Dynamic Pickup-and-Delivery Routing with Early-Arrival Waiting Limits and Station Relocation

Agrima Khanna*, Sophie Pavia*, Fangqi Liu*,
Ayan Mukhopadhyay*, Abhishek Dubey*
*Vanderbilt University
Nashville, TN, USA

***Abstract*—Demand-responsive transport (DRT) systems provide flexible shared mobility that can be modeled as a dynamic Pickup and Delivery Problem with Time Windows (PDPTW). While routing in dynamic PDPTW has been extensively studied, early-arrival waiting limits and vehicle stationing remain largely overlooked. In practice, vehicles cannot wait indefinitely at pickup locations due to curbside constraints, requiring relocation to designated staging areas. Moreover, idle vehicles should be proactively positioned to anticipate uncertain future demand. Existing approaches typically treat routing and relocation as separate decisions, leading to suboptimal performance. To address these challenges, we introduce the Dynamic Vehicle Routing Problem with Stationing (DVRP-S), which extends PDPTW by integrating early-arrival waiting limits and station assignment within a unified framework. We formulate DVRP-S as a route-based Markov Decision Process (MDP) to capture sequential decision-making under stochastic demand. We propose MC-DVRPS, a two-stage solution that combines a rolling-horizon routing solver with a Monte Carlo Tree Search (MCTS) module for routing-aware idle vehicle relocation. By jointly optimizing routing and stationing under uncertainty, our approach reduces unnecessary empty travel and improves service efficiency. Experiments using real-world data from a mid-sized U.S. public transit agency demonstrate significant improvements in service rates over state-of-the-art dynamic PDPTW baselines.**

***Index Terms*—Dynamic Vehicle Routing, Monte Carlo Tree Search, Smart Transportartion, Proactive Stationing, PDPTW.**

I. Introduction

Demand-responsive transport (DRT) systems, including services such as microtransit and paratransit, are an increasingly important component of modern urban mobility, providing flexible, shared transportation that complements traditional fixed-route transit networks [1]. Unlike conventional bus services, DRT platforms dynamically route vehicles to serve pickup and drop-off requests that may arrive both in advance and in real time. This operational setting requires continuous decision-making under capacity and temporal constraints. From an optimization perspective, DRT routing can be modeled as a dynamic Pickup and Delivery Problem with Time Windows (PDPTW), where requests arrive online and must be assigned under capacity and temporal constraints [2]. The problem is NP-hard [3] and further complicated by uncertainty in future demand. Although prior work has demonstrated scalable real-time ride-matching for dynamic PDPTW [4], most formulations assume unrestricted waiting upon early arrival at pickup locations. In practice, curbside management in DRT systems imposes early-arrival waiting limits. Studies report that ride-hailing and shared mobility services intensify competition for limited curb space [5], and vehicles may double-park or block traffic lanes while waiting for passengers [6]. Without coordinated stationing infrastructure, fleets may cruise urban streets to avoid parking, further contributing to congestion and unnecessary vehicle travel [7]. As a result, agencies often impose early-arrival waiting limits at pickup locations, requiring vehicles predicted to arrive too early to station at designated parking areas rather than idle curbside. These operational constraints are not captured in classical PDPTW formulations, which allow unrestricted waiting upon early arrival. When early-arrival waiting limits are imposed, relocation becomes unavoidable and can no longer be treated as an auxiliary adjustment. Instead, relocation must be embedded directly into routing decisions. Without coordinated optimization, naive strategies such as returning vehicles to a depot can significantly increase deadheading [5, 8], leading to higher operational costs. Consequently, vehicle routing, relocation, and scheduling become intrinsically coupled, fundamentally altering the structure of the dynamic PDPTW. This structural coupling is further complicated in partially dynamic environments, where new requests arrive online while previously accepted trips must still be honored. Dynamic rerouting must preserve service commitments for assigned and onboard passengers while maintaining feasibility under capacity and waiting-limit constraints. Moreover, routing decisions cannot remain purely reactive. Under uncertain future demand, idle vehicles must be proactively repositioned to promising staging locations to improve future service efficiency. Although prior work has explored vehicle rebalancing under demand imbalance [9, 10, 11] and systems such as A-RTRS [12] integrate forecast-driven relocation, repositioning is typically treated as a separate optimization stage decoupled from routing feasibility and operational constraints. To address curbside-driven waiting constraints and proactive vehicle positioning in DRT systems, we study the Dynamic Vehicle Routing Problem with Stationing (DVRP-S). DVRP-S extends classical PDPTW by introducing early arrival waiting limits and station relocation decisions, making station selection an integral component of routing. We first present a deterministic MILP formulation of DVRP-S that captures routing feasibility and waiting-limit constraints. We then extend the model to a partially dynamic setting with sequential request arrivals and uncertain demand, formulating DVRP-S as a route-based

Markov Decision Process (MDP) [13]. To solve the dynamic DVRP-S, we propose a two-stage decision framework, MC-DVRPS. At each decision epoch, a rolling-horizon PDPTW-S solver updates routing decisions under waiting-limit and capacity constraints, while a Monte Carlo Tree Search (MCTS) module determines proactive station assignments for idle vehicles. MCTS evaluates candidate stationing actions via routing-aware rollouts. For each sampled future demand realization, routing decisions are recomputed to estimate downstream service performance and operational cost. This simulation-based lookahead enables effective decision-making under stochastic uncertainty [14, 15, 16]. Our main contributions are:

- **DVRP-S with early-arrival waiting limits.** We introduce DVRP-S by incorporating early-arrival waiting limits and station relocation directly into dynamic routing.
- **MDP-based dynamic model.** We formulate DVRP-S as a route-based MDP that integrates routing feasibility with proactive stationing under stochastic demand.
- **Routing-aware MCTS framework.** We propose MC-DVRPS, a two-stage approach that combines a rolling-horizon routing solver with routing-aware MCTS for idle vehicle relocation.
- **Empirical Evaluation.** Using real-world transit data from a mid-sized U.S. city, we compare DVRP-S against myopic and state-of-the-art dynamic VRP baselines with relocation, demonstrating significant improvements in service rates under resource-constrained settings.

II. Related Work

The dynamic PDPTW requires vehicles to serve origin-destination pairs under capacity and time-window constraints with online request arrivals [3, 2, 17]. Recent extensions incorporate operational constraints such as driver breaks [18], and Alonso-Mora et al. [4] introduced shareability graphs for scalable real-time ride-matching. However, these formulations allow unrestricted waiting at pickup locations and do not address *where* idle vehicles should wait, a constraint imposed by transit agencies enforcing early-arrival waiting limits.

Vehicle relocation addresses supply-demand imbalances through zone-based forecasting [9], robust optimization [10], network-flow models [11], and MPC-based redistribution [12]. These methods treat relocation as a separate stage decoupled from routing, so routing decisions are made without awareness of relocation obligations or waiting limits. RL-based approaches [19, 20] learn joint dispatching and relocation policies but operate over spatial zones on large commercial fleets and do not model time windows or ride-pooling, making them a poor fit for small DRT fleets where individual vehicle placement has outsized impact. MCTS has been applied to fleet composition [16] and online paratransit scheduling [15], demonstrating that simulation-based lookahead outperforms myopic policies in resource-constrained settings, but these works assume vehicles can idle freely and do not incorporate stationing constraints.

Across all methods, two gaps persist: (i) relocation and routing are solved independently, so station detours are not accounted for during route construction, and (ii) no existing formulation embeds early-arrival waiting limits as hard constraints within routing. MC-DVRPS addresses both by integrating stationing constraints directly into routing (DVRP-S) and using MCTS with routing-aware rollouts to select station assignments by simulating request-level demand forward to end of day.

III. Problem Statement

This section first presents a deterministic mixed-integer linear programming (MILP) formulation of the Dynamic Vehicle Routing Problem with Stationing (DVRP-S) under early-arrival waiting limits. In this setting, all requests are assumed to be known in advance, and the formulation captures routing feasibility, waiting-limit constraints, and station assignment decisions. We then extend DVRP-S to a partially dynamic setting in which requests arrive sequentially, and future demand is uncertain. To model sequential decision-making under stochastic arrivals, we formulate the problem as a route-based Markov Decision Process (MDP).

A. DVRP-S Problem Formulation

We consider an online pickup-and-delivery problem with time windows and stationing constraints, referred to as DVRP-S, which builds upon the classical PDPTW formulation [21]. At each decision epoch, the system observes the current fleet state together with the set of active requests and computes updated routes via replanning.

Nodes and Request Structure. At each decision epoch, the routing solver receives an instance consisting of the currently active requests. Let P denote the set of pickup nodes in the current epoch. For each pickup $i \in P$, there exists a corresponding dropoff node $n+i$, and we define $D = \{\, n+i : i \in P \,\}$. The full node set is $N = P \cup D$. Each node $i \in N$ has service time s_i and time window $[a_i, b_i]$. Each pickup $i \in P$ has demand $d_i > 0$, and the load change at node i is defined as $\ell_i = d_i$ for pickups and $\ell_{n+i} = -d_i$ for dropoffs.

Online Request States. In the online setting, requests may span multiple decision epochs. Vehicles may already carry passengers from prior epochs; these are modeled as dropoff-only nodes D_0 that must be served and remain locked to their assigned vehicles. Accordingly, the node set becomes $N = P \cup D \cup D_0$. In addition, each pickup $i \in P$ is associated with an indicator $tag(i) \in \{0, 1\}$, where $tag(i) = 1$ denotes a previously accepted request whose pickup has not yet occurred. Such committed requests must be served in the current replanning instance.

Vehicle. Each vehicle $k \in K$ is characterized by its current location $o(k)$, its current onboard load $\bar{y}_k \geq 0$, and its maximum capacity C_k. We denote by A the set of feasible arcs, i.e., all directed pairs (i, j) representing allowable vehicle movements (including arcs from depots to nodes, between nodes, and from nodes to depots) over the active node set. For vehicle k, the travel time and travel cost on arc $(i, j) \in A$ are denoted by τ_{ijk} and c_{ijk}, respectively.

Stations. The service area contains a set of designated stations $\mathcal{S} = \{s_1, \ldots, s_{|\mathcal{S}|}\}$ where vehicles are permitted to idle when not actively serving passengers. For each feasible arc $(i, j) \in A$ and vehicle k, we assume that a station visit is implemented by detouring via the station that minimizes the incremental travel time from i to j, i.e., $\arg\min_{s\in\mathcal{S}}\big(\tau_{isk} + \tau_{sjk} - \tau_{ijk}\big)$. We precompute the resulting additional travel time and cost relative to traveling directly from i to j, and denote them by τ^{stn}_{ijk} and c^{stn}_{ijk}, respectively.

Early-Arrival Waiting Limit. An early-arrival waiting limit $\delta > 0$ is imposed. If an empty vehicle would arrive at its next scheduled node more than δ time units before the beginning of its service window, it is prohibited from waiting in place and must instead visit a station before proceeding.

Decision Variables. We introduce two binary routing variables. The routing variable $x_{ijk} \in \{0, 1\}$ equals 1 if vehicle k travels directly from node i to node j, and 0 otherwise. The stationing variable $\sigma_{ijk} \in \{0, 1\}$ equals 1 if vehicle k detours via a station when traversing arc (i, j), and 0 otherwise.

To track vehicle state evolution along a route, we introduce continuous variables $y_{ik} \geq 0$ representing the onboard load of vehicle k after serving node i, and $t_{ik} \geq 0$ representing the service start time of vehicle k at node i.

1) Traditional PDPTW Constraints: We next summarize the standard PDPTW constraints governing routing, capacity, and time feasibility.

Routing and flow conservation. Vehicles must start and end at their depots and satisfy flow conservation:

$$\sum_{j\in P\cup\{d(k)\}} x_{o(k),j,k} = 1, \qquad \forall k \in K, \tag{1}$$

$$\sum_{i\in D\cup\{o(k)\}} x_{i,d(k),k} = 1, \qquad \forall k \in K, \tag{2}$$

$$\sum_{i\in N\cup\{o(k)\}} x_{ijk} = \sum_{i\in N\cup\{d(k)\}} x_{jik}, \forall k \in K,\ j \in N. \tag{3}$$

Pickup–delivery pairing is enforced as

$$\sum_{j\in N} x_{ijk} = \sum_{j\in N} x_{j,n+i,k}, \quad \forall k \in K,\ i \in P. \tag{4}$$

A dropoff may be visited only if its pickup is served:

$$\sum_{k\in K} \sum_{j\in N\cup\{d(k)\}} x_{n+i,j,k} \leq r_i, \quad \forall i \in P. \tag{5}$$

Capacity constraints. The variable $y_{ik} \geq 0$ denotes the onboard load of vehicle k after serving node i:

$$y_{o(k),k} = \bar{y}_k, \qquad \forall k \in K, \tag{6}$$

$$(x_{ijk} = 1) \Rightarrow y_{ik} + \ell_j = y_{jk}, \forall k \in K,\ (i, j) \in A, \tag{7}$$

$$0 \leq y_{ik} \leq C_k, \qquad \forall k \in K,\ i \in N. \tag{8}$$

Time Window Constraints. Let $t_{ik} \geq 0$ denote the service start time of vehicle k at node i. *(Indices.)* Unless stated otherwise, constraints below hold for all valid $k \in K$, $i \in N$, and $(i, j) \in A$.

$$t_{o(k),k} = 0, \tag{9}$$

$$(x_{ijk} = 1) \Rightarrow t_{ik} + s_i + \tau_{ijk} + \sigma_{ijk}\tau^{\text{stn}}_{ijk} \leq t_{jk}, \tag{10}$$

$$a_i \leq t_{ik} \leq b_i. \tag{11}$$

Pickup–delivery precedence is enforced as

$$t_{ik} + \tau_{i,n+i,k} + \sigma_{i,n+i,k}\tau^{\text{stn}}_{i,n+i,k} \leq t_{n+i,k}, \quad \forall k \in K,\ i \in P. \tag{12}$$

2) Stationing constraints: Beyond the traditional PDPTW constraints, DVRP-S incorporates stationing requirements to regulate vehicle idling behavior. The stationing variable σ_{ijk} is governed by two conditions: (i) only empty vehicles may detour via a station, and (ii) an empty vehicle must visit a station if the anticipated idle time before the next service exceeds a threshold δ. Unless otherwise stated, the following constraints hold for all $k \in K$ and $(i, j) \in A$.

$$\sigma_{ijk} = 1 \Rightarrow y_{ik} = 0 \tag{13}$$

$$(x_{ijk} = 1) \wedge (y_{ik} = 0) \wedge (t_{jk} - t_{ik} - s_i - \tau_{ijk} \geq \delta) \Rightarrow \sigma_{ijk} = 1 \tag{14}$$

Constraint (13) ensures that a station detour is only allowed when the vehicle is empty. Constraint (14) enforces a mandatory station visit whenever an empty vehicle would otherwise experience an idle gap exceeding δ between consecutive services.

3) Online Consistency Constraints: Because routing decisions are recomputed in a rolling-horizon manner, additional constraints are required to ensure feasibility and service continuity across decision epochs.

Partial requests. Passengers already onboard at the beginning of the current epoch are represented by dropoff-only nodes D_0, each locked to its assigned vehicle. These requests must remain feasible under replanning and therefore must be completed by the same vehicle. For each such node $i \in D_0$ assigned to vehicle k_i, we enforce

$$\sum_{j\in N\cup\{d(k_i)\}} x_{ijk_i} = 1, \quad \forall i \in D_0. \tag{15}$$

Committed requests. Pickup requests previously accepted but not yet served (i.e., $tag(i) = 1$) must be honored in the current replanning instance. While these requests may be reassigned to a different vehicle if beneficial, they cannot be rejected. Accordingly, we impose

$$\sum_{k\in K} \sum_{j\in N\cup\{d(k)\}} x_{ijk} = 1, \quad \forall i \in P \text{ with } tag(i) = 1. \tag{16}$$

This ensures cross-epoch consistency and prevents confirmed requests from being dropped during replanning.

4) Objective: We introduce a serviceability variable r_i for each pickup node $i \in P$, where $r_i = 1$ if request i is served and $r_i = 0$ otherwise:

$$r_i = \sum_{k\in K} \sum_{j\in N\cup\{d(k)\}} x_{ijk}, \quad \forall i \in P. \tag{17}$$

At decision epoch t, we maximize the number of served pickup requests in the active set P_t:

$$\min \ \Big(|P_t| - \sum_{i \in P_t} r_i\Big). \tag{18}$$

B. Event-Driven MDP Formulation of DVRP-S

We consider DVRP-S in an online setting where requests arrive sequentially and future demand is uncertain. Decision epochs are event-driven and occur whenever a new request arrives. At each decision epoch t, the system observes the current fleet state and active requests, and recomputes routing and relocation decisions. Once executed, decisions are irrevocable (e.g., serving a pickup or initiating a relocation), while subsequent arrivals may disrupt the current plan. This sequential decision-making under uncertainty motivates an event-driven Markov Decision Process (MDP).

State. At decision epoch t, the state $s_t \in \mathcal{X}$ is

$$s_t = \Big(\{o_t(k),\, \bar{y}_t(k)\}_{k \in K},\ P_t,\ D_{0,t},\ \{tag_t(i)\}_{i \in P_t},\ \mathcal{S},\ \delta\Big). \tag{19}$$

Here, $o_t(k)$ denotes the current location of vehicle k (a request node or a station) and $\bar{y}_t(k)$ is its onboard load. P_t is the active pickup set, $D_{0,t}$ is the set of dropoff-only nodes for onboard passengers, and $tag_t(i)$ indicates whether pickup i is committed. We define the corresponding dropoff set as $D_t = \{\, n{+}i : i \in P_t \,\}$ and the active node set as $N_t = P_t \cup D_t \cup D_{0,t}$. The station set $\mathcal{S}$ and waiting threshold δ are fixed parameters.

Action. At decision epoch t, an action $a_t \in \mathcal{A}(s_t)$ specifies a replanned fleet schedule over N_t by selecting route arcs $\{x_{ijk}\}$ and station-detour indicators $\{\sigma_{ijk}\}$. The resulting service times $\{t_{ik}\}$ and loads $\{y_{ik}\}$ are induced by these decisions through the DVRP-S feasibility constraints (capacity, time windows, waiting limits, and commitment constraints for $tag_t(i) = 1$). Thus, actions jointly capture reactive request assignment and proactive vehicle positioning.

Transition. The system evolves in an event-driven manner. Given (s_t, a_t), vehicles execute the prescribed routing and stationing decisions until the next arrival event. Between events, fleet progression is deterministic: vehicle locations, loads, and service completion evolve according to travel times and service durations implied by a_t.

Let $z_i \in \{0, 1\}$ indicate whether pickup $i \in P_t$ is served under a_t (and hence generates a completed pickup and a corresponding dropoff). Served pickups are removed once executed, while onboard passengers remain represented via dropoff-only nodes until delivery.

Uncertainty arises from exogenous demand. New pickup requests arrive according to a stochastic spatio-temporal arrival process generating random arrival times, origins, destinations, and time windows. At the next decision epoch $t{+}1$, the state is updated by (i) advancing the fleet deterministically under a_t and (ii) adding newly arrived requests and updating $D_{0,t+1}$ accordingly:

$$s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t), \tag{20}$$

where $\mathcal{P}$ combines deterministic fleet evolution and the stochastic arrival process.

Reward. At decision epoch t, the immediate reward equals the number of pickups served under a_t:

$$g(s_t, a_t) = \sum_{i \in P_t} z_i. \tag{21}$$

Policy. A policy π maps each state s_t to a feasible action $a_t \in \mathcal{A}(s_t)$, where $\mathcal{A}(s_t)$ denotes the set of routing and stationing decisions satisfying the DVRP-S feasibility constraints. Under policy π, actions are selected as $a_t = \pi(s_t)$.

Objective. The objective is to find a policy π^* that maximizes the expected cumulative reward over one operating day:

$$\pi^* = \arg\max_{\pi} \ \mathbb{E}_{\pi}\left[\sum_{t=1}^{T} g(s_t, a_t)\right], \tag{22}$$

where the expectation is taken with respect to the stochastic arrival process.

IV. Methodology

Solving the DVRP-S MDP entails jointly optimizing routing for active requests and relocation for future stochastic demand. Such joint optimization is computationally intractable in real time: routing is NP-hard, relocation decisions scale exponentially with the number of idle vehicles, and relocation choices directly affect future routing feasibility. We therefore adopt a structured decomposition that separates immediate routing optimization from anticipatory relocation, achieving computational tractability while preserving forward-looking decision quality. To operationalize this decomposition, we propose *MCTS-Based Dynamic Vehicle Routing with Stationing (MC-DVRPS)*, a two-stage policy for real-time fleet control under stochastic demand. The overall workflow is illustrated in Figure 1. At each decision epoch triggered by a new request, MC-DVRPS first solves a *Rolling-Horizon DVRP-S* instance to update routes for active requests, and then applies *MCTS-Based Vehicle Relocation* to relocate idle vehicles in anticipation of future arrivals. In this way, the policy balances immediate service maximization with anticipatory relocation under demand uncertainty.

A. Phase 1: Rolling-Horizon DVRP-S

At each decision epoch t, we solve a rolling-horizon DVRP-S instance (Section III-A) based on the current state s_t. The subproblem accounts for vehicle locations $o_t(k)$, onboard loads $\bar{y}_t(k)$, pending dropoffs $D_{0,t}$, confirmed requests, and newly arrived pickups with their time windows. Routes are optimized over the active pickup set P_t, without anticipating future demand. This phase maximizes immediate service feasibility under the current system conditions. We solve this subproblem using the Hexaly optimizer [22], with the objective in (18), which prioritizes maximizing served requests while minimizing travel cost. A time limit of 30 seconds per epoch is imposed to ensure real-time operation. The solver returns updated route plans for all vehicles and determines which active requests are feasibly served under the imposed constraints. Requests that are served in the solution are committed, while infeasible requests are rejected.

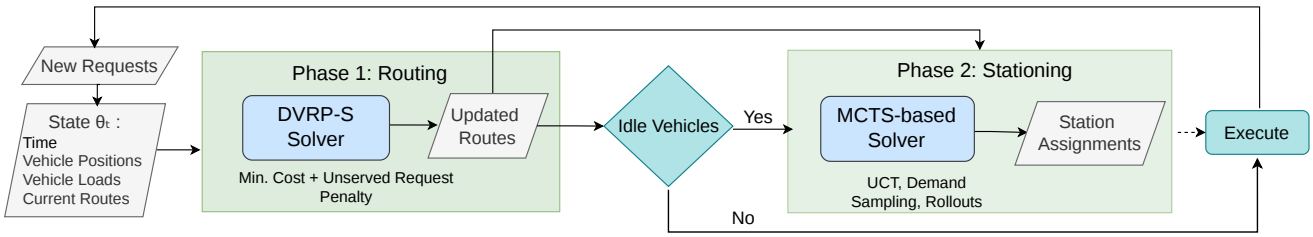


Fig. 1: Decision epoch. A new request and system state enter the DVRP-S solver (Phase 1), producing updated routes. If idle vehicles exist, MCTS (Phase 2) selects station assignments. Routes are executed until the next request arrives.

B. Phase 2: MCTS-based Vehicle Relocation

After the routing phase, the system identifies the set of vehicles eligible for stationing decisions. A vehicle v is considered for relocation if it has zero onboard load and is expected to remain idle beyond the allowable waiting horizon, i.e., its next scheduled service time exceeds δ from the current time. Each such vehicle is currently stationed at some location $s_v^{\text{cur}} \in \mathcal{S}$. The stationing phase determines whether to *relocate* these vehicles across the station network. For each eligible vehicle v, we select a (possibly unchanged) destination station $s_v \in \mathcal{S}$, yielding a relocation plan $\mathbf{s}_t = \{s_v\}_{v \in \mathcal{V}_t^{\text{reloc}}}$. We apply Monte Carlo Tree Search (MCTS) [14] to select these station assignments. The search evaluates candidate relocation plans by simulating future request arrivals sampled from a learned generative model trained on historical data.

1) Search Tree Structure: When multiple idle vehicles require stationing, they are processed sequentially in order of decreasing idle time, with the time budget divided equally among them. For each vehicle, a separate search tree is constructed: the root reflects the current system state, including updated positions of any vehicles already assigned in the current epoch, and each child of the root corresponds to a candidate station in $\mathcal{S}$. This sequential scheme ensures that later assignments account for earlier ones while keeping the branching factor at $|\mathcal{S}|$ per search.

Tree traversal uses the standard Upper Confidence Bound for Trees (UCT) criterion. Let $Q(n')$ denote the cumulative simulation reward accumulated through node n', $N(n')$ the number of times node n' has been visited, and $c > 0$ the exploration constant that controls the trade-off between exploiting high-reward nodes and exploring less-visited ones. The UCT selection rule is:

$$n^* = \arg \max_{n' \in \textit{children}(n)} \left[\frac{Q(n')}{N(n')} + c\sqrt{\frac{\ln N(n)}{N(n')}} \right] \tag{23}$$

We set $c = \sqrt{2}$ in all experiments, following the theoretical recommendation of Kocsis and Szepesvári [14]. Upon reaching a node that is not fully expanded, a child is added for an untried station assignment.

2) Simulation: From an expanded node, we simulate the system forward to the end of the operating day to estimate the long-term value of a partial assignment:

1) *Complete the assignment*: Assign any remaining idle vehicles to their nearest station.
2) *Sample future requests*: We draw request chains from a generative demand model, retaining only requests occurring after the current time t_k.
3) *Simulate decisions*: For each sampled request arriving in sequence, apply a first-insertion heuristic [23] for routing and nearest-station assignment for stationing. The first-insertion heuristic attempts to insert each new request into the route that minimizes the increase in total travel cost while satisfying time-window and capacity constraints; if no feasible insertion exists, the request is rejected.
4) *Compute reward*: Evaluate the service rate over the simulation horizon:

$$\hat{R} = \frac{|\{r : r \text{ served in simulation}\}|}{|\{r : r \text{ arrived in simulation}\}|} \tag{24}$$

The simulation reward $\hat{R}$ is backpropagated along the path from the expanded node to the root, updating visit counts and cumulative rewards at each ancestor.

Generative Demand Model. Following Wilbur et al. [15], we sample future demand from a hierarchical model: a Gaussian over daily request counts and a frequency-weighted pool of observed trips. Synthetic days are pre-generated offline into a library of request chains. At each epoch, three chains are drawn and used to seed parallel MCTS trees whose scores are averaged to hedge against demand uncertainty.

3) Station Selection: After exhausting the time budget, we select the assignment corresponding to the most-visited child: $\mathbf{s}^* = \arg\max_{\mathbf{s}} N(n_{\mathbf{s}})$. We select by visit count rather than mean reward, as visit count is less sensitive to noisy simulation estimates [14]. The vehicle's position is immediately updated to s_v^* before running MCTS for the next idle vehicle. After all idle vehicles have been assigned, the updated positions become the starting locations for the routing solver at subsequent epochs.

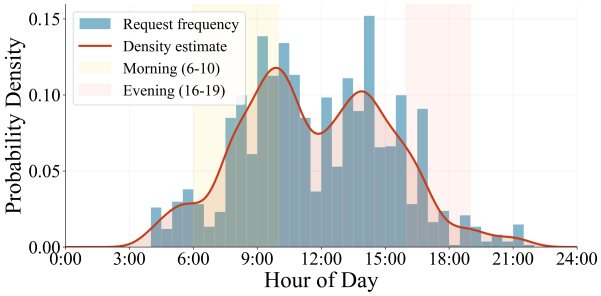
V. Experimental Setup

Table I summarizes all experiment parameters.

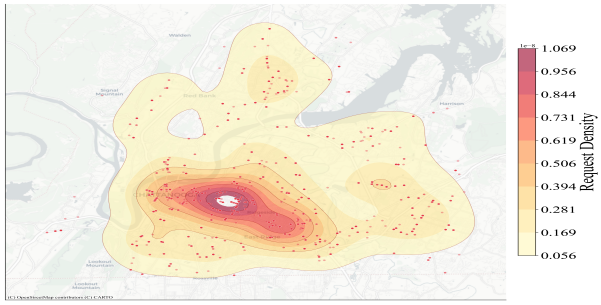
1) Dataset: We use six months of paratransit trip data (January–July 2021) from the Chattanooga Area Regional Transportation Authority (CARTA), comprising 25,843 requests (179 ± 23 per day). Figure 2 shows the spatio-temporal demand distribution where arrivals follow a bimodal pattern

TABLE I: Experiment parameters

Parameter	Value
Region	Chattanooga, TN ($\sim$940 km^2)
Data period	January–July 2021 (6 months)
Total requests	25,843
Requests per day	179 ± 23 (min 144, max 212)
Road network	OpenStreetMap (free-flow speeds)
Test days per configuration	15 (randomly sampled)
Fleet sizes $\lvert K\rvert$	$\{3, 5, 7\}$
Station counts $\lvert\mathcal{S}\rvert$	$\{1, 5, 10\}$
Vehicle capacity C_k	8 passengers
Idle-time threshold δ	900 s (15 min)
Routing solver budget	30 s per epoch
MCTS budget	30 s per epoch
Exploration constant c	$\sqrt{2}$
Demand chains n_{chains}	3



(a) Temporal distribution of request arrivals.



(b) Spatial density of pickup locations.

Fig. 2: Spatio-temporal request patterns in the dataset

peaking mid-morning and mid-afternoon, with pickups concentrated in the downtown core and a secondary northern cluster. This spatial concentration motivates the demand-weighted station placement described below.

2) Road Network: We extracted a routing graph from OpenStreetMap [24] using OSMNX [25] with free-flow travel times as edge weights. Shortest paths between all nodes were precomputed for constant-time lookups.

3) Station Selection: Station locations were selected via demand-weighted DBSCAN clustering over pickup locations, enforcing a minimum inter-station distance. Cluster centers were mapped to the nearest feasible parking area identified in conjunction with the transit agency.

4) Configurations: We evaluate two experiment sets: (1) varying fleet size $|K| \in \{3, 5, 7\}$ with $|\mathcal{S}| = 5$ stations, and (2) varying stations $|\mathcal{S}| \in \{1, 5, 10\}$ with $|K| = 5$ vehicles. All vehicles have capacity $C_k = 8$ and idle-time threshold $\delta = 900$ s.

5) Evaluation Procedure: For each configuration, 15 operating days are sampled uniformly at random; the same days are used across all methods. Real requests are replayed chronologically as online arrivals. The Hexaly routing solver receives a 30-second budget per epoch; MCTS runs an additional 30 seconds when idle vehicles require stationing. MCTS uses exploration constant $c = \sqrt{2}$, $n_{\text{chains}} = 3$ demand chains via root parallelization, first-insertion rollouts with nearest-station assignment, most-visited-child selection, and vehicles ordered by decreasing idle time. Results are reported as boxplots where each point is one test day.

6) Metrics: We report two metrics. **Service rate**: fraction of requests served ($\sum_{i \in P} r_i / n$). **VMT/PMT ratio**: total vehicle miles traveled divided by passenger miles traveled. VMT includes all fleet miles (loaded, deadheading, relocation, depot returns); PMT counts only miles with passengers aboard. A ratio of 1 indicates perfect efficiency; higher values reflect greater deadheading.

7) Baselines: All baselines use the same vehicles, capacity, and stations as MC-DVRPS.

- **Myopic:** Assigns requests via the RTV-graph framework [4], selecting the lowest-cost feasible assignment. No stationing is performed; requests triggering a waiting-limit violation are rejected.
- **Greedy Stationing Myopic:** Uses the same myopic RTV-graph routing as above, with a greedy stationing heuristic that selects the station minimizing total travel time (current location $\rightarrow$ station $\rightarrow$ next request).
- **Greedy Stationing MCTS:** Uses MCTS for routing [15] with the same greedy stationing heuristic.
- **A-RTRS [12]:** Integrates dispatching with MPC-based relocation using learned zone-level demand forecasts. We adapt it to our setting by mapping zone-level relocation decisions to the nearest station.

VI. Results

We evaluate all methods across two experiment sets: varying fleet size (3, 5, and 7 vehicles with 5 stations) and varying station count (1, 5, and 10 stations with 5 vehicles). Figure 3 summarizes all results.

a) Service Rate Across Fleet Sizes.: As shown in Figure 3(a), MC-DVRPS achieves the highest median service rate across all fleet sizes. With 5 vehicles, MC-DVRPS serves approximately 90% of requests, compared to roughly 88% for A-RTRS and 65% for Greedy Stationing MCTS. The gap is most pronounced at 3 vehicles, where MC-DVRPS reaches a median of approximately 72% which is more than double the next-best non-anticipatory method. This demonstrates that simulation-based lookahead is most valuable when fleet resources are scarce: by anticipating future demand through sampled request chains, MC-DVRPS positions vehicles where they can feasibly serve upcoming requests rather than stranding them at suboptimal stations. At 7 vehicles, MC-DVRPS still leads but all non-myopic methods converge above 90%,

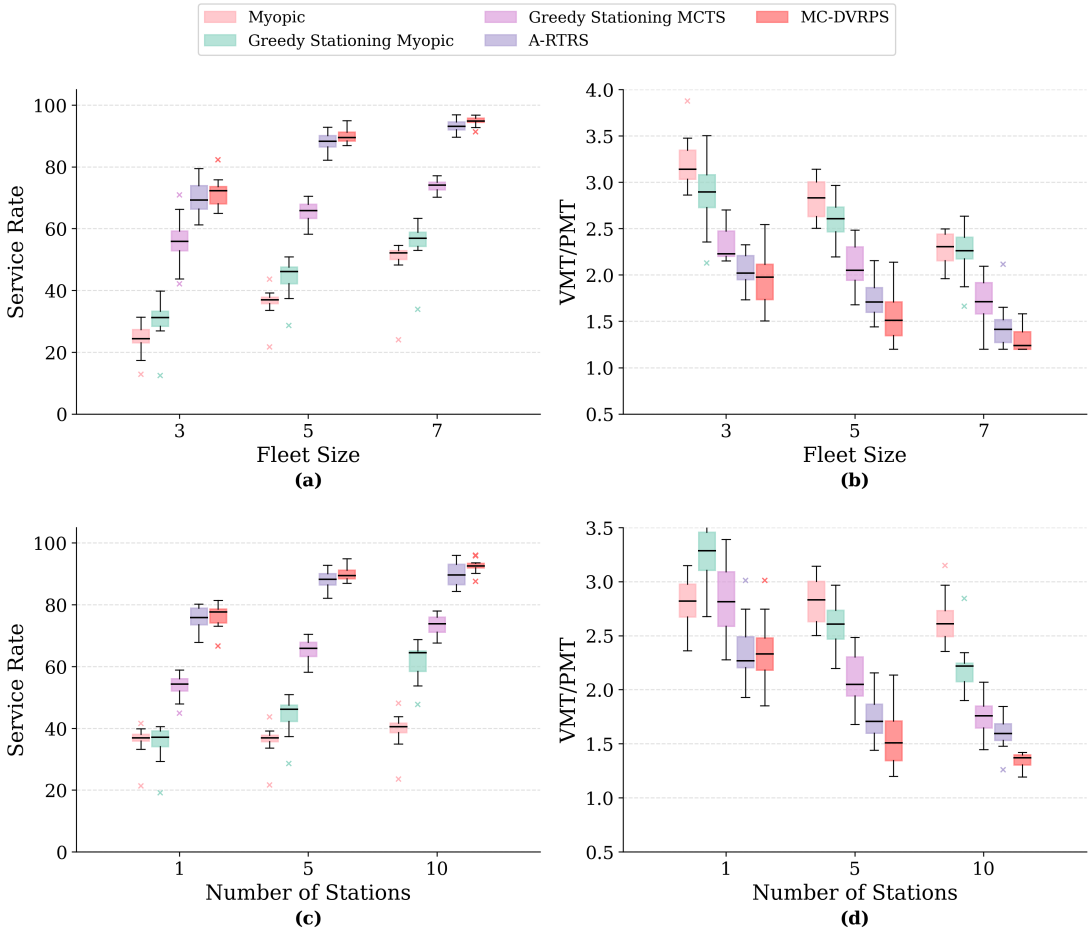


Fig. 3: Performance across all configurations. Top row: varying fleet size ($|K| \in \{3, 5, 7\}$, $|\mathcal{S}| = 5$ stations). Bottom row: varying station count ($|\mathcal{S}| \in \{1, 5, 10\}$, $|K| = 5$ vehicles). Left column: service rate (% requests served; higher is better). Right column: VMT/PMT ratio (lower is better). Each boxplot summarizes 15 test days.

indicating that the computational investment in MCTS yields the greatest marginal benefit in resource-constrained regimes where each stationing decision has outsized impact on downstream serviceability.

b) VMT/PMT Across Fleet Sizes.: Figure 3(b) shows that MC-DVRPS consistently achieves the lowest VMT/PMT ratio. At 5 vehicles, MC-DVRPS achieves a median of approximately 1.5, compared to 1.7 for A-RTRS, 2.0 for Greedy Stationing MCTS, and 2.85 for Myopic. The advantage persists at 7 vehicles (median ≈1.2 vs. ≈1.4 for A-RTRS). This efficiency gain is a direct consequence of routing-aware rollouts: because each MCTS simulation evaluates station assignments by running the full routing process forward, MC-DVRPS selects positions that minimize future empty travel, not just immediate detour cost. The simultaneous improvement in both service rate and VMT/PMT confirms that anticipatory stationing reduces deadheading that would otherwise cause time-window violations and cascading rejections.

c) Service Rate Across Different Numbers of Stations: Figure 3(c) reveals that increasing the number of stations steadily improves performance for all methods. At 1 station, even MC-DVRPS achieves only ≈75%, as all vehicles are funneled to the same location regardless of where future demand is concentrated, eliminating the spatial advantage of intelligent stationing. Moving from 1 to 5 stations yields the largest gain for MC-DVRPS, reaching ≈90%. At 10 stations, MC-DVRPS reaches ≈93%, with minimal returns consistent with the concentrated demand distribution shown in Figure 2: 5 well-placed stations already capture most of the spatial benefit in this service area, and additional stations offer only marginal reductions in detour cost.

d) VMT/PMT Across Different Numbers of Stations.: As shown in Figure 3(d), routing efficiency improves monotonically with available stations. MC-DVRPS achieves a median VMT/PMT of approximately 2.3 at 1 station, 1.5 at 5 stations, and 1.4 at 10 stations, maintaining the lowest ratio across all configurations. The largest efficiency gain occurs between 1 and 5 stations, where the ability to route idle vehicles to spatially distributed stations substantially reduces empty miles. At $|\mathcal{S}| = 10$, MC-DVRPS approaches a VMT/PMT of 1.4, suggesting that with sufficient station coverage and anticipatory positioning, nearly all vehicle travel is productive. The

narrowing gap between 5 and 10 stations further supports that the demand-weighted station placement captures the dominant spatial structure of this service area, and that MC-DVRPS effectively exploits even a modest number of stations through its lookahead mechanism.

VII. Conclusion

We introduced the Dynamic Vehicle Routing Problem with Stationing (DVRP-S), which embeds early-arrival waiting limits and station relocation decisions directly into the dynamic PDPTW, fundamentally coupling routing feasibility with vehicle repositioning. We presented a deterministic MILP formulation and extended DVRP-S to a partially dynamic setting via a route-based MDP. To solve the dynamic problem, we proposed MC-DVRPS, a two-stage framework that combines rolling-horizon routing with routing-aware Monte Carlo Tree Search (MCTS) for proactive stationing under stochastic demand. Experiments on real-world DRT data demonstrate that MC-DVRPS consistently achieves the highest service rates and lowest VMT/PMT ratios, with the largest gains in resource-constrained settings where anticipatory positioning is most impactful. Current limitations include the static travel time model, which assumes free-flow speeds despite real-world time-of-day variations—and the computational cost of MCTS rollouts that grows with the number of eligible vehicles and stations; future work will explore learned rollout policies, time-dependent speed profiles, and hierarchical station clustering to improve scalability and realism.

References

[1] P. Vansteenwegen, L. Melis, D. Aktaş, K. Sørensen, F. Vieira, and B. Montenegro, "A survey on demand-responsive public bus systems," *Transportation Research Part C*, 2022.

[2] G. Berbeglia, J.-F. Cordeau, and G. Laporte, "Dynamic pickup and delivery problems," *European Journal of Operational Research*, 2010.

[3] Y. Dumas, J. Desrosiers, and F. Soumis, "The pickup and delivery problem with time windows," *European Journal of Operational Research*, 1991.

[4] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proc. National Academy of Sciences*, 2017.

[5] A. Henao and W. Marshall, "The impact of ride-hailing on parking (and vice versa)," *Journal of Transport and Land Use*, 2019.

[6] B. Grush, "Pick-up and drop-off at the curb," Urban Robotics Foundation, 2025.

[7] A. Millard-Ball, "The autonomous vehicle parking problem," *Transport Policy*, 2019.

[8] A. Brown, V. Mukhija, and D. Shoup, "Where ridehail drivers go between trips," *Transportation*, 2023.

[9] A. Wallar, M. Van Der Zee, J. Alonso-Mora, and D. Rus, "Vehicle rebalancing for mobility-on-demand systems with ride-sharing," in *IEEE/RSJ IROS*, 2018.

[10] L. He, Z. Hu, and M. Zhang, "Robust repositioning for vehicle sharing," *Manufacturing & Service Operations Management*, 2020.

[11] R. Iglesias, F. Rossi, R. Zhang, and M. Pavone, "A BCMP network approach to modeling and controlling autonomous mobility-on-demand systems," *Intl. Journal of Robotics Research*, 2019.

[12] C. Riley, P. van Hentenryck, and E. Yuan, "Real-time dispatching of large-scale ride-sharing systems: Integrating optimization, machine learning, and model predictive control," in *IJCAI*, 2020.

[13] M. Ulmer, "Route-based Markov decision processes for dynamic vehicle routing problems," Technische Universität Braunschweig, Tech. Rep., 2017.

[14] C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of Monte Carlo tree search methods," *IEEE Trans. Computational Intelligence and AI in Games*, 2012.

[15] M. Wilbur, S. Kadir, Y. Kim, G. Pettet, A. Mukhopadhyay, P. Pugliese, S. Samaranayake, A. Laszka, and A. Dubey, "An online approach to solve the dynamic vehicle routing problem with stochastic trip requests for paratransit services," in *ACM/IEEE ICCPS*, 2022.

[16] T. Baltussen, M. Goutham, O. Ashour, S. Meissner, A. Wortmann, and A. Fay, "A parallel Monte-Carlo tree search-based metaheuristic for optimal fleet composition considering vehicle routing," in *IEEE CASE*, 2023.

[17] H. Psaraftis, M. Wen, and C. Kontovas, "Dynamic vehicle routing problems: Three decades and counting," *Networks*, 2016.

[18] A. Khanna, F. Liu, S. Gupta, S. Pavia, A. Mukhopadhyay, and A. Dubey, "PDPTW-DB: MILP-based offline route planning for PDPTW with driver breaks," in *ACM ICDCN*, 2025.

[19] J. Holler, R. Vuorio, Z. Qin, X. Tang, Y. Jiao, T. Jin, S. Singh, C. Wang, and J. Ye, "Deep reinforcement learning for multi-driver vehicle dispatching and repositioning problem," in *IEEE International Conference on Data Mining (ICDM)*, 2019.

[20] Y. Jiao, X. Tang, Z. Qin, S. Li, F. Zhang, H. Zhu, and J. Ye, "Real-world ride-hailing vehicle repositioning using deep reinforcement learning," *Transportation Research Part C: Emerging Technologies*, 2021.

[21] P. Toth and D. Vigo, *The Vehicle Routing Problem*. SIAM, 2002.

[22] Hexaly, "Hexaly optimizer," 2024, version 12.5. [Online]. Available: https://www.hexaly.com

[23] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, 1987.

[24] OpenStreetMap contributors, "OpenStreetMap," https://www.openstreetmap.org, 2021, accessed: 2021.

[25] G. Boeing, "OSMnx," *Computers, Environment and Urban Systems*, 2017.