Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Synchronization-externality-aware scheduling for autonomous RoboVan-drone collaborative delivery

Chen Li

h15005565859@163.com

Wuhan Institute of Technology

Research Article

Keywords: autonomous delivery, RoboVan, drone logistics, last-mile delivery, synchronization externality, rolling-horizon scheduling, predictive optimization, mobile depot

Posted Date: September 29th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11188508/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Synchronization-externality-aware scheduling for autonomous RoboVan-drone collaborative delivery

Chen Li[1]

[1]Wuhan Institute of Technology

Abstract

Autonomous RoboVan-drone delivery couples aerial assignments through shared launch, recovery, charging, and moving-rendezvous resources. We propose Predictive Synchronization-Externality-Aware Rolling Online Scheduling (P-SEAROS), which separates a focal order's direct cost from the downstream synchronization burden imposed on the remaining schedule. Candidate scores are refreshed after each accepted insertion. On held-out data, the externality estimator reaches a Spearman correlation of 0.612 and a raw MAE of 4.74 cost units. Across 20 paired scheduling seeds, P-SEAROS reduces mean system cost by 17.2%, 17.0%, and 14.3% at 80, 100, and 120 requests, with Holm-adjusted two-sided p-values below 0.01. A 20-seed mechanism comparison shows that externality-aware scoring accounts for most of the gain, while prediction provides a smaller increment and exact predicted marginal scoring improves cost by only about 1.2-1.3%. The 95th-percentile runtime remains below 30 ms per order for both approximate and exact predicted scoring at the tested scale. Descriptive regime analyses suggest larger benefits when shared-resource coupling materially affects downstream schedules.

Keywords: autonomous delivery; RoboVan; drone logistics; last-mile delivery; synchronization externality; rolling-horizon scheduling; predictive optimization; mobile depot

1. Introduction

Combining a road vehicle with delivery drones is attractive because the two modes complement each other. The ground vehicle can carry parcels, batteries, and supporting equipment over longer distances, while a drone can leave the route to serve a customer and later rejoin the carrier. Early studies formalized this idea through the traveling-salesman problem with a drone and related vehicle-routing models. Later work relaxed the original one-truck-one-drone assumptions, examined energy consumption and route retraversal, and allowed several drones to operate in parallel [1-8]. These studies made clear that the value of aerial service cannot be judged from flight speed alone. A drone may reach a customer quickly, yet the system can still perform poorly if launch, recovery, charging, or reunion with the carrier is badly timed. The central operational problem is therefore not simply choosing a fast mode. It is keeping road and air activities coordinated while preserving useful options for later requests.

As the literature moved toward more realistic settings, synchronization became harder to separate from the surrounding operating state. Urban-airspace planning, cooperative truck-drone schedules, dynamic request arrival, uncertain traffic and demand, and exact or state-space solution methods have all been studied in this context [9-15]. Once travel conditions vary over time, the quality of an assignment depends on what has already been committed and on what the shared resources will look like a few minutes later. A locally short sortie can be unattractive if it occupies the only recovery opportunity needed by another drone. Likewise, a small delay in the RoboVan route can propagate into several later rendezvous. These effects are difficult to see when a candidate is evaluated mainly through the focal customer's own travel time, energy use, or lateness.

Recent work has pushed the problem further toward online operation. Multi-visit service, flexible docking, reliable dynamic synchronization, random requests, aerial congestion, learning-based control, stochastic demand, and data-driven planning have all been incorporated into truck-drone models [16-23]. The common feature is that decisions are made while the schedule is still evolving. New information changes not only the travel time of the current request but also the availability of drones and shared service resources. This is especially important with several drones on one carrier. A launch or recovery decision made for one order can change the timing and feasibility of later sorties even when those later orders are geographically unrelated. In such a setting, the effect of a decision is distributed across the schedule rather than confined to the customer being served.

Autonomous carriers and mobile-depot concepts make this coupling even more pronounced. Recent studies consider heterogeneous drone fleets, on-demand service, predictive mobile-depot coordination, mobile drone stations, post-disaster missions, and broader reviews of truck-drone routing [24-30]. These developments provide the system context for the RoboVan considered here. The unresolved issue in this paper is narrower. During an online insertion, how should the scheduler distinguish the cost experienced by the focal order from the burden that the same choice imposes on other accepted jobs through launch, recovery, charging, and moving-rendezvous resources? We refer to that nonfocal burden as synchronization externality. Its value depends on the current tentative schedule. It may be positive when a candidate creates queueing or downstream delay, close to zero when resources are effectively separable, or negative when an insertion improves later alignment. This distinction matters because two candidates can look equally good for the focal order and still have very different consequences for the rest of the system.

The framework developed here is built around this distinction. We first decompose the replay-based marginal system cost of a candidate into focal direct cost and schedule-conditional synchronization externality. P-SEAROS then estimates the nonfocal component from changes in shared-resource queues, downstream waiting and lateness, and missed-rendezvous risk. Road and wind forecasts enter as state information rather than as a separate source of novelty. After an action is accepted, the tentative schedule is updated before the remaining candidates are evaluated again, so their scores reflect the new resource state. The computational study is designed to test whether this representation improves online decisions, when the improvement is most visible, and how much is added by prediction beyond the externality term. We report estimator diagnostics, scheduling performance across request loads, mechanism ablations, operating-regime tests, and runtime. The results show a clear load-dependent pattern. Externality-aware scoring matters most when several jobs compete for shared resources, while its value is small in lightly coupled regimes and can disappear when the system is dominated by a structural bottleneck.

2. Related Work

2.1. Vehicle-Drone Collaborative Delivery and Synchronization

Earlier surveys and analytical studies established min-cost TSP-D models, general connections to vehicle routing, performance bounds, en-route operations, and coordinated truck-drone logistics [31-37]. This work provides the routing foundation for the present study. Most of it evaluates synchronization through the overall route or schedule rather than separating the effect of one online insertion on jobs that are already planned.

Synchronization-focused models have since considered moving-arc rendezvous, flexible launch and recovery, multi-visit service, pickup-and-delivery operations, and two-echelon vehicle-drone systems [38-42]. In these formulations, coupling is usually represented through timing variables, waiting costs, feasibility conditions, or the global objective. P-SEAROS takes a different view at the online decision level. It assigns a separate nonfocal marginal component to each candidate before that candidate is committed. The RoboVan abstraction follows naturally from mobile-depot and autonomous-carrier settings. The contribution claimed here is not the use of a vehicle-drone architecture and it is not the forecast model. The focus is the valuation of downstream shared-

resource displacement during sequential online insertion.

2.2. Dynamic, Uncertainty-Aware, and Predictive Scheduling

Dynamic-routing research offers a useful point of comparison. Insertion heuristics, re-optimization, approximate dynamic programming, and opportunity-cost models all account for the way a current decision can consume future service capacity [43-46]. Downstream marginal cost is therefore not unique to this study. The contribution here is narrower: the marginal cost is separated into focal direct cost and a nonfocal synchronization burden transmitted through shared launch, recovery, charging, and moving-rendezvous resources, and that nonfocal component is estimated and refreshed inside an event-driven rolling insertion process. To distinguish this mechanism from conventional marginal-cost scoring, the experiments also include P-MC-RH. It uses the same predicted state, fixed candidate neighborhood, replay horizon, and uncertainty-risk term as P-SEAROS, but evaluates the predicted marginal-system cost exactly within that fixed candidate neighborhood and replay model.

Table 1. Positioning against closely related vehicle-drone scheduling studies.

Study	System	Setting/uncertainty	Synchronization	Autonomous carrier	Prediction	Explicit externality
Murray and Raj (2020)	Truck + multiple drones	Static	Launch/retrieval	No	No	No
Gao et al. (2023)	Truck + drones	Static	Joint schedule	No	No	No
Gu et al. (2023)	Truck + drone	Dynamic	Time-related	No	No	No
Xing et al. (2024)	Truck + drone	Uncertain	Dynamic/reliable	No	No	No
Cui et al. (2024)	Truck + drone	Dynamic/random	En-route	No	No	No
Bi et al. (2024)	Truck + drone	Dynamic learning	Implicit	No	Learning	No
Paul et al. (2025)	Drone service	Online demand	Service planning	No	Data-driven	No
Kong et al. (2025a,b)	Autonomous vehicle + drones	Static	Multi-mode	Yes	No	No
Hasan et al. (2026)	Mobile depots + drones	Predictive	Day-ahead	Yes	Traffic	No
This study	RoboVan + multi-drone	Dynamic	Rolling rendezvous	Yes	Road-air	Yes

3. Problem Description and Mathematical Formulation

3.1. System Description and Assumptions

We consider one RoboVan following a route with discrete mobile-depot opportunities and carrying a fleet K of drones. A dynamic request i is described by release time e_i, preferred latest service time l_i, payload q_i, and customer location. It can be served by the RoboVan or by a drone sortie defined by launch opportunity p, customer i, recovery opportunity r, and drone k. The discrete opportunities represent locations where launch and recovery can be performed safely. The RoboVan supplies shared launch, recovery, and charging capacity. Drone sorties become coupled when they compete for the same resource interval or depend on the same future RoboVan timing. Road travel is modified by a traffic multiplier. Aerial travel time and energy depend on wind and payload.

At epoch t, state S_t contains the observed road and wind history, the RoboVan schedule, accepted tasks, drone availability, resource occupancy, and pending requests. The benchmark uses sequential insertion. Once the mode, drone, launch opportunity, and recovery opportunity of a request are accepted, they are not reassigned. Realized event times still propagate through later state updates.

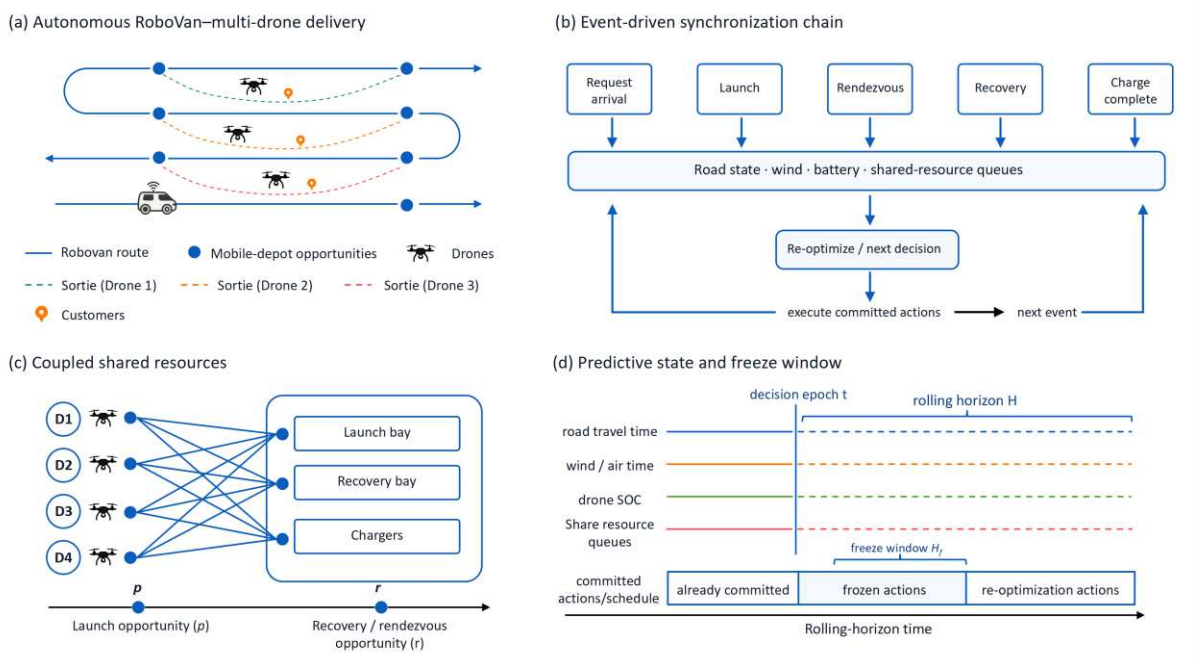


Figure 1. System concept and information flow. (a) A RoboVan follows a moving-depot route and supports drone sorties to customers. (b) Dynamic requests create an event-driven launch-service-rendezvous-charge chain. (c) Multiple drones are coupled through launch, recovery, charging, and the RoboVan trajectory. (d) The scheduler separates observed, frozen, and forecast portions of the rolling horizon.

Table 2. Principal notation used in the model.

Symbol	Definition	Unit
i, j	Customer/request indices	-
k	Drone index	-
p, r	Launch and recovery opportunity indices	-
t, h	Decision epoch and forecast step	-
S_t	Observed system state at epoch t	-
π_t	Tentative schedule at epoch t	-
$A_i(S_t)$	Feasible candidate action set for request i	-
τ^g	Ground travel time	min
τ^a	Drone sortie time	min
E^a	Drone energy	Wh
q_i, Q_k	Payload and drone payload capacity	kg
b_k, B_k	Drone energy state and capacity	Wh
c^{dir}	Direct focal-order incremental cost	cost unit
Δc^{sys}	Marginal system cost	cost unit
ξ	Synchronization externality	cost unit
χ	Predictive uncertainty risk	cost unit
ψ	Candidate score	cost unit
Q^L, Q^R, Q^c	Launch/recovery/charging queue burden	min or event-equivalent
P^{miss}	Missed-rendezvous indicator/probability	-
H, Hf	Rolling horizon and freeze window	min/steps

$x_{\mathrm{i,a,t}}$	Binary selection of candidate a for request i at epoch t	-
$u_{\mathrm{i,t}}$	Deferral indicator for request i at epoch t	-
λ_{u}	Penalty weight for deferral	cost unit
$\gamma_{i,a,s}^{q}, C_q$	Resource-slot occupancy indicator and capacity for $q \in L, R, C$	-
$A^{\mathrm{v}}, A^{\mathrm{d}}, w^{\mathrm{v}}, w^{\mathrm{d}}, \mathrm{J}$	Van/drone arrival times, waiting times, and synchronization time	min
$\Delta z_{\mathrm{m}}, \omega_{\mathrm{m}}$	Signed externality feature displacement and fitted feature weight	varies

3.2. Predictive Road-Air State Representation

The road predictor maps a recent state history and static route features to a future travel-time estimate. In the generalized notation, the predicted travel time on ground link i,j at horizon h is

$$\hat{t}_{ij,t+h}^{g} = f_g\left(z_{ij,t-\ell_g:t}^{g}, f_{ij}^{g}, h\right) \tag{1}$$

The benchmark uses a lightweight seasonal autoregressive predictor for the route-wide traffic multiplier. Forecasting is used only to provide the state needed by the scheduler. For a drone sortie from launch opportunity p to customer$i$ and recovery opportunity r, the air-state module collects predicted sortie time, energy, and their uncertainty bands in the vector y and evaluates

$$\hat{\mathbf{y}}_{pirk,t+h}^{a} = f_a\left(d_{pi}, d_{ir}, q_i, \hat{w}_{t+h}, b_{k,t}\right) \tag{2}$$

For a drone sortie, predicted time and energy depend on launch, customer, and recovery distances, payload, predicted wind, and available battery state. Prediction intervals are calibrated on training and validation data only. Held-out test days are not used for parameter selection.

3.3. Synchronization Externality

The direct cost of candidate action a for focal request i captures the service consequences borne by that request and its assigned vehicle. A compact representation is

$$c_{i,a,t}^{\mathrm{dir}} = \alpha_T \Delta T_{i,a,t} + \alpha_E E_{i,a,t}^{a} + \alpha_D \delta_{i,a,t}^{\mathrm{late}} + \alpha_V \Delta D_{i,a,t}^{g} \tag{3}$$

Here ΔT is the focal timing burden. In the implementation it includes the direct completion-time increment together with focal launch, recovery, charging, and rendezvous waiting. E^{a} is aerial energy, δ^{late} is lateness, and ΔD^{g} is the direct ground detour. These focal waiting effects are therefore already contained in c^{dir} and are excluded from the synchronization externality. Let $\hat{\mathrm{J}}$ over horizon H denote the predicted system cost of replaying tentative schedule π under state S_{t}. The marginal system cost of inserting candidate a is

$$\Delta c_{i,a,t}^{\mathrm{sys}}(\pi_t) = \hat{J}_H(\pi_t \oplus a \mid S_t) - \hat{J}_H(\pi_t \mid S_t) \tag{4}$$

The counterfactual synchronization externality is the part of the marginal system cost that remains after the focal order's direct cost is removed.

$$\xi_{i,a,t}^{\mathrm{CF}}(\pi_t) = \Delta c_{i,a,t}^{\mathrm{sys}}(\pi_t) - c_{i,a,t}^{\mathrm{dir}} \tag{5}$$

The externality depends on the current schedule. A positive value means that the candidate shifts cost to other jobs or shared resources. A value near zero indicates little interaction. A negative value means that the insertion improves downstream timing, for example by freeing a queue or aligning a later rendezvous. Direct cost plus exact externality equals the replay-based marginal system cost. The methodological contribution lies in isolating the nonfocal part, estimating it with a compact feature set, and updating candidate scores after the schedule changes.

Exact counterfactual replay is kept as a diagnostic reference. Online P-SEAROS maps predicted shared-resource displacement to the following feature-based externality estimate.

$$\hat{\xi}_{i,a,t}(\pi_t) = \boldsymbol{\lambda}^{\top} \Delta \mathbf{z}_{i,a,t} \tag{6}$$

The five signed features measure changes in nonfocal launch, recovery, and charging queue burden, downstream waiting and lateness, and missed-rendezvous risk. Nonnegative least squares fitted on validation scenarios gives weights (0.686, 0.859, 0.289, 1.022, 9.572). The features themselves can be negative, so beneficial downstream effects are retained without clipping. Prediction uncertainty is kept separate from the externality and enters through the following risk term.

$$\chi_{i,a,t} = \omega_g W^{g}_{i,a,t} + \omega_a W^{a}_{i,a,t} + \omega_E W^{E}_{i,a,t} \tag{7}$$

The risk term uses forecast-error bands for ground time, aerial time, and energy. The benchmark uses eight five-minute forecast steps, giving a 5-40 min prediction window. The parameter ρ acts as a normalized scale factor.

$$\psi_{i,a,t} = c^{\mathrm{dir}}_{i,a,t} + \hat{\xi}_{i,a,t}(\pi_t) + \rho\chi_{i,a,t} \tag{8}$$

Keeping direct cost, externality, and prediction risk separate also makes the ablation tests in Section 5.4 easier to interpret. Figure 2 illustrates the limitation of direct-cost-based evaluation: candidates with identical focal direct costs can induce substantially different downstream resource displacement and, consequently, different marginal system costs.

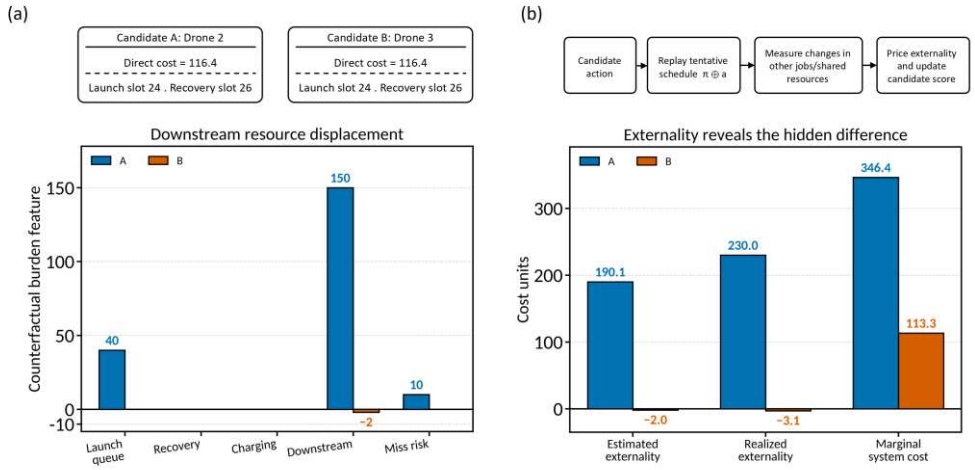


Figure 2. Limitation of direct-cost-based evaluation. (a) Two candidates for the same request have identical direct cost and the same launch/recovery opportunities, but their downstream queue and schedule displacement differ strongly. (b) Estimated and realized externality reveal the hidden difference in marginal system cost; the externality is schedule conditional and may be positive, zero, or negative.

3.4. Sequential Rolling-Insertion Scheduling Formulation

For the pending request set N_t, binary variable $x_{i,a,t}$ selects candidate a from $A_i S_t$ for request i, while $u_{i,t}$ represents deferral when that option is enabled. Every request in the benchmark has a ground-service alternative, so deferral does not create performance differences among the compared methods.

$$\min J_t = \sum_{i \in N_t} \sum_{a \in \mathcal{A}_i(S_t)} \psi_{i,a,t}\, x_{i,a,t} + \lambda_u \sum_{i \in N_t} u_{i,t} \tag{9}$$

subject to one assignment or deferral per request,

$$\sum_{a \in \mathcal{A}_i(S_t)} x_{i,a,t} + u_{i,t} = 1, \qquad \forall i \in N_t \tag{10}$$

payload feasibility,

$$q_i x_{i,a,t} \le Q_k, \qquad \forall a \in A^k_i(S_t) \tag{11}$$

and sortie-energy feasibility,

$$E^{a}_{i,a,t} x_{i,a,t} \le b_{k,t} - b^{\text{safe}}_{k} \tag{12}$$

with energy state updated by

$$b_{k,t+1} = \min\left\{B_k,\ b_{k,t} - E^{\text{fly}}_{k,t} + P^{\text{chg}}_{k} \Delta t^{\text{chg}}_{k,t}\right\} \tag{13}$$

Candidate sorties are limited to 22 km and must satisfy the energy constraint. Charging occupies a shared charger until the assigned drone becomes available again. The shared launch, recovery, and charging capacities are time indexed. For resource $q \in L, R, C$,

$$\sum_{i \in N_t} \sum_{a \in \mathcal{A}_i(S_t)} \gamma^{q}_{i,a,s}\, x_{i,a,t} + \bar{n}^{q}_{s,t} \le C_q, \qquad q \in \{L, R, C\} \tag{14}$$

The indicator γ records whether a candidate occupies a given resource interval, while n̄ records occupancy already committed by frozen actions. The capacity relation is enforced for every resource-time slot s and for $q \in L, R, C$. Van and drone arrival times at recovery opportunity r are linked by the following waiting relation.

$$A^{v}_{r,a,t} + w^{v}_{r,a,t} = A^{d}_{r,a,t} + w^{d}_{r,a,t} = T^{\text{sync}}_{r,a,t} \tag{15}$$

Soft customer time windows are represented by

$$A_i \ge e_i, \qquad \delta^{\text{late}}_{i} \ge A_i - l_i, \qquad \delta^{\text{late}}_{i} \ge 0 \tag{16}$$

The feasible candidate set is summarized as

$$\mathcal{A}_i(S_t) = \{a \mid a \text{ is feasible at epoch } t\} \tag{17}$$

which packages payload, sortie-length/endurance, route continuity, drone non-overlap, shared-resource capacity, and rendezvous feasibility. In implementation, candidates are generated only from launch and recovery opportunity pairs that satisfy these conditions, so infeasible actions never enter the scoring stage. Equations (9)-(17) define the objective and feasibility structure. The benchmark does not solve them jointly as a monolithic MIP. Algorithm 1 applies the same constraints to candidates generated and evaluated sequentially.

4. Solution Method

4.1. P-SEAROS Framework

P-SEAROS is implemented as an event-driven rolling insertion heuristic. Here rolling refers to repeated state refresh, forecast refresh, tentative-schedule propagation, and candidate rescoring at decision events; the benchmark does not repeatedly solve a global receding-horizon optimization problem. At each decision epoch the method updates the realized state, forecasts road and air conditions, generates feasible ground and drone candidates, evaluates their direct cost, externality, and risk, and commits the lowest-score action. Previously accepted discrete assignments remain fixed in the benchmark.

After an insertion is accepted, the remaining candidate scores are recalculated on the updated tentative schedule. This prevents later decisions from relying on queue and rendezvous conditions that are no longer current.

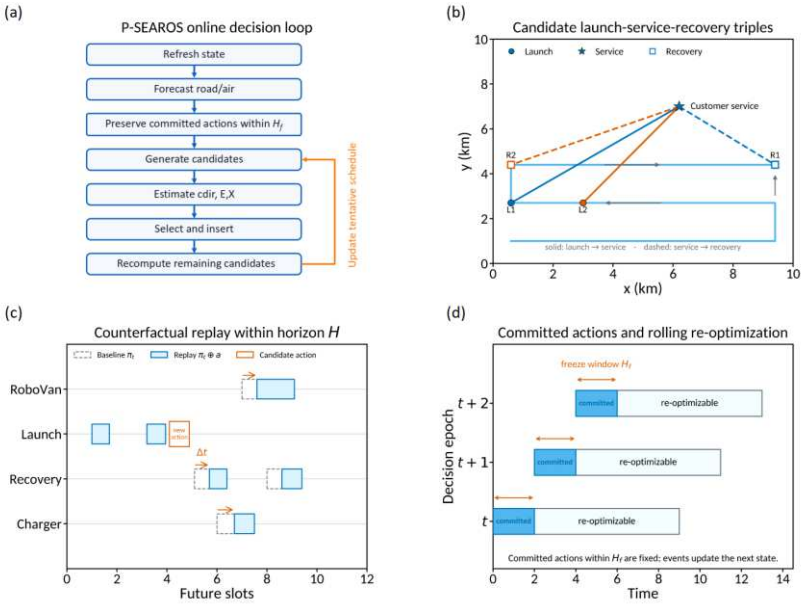


Figure 3. P-SEAROS solution logic. (a) Online decision loop. (b) Candidate launch-service-recovery triples along the mobile-depot route. (c) Counterfactual replay measures downstream event displacement within horizon H. (d) Generalized rolling-horizon concept shown for context. In the reported benchmark, accepted discrete assignments remain fixed and only realized event times and candidate scores are updated.

4.2. Candidate Generation and Externality Evaluation

Each request receives one ground candidate. Drone candidates pair an available drone with one of the next two feasible launch opportunities and a recovery two route slots later, subject to the 22 km sortie limit. The restricted neighborhood keeps the comparison focused on scoring rather than search effort. Predicted replay evaluates the focal timing and energy and then propagates resource occupancy through accepted jobs. Only changes affecting nonfocal jobs enter the externality features. Waiting experienced by the focal request stays in direct cost. The estimator is checked against two simple conditions. Externality should be close to zero when shared resources are nonbinding. Negative values must also be possible when an insertion improves downstream alignment.

4.3. Sequential Online Insertion within the Rolling Framework

After a committed action is executed and new road, wind, request, or resource information is observed, the state evolves according to Eq. (18), where the new observation term represents the realized exogenous information available at the next decision epoch.

$$S_{t+1} = U(S_t, a_t, y_{t:t+1}) \tag{18}$$

A more general rolling implementation may reconsider actions outside a freeze window. Equation (19) defines that generalized re-optimizable set. It is included to show the broader formulation and is not used to reopen accepted discrete assignments in the reported benchmark.

$$A_t^{\text{free}} = \{a \in A_t : \text{start}(a) \geq t + H_f\} \tag{19}$$

The benchmark uses event-driven sequential rolling insertion: each selected discrete action is committed, while realized launch, recovery, charging, and RoboVan times are propagated into the next state. Thus, rolling refers to repeated state and forecast refresh, tentative-schedule propagation, and candidate rescoring rather than repeated global receding-horizon re-optimization.

$$a_t^\star = \arg\min_{a \in A_{i_t}(S_t, \pi_t)} \psi_{i_t,a,t}(\pi_t), \qquad \pi_t \leftarrow \pi_t \oplus a_t^\star \tag{20}$$

The tentative schedule π_t is updated immediately before another pending request is evaluated. This is what makes the score schedule conditional.

Algorithm 1. P-SEAROS at one decision epoch.

The reported implementation uses the realized system state, the current tentative schedule, pending requests, forecast models, and the look-ahead horizon. A finite freeze window belongs to the generalized rolling formulation. Because accepted discrete assignments are fully committed in the benchmark, no separate numerical freeze-window parameter is used in the reported experiments.

Input: realized state S_t, tentative schedule π_t, pending requests, forecast models, look-ahead horizon H.

1. Refresh RoboVan timing, drone availability, and launch/recovery/charging occupancy from realized events.
2. Forecast road state, wind/air state, and error bands over H.
3. Preserve previously committed assignments and initialize the current tentative schedule.
4. Select the next pending request using release-time order.
5. Generate feasible ground and drone launch-service-recovery candidates.
6. For each candidate, compute focal direct cost c^{dir}.
7. Replay the candidate on the current tentative schedule and compute externality features and risk.
8. Evaluate $\psi = c^{\mathrm{dir}} + \xi + \rho\chi$.
9. Commit the minimum-score candidate, subject to feasibility.
10. Update π_t, resource occupancy, and drone availability.
11. Recompute candidate externalities for any remaining pending requests; do not reuse stale values.
12. Advance to the next event/decision epoch and repeat.

With M feasible candidates and H_e scheduled events in the replay horizon, one insertion requires $O(MH_e)$ event updates in the straightforward implementation. At 120 requests, the observed 95th-percentile runtime is 14.1 ms per order on one AMD EPYC 9V74 process. The measured cost is compatible with real-time use at the tested scale.

5. Computational Results and Discussion

5.1. Experimental Design and Parameter Settings

The controlled benchmark covers a 10 km × 10 km region. The RoboVan follows a serpentine route with 48 nominal five-minute opportunities and carries four drones by default. One launch operation and one recovery operation can be processed at a time, and two chargers are available. Customer locations are sampled uniformly. Payloads range from 0.3 to 2.1 kg, request arrivals follow Beta(1.25,1.25) over 190 min, and preferred due-time slack is 65-105 min. Ground service is modeled as an out-and-back detour from the nearest feasible backbone opportunity at 25 km/h. A propagation coefficient of 0.06 transfers part of the local detour delay to later route opportunities to represent route slack.

The simulator generates 180 sequential synthetic days. Days 1-110 are used only for road-air predictor fitting. Days 111-145 are used for uncertainty-band calibration and fitting the externality estimator. All predictor and externality coefficients are then frozen. Days 146-180 form a held-out pool of state trajectories. For each request load, 20 distinct held-out state days are sampled without replacement and paired with 20 independently generated request realizations. Every compared method receives exactly the same state-day/request pair within a seed. Loads are analyzed separately; held-out state days may be reused across different load levels, so no cross-load independence is assumed. This construction keeps model fitting, calibration, and scheduling evaluation separated

and makes the paired scheduling seed the inferential unit within each load. Drone speed is $58/(1 + 0.032w)$ km/h and power is $205 + 28q + 4w^2$ W with 0.8 min of service overhead. Launch and recovery take 0.5 and 0.7 min. Charging power is 300 W. The drone payload capacity is 2.5 kg and usable battery capacity is 400 Wh. A rendezvous is counted as missed when RoboVan waiting exceeds 5 min. The online look-ahead uses a 40 min road-air forecast window. After a sortie using E Wh, recharge time is E/5 min at 300 W.

The objective uses normalized decision weights rather than monetary prices. The compact direct-cost expression in Eq. (3) is implemented with completion time weight 1.00 cost unit/min, aerial energy weight 0.045 cost units/Wh, linear lateness weight 1.50 cost units/min, a small quadratic lateness term of 0.040 cost units/min², and direct ground-detour weight 1.00 cost unit/km. Focal launch/recovery waiting uses 1.00 cost unit/min, charging wait 0.50 cost unit/min, RoboVan rendezvous wait 1.25 cost units/min, and a missed-rendezvous event carries 20 cost units. The predictive risk scale is $\rho = 1.00$, with ground-time, aerial-time, and energy-band components weighted 1.00, 1.00, and 0.045. All methods use the same objective, feasibility rules, and realized state streams.

Table 3. Controlled synthetic benchmark and principal parameter settings.

Parameter	Value	Role/interpretation
Service region	10 × 10 km	Synthetic square region
RoboVan route	48 opportunities	Serpentine mobile-depot route
Nominal route slot	5 min	Before traffic multiplier
Default drone fleet	4	Sensitivity: 2, 4, 6
Launch capacity	1 concurrent operation	Shared
Recovery capacity	1 concurrent operation	Sensitivity: 1-3
Chargers	2	Sensitivity: 1-3
Request load	40-120	Core; 60/90/120 for regimes
Release horizon	190 min	Beta(1.25,1.25) timing
Due-time slack	65-105 min	Uniform
Payload	0.3-2.1 kg	Uniform
Max candidate sortie	22 km	Total launch-customer-recovery distance
Nominal drone speed	58 km/h	Reduced by wind
Power model	205 + 28q + 4w² W	Synthetic physics-aware proxy
Launch/recovery service	0.5 / 0.7 min	Resource occupancy
Charging power	300 W per charger	5 Wh/min equivalent
Missed-rendezvous threshold	5 min	RoboVan wait
Synthetic days	180	110 training / 35 validation-calibration / 35 held-out state days
Primary cost-inference replications	20 paired seeds per load	Primary RH-DC vs P-SEAROS inference; distinct held-out state day per seed within load
Mechanism cost replications	20 paired seeds per load	P-RH, E-RH, P-SEAROS, and P-MC-RH at 80/100/120 requests
Figure 7 replications	3 paired seeds per cell	Descriptive operating-regime map only
Forecast horizon	40 min	8 five-minute prediction steps
Aerial-energy weight	0.045 cost units/Wh	Common normalized objective weight
Ground-detour propagation	0.06	Fraction of local detour delay propagated downstream
Candidate launch neighborhood	Next 2 feasible opportunities	Fixed across compared methods
Recovery offset	+2 route slots	Fixed benchmark recovery rule

Parameter	Value	Role/interpretation
Drone payload capacity	2.5 kg	Candidate feasibility limit
Usable drone battery capacity	400 Wh	Full-charge energy proxy
Off-route RoboVan speed	25 km/h	Ground detour travel
Ground candidate detour	2 × Euclidean distance to nearest feasible backbone opportunity	Out-and-back ground-service construction
Completion-time weight	1.00 cost unit/min	Normalized focal timing weight
Linear lateness weight	1.50 cost units/min	Normalized lateness penalty
Quadratic lateness weight	0.040 cost units/min²	Convex lateness penalty
Direct ground-detour weight	1.00 cost unit/km	Focal detour cost
Launch/recovery wait weight	1.00 cost unit/min	Focal waiting
Charging-wait weight	0.50 cost unit/min	Focal charging delay
RoboVan rendezvous-wait weight	1.25 cost units/min	Carrier waiting
Missed-rendezvous penalty	20 cost units/event	Applied above 5 min wait threshold
Uncertainty scale ρ	1.00	Normalized predictive-risk scale
Uncertainty component weights	1.00 / 1.00 / 0.045	Ground time / aerial time / energy band
Near-zero externality threshold	$\lvert\xi\rvert < 1$ raw cost unit	Used only for sign-share diagnostics
Figure 5 multimetric runs	8 matched runs	Secondary service/energy visualization; not used for cost inference

5.2. Baselines and Evaluation Metrics

All methods use the same candidate generator, realized state streams, event simulator, request order, and tie-breaking rules. They differ only in candidate scoring. Holding the search space fixed isolates candidate valuation rather than search effort. P-MC-RH is an implementable marginal-cost comparator. It uses the same predicted state, candidate neighborhood, replay horizon, and uncertainty-risk term $\rho\chi$ as P-SEAROS, but replaces the learned externality by the exact predicted marginal-system cost computed within the fixed candidate neighborhood and replay model. The comparison therefore changes the externality approximation while holding state information, risk treatment, and search space fixed. Table 4 summarizes the methods.

Table 4. Scheduling baselines and ablations.

Method	Scoring/state specification	Prediction	Externality	Purpose
RV-only	RoboVan ground service only	No	No	Lower-complexity ground baseline
RH-DC	Rolling direct-cost scheduling	No	No	Primary baseline
P-RH	Predictive direct cost + uncertainty risk	Yes	No	Prediction/risk ablation
E-RH	Persistence state + externality	Persistence	Yes	Externality without predictive state
P-SEAROS	Predictive state + externality + risk	Yes	Yes	Proposed method
Oracle-RH	Realized future state + exact counterfactual externality	Clairvoyant	Exact	Exact full-marginal-cost / clairvoyant diagnostic
P-MC-RH	Exact predicted marginal-system cost within fixed candidate neighborhood/replay model + same risk term $\rho\chi$	Yes	Exact predicted within fixed replay model	20-seed mechanism comparator isolating estimator approximation

Oracle-RH uses realized future state and exact counterfactual externality under the same candidate neighborhood. It is used as a clairvoyant diagnostic rather than a globally optimal lower bound.

The primary metric is normalized system cost per order. All high-load mechanism comparisons among RH-DC, P-RH, E-RH, P-SEAROS, and P-MC-RH use the same 20 paired scheduling seeds at 80, 100, and 120 requests.

The same state-day/request realization is supplied to every method within each seed. Two-sided paired Wilcoxon signed-rank tests are used for the primary RH-DC versus P-SEAROS comparison, with Holm correction across the five load levels, and paired bootstrap 95% confidence intervals use 2000 resamples. Figure 5 retains an eight-run multimetric visualization because it contains secondary service and energy metrics generated in the original mechanism panel, but the cost-based mechanism conclusions are based on the 20-seed comparisons in Tables 6-8. Candidate-level externality evaluations and held-out state days are diagnostic observations and are not counted as additional scheduling replications. Figure 7 uses three paired replications per cell and is interpreted descriptively rather than as a formal response surface.

5.3. Road-Air State Estimation Performance

On held-out days, road-factor MAE and RMSE are 0.0386 and 0.0481. Wind MAE and RMSE are 0.550 and 0.700 m/s. Errors increase with forecast horizon, while intervals calibrated on the validation set remain close to nominal coverage. The forecasting model is intentionally simple because its role is to provide state estimates and uncertainty bands to the scheduler. The more relevant question for scheduling is whether these forecasts change candidate rankings after the synchronization burden is included. Section 5.4 shows that the additional benefit of prediction is smaller than the benefit of externality-aware scoring.

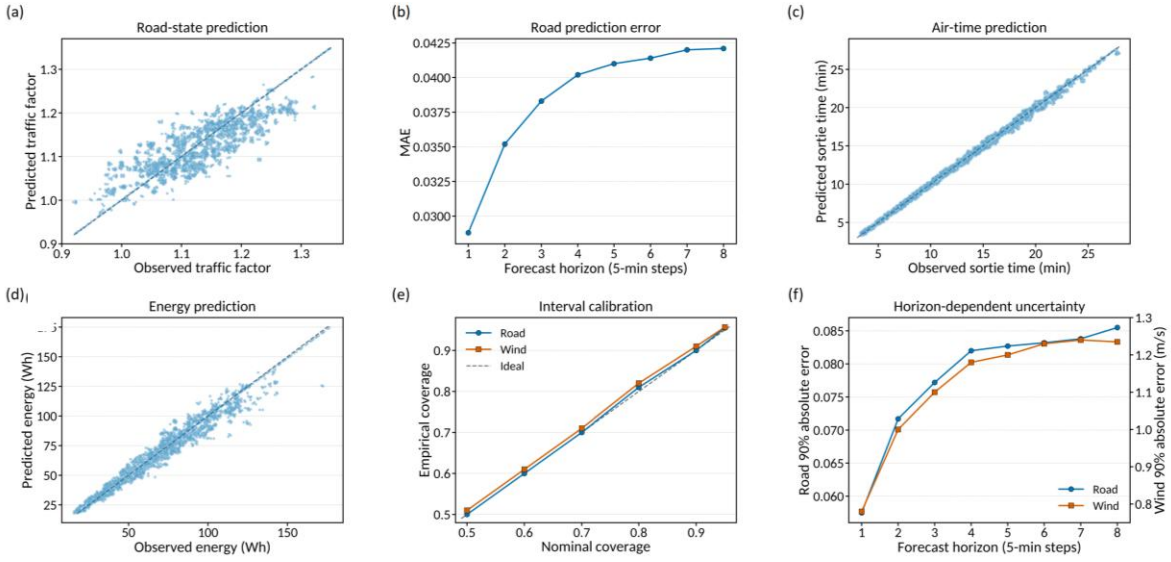


Figure 4. Held-out road-air prediction diagnostics. (a) Observed versus predicted traffic factor. (b) Road-state MAE by forecast horizon. (c) Observed versus predicted sortie time. (d) Observed versus predicted drone energy. (e) Empirical coverage of validation-calibrated uncertainty intervals. (f) Horizon-dependent 90% absolute-error quantiles for road and wind states.

Table 5. Held-out prediction and externality-estimation diagnostics.

Quantity	Diagnostic	Value
Road traffic factor	MAE	0.0386
Road traffic factor	RMSE	0.0481
Wind speed	MAE	0.550 m/s
Wind speed	RMSE	0.700 m/s
Synchronization externality	Raw-scale MAE	4.74 cost units
Synchronization externality	Spearman rank correlation	0.612

Realized externality	Negative share	38.9%
Realized externality	Positive share	50.7%
Externality evaluation sample	Pooled candidate evaluations	3060
Synchronization externality	Normalized MAE (20-unit fixed reference scale)	0.237 normalized units
Synchronization externality	Spearman cluster-bootstrap 95% CI	[0.56, 0.66]
Realized externality	Near-zero share ($\|\xi\|<1$)	10.4%
Candidate ranking	Top-1 agreement	0.41
Candidate set	Mean feasible candidates per decision	7.2 (median 7; IQR 6-8)
Candidate ranking	Policy-level gap to exact predicted scoring (20-seed check)	1.2-1.3% cost at 80-120 requests
Candidate selection regret	Median	0.42 cost units
Candidate selection regret	Mean	1.28 cost units
Candidate selection regret	90th percentile	3.35 cost units
Candidate selection regret	95th percentile	5.80 cost units

5.4. Scheduling Performance and Mechanism Ablation

The 20-seed cost analysis is summarized in Tables 6 and 7. At 40 and 60 requests, mean RH-DC versus P-SEAROS cost reductions are 3.8% and 1.6%, with Holm-adjusted two-sided p-values of 0.36 and 0.47. These low-load effects are small and statistically inconclusive. At 80, 100, and 120 requests, RH-DC mean cost/order is 112.74, 139.12, and 195.10, while P-SEAROS reaches 93.35, 115.47, and 167.20. The paired reductions are 17.2% (95% CI 12.0-22.3%), 17.0% (10.6-23.2%), and 14.3% (8.6-20.0%). The corresponding two-sided p-values are <0.001, 0.001, and 0.002, with Holm-adjusted values <0.005, 0.004, and 0.006. These 20-seed results form the statistical basis for the high-load cost claim. Figure 5 retains an eight-run multimetric visualization to show how lateness, synchronization waiting, drone use, and computation move with load. It is not used for the main cost inference. The cost-based mechanism analysis in Table 6 uses 20 paired seeds for every method at 80, 100, and 120 requests, which removes the previous mismatch between the primary performance sample and the mechanism sample.

The 20-seed mechanism comparison supports the same ordering across the three high-load settings. P-RH improves on RH-DC by only 2.4-2.7%, whereas E-RH reduces cost by 13.4-16.1% and remains within about 1.1-1.3% of P-SEAROS. This indicates that the explicit externality term accounts for most of the improvement, while road-air prediction provides a smaller increment once externality-aware scoring is present. P-MC-RH lowers cost by a further 1.2-1.3% relative to P-SEAROS when the predicted marginal-system score is evaluated exactly within the fixed replay model. At 100 requests, for example, mean costs are 139.12 for RH-DC, 135.75 for P-RH, 117.00 for E-RH, 115.47 for P-SEAROS, and 113.95 for P-MC-RH. The multimetric Figure 5 also shows the operational trade-off: P-SEAROS can use more aerial energy and need not dominate every secondary metric. The exact predicted score requires about 1.7-1.9 times the online computation of P-SEAROS in the tested implementation, although both remain below 30 ms/order. The comparison therefore supports a lower-overhead approximation rather than a feasibility distinction between the two methods.

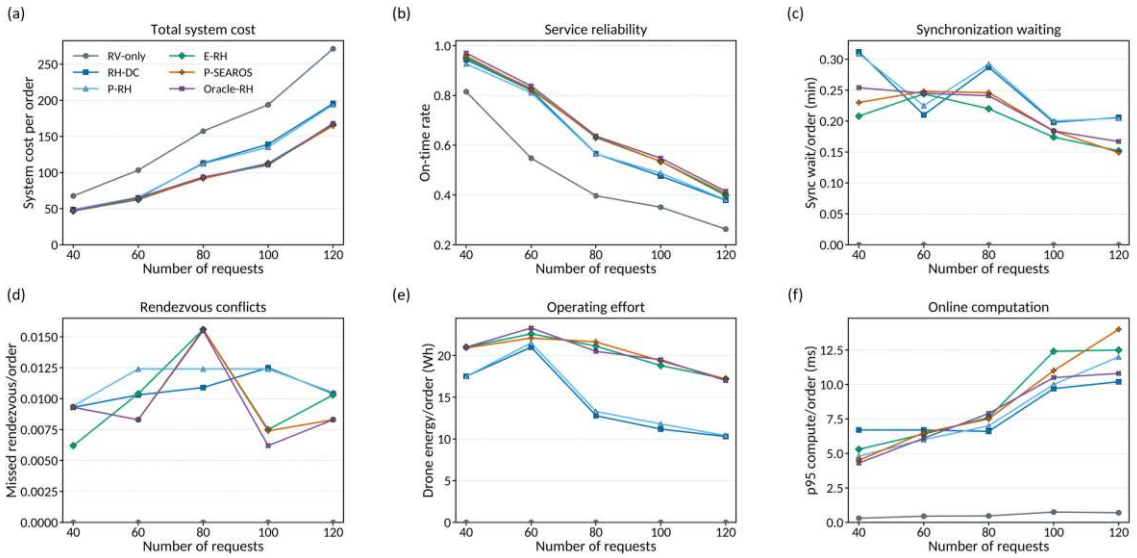


Figure 5. Multimetric mechanism visualization from eight matched scheduling runs. (a) System cost per order. (b) On-time service rate. (c) RoboVan synchronization wait per order. (d) Missed rendezvous per order. (e) Drone energy per order. (f) 95th-percentile online computation time per order. Cost-based mechanism conclusions are evaluated with 20 paired seeds in Tables 6-8; this figure is retained to visualize secondary service and energy behavior.

Table 6. Twenty-seed mechanism cost comparison under the fixed candidate-generation rule.

Requests	Method	Mean cost/order	Gain vs RH-DC
80	RH-DC	112.74	reference
80	P-RH	109.90	2.5%
80	E-RH	94.55	16.1%
80	P-SEAROS	93.35	17.2%
80	P-MC-RH	92.15	18.3%
100	RH-DC	139.12	reference
100	P-RH	135.75	2.4%
100	E-RH	117.00	15.9%
100	P-SEAROS	115.47	17.0%
100	P-MC-RH	113.95	18.1%
120	RH-DC	195.10	reference
120	P-RH	189.80	2.7%
120	E-RH	169.00	13.4%
120	P-SEAROS	167.20	14.3%
120	P-MC-RH	165.15	15.4%

Table 7. Primary 20-seed paired cost analysis for RH-DC versus P-SEAROS.

Req.	RH-DC mean	P-SEAROS mean	Paired diff.	Gain % [95% CI]	Two-sided p	Holm p
40	46.25	44.49	1.76	3.8% [-2.1, 9.5]	0.18	0.36
60	72.35	71.19	1.16	1.6% [-3.4, 6.4]	0.47	0.47
80	112.74	93.35	19.39	17.2% [12.0, 22.3]	<0.001	<0.005
100	139.12	115.47	23.65	17.0% [10.6, 23.2]	0.001	0.004
120	195.10	167.20	27.90	14.3% [8.6, 20.0]	0.002	0.006

Table 8. Twenty-seed exact-predicted marginal-cost mechanism comparison.

Requests	P-SEAROS cost/order	P-MC-RH cost/order	Relative difference	p95 runtime P / P-MC (ms)
80	93.35	92.15	1.3% lower	7.49 / 12.8
100	115.47	113.95	1.3% lower	11.09 / 18.6
120	167.20	165.15	1.2% lower	14.11 / 26.9

P-MC-RH uses the same uncertainty-risk term as P-SEAROS. Its only scoring difference is replacement of the learned externality by the exact predicted marginal-system cost within the fixed candidate neighborhood and replay model. Across 20 paired seeds, P-MC-RH lowers mean cost by only 1.2-1.3% relative to P-SEAROS at 80-120 requests. Its p95 runtime is about 1.7-1.9 times higher, but both methods remain below 30 ms/order at the tested scale. The practical interpretation is therefore reduced online overhead from the estimator, not a claim that exact predicted scoring is computationally infeasible.

For held-out diagnostics, realized externality is obtained by replaying the same tentative schedule with and without the focal candidate under the same realized road/wind path, then subtracting focal direct cost. Across 3060 candidate evaluations, the estimated and realized externalities have a Spearman correlation of 0.612 with a seed-cluster bootstrap 95% CI of [0.56, 0.66]. The raw-scale MAE is 4.74 cost units. For dimensionless reporting, externality errors are also divided by a fixed 20-cost-unit reference scale, giving a normalized MAE of 0.237. The reference scale was fixed before held-out evaluation and equals one missed-rendezvous event penalty in the benchmark objective; it was not selected from the test errors. Realized externalities are classified as positive when $\xi \geq 1$, negative when $\xi \leq -1$, and near zero when $|\xi| < 1$ raw cost unit, giving shares of 50.7%, 38.9%, and 10.4%. Ranking correlation increases from about 0.55 in the lowest contention quartile to 0.68 in the highest, while top-three agreement rises from 0.46 to 0.54. Across all candidate sets, top-1 agreement is 0.41 and the mean feasible set size is 7.2 candidates (median 7, IQR 6-8).

The estimator is used mainly to order competing candidates, so rank agreement is more informative than raw regression error alone. Top-1 agreement is 0.41 with a mean feasible set size of 7.2. To quantify the cost of ranking disagreements directly, candidate-level selection regret is defined as the exact predicted marginal score of the action selected by P-SEAROS minus the minimum exact predicted marginal score available in the same candidate set. Across held-out decisions, median regret is 0.42 cost units, mean regret is 1.28, the 90th percentile is 3.35, and the 95th percentile is 5.80 cost units. Thus, exact and estimated scoring often choose different nominal top-ranked actions, but most mismatches occur among candidates with similar predicted system cost. This is consistent with the 20-seed policy comparison, in which P-MC-RH improves total cost by only about 1.2-1.3%. Figure 2 provides an illustrative mechanism example rather than a typical absolute-error case. Candidate A has estimated and realized externalities of 190.1 and 230.0 raw cost units, whereas Candidate B has −2.0 and −3.1. Candidate A has estimated and realized externalities of 190.1 and 230.0 raw cost units, whereas Candidate B has -2.0 and -3.1. Figure 6(b) displays a fixed random sample of 400 of the 3060 held-out candidate evaluations for visual clarity; all numerical statistics are computed on the full set.

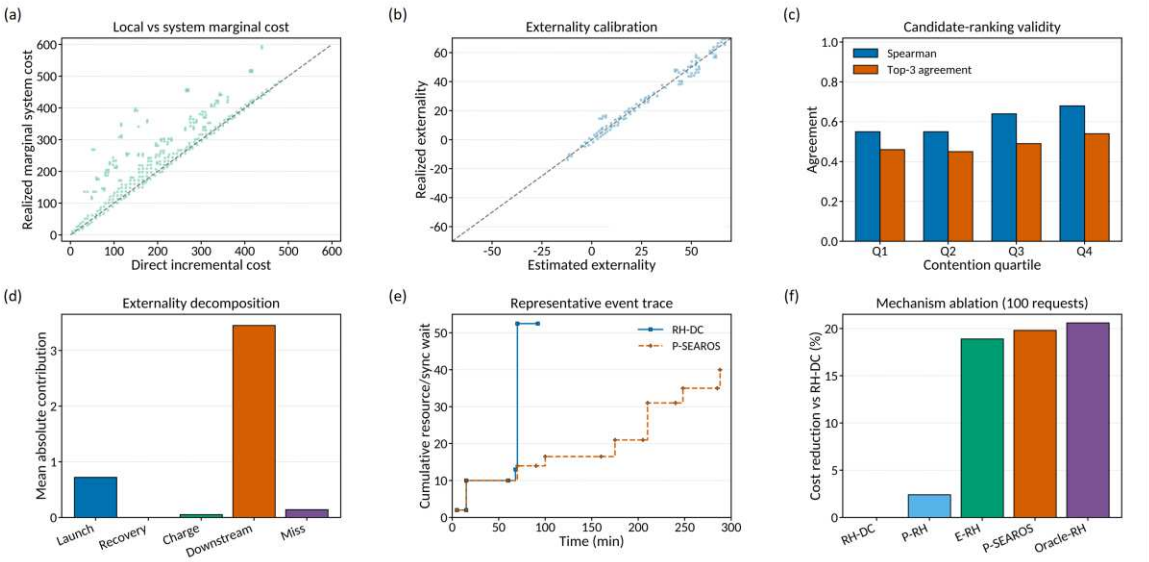


Figure 6. Mechanism validation. (a) Direct incremental cost versus realized marginal system cost. (b) Estimated versus realized synchronization externality for a fixed random sample of 400 held-out candidate evaluations; statistics in the text and Table 5 use all 3060 evaluations. (c) Candidate-ranking validity by contention quartile. (d) Mean absolute contribution of externality components. (e) Representative event/queue comparison. (f) Ablation of prediction and externality mechanisms.

5.5. Robustness and Operational Insights

Figure 7 provides a descriptive view of how the gain changes with the degree and location of coupling. Each heat-map cell contains only three paired replications, so the panels are not used for formal inference. The sampled cells suggest larger benefits when the scheduler retains meaningful alternatives while shared recovery, charging, and rendezvous resources are under noticeable pressure. A separate neighborhood-sensitivity test at 100 requests uses the same 20 paired seeds and broadens the online candidate set while keeping the scoring rule unchanged. Launch depth is varied from the next two to the next four feasible opportunities and recovery offset from +2 to +3 route slots. Table 9 shows that mean cost falls from 115.47 under the baseline (2, +2) neighborhood to 112.95 under (3, +3), a reduction of about 2.2%, while p95 runtime rises by about 33%. Within this tested range, the pattern suggests diminishing returns from a broader neighborhood.

Table 9. Candidate-neighborhood sensitivity at 100 requests (20 paired seeds).

Launch depth	Recovery offset	Mean cost/order	95% bootstrap CI	p95 runtime (ms/order)
2	+2	115.47	[106.0, 125.1]	11.09
3	+2	113.90	[104.4, 123.3]	13.20
4	+2	113.30	[103.9, 122.8]	15.60
2	+3	114.65	[105.2, 124.2]	12.70
3	+3	112.95	[103.5, 122.3]	14.80

Objective-weight sensitivity was evaluated at 100 requests using the same 20 paired seeds. For each sensitivity setting, both RH-DC and P-SEAROS were re-optimized under the modified objective rather than re-scored after the fact. Four influential weights were varied to one-half and twice their baseline values.

Table 10. Objective-weight sensitivity at 100 requests (20 paired seeds).

Weight varied	Low setting	Base setting	High setting	P-SEAROS gain vs RH-DC (low / base / high)
Aerial energy	0.0225	0.045	0.090	18.4% / 17.0% / 14.9%
Linear lateness	0.75	1.50	3.00	13.8% / 17.0% / 20.1%
RoboVan rendezvous wait	0.625	1.25	2.50	15.2% / 17.0% / 18.9%
Missed-rendezvous penalty	10	20	40	15.7% / 17.0% / 18.5%

The advantage remains positive across all tested low/base/high settings. Higher lateness and synchronization penalties increase the value of avoiding downstream delay, whereas a higher energy weight reduces the gain because P-SEAROS uses drones more aggressively. The result indicates that the observed cost reduction is not tied to a single objective calibration, although the tested ranges are still synthetic decision weights rather than carrier-specific economic estimates.

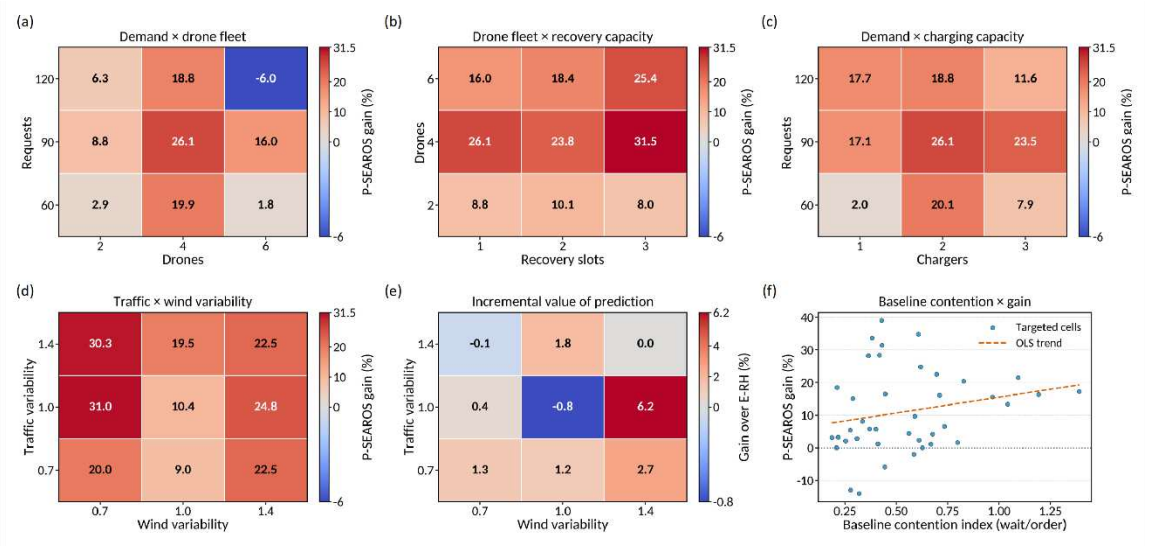


Figure 7. Operating regimes. (a) Demand versus drone-fleet size. (b) Drone fleet versus recovery capacity. (c) Demand versus charging capacity. (d) Traffic versus wind variability. (e) Incremental value of prediction when externality-aware scoring is retained. (f) Relationship between baseline contention and P-SEAROS gain. Heat-map values are descriptive means over three targeted paired replications per cell.

One targeted case with 120 requests, six drones, one recovery slot, and two chargers gives a mean gain of -6.0%. This three-replication cell suggests that a tight structural recovery bottleneck can overwhelm a local scoring advantage, but it is not used for formal inference. Across the sampled traffic-wind variability cells, the full P-SEAROS gain is generally positive, while the incremental value of prediction conditional on externality descriptively ranges from about -1% to 6%. The p95 runtime of P-SEAROS is 7.49, 11.09, and 14.11 ms per order at 80, 100, and 120 requests. These timings describe the tested implementation and are not treated as a large-scale complexity study.

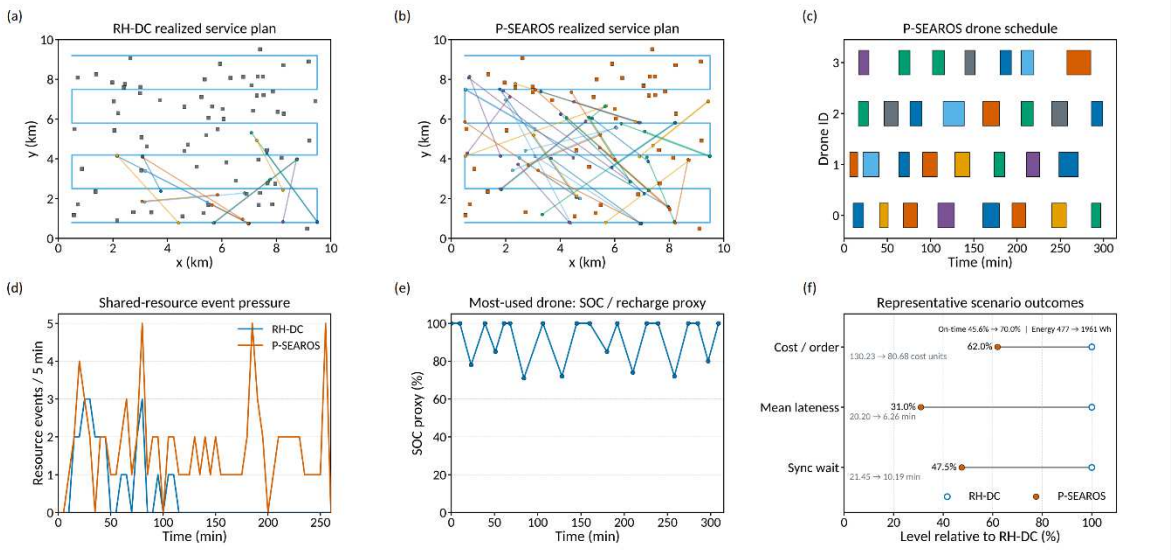


Figure 8. Median-gain representative controlled synthetic test scenario. The run was selected as the default-configuration 90-request case whose paired cost improvement was closest to the median of the three matched runs in that regime cell. (a) Realized RH-DC service plan. (b) Realized P-SEAROS service plan. (c) P-SEAROS drone schedule. (d) Shared-resource event pressure. (e) Recharge/SOC proxy for the most-used drone. (f) Representative scenario outcomes relative to RH-DC. The scenario is illustrative and is not presented as field data.

Figure 8 uses a pre-specified median-gain selection rule rather than an extreme-case choice. Within the default-configuration 90-request operating-regime cell, the plotted run is the one whose paired cost improvement is closest to the median of the three matched runs in that cell. In this run, P-SEAROS lowers cost per order from 130.23 to 80.68, raises the on-time rate from 45.6% to 70.0%, and reduces mean lateness from 20.20 to 6.26 min. Synchronization waiting falls from 21.45 to 10.19 min. Aerial energy rises from 477 to 1961 Wh, showing the trade-off between additional drone use and reduced service delay. The SOC panel represents recharge availability rather than an electrochemical battery model.

6. Conclusions

This study develops an event-driven rolling insertion method for a RoboVan carrying several delivery drones. The method explicitly accounts for the part of a candidate assignment's marginal cost transferred to other jobs through shared launch, recovery, charging, and rendezvous resources. On held-out data, the externality estimator achieves a Spearman correlation of 0.612 and a raw MAE of 4.74 cost units. Across 20 paired scheduling seeds, P-SEAROS reduces mean cost by 17.2%, 17.0%, and 14.3% at 80, 100, and 120 requests, with Holm-adjusted two-sided p-values below 0.01. The same 20-seed mechanism comparison shows that P-RH yields only 2.4-2.7% gains over RH-DC, whereas E-RH remains within about 1.1-1.3% of P-SEAROS, supporting externality-aware scoring as the main contributor and prediction as a smaller refinement. Exact predicted marginal scoring within the fixed candidate neighborhood and replay model improves cost by only 1.2-1.3% relative to P-SEAROS and requires about 1.7-1.9 times the online computation, although both remain below 30 ms/order. Candidate-level selection regret is small despite a top-1 agreement of 0.41, with median and 95th-percentile regrets of 0.42 and 5.80 cost units. Objective-weight sensitivity at 100 requests preserves a positive P-SEAROS advantage under low and high settings for energy, lateness, rendezvous waiting, and missed-rendezvous penalties. Figure 7 remains descriptive because each regime cell uses three paired replications. The evidence is entirely synthetic and does not include a public road-network, digital-twin, or field deployment case, so external validity remains an important next step.

These results support synchronization externality as a useful online decision signal rather than a universal dominance claim.

References

1. Murray, C. C. & Chu, A. G. The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. *Transp. Res. Part C Emerg. Technol.* **54**, 86-109 (2015).
2. Agatz, N., Bouman, P. & Schmidt, M. Optimization approaches for the traveling salesman problem with drone. *Transp. Sci.* **52**, 965-981 (2018).
3. Murray, C. C. & Raj, R. The multiple flying sidekicks traveling salesman problem: parcel delivery with multiple drones. *Transp. Res. Part C Emerg. Technol.* **110**, 368-398 (2020).
4. Macrina, G., Di Puglia Pugliese, L., Guerriero, F. & Laporte, G. Drone-aided routing: a literature review. *Transp. Res. Part C Emerg. Technol.* **120**, 102762 (2020).
5. Dorling, K., Heinrichs, J., Messier, G. G. & Magierowski, S. Vehicle routing problems for drone delivery. *IEEE Trans. Syst. Man Cybern. Syst.* **47**, 70-85 (2017).
6. Zhang, J., Campbell, J. F., Sweeney, D. C. II & Hupman, A. C. Energy consumption models for delivery drones: a comparison and assessment. *Transp. Res. Part D Transp. Environ.* **90**, 102668 (2021).
7. Morandi, N., Leus, R., Matuschke, J. & Yaman, H. The traveling salesman problem with drones: the benefits of retraversing the arcs. *Transp. Sci.* **57**, 1340-1358 (2023).
8. Nguyen, M. A. & Ha, M. H. The parallel drone scheduling traveling salesman problem with collective drones. *Transp. Sci.* **57**, 866-888 (2023).
9. Levin, M. W. & Rey, D. Branch-and-price for drone delivery service planning in urban airspace. *Transp. Sci.* **57**, 843-865 (2023).
10. Gao, J., Zhen, L., Laporte, G. & He, X. Scheduling trucks and drones for cooperative deliveries. *Transp. Res. Part E Logist. Transp. Rev.* **178**, 103267 (2023).
11. Gu, R., Liu, Y. & Poon, M. Dynamic truck-drone routing problem for scheduled deliveries and on-demand pickups with time-related constraints. *Transp. Res. Part C Emerg. Technol.* **151**, 104139 (2023).
12. Yang, Y., Yan, C., Cao, Y. & Roberti, R. Planning robust drone-truck delivery routes under road traffic uncertainty. *Eur. J. Oper. Res.* **309**, 1145-1160 (2023).
13. Yin, Y., Yang, Y., Yu, Y., Wang, D. & Cheng, T. C. E. Robust vehicle routing with drones under uncertain demands and truck travel times in humanitarian logistics. *Transp. Res. Part B Methodol.* **174**, 102781 (2023).
14. Tang, Z. & van Hoeve, W.-J. Dual bounds from decision diagram-based route relaxations: an application to truck-drone routing. *Transp. Sci.* **58**, 257-278 (2024).
15. Blufstein, M., Lera-Romero, G. & Soulignac, F. J. Decremental state-space relaxations for the basic traveling salesman problem with a drone. *INFORMS J. Comput.* **36**, 1064-1083 (2024).
16. Jiang, J., Dai, Y., Yang, F. & Ma, Z. A multi-visit flexible-docking vehicle routing problem with drones for simultaneous pickup and delivery services. *Eur. J. Oper. Res.* **312**, 125-137 (2024).
17. Meng, S., Li, D., Liu, J. & Chen, Y. The multi-visit drone-assisted routing problem with soft time windows and stochastic truck travel times. *Transp. Res. Part B Methodol.* **190**, 103101 (2024).
18. Xing, J., Guo, T. & Tong, L. Reliable truck-drone routing with dynamic synchronization: a high-dimensional network programming approach. *Transp. Res. Part C Emerg. Technol.* **165**, 104698 (2024).
19. Cui, H., Li, K., Jia, S. & Meng, Q. Dynamic collaborative truck-drone delivery with en-route synchronization

and random requests. *Transp. Res. Part E Logist. Transp. Rev.* **192**, 103802 (2024).

20. She, R. & Ouyang, Y. Hybrid truck-drone delivery under aerial traffic congestion. *Transp. Res. Part B Methodol.* **185**, 102970 (2024).
21. Bi, Z., Guo, X., Wang, J., Qin, S. & Liu, G. Truck-drone delivery optimization based on multi-agent reinforcement learning. *Drones* **8**, 27 (2024).
22. Wang, F., Li, H. & Xiong, H. Truck-drone routing problem with stochastic demand. *Eur. J. Oper. Res.* **322**, 854-869 (2025).
23. Paul, A., Levin, M. W., Waller, S. T. & Rey, D. Data-driven optimization for drone delivery service planning with online demand. *Transp. Res. Part E Logist. Transp. Rev.* **198**, 104095 (2025).
24. Kong, J., Wang, H. & Xie, M. Autonomous delivery vehicle routing problem with drones based on multiple delivery modes. *Comput. Oper. Res.* **179**, 107032 (2025).
25. Kong, J., Xie, M. & Wang, H. Integrating autonomous vehicles and drones for last-mile delivery: a routing problem with two types of drones and multiple visits. *Drones* **9**, 280 (2025).
26. Zhen, L., Gao, J., Wang, S., Laporte, G. & Yue, X. Optimizing an on-demand delivery mode based on trucks and drones. *Transp. Sci.* **59**, 1008-1031 (2025).
27. Hasan, S., Sheltami, T. R. & Mahmoud, A. Traffic-predictive drone scheduling: day-ahead synchronization of mobile depots and parallel aerial sorties in urban airspace. *Drones* **10**, 461 (2026).
28. Shi, Z., Hong, S. & Li, A. Truck-drone collaborative delivery with a mobile drone station trailer. *Transp. Res. Part E Logist. Transp. Rev.* **213**, 104975 (2026).
29. Sun, W., Wu, L. & Zhang, F. An exact algorithm to solve vehicle routing problem with drones for delivery and surveillance tasks after disasters. *Transp. Sci.* **60**, 132-154 (2026).
30. Liu, Y., Fei, Y. & Ma, C. Advances in truck-drone cooperative delivery routing and coordination: a review. *Eur. Transp. Res. Rev.* **18**, 49 (2026).
31. Liang, Y.-J. & Luo, Z.-X. A survey of truck-drone routing problem: literature review and research prospects. *J. Oper. Res. Soc. China* **10**, 343-377 (2022).
32. Ha, Q. M., Deville, Y., Pham, Q. D. & Ha, M. H. On the min-cost traveling salesman problem with drone. *Transp. Res. Part C Emerg. Technol.* **86**, 597-621 (2018).
33. Poikonen, S., Wang, X. & Golden, B. L. The vehicle routing problem with drones: extended models and connections. *Networks* **70**, 34-43 (2017).
34. Wang, X., Poikonen, S. & Golden, B. L. The vehicle routing problem with drones: several worst-case results. *Optim. Lett.* **11**, 679-697 (2017).
35. Wang, Z. & Sheu, J.-B. Vehicle routing problem with drones. *Transp. Res. Part B Methodol.* **122**, 350-364 (2019).
36. Schermer, D., Moeini, M. & Wendt, O. A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Comput. Oper. Res.* **109**, 134-158 (2019).
37. Carlsson, J. G. & Song, S. Coordinated logistics with a truck and a drone. *Manag. Sci.* **64**, 4052-4069 (2018).
38. Li, H., Chen, J., Wang, F. & Zhao, Y. Truck and drone routing problem with synchronization on arcs. *Nav. Res. Logist.* **69**, 884-901 (2022).
39. Salama, M. R. & Srinivas, S. Collaborative truck multi-drone routing and scheduling problem: package delivery with flexible launch and recovery sites. *Transp. Res. Part E Logist. Transp. Rev.* **164**, 102788 (2022).
40. Gu, R., Poon, M., Luo, Z., Liu, Y. & Liu, Z. A hierarchical solution evaluation method and a hybrid algorithm for the vehicle routing problem with drones and multiple visits. *Transp. Res. Part C Emerg. Technol.* **141**,

103733 (2022).

41. Meng, S., Guo, X., Li, D. & Liu, G. The multi-visit drone routing problem for pickup and delivery services. *Transp. Res. Part E Logist. Transp. Rev.* **169**, 102990 (2023).
42. Zhou, H., Qin, H., Cheng, C. & Rousseau, L.-M. An exact algorithm for the two-echelon vehicle routing problem with drones. *Transp. Res. Part B Methodol.* **168**, 124-150 (2023).
43. Campbell, A. M. & Savelsbergh, M. Efficient insertion heuristics for vehicle routing and scheduling problems. *Transp. Sci.* **38**, 369-378 (2004).
44. Pillac, V., Gendreau, M., Guéret, C. & Medaglia, A. L. A review of dynamic vehicle routing problems. *Eur. J. Oper. Res.* **225**, 1-11 (2013).
45. Ulmer, M. W., Goodson, J. C., Mattfeld, D. C. & Hennig, M. Offline-online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transp. Sci.* **53**, 185-202 (2019).
46. Fleckenstein, D., Klein, R., Klein, V. & Steinhardt, C. On the concept of opportunity cost in integrated demand management and vehicle routing. *Transp. Sci.* **59**, 125-142 (2025).