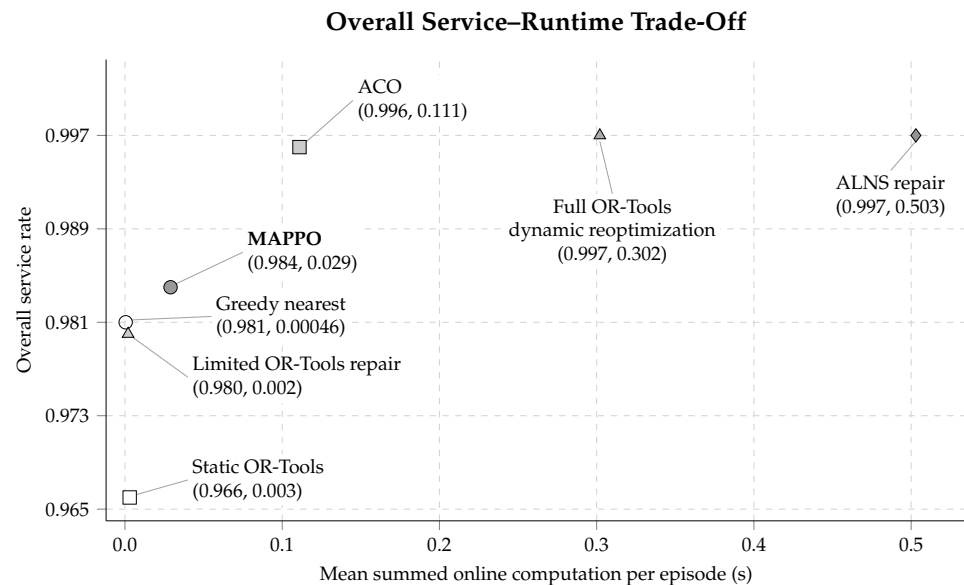


Figure 3. Overall service and runtime from Table 9. MAPPO provides a low-latency state-conditioned policy, whereas ACO, ALNS repair, and Full OR-Tools dynamic reoptimization obtain higher service with greater computation in the evaluated implementation.

7.5. Robustness and Scalability

The frozen policy was tested under fleet, demand, capacity, candidate-density, and severity shifts. Service remained above 0.94 with two vehicles or 0.8 times the reference capacity and reached 0.919 at 1.4 times the demand; computation grew approximately linearly from two to eight vehicles (Table 12).

Table 12. Zero-shot robustness and scalability of the frozen MAPPO policy.

Evaluation Condition	Service	Failed Packages	Recovery (min)	Runtime (s)
Fleet: 2 vehicles	0.941	318.400	1008.600	0.018
Fleet: 4 vehicles (reference)	0.984	86.328	862.487	0.029
Fleet: 6 vehicles	0.990	55.200	809.800	0.041
Fleet: 8 vehicles	0.992	44.600	785.300	0.053
Demand: 0.8×	0.992	34.900	774.200	0.028
Demand: 1.2×	0.957	276.500	997.100	0.030
Demand: 1.4×	0.919	604.800	1143.700	0.031
Capacity: 0.8×	0.944	298.600	1015.400	0.029
Capacity: 1.2×	0.991	48.900	808.600	0.029
Candidates: 0.75× density	0.974	139.800	912.200	0.028
Candidates: 1.25× density	0.986	76.400	848.500	0.030
Severe compound disruption	0.930	397.260	1062.830	0.031

7.6. Top- k Sensitivity

Matched policies at $k \in \{5, 10, 15, 20\}$ show that service increased from 0.968 to 0.984 between $k = 5$ and $k = 10$, while gains beyond ten were smaller and runtime increased. The selected $k = 10$ is therefore a reasonable performance–coverage–runtime balance for this setup (Table 13).

Table 13. Sensitivity to the structural candidate-set size.

<i>k</i>	Service	Failed Packages	Recovery (min)	Runtime (s)	Coverage
5	0.968	172.300	907.400	0.023	71.4%
10	0.984	86.328	862.487	0.029	86.8%
15	0.986	76.100	851.200	0.036	94.1%
20	0.987	70.200	846.000	0.044	97.6%

7.7. Feature and Component Ablation

Table 14 reports independently retrained ablation models. Removing the action mask collapsed service to 0%, confirming its structural role in the discrete action representation. Removing closure, forecast, and risk information reduced service to 0.453, 0.924, and 0.944, respectively; the extended observation ablation reached 0.973. Centralized-critic and environmental-reward variants remained close to the 0.985 reference on service, with their intended effects examined below.

Table 14. MAPPO ablation summary.

Variant	Service	Failed Packages	Recovery (min)	N/R
Full proposed MAPPO	0.985	88.254	856.396	8.3%
Without environmental reward	0.988	69.132	841.705	7.5%
Without action masks	0.000	5333.520	N/R	100.0%
Without extended observation layer	0.973	154.962	851.478	14.2%
Without centralized critic	0.987	73.958	839.849	7.5%
Without demand forecast in state	0.924	400.247	863.608	36.7%
Without risk features in state	0.944	306.285	895.700	29.2%
Without road-closure features	0.453	2915.100	474.101	79.2%

Note: Recovery means use recovered episodes; N/R is the share for which 95% service recovery was not reached within the episode horizon.

7.8. Environmental-Reward and CTDE Separation

Removing the environmental term increased energy from 202.121 to 207.864 kWh and the distance index by 2.7%, while service changed from 0.985 to 0.988 (Table 15). In the architecture separation, a shared actor with a local critic achieved 0.987 service, the centralized-critic reference achieved 0.985, and independent actor–critic pairs achieved 0.977 (Table 16). The clearest service contribution therefore comes from shared policy learning under the team reward; centralized value learning remains part of the training framework without improving the primary service metric in this setting.

Table 15. Energy outcome for the environmental-reward ablation.

Variant	Service	Energy (kWh)	Distance Index	Runtime (s)
Full proposed MAPPO	0.985	202.121	1.000	0.029
Without environmental reward	0.988	207.864	1.027	0.029

Table 16. Separation of centralized value learning and parameter sharing.

Training Architecture	Service	Failed Packages	Recovery (min)	Runtime (s)
Shared actor + centralized critic	0.985	88.254	856.396	0.029
Shared actor + local critic	0.987	73.958	839.849	0.029
Independent actors + independent critics	0.977	129.600	889.300	0.036

7.9. Statistical Comparisons

Record-level paired checks preserve both principal orderings: MAPPO improves service, failures, and restricted recovery relative to static OR-Tools, while Full OR-Tools dynamic reoptimization leads MAPPO on the same metrics (Appendix A, Table A1).

7.10. Dependence-Aware Inference and Recovery Thresholds

Crossed seed/day inference confirms both orderings for service, failed packages, and restricted recovery: all ten seed means and all twelve day means agree in direction, the Holm-adjusted seed-level $p = 0.012$, and every crossed 95% interval excludes zero (Table 17). The MAPPO advantage over static planning also persists at 90%, 95%, and 98% recovery thresholds (Appendix A, Table A2).

Table 17. Crossed seed/day dependence-aware confirmation; difference is MAPPO minus comparator.

Comparison	Metric	Mean Difference	Seed Holm p	Day Sign p	Crossed 95% CI
MAPPO vs. static OR-Tools	Service	0.0182	0.012	0.00049	[0.0110, 0.0255]
	Failed packages	−94.1325	0.012	0.00049	[−132.6000, −58.4000]
	Restricted recovery	−193.223	0.012	0.00049	[−278.500, −119.800]
MAPPO vs. Full OR-Tools dynamic re-optimization	Service	−0.0129	0.012	0.00049	[−0.0213, −0.0061]
	Failed packages	72.3844	0.012	0.00049	[45.8000, 105.7000]
	Restricted recovery	166.114	0.012	0.00049	[101.900, 236.400]

8. Discussion

8.1. Response to the Research Gap

The framework fills the identified gap by integrating cooperative policy execution, dynamic consolidation-point observations, compound-disruption features, and structural feasibility in one routing environment. Its comparison with static planning, limited repair, metaheuristics, and complete reoptimization locates policy inference within the broader dynamic-routing spectrum [3,4].

8.2. Interpretation of Overall Performance

For RQ1, MAPPO's higher service, fewer failures, and faster recovery relative to static planning show the value of refreshing closure, risk, forecast, and remaining-demand information at each decision epoch. The larger severe-scenario gap reinforces this mechanism and agrees with anticipatory dynamic-routing research [21,23,26]. For RQ2, ACO, ALNS, and Full OR-Tools dynamic reoptimization achieve higher service by searching the revealed instance, while MAPPO reuses an amortized decision policy [14–16].

8.3. Comparison with Static and Adaptive Optimization

Static OR-Tools isolates the cost of retaining a pre-disruption plan. ALNS and Full OR-Tools dynamic reoptimization exploit revealed information through route repair or broad replanning, consistent with established strengths of adaptive search [27,28]. ACO occupies a middle position through short population-based search. The resulting order is methodologically coherent: greater online search authority increases service, while policy inference reduces computation. Related comparisons likewise show that the balance depends on instance structure, objectives, and reoptimization design [22].

8.4. Value of Solver-Independent Policy Execution

Solver-independent execution provides a direct state-conditioned action through observation construction, masking, and a neural forward pass. MAPPO's 0.029 s episode total is 3.8–17.3 times below the higher-service adaptive methods, yet every compared

method remains subsecond on the test hardware. The practical value of that margin is strongest under repeated or parallel decisions, constrained control cycles, or warm-start use; a single ordinary dispatch can instead prioritize the additional service from adaptive search.

8.5. Role of Demand, Risk, Closure, and Feasibility Information

For RQ3, closure information produced the largest feature effect, followed by the forecast and risk features; this ordering supports explicit representation of both immediate network admissibility and anticipatory demand [4,23]. The zero-service no-mask result confirms that masking supplies structural admissibility for the chosen action space [20]. The top- k test further shows that $k = 10$ captures most of the observed gain before runtime rises.

The architecture separation refines the role of MAPPO/CTDE. Shared policy learning under a team reward contributes more clearly to service than critic centralization: the local-critic shared actor slightly exceeds the centralized reference, whereas independent policies perform worse. This pattern is consistent with the sensitivity of cooperative PPO to implementation and critic design [19,45]. The environmental term affects operational energy rather than the primary service ordering.

8.6. Managerial Decision Implications

Table 18 translates the empirical ordering into method-selection conditions. MAPPO suits repeated state-conditioned decisions; repair preserves usable plan segments; and broader adaptive search suits cases in which marginal service and replanning authority dominate the small absolute runtime difference.

Table 18. Operational conditions guiding method selection.

Operating Condition	Preferred Response	Managerial Rationale
Repeated subsecond decisions; familiar state distribution	MAPPO	Direct state-conditioned action with a stable on-line computation path
Localized disruption; incumbent route largely usable	Limited repair	Preserves plan stability and minimizes route communication
Moderate computation window; marginal service has high value	ACO or ALNS	Searches the revealed instance for additional service
Major demand or network shock; broad replanning allowed	Full OR-Tools dynamic reoptimization	Uses full authority over the unexecuted plan
Out-of-distribution state or uncertain feasibility data	Optimizer or human-supervised fallback	Avoids relying solely on an extrapolating learned policy
High-frequency operation with periodic planning cycles	Hybrid hierarchy	MAPPO supplies immediate actions; search periodically restores route quality

Operators can encode the trade-off through trigger rules based on predicted service, disruption magnitude, available computation, and permitted route changes.

8.7. Parcel-Locker and Urban-Logistics Implications

The analytical candidate consolidation points create shared demand whose assignment couples vehicle capacity and future service. Physical parcel-locker deployment additionally depends on site access, compartment capacity, occupancy, recipient choice, and pickup behavior [6,33,36]. Field and optimization studies confirm that these factors shape realized benefits [7,24,40]; they form the deployment layer for extending the present routing framework.

8.8. Environmental Boundary

The environmental result is a routing-level electric-fleet proxy: it reports operational energy and assumes zero tailpipe CO₂. Charging, grid, production, infrastructure, and recipient-travel effects define the separate lifecycle assessment layer [41,55].

8.9. Implications for Future Hybrid Learning–Optimization Systems

The results support a hybrid hierarchy in which MAPPO supplies immediate actions or warm starts, ALNS performs targeted repair, and Full OR-Tools dynamic reoptimization handles periodic or major-shock replanning. Learning-guided constraint processing and ALNS already provide relevant design precedents [48,56]. For multi-carrier deployment, blockchain-backed event records offer a complementary mechanism for traceable handoffs with crowdsourced carriers [57].

9. Limitations and Validity Threats

9.1. Construct Validity

The operational metrics emphasize service, failures, recovery, and computation. Aggregation omits customer-level time windows and facility states, while risk and closure variables approximate road reliability. Recovery uses the first revealed disruption and a 95% service threshold, with threshold sensitivity reported in Appendix A Table A2. Solver independence applies to frozen-policy execution; clustering, forecasts, and the saved OSRM matrix remain preprocessing inputs.

9.2. Internal Validity

Paired instances, one-thread execution, a common timing boundary, validation-only tuning, and fixed stopping rules support comparison, while algorithm configuration continues to shape quality and runtime. Independently retrained ablations include stochastic optimization and component interactions; Tables 13–16 provide the principal structural checks.

9.3. Statistical Validity

Ten seeds crossed with 12 held-out days provide 120 seed–day cells. The two-way bootstrap resamples both factors and preserves matched methods and scenarios; seed- and day-aggregated directions agree for all primary contrasts (Table 17 and Appendix A Table A1). Inference remains tied to the held-out period and calibrated scenario generator.

9.4. External Validity

The analytical candidate consolidation points, Los Angeles network, cooperative homogeneous fleet, and calibrated disruptions define the transfer boundary. New cities require local point construction, road matrices, demand and risk processing, and policy validation; installed parcel-locker operations additionally require facility and customer states. Zero-shot tests cover fleet, demand, capacity, density, and severity shifts (Table 12). Runtime is hardware-dependent, and the electric result remains the routing-level proxy defined in Section 4.8.

10. Conclusions

This study presents a problem-specific MAPPO framework that integrates shared multi-agent action selection, dynamic consolidation-point observations, fleet-level value learning, compound-disruption information, and structural feasibility for routing through analytical candidate parcel-locker consolidation points.

MAPPO served 98.4% of demand versus 96.6% for static OR-Tools, reduced failed packages by 52.2%, and increased severe-scenario service from 89.4% to 93.0%. Its summed

online computation time was 0.029 s per episode, while higher-service adaptive methods reached 99.6–99.7% at 0.111–0.503 s. All methods remained subsecond on the test hardware, placing MAPPO's value in computational headroom for repeated or tightly timed decisions. Ablations identify closure, forecast, risk, shared policy learning, and explicit admissibility as the principal framework components.

The results establish a quality–computation trade-off between direct policy execution and adaptive search. Transfer to installed parcel-locker operations requires facility and customer states, broader geographic validation, and field calibration; the environmental result remains an operational routing proxy. Future work can combine MAPPO actions or warm starts with event-triggered ALNS and Full OR-Tools dynamic reoptimization.

Author Contributions: Conceptualization, M.-A.E., K.L.N. and J.B.; methodology, M.-A.E., K.L.N. and J.B.; software, M.-A.E.; validation, M.-A.E. and J.B.; formal analysis, K.L.N. and J.B.; investigation, M.-A.E. and K.L.N.; data curation, M.-A.E.; writing—original draft preparation, K.L.N.; writing—review and editing, K.L.N. and J.B.; visualization, M.-A.E.; supervision, J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received funding from Université Mohammed VI Polytechnique (UM6P).

Data Availability Statement: The processed data and supporting research materials are available from the corresponding author upon reasonable request. Unrestricted public release is not currently possible because these materials form part of an ongoing research program and remain subject to project-development and intellectual-property considerations. Requests will be evaluated in accordance with these constraints.

Acknowledgments: This work was conducted as part of the MILEX project and greatly benefited from the unwavering support of the Hassan II University of Casablanca, Ecole Centrale Casablanca, and Mohammed VI Polytechnic University. The collaborative and constructive involvement of these institutions played a pivotal role in the successful realization of this work, and we express our profound gratitude for their invaluable contributions throughout the research process.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACO	Ant Colony Optimization
ADP	Alternative Delivery Point
ALNS	Adaptive Large Neighborhood Search
CTDE	Centralized Training with Decentralized Execution
GAE	Generalized Advantage Estimation
MAPPO	Multi-Agent Proximal Policy Optimization
MARL	Multi-Agent Reinforcement Learning
OR	Operations Research
OSRM	Open Source Routing Machine
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
VRP	Vehicle Routing Problem

Appendix A. Additional Statistical and Sensitivity Results

The record-level analysis is retained as a precision check alongside the dependence-aware inference reported in the main Results section. Recovery-threshold sensitivity evaluates whether the principal comparison changes under alternative definitions of near-complete service.

Table A1. Record-level paired precision check; difference is proposed MAPPO minus comparator.

Comparison	Metric	Mean Difference	Wilcoxon p	Bootstrap 95% CI
MAPPO vs. static OR-Tools	Service	0.0182	3.70×10^{-22}	[0.0149, 0.0216]
MAPPO vs. static OR-Tools	Failed packages	−94.1325	2.61×10^{-21}	[−112.1395, −76.9017]
MAPPO vs. static OR-Tools	Restricted recovery	−193.223	4.50×10^{-27}	[−222.4419, −163.6355]
MAPPO vs. Full OR-Tools dynamic reoptimization	Service	−0.0129	4.92×10^{-12}	[−0.0166, −0.0098]
MAPPO vs. Full OR-Tools dynamic reoptimization	Failed packages	72.3844	5.42×10^{-14}	[55.1139, 92.3146]
MAPPO vs. Full OR-Tools dynamic reoptimization	Restricted recovery	166.114	6.17×10^{-70}	[148.3997, 183.8188]

Table A2. Recovery-threshold sensitivity. Means use recovered episodes; N/R is the share not reaching the threshold by the horizon.

Threshold	Method	Recovery (min)	N/R Share
90%	Proposed MAPPO	721.400	1.7%
	OR-Tools static	861.300	4.2%
95%	Proposed MAPPO	862.487	9.2%
	OR-Tools static	1055.710	18.3%
98%	Proposed MAPPO	1074.800	21.7%
	OR-Tools static	1288.900	34.2%

References

1. Laporte, G. The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms. *Eur. J. Oper. Res.* **1992**, *59*, 345–358. [\[CrossRef\]](#)
2. Vidal, T.; Laporte, G.; Matl, P. A Concise Guide to Existing and Emerging Vehicle Routing Problem Variants. *Eur. J. Oper. Res.* **2020**, *286*, 401–416. [\[CrossRef\]](#)
3. Pillac, V.; Gendreau, M.; Gueret, C.; Medaglia, A.L. A Review of Dynamic Vehicle Routing Problems. *Eur. J. Oper. Res.* **2013**, *225*, 1–11. [\[CrossRef\]](#)
4. Soeffker, N.; Ulmer, M.W.; Mattfeld, D.C. Stochastic Dynamic Vehicle Routing in the Light of Prescriptive Analytics: A Review. *Eur. J. Oper. Res.* **2022**, *298*, 801–820. [\[CrossRef\]](#)
5. Boysen, N.; Fedtke, S.; Schwerdfeger, S. Last-Mile Delivery Concepts: A Survey from an Operational Research Perspective. *OR Spectr.* **2021**, *43*, 1–58. [\[CrossRef\]](#)
6. Tilk, C.; Olkis, K.; Irnich, S. The Last-Mile Vehicle Routing Problem with Delivery Options. *OR Spectr.* **2021**, *43*, 877–904. [\[CrossRef\]](#)
7. Dell’Amico, M.; Montemanni, R.; Novellani, S. Pickup and Delivery with Lockers. *Transp. Res. Part C Emerg. Technol.* **2023**, *148*, 104022. [\[CrossRef\]](#)
8. Dorigo, M.; Gambardella, L.M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Trans. Evol. Comput.* **1997**, *1*, 53–66. [\[CrossRef\]](#)
9. Ropke, S.; Pisinger, D. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transp. Sci.* **2006**, *40*, 455–472. [\[CrossRef\]](#)
10. Pisinger, D.; Ropke, S. A General Heuristic for Vehicle Routing Problems. *Comput. Oper. Res.* **2007**, *34*, 2403–2435. [\[CrossRef\]](#)
11. Nazari, M.; Oroojlooy, A.; Snyder, L.V.; Takac, M. Reinforcement Learning for Solving the Vehicle Routing Problem. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; Volume 31.
12. Kool, W.; van Hoof, H.; Welling, M. Attention, Learn to Solve Routing Problems! In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
13. da Costa, P.; Rhuggenaath, J.; Zhang, Y.; Akcay, A.; Kaymak, U. Learning 2-Opt Heuristics for Routing Problems via Deep Reinforcement Learning. *SN Comput. Sci.* **2021**, *2*, 388. [\[CrossRef\]](#)
14. Bengio, Y.; Lodi, A.; Prouvost, A. Machine Learning for Combinatorial Optimization: A Methodological Tour d’Horizon. *Eur. J. Oper. Res.* **2021**, *290*, 405–421. [\[CrossRef\]](#)
15. Mazyavkina, N.; Sviridov, S.; Ivanov, S.; Burnaev, E. Reinforcement Learning for Combinatorial Optimization: A Survey. *Comput. Oper. Res.* **2021**, *134*, 105400. [\[CrossRef\]](#)
16. Wang, Q.; Tang, C. Deep Reinforcement Learning for Transportation Network Combinatorial Optimization: A Survey. *Knowl. Based Syst.* **2021**, *233*, 107526. [\[CrossRef\]](#)
17. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.

18. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347. [[CrossRef](#)]
19. Yu, C.; Velu, A.; Vinitzky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; Volume 35, pp. 24611–24624.
20. Huang, S.; Ontanon, S. Exploring the Use of Invalid Action Masking in Reinforcement Learning: A Comparative Study of On-Policy and Off-Policy Algorithms in Real-Time Strategy Games. *Appl. Sci.* **2023**, *13*, 8283. [[CrossRef](#)]
21. Basso, R.; Kulcsar, B.; Sanchez-Diaz, I.; Qu, X. Dynamic Stochastic Electric Vehicle Routing with Safe Reinforcement Learning. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *157*, 102496. [[CrossRef](#)]
22. Kullman, N.D.; Cousineau, M.; Goodson, J.C.; Mendoza, J.E. Dynamic Ride-Hailing with Electric Vehicles. *Transp. Sci.* **2022**, *56*, 775–794. [[CrossRef](#)]
23. Zhou, C.; Ma, J.; Douge, L.; Chew, E.P.; Lee, L.H. Reinforcement Learning-Based Approach for Dynamic Vehicle Routing Problem with Stochastic Demand. *Comput. Ind. Eng.* **2023**, *182*, 109443. [[CrossRef](#)]
24. Janinhoff, L.; Klein, R.; Scholz, D. Multitrip Vehicle Routing with Delivery Options: A Data-Driven Application to the Parcel Industry. *OR Spectr.* **2024**, *46*, 241–294. [[CrossRef](#)]
25. Wang, Y.; Qiu, D.; He, Y.; Zhou, Q.; Strbac, G. Multi-Agent Reinforcement Learning for Electric Vehicle Decarbonized Routing and Scheduling. *Energy* **2023**, *284*, 129335. [[CrossRef](#)]
26. Zhang, J.; Luo, K.; Florio, A.M.; van Woensel, T. Solving Large-Scale Dynamic Vehicle Routing Problems with Stochastic Requests. *Eur. J. Oper. Res.* **2023**, *306*, 596–614. [[CrossRef](#)]
27. Windras Mara, S.T.; Norcahyo, R.; Jodiawan, P.; Lusiantoro, L.; Rifai, A.P. A Survey of Adaptive Large Neighborhood Search Algorithms and Applications. *Comput. Oper. Res.* **2022**, *146*, 105903. [[CrossRef](#)]
28. Voigt, S. A Review and Ranking of Operators in Adaptive Large Neighborhood Search for Vehicle Routing Problems. *Eur. J. Oper. Res.* **2025**, *322*, 357–375. [[CrossRef](#)]
29. Gómez-Montoya, R.A.; Cano, J.A.; Cortés, P.; Salazar, F. A Discrete Particle Swarm Optimization to Solve the Put-Away Routing Problem in Distribution Centres. *Computation* **2020**, *8*, 99. [[CrossRef](#)]
30. Pan, W.; Liu, S.Q. Deep Reinforcement Learning for the Dynamic and Uncertain Vehicle Routing Problem. *Appl. Intell.* **2023**, *53*, 405–422. [[CrossRef](#)]
31. Ejjanfi, M.A.; Benhra, J. Reinforcement Learning Approach Based on Proximal Policy Optimization Algorithm for Efficient Last Mile Delivery Using Smart Lockers. *Inspire Smart Syst. J.* **2026**, *1*, 9–25. [[CrossRef](#)]
32. Grabenschweiger, J.; Doerner, K.F.; Hartl, R.F.; Savelsbergh, M.W.P. The Vehicle Routing Problem with Heterogeneous Locker Boxes. *Cent. Eur. J. Oper. Res.* **2021**, *29*, 113–142. [[CrossRef](#)]
33. Lyu, G.; Teo, C.P. Last Mile Innovation: The Case of the Locker Alliance Network. *Manuf. Serv. Oper. Manag.* **2022**, *24*, 2425–2443. [[CrossRef](#)]
34. Schaefer, J.S.; Figliozzi, M. Spatial Accessibility and Equity Analysis of Amazon Parcel Lockers Facilities. *J. Transp. Geogr.* **2021**, *97*, 103212. [[CrossRef](#)] [[PubMed](#)]
35. Raviv, T. The Service Points' Location and Capacity Problem. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *176*, 103216. [[CrossRef](#)]
36. Ma, B.; Teo, C.C.; Wong, Y.D. Location Analysis of Parcel Locker Network: Effects of Spatial Characteristics on Operational Performance. *Transp. Res. Part E Logist. Transp. Rev.* **2024**, *192*, 103776. [[CrossRef](#)]
37. Li, J.; Ensafian, H.; Bell, M.G.H.; Geers, D.G. Deploying Autonomous Mobile Lockers in a Two-Echelon Parcel Operation. *Transp. Res. Part C Emerg. Technol.* **2021**, *128*, 103155. [[CrossRef](#)]
38. Liu, Y.; Ye, Q.; Escribano-Macias, J.; Feng, Y.; Candela, E.; Angeloudis, P. Route Planning for Last-Mile Deliveries Using Mobile Parcel Lockers: A Hybrid Q-Learning Network Approach. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *177*, 103234. [[CrossRef](#)]
39. Wang, Y.; Bi, M.; Lai, J.; Wang, C.; Chen, Y.; Holguin-Veras, J. Recourse Strategy for the Routing Problem of Mobile Parcel Lockers with Time Windows under Uncertain Demands. *Eur. J. Oper. Res.* **2024**, *316*, 942–957. [[CrossRef](#)]
40. Ranjbari, A.; Diehl, C.; Dalla Chiara, G.; Goodchild, A. Do Parcel Lockers Reduce Delivery Times? Evidence from the Field. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *172*, 103070. [[CrossRef](#)]
41. Niemeijer, R.; Buijs, P. A Greener Last Mile: Analyzing the Carbon Emission Impact of Pickup Points in Last-Mile Parcel Delivery. *Renew. Sustain. Energy Rev.* **2023**, *186*, 113630. [[CrossRef](#)]
42. Janinhoff, L.; Klein, R.; Sailer, D.; Schoppa, J.M. Out-of-Home Delivery in Last-Mile Logistics: A Review. *Comput. Oper. Res.* **2024**, *168*, 106686. [[CrossRef](#)]
43. Mohri, S.S.; Ghaderi, H.; van Woensel, T.; Mohammadi, M.; Nassir, N.; Thompson, R.G. Contextualizing Alternative Delivery Points in Last Mile Delivery. *Transp. Res. Part E Logist. Transp. Rev.* **2024**, *192*, 103787. [[CrossRef](#)]
44. Zhang, Q.; Demir, E. Parcel Locker Solutions for Last Mile Delivery: A Systematic Literature Review and Future Research Directions. *Front. Future Transp.* **2025**, *6*, 1654621. [[CrossRef](#)]
45. Gronauer, S.; Diepold, K. Multi-Agent Deep Reinforcement Learning: A Survey. *Artif. Intell. Rev.* **2022**, *55*, 895–943. [[CrossRef](#)]

46. Arishi, A.; Krishnan, K. A Multi-Agent Deep Reinforcement Learning Approach for Solving the Multi-Depot Vehicle Routing Problem. *J. Manag. Anal.* **2023**, *10*, 493–515. [CrossRef]
47. Mak, S.; Xu, L.; Pearce, T.; Ostroumov, M.; Brintrup, A. Fair Collaborative Vehicle Routing: A Deep Multi-Agent Reinforcement Learning Approach. *Transp. Res. Part C Emerg. Technol.* **2023**, *157*, 104376. [CrossRef]
48. Cappart, Q.; Moisan, T.; Rousseau, L.M.; Premont-Schwarz, I.; Cire, A.A. Combining Reinforcement Learning and Constraint Programming for Combinatorial Optimization. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 3677–3687. [CrossRef]
49. Hewamalage, H.; Ackermann, K.; Bergmeir, C. Forecast Evaluation for Data Scientists: Common Pitfalls and Best Practices. *Data Min. Knowl. Discov.* **2023**, *37*, 788–832. [CrossRef] [PubMed]
50. City of Los Angeles. Traffic Collision Data from 2010 to Present. Los Angeles Open Data Portal. 2026. Available online: <https://data.lacity.org/d/d5tf-ez2w> (accessed on 15 July 2026).
51. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; Abbeel, P. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
52. Google for Developers. OR-Tools Routing Options. 2026. Available online: https://developers.google.com/optimization/routing/routing_options (accessed on 2 September 2026).
53. Royston, P.; Parmar, M.K.B. Restricted Mean Survival Time: An Alternative to the Hazard Ratio for the Design and Analysis of Randomized Trials with a Time-to-Event Outcome. *BMC Med. Res. Methodol.* **2013**, *13*, 152. [CrossRef] [PubMed]
54. Owen, A.B. The Pigeonhole Bootstrap. *Ann. Appl. Stat.* **2007**, *1*, 386–411. [CrossRef]
55. Moghdani, R.; Salimifard, K.; Demir, E.; Benyettou, A. The Green Vehicle Routing Problem: A Systematic Literature Review. *J. Clean. Prod.* **2021**, *279*, 123691. [CrossRef]
56. Wang, C.; Cao, M.; Jiang, H.; Xiang, X.; Zhang, X. A Deep Reinforcement Learning-Based Adaptive Large Neighborhood Search for Capacitated Electric Vehicle Routing Problems. *IEEE Trans. Emerg. Top. Comput. Intell.* **2025**, *9*, 131–144. [CrossRef]
57. Nadime, K.L.; Benhra, J.; Benabbou, R.; Mouatassim, S. A Blockchain-Powered Framework for Traceable and Secure Pharmaceutical Delivery with Crowdsourced Logistics. In *Proceedings of the Innovative Intelligent Industrial Production and Logistics; Communications in Computer and Information Science*; Springer: Cham, Switzerland, 2025; Volume 2372, pp. 367–384. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.