How Robust Reinforcement Learning Enables Courier-Friendly Route Planning for Last-Mile Delivery?

Ziying Jia [1] Zeyu Dong [2] Miao Yin [1] Sihong He [1]

Abstract

Last-mile delivery (LMD) systems increasingly face dynamic customer demands that introduce uncertainty and lead to unstable delivery routes, reducing efficiency and placing cognitive burdens on couriers. To address this, we propose R^3S^2Route, a Robust Regularizer-enhanced RL-based Smooth and Stable Routing Algorithm that learns courier-friendly policies under state uncertainty. Our method adopts an actor-critic reinforcement learning framework and incorporates a robustness regularizer to penalize policy sensitivity to input perturbations. We formally define route smoothness and stability as courier-friendliness metrics, and integrate them into the learning framework to produce routing policies that are both geometrically intuitive and keep spatial-temporal consistent. Experimental results demonstrate that R^3S^2Route achieves up to 59.68% improvement in route smoothness and 14.29% in route stability, while maintaining low travel distances and time-window violation rates, outperforming several baselines in dynamic delivery environments.

1. Introduction

The last-mile delivery (LMD) problem focuses on optimizing the delivery sequence in the final stage of the supply chain, where goods are transported from distribution centers to end customers by couriers (Boysen et al., 2021). This stage is widely considered the most complex and costly segment of the delivery process (Macioszek, 2018). With the development of IoT technologies and e-commerce platforms, delivery companies are providing increasingly flexible services, allowing customers to choose preferred delivery times and locations, monitor real-time package status, request door-to-door pickup services, and adjust delivery/pickup options in real time (Wanganoo & Patil, 2020).

Though the increased service flexibility enhances customer experience and satisfaction, it also introduces greater uncertainty to LMD, posing significant challenges for robust delivery route planning and effective human-in-the-loop execution. For example, customers using services like Amazon Prime frequently change their delivery preferences, which drives the logistics system to dynamically re-optimize routes. The frequent route changes without considering route stability introduce cognitive burdens on couriers, reducing acceptance of system instructions and diminishing the overall efficiency of the delivery process.

However, robust route planning that explicitly considers courier-friendly decision execution remains challenging and underexplored. Most existing approaches focus on computational efficiency, robustness to dynamic factors and uncertain predictions, and fair workload or monetary incentive to couriers utilizing exact optimization techniques (Lysgaard et al., 2004; Gauvin et al., 2014), heuristic and metaheuristic methods (Pisinger & Ropke, 2007; Li & Li, 2020; Matl et al., 2018), and learning-based algorithms (Nazari et al., 2018; Pan & Liu, 2023). Without dedicated design, merely incorporating courier-friendliness metrics into these methods fails to achieve simultaneous optimization of courier-friendliness and robustness due to their inherently conflicting objectives. Robust optimization methods typically aim to optimize for worst-case scenarios, often resulting in aggressive uncertain variables estimation during re-routing. As a result, the newly generated routes may differ significantly from previous ones. In contrast, courier-friendliness emphasizes route stability and alignment with human preferences—such as minimizing zigzag patterns. Moreover, courier-friendliness metrics such as route stability and smoothness are often nonlinear, non-differentiable, and difficult to formalize. This makes it challenging to incorporate them into traditional gradient-based or optimization frameworks.

To address the aforementioned challenges, we propose a reinforcement learning (RL)-based robust LMD framework

[1]Department of Computer Science and Engineering, University of Texas at Arlington, TX, US [2]Department of Applied Math and Statistics, Stony Brook University, NY, US. Correspondence to: Ziying Jia, Zeyu Dong, Miao Yin, Sihong He <zxj3060@mavs.uta.edu, zeyu.dong@stonybrook.edu, miao.yin@uta.edu, sihong.he@uta.edu>.

ICML 2025 Workshop on Programmatic Representations for Agent Learning, Vancouver, Canada. Copyright 2025 by the author(s).

that incorporates courier-friendliness into robust route planning under state uncertainty. Our framework is designed to handle both environmental uncertainties and route constraints in a unified manner. Specifically, we formally define two key metrics—route stability and route smoothness—to quantify courier-friendliness. Unlike traditional optimization methods, our RL-based approach is not limited by the nonlinearity or non-differentiability of these metrics. By carefully designing the reward function and optimization algorithm within the RL framework, we avoid the limitations of linear combination-based multi-objective optimization, which often fails to balance conflicting objectives effectively. Our main **contributions** are as follows:

(1) To the best of our knowledge, we are the first to formulate the robust courier-friendly last-mile delivery (LMD) problem under prediction error and dynamic environments as a robust reinforcement learning problem under state uncertainty. Via a proper design of agent, state, action, reward and robust policy, we set the goal of the problem as producing robust courier-friendly routing policy in terms of route smoothness and stability.

(2) We design a **R**obust **R**egularizer-enhanced **R**L-based **S**mooth and **S**table Routing Algorithm ($\mathrm{R}^3\mathrm{S}^2$Route) to learn smooth and stable routing policies resilient to dynamic delivery uncertainties. It adopts an actor-critic framework and a robustness regularizer that constrains policy sensitivity to input perturbations.

(3) We run experiments to demonstrate the effectiveness of $\mathrm{R}^3\mathrm{S}^2$Route. Experiments show that our proposed algorithm performs better in terms of route smoothness and stability while maintaining low travel distance and time window violation rates. Specifically, compared to Tabu Search, $\mathrm{R}^3\mathrm{S}^2$Route increases route smoothness by 31.61% and improves route stability by 6.67%; compared to RL-based baseline, $\mathrm{R}^3\mathrm{S}^2$Route increases route smoothness by 59.68% and improves route stability by 14.29%.

2. Related Work

Last-mile delivery problem is traditionally modeled as a Vehicle Routing Problem (VRP), which aims to determine optimal route to minimize travel cost and service time. Classical solutions include exact optimization methods, heuristics, and metaheuristics. Exact optimization methods are often restricted to small-scale applications due to the NP-hard nature of VRP (Kumar & Panneerselvam, 2012). Heuristics and metaheuristics offer scalable solutions but do not guarantee global optimality (Breedam, 2001).

Reinforcement Learning (RL) has emerged as a powerful paradigm for solving VRP. RL formulates the VRP as a sequential decision-making process, where an agent learns to make optimal routing decisions by interacting with the environment. Early works introduce a pointer network trained via policy gradient methods to construct routes incrementally (Bello et al., 2017; Nazari et al., 2018). Recent advances leverage deep RL techniques, such as Deep Q-Learning (DQN) (Pan & Liu, 2023; Cai et al., 2024) and Proximal Policy Optimization (PPO) (Ara et al., 2023; Foa et al., 2022), to improve scalability and solution quality. Moreover,, Graph Neural Networks (GNN) have been integrated with deep RL to capture spatial and structural relationships in routing graphs, enhancing the representation of customer locations and vehicle paths (Tien & Qi-lee, 2022). While RL-based methods demonstrate adaptability and scalability, they often lack robustness to real-time disruptions and fail to account for courier-centric considerations such as route smoothness and stability.

The emerging field of robust RL provides a promising opportunity to address these limitations. It extends traditional RL by optimizing policies that are resilient to model uncertainties, adversarial perturbations, and environmental shifts (Moos et al., 2022). One foundational approach is Minmax Optimization, where the agent maximizes its reward while accounting for worst-case environmental shifts (Li et al., 2019). This concept extends to Adversarial Training, which perturbs state transitions during learning to harden policies against unexpected changes (Pinto et al., 2017). Additionally, Distributionally Robust Optimization (DRO) aims to maintain policy stability by considering worst-case distributions within uncertainty sets (Lin et al., 2022). This is further refined by Wasserstein robust RL, which leverages the Wasserstein distance to model uncertainty in transition dynamics and rewards (Abdullah et al., 2019). Moreover, Policy Regularization Techniques directly penalize policies that are overly sensitive to environmental changes (Zhang et al., 2021). Our proposed framework leverages robust RL principles by applying a robust policy regularizer to enhance route stability under real-time updates, addressing both environmental unpredictability and courier-centric requirements.

3. Method

In this section, we address the problem of real-time robust and courier-friendly last-mile delivery route optimization in a dynamic environment. We formulate it as a Reinforcement Learning (RL) problem. To solve this problem, we propose a framework that combines a multi-objective reward and a robust policy regularizer to enhance both route smoothness and stability.

3.1. Preliminary: Reinforcement Learning

An RL problem is typically modeled as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $\mathcal{P}(s'|s, a)$ is the transition probability function, describing the likelihood of moving to state s' after taking action a in state s, $\mathcal{R}(s, a)$ is the reward function, specifying the immediate reward received after executing action a in state s, and $\gamma \in (0, 1)$ is the discount factor, balancing immediate and future rewards. At each time step t, the agent observes the current state s_t, selects an action a_t according to its policy $\pi(a_t|s_t)$, receives a reward $r_t = \mathcal{R}(s_t, a_t)$, and transitions to a new state s_{t+1} according to the state transition probability $\mathcal{P}(\cdot|s_t, a_t)$. The policy π is defined as a mapping $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, where $\Delta(\mathcal{A})$ denotes the set of probability distributions over the action space $\mathcal{A}$. The objective of the agent is to learn a policy π that maximizes the expected cumulative discounted reward, i.e. $\max_\pi V^\pi$, where $V^\pi = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$. An optimal policy π^* is defined as $\pi^* = \arg\max_\pi V^\pi$.

3.2. Problem Statement

We consider the problem of real-time robust and courier-friendly last-mile delivery route optimization. The goal is to find a robust and optimal policy that optimizes courier-centric factors—specifically, route smoothness and stability—with traditional efficiency metrics, including total travel distance and time window violation rate. Meanwhile, the policy should be robust to uncertainties caused by demand modifications, time window adjustments, new delivery requests, and cancellations.

We represent the geographic structure of the delivery area as a graph. We divide a day into T equal-length time intervals, defined by discrete time points $t = 0, 1, \ldots, T$, where each interval spans the period $[t, t + 1)$. At the beginning of the day $t = 0$, the courier is assigned an initial set of delivery requests and constructs an initial corresponding delivery route. When the courier completes a delivery, the system updates the demand of the served customer. Meanwhile, the customers may introduce real-time updates into the system, including customer demand changes, time window adjustments, new delivery requests, and cancellations. At each time point $t > 0$, the courier observes the updated set of requests and recalculates the delivery route accordingly.

3.3. RL-based Robust and Courier-Friendly Last-Mile Delivery Problem Formulation

We formulate the Robust and Courier-Friendly Last-Mile Delivery Problem (RCF-LMDP) as a RL problem $G = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. We model the courier as a decision-making agent responsible for calculating delivery routes in a dynamic environment. The definitions of the state, action, state transition, executed route, courier-friendly metrics, reward function, state-robust policy, and goal are introduced as follows.

State s_t: The state at time t is $s_t = \{loc^t_{\text{courier}}, q^t, cus^t\} \in \mathcal{S}$, where loc^t_{courier} denotes the current location of the courier, $q^t \in [0, Q]$ represents the remaining capacity of the delivery vehicle, where Q is the maximum capacity, and $cus^t = \{cus^{1,t}, cus^{2,t}, \ldots, cus^{I,t}\}$ is the set of uncompleted delivery requests, where $i \in [1, I]$ is the index of customers who have not yet been served. Each delivery request $cus^{i,t} \in cus^t$ is represented as $cus^{i,t} = \{loc^i, d^{i,t}, T_s^{i,t}, T_e^{i,t}\}$, where loc^i indicates the customer's location, $d^{i,t}$ is the customer demand, and $[T_s^{i,t}, T_e^{i,t}]$ denotes the time window during which the delivery must be completed.

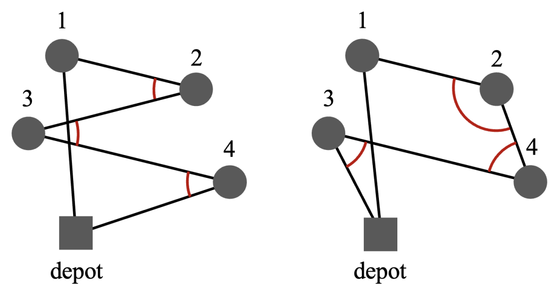
Action a_t: At each time point t, the action a_t corresponds to a delivery route calculated by the courier to serve the current set of uncompleted requests. Specially, the route is defined as $a_t = (n_0, n_1, \ldots, n_I)$, where n_0 denotes the courier's current position, each n_i for $i > 0$ represents a customer location selected from the active request set cus^t, and I is the total number of unserved customers. Unlike classical formulations where the action space is limited to selecting a single next customer, our framework defines the action as a route. This is because generating a route at each decision point allows the model to reason globally about route quality, capturing long-term trade-offs in efficiency, route smoothness, route stability, and time window compliance, rather than making purely myopic decisions.

State Transition $\mathcal{P}$: At time t, the courier executes action a_t by traversing the sequence of nodes and fulfilling customer requests. The MDP state transition is driven by two types of changes: (1) *intrinsic changes* resulting from customer behaviors (e.g., demand modifications, time window adjustments, new delivery requests, or cancellations), and (2) *exogenous changes* caused by the courier's actions, such as completing deliveries and moving between locations. When no intrinsic changes occur, the state transition involves only updates to the courier's location, removal of served requests, and adjustment of the remaining vehicle capacity. Otherwise, customer-driven updates additionally modify the request set, affecting the next state s_{t+1}. Therefore, to avoid unnecessary and redundant computation, we only allow the agent to compute a new action when intrinsic changes occur.

Executed Route $\bar{a}_t$: Due to the dynamic nature of customer-driven intrinsic changes, a new action a_t is recomputed each time such changes occur. However, since such changes may interrupt the execution of the current route, only a prefix of each planned route is typically executed before the route is replaced. The executed route up to time t, denoted by $\bar{a}_t$, is thus defined as the concatenation of the actually traversed segments of all previously planned actions:

$$\bar{a}_t = \text{concat}(\text{prefix}(a_{\tau_1}), \text{prefix}(a_{\tau_2}), \ldots, \text{prefix}(a_{\tau_k}))$$

where $\tau_1, \tau_2, \ldots, \tau_k < t$ are the time steps at which intrinsic changes triggered route replanning, and $\text{prefix}(a_{\tau_i})$ denotes



(a) Low Smoothness: sharp turns and zigzag patterns

(b) High Smoothness: larger turning angles and smoother paths

Figure 1. Illustration of Route Smoothness

the portion of the route that was actually executed before the next change occurred. A special case is that prefix$(a_t) = a_t$ at time t. Therefore, $\bar{a}_T$ is the ultimate executed route.

Courier-Friendly Metrics (1): Route Smoothness. To capture how geometrically intuitive and cognitively manageable a delivery route is for human couriers, we propose to use *route smoothness*. This metric reflects the geometric continuity of a delivery route and is designed to reduce sharp turns and zigzag patterns, which are known to increase navigation complexity and mental fatigue for couriers. Given a route $a = (n_0, n_1, \ldots, n_I)$, we measure the smoothness of the route as the average turning angle:

$$r_s(a) = \frac{1}{I-1}(\sum_{i=1}^{I-1} |\omega_i|),$$

where each angle ω_i is formed by the triplet (n_{i-1}, n_i, n_{i+1}). To illustrate how this metric is computed, consider a set of customers $\{1, 2, 3, 4\}$. Given a route over the customer set, $a^{(a)} = (\text{depot}, 1, 2, 3, 4, \text{depot})$, as shown in Figure 1a, the route smoothness is calculated by summing the turning angles at node 1, 2, 3 and 4. Given an alternative route over the same customer set, $a^{(b)} = (\text{depot}, 1, 2, 4, 3, \text{depot})$, as illustrated in Figure 1b, the corresponding turning angles highlighted in red demonstrate how different route orderings influence smoothness at the same node. As evident from the comparison, smoother routes (Figure 1b) exhibit gentler turning angles, while less smooth routes (Figure 1a) involve abrupt directional changes.

In our formulation, a larger average turning angle indicates a smoother and more intuitive path. Therefore, a higher value of r_s corresponds to a more courier-friendly route, promoting navigational ease and route adherence. First, smooth routes reduce the *cognitive load* on couriers by providing more consistent and predictable directional flow, which is easier to follow using spatial memory or simple visual cues. Second, they require fewer micro-decisions at intersections or forks, minimizing hesitation or navigation errors. Third, smoother paths are more likely to align with the courier's own heuristics or intuitions about "good routes," thereby increasing the likelihood of adherence to algorithm-generated plans. In contrast, routes with excessive zigzagging or sharp turns can feel counterintuitive or mentally taxing, encouraging deviation and reducing overall compliance.

Reward function $\mathcal{R}$: Our goal is to find a delivery route that achieves an effective long-term trade-off among operational efficiency, route quality, and customer satisfaction. To this end, we define the episode-level reward function $\mathcal{R}(s, a)$ as a weighted sum of the route distance r_d, route smoothness r_s and time-window violation rate following the route r_v, i.e.,

$$\mathcal{R}(s, a) = \alpha r_s(\bar{a}_T) - \beta r_d(\bar{a}_T) - \gamma r_v(s, \bar{a}_T),$$

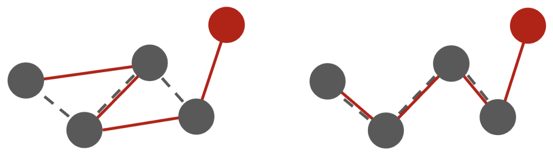
where $s \in \mathcal{S}$ and for any route $a = (n_0, n_1, \ldots, n_I)$,

$$r_d(a) = \sum_{i=0}^{I-1} \text{dist}(n_i, n_{i+1}),$$

dist(n_i, n_{i+1}) is the distance between customers n_i and n_{i+1}.

$$r_v(s, a) = \frac{1}{I} \sum_{i=1}^{I} \mathbb{I}\left[t^i_{\text{arr}} > T^i_e \ \vee \ t^i_{\text{arr}} < T^i_s\right]$$

is the proportion of deliveries violating time window constraints, where t^i_{arr} denotes the arrival time at customer n_i, $[T^i_s, T^i_e]$ is the corresponding valid time window, and the indicator function $\mathbb{I}[\cdot]$ equals 1 if the time window is violated and 0 otherwise. The non-negative coefficients α, β, and γ control the relative importance of route distance, smoothness, and violation rate. By maximizing the total discounted rewards, the agent is encouraged to generate routes that are not only short in distance but also geometrically smooth and temporally feasible, effectively balancing efficiency, courier-friendliness, and service quality. Reinforcement learning provides a flexible and unified framework for multi-objective optimization, where non-linear and non-differentiable criteria can be jointly optimized through reward shaping.



(a) Low Stability (similarity $= \frac{1}{7}$): significant changes after updates

(b) High Stability (similarity $= \frac{3}{7}$): minimal route modifications after updates

Figure 2. Illustration of Route Stability

Courier-Friendly Metric (2): Route Stability. While route smoothness focuses on quantifying zigzag patterns, route stability captures the spatial-temporal consistency of planned paths in dynamic environments. Frequent changes in routing decisions—especially in response to real-time customer updates—can increase couriers' cognitive burden, and reduce adherence to system-generated routes. To promote trust and usability, it is desirable for updated routes to remain as consistent as possible with prior routes.

Formally, given two executed routes $\bar{a}_t = (n_0, n_1, \dots, n_I)$ and $\bar{a}_{t'} = (n'_0, n'_1, \dots, n'_I)$, we define the route stability $stab(a, a')$ as the Jaccard similarity between the directed edge sets:

$$stab(\bar{a}_t, \bar{a}_{t'}) = \frac{|E \cap E'|}{|E \cup E'|},$$

where $E = \{< n_i, n_{i+1} > | \ 0 \le i < I\} \cup \{< n_I, n_0 >\}$, $< n_i, n_{i+1} >$ is an edge connecting n_i and n_{i+1}, and E' is defined analogously for a'. A higher value of $stab(a, a')$ indicates a larger overlap between these two routes a and a', reflecting a more stable and courier-friendly routing decision. In Figure 2, gray nodes represent existing customers, and the red node indicates a newly arriving customer. The route before the newly arriving request is shown with gray dashed lines, while the updated route is shown with red solid lines. As illustrated, more stable routes (Figure 2b) involve minimal deviations from the original plan, whereas less stable routes (Figure 2a) exhibit substantial changes in response to the new request. This metric is particularly relevant in real-time delivery scenarios where customer requests evolve frequently. By preserving route structure as much as possible across updates, stability promotes easier execution and greater system reliability.

State-Robust Policy: A routing policy π is said to be *state-robust* with respect to a perturbation set Δ if for any state $s \in \mathcal{S}$, and any admissible perturbation $\delta \in \Delta$, the routing decision produced by $\pi(s + \delta)$ remains close to $\pi(s)$ in terms of some distance function. That is,

$$\forall s \in \mathcal{S},\ \forall \delta \in \Delta, \quad d\left(\pi(s), \pi(s+\delta)\right) \le \epsilon,$$

where $d(\cdot, \cdot)$ denotes a distance between routes and $\epsilon > 0$ is a robustness tolerance. This definition endows the policy with an intrinsic robustness to state uncertainty: as long as the state perturbations remain within the admissible set Δ, the resulting routing decisions will not deviate significantly from those computed under nominal conditions or accurate state information. For example, the policy will be robust to small shifts in customer time windows, certain fluctuations in predicted demand, or localized cancellations.

Connecting State Robustness and Route Stability: We further observe that this state-robustness definition closely aligns with our goal of achieving high route stability during dynamic re-routing. In both cases, the central objective is to prevent small fluctuations in customer-related information from inducing large, disruptive changes in routing behavior. By learning a policy that reacts smoothly to bounded input uncertainty, we implicitly promote spatial-temporal consistency in delivery routes, which is an essential property for maintaining courier trust and operational efficiency in real-time last-mile logistics. In the context of last-mile delivery, such a state-robust policy induces a form of *stable routing*. Thus, by utilizing state-robust RL training techniques, we expect to obtain a robust and courier-friendly (mainly stable) routing policy.

Goal: The goal of our robust last-mile delivery routing problem is to find an optimal policy π^* that maximizes the expected route-level reward while enforcing state-robustness:

$$\max_{\pi} \mathbb{E}[\mathcal{R}(s_t, a_t) \mid a_t \sim \pi(s_t)] \qquad (1)$$
$$\text{s.t.} \quad \forall s \in \mathcal{S},\ \forall \delta \in \Delta, \quad d\left(\pi(s), \pi(s+\delta)\right) \le \epsilon$$

where Δ and ϵ are predefined perturbation set and robustness threshold. We will introduce our algorithm that solving this optimization problem in the next section.

Algorithm 1 : **R**obust **R**egularizer-enhanced **R**L-based **S**mooth and **S**table routing algorithm (R^{3}S^{2}Route)

```
1:  Input $\psi, \Delta, \eta$. Initialize $\theta, \phi$.
2:  for episode $i = 1, 2, \dots, 30000$ do
3:    Initialize empty executed route: $\bar{a} \leftarrow [\,]$.
4:    Observe $s_0$, sample $a_0 \sim \pi_\theta(\cdot \mid s_0)$.
5:    for time $t = 0, 1, \dots, T-1$ do
6:      if customer-driven updates occur then
7:        Update executed route: $\bar{a} \leftarrow \bar{a} \| \text{prefix}(a_t)$.
8:        Observe $s_{t+1}$, sample $a_{t+1} \sim \pi_\theta(\cdot \mid s_{t+1})$.
9:      else
10:       Continue previous route: $a_{t+1} = a_t$.
11:     end if
12:   end for
13:   Update executed route: $\bar{a} \leftarrow \bar{a} \| \text{prefix}(a_T)$.
14:   Observe reward $\mathcal{R}(s, \bar{a})$.
15:   Update critic: $\phi \leftarrow \phi - \eta \nabla_\phi L^{\text{Critic}}(\phi)$.
16:   Update actor: $\theta \leftarrow \theta + \eta \nabla_\theta L^{\text{Actor}}(\theta)$.
17: end for
18: Output $\theta, \phi$.
```

4. Robust Regularizer-enhanced RL-based Smooth and Stable Routing Algorithm

Traditional robust reinforcement learning often adopts a min-max optimization framework, where the policy is trained to perform well under worst-case realizations of environmental uncertainty. While this formulation offers strong theoretical guarantees, it typically leads to overly conservative policies that may sacrifice average-case performance for robustness. Moreover, the adversarial training required in max-min setups is often unstable and computationally expensive, especially in high-dimensional or real-time decision problems like last-mile delivery. In contrast, incorporating robustness as a regularization term into the objective offers a more flexible and efficient alternative. This approach allows us to balance robustness with other performance metrics (e.g., smoothness, distance) through tunable coefficients. Regularization-based methods are generally easier to implement, more stable during training, and better suited for scenarios where bounded uncertainty is present but full adversarial modeling is either intractable or unnecessary. Additionally, they often lead to policies that are less conservative and more adaptable in practical settings.

Therefore, we propose a **R**obust **R**egularizer-enhanced **R**L-based **S**mooth and **S**table routing algorithm (R^{3}S^{2}Route) to solve problem (1) and train robust policies. We adopt an actor-critic reinforcement learning framework in R^{3}S^{2}Route. The actor, parameterized by θ, represents a stochastic policy $\pi_\theta(a|s)$ that outputs routing decisions given the current state. The critic, parameterized by ϕ, estimates the value function $V_\phi(s)$, which serves as a baseline for advantage estimation during policy updates. Both the actor and critic can be approximated using neural networks. The critic is trained by minimizing the mean-squared error between the predicted value and a Monte Carlo or TD return target:

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

where $\hat{V}_t$ may be computed using a TD target $\hat{V}_t = r_t + \gamma V_\phi(s_{t+1})$, or an n-step return.

Robust Regularized Policy Optimization: Policy learning is performed using the standard Proximal Policy Optimization (PPO) objective, augmented with a robustness regularizer to enhance the policy's robustness to state uncertainty. Specifically, the actor is trained to maximize the following regularized objective:

$$\begin{aligned} L^{\text{Actor}}(\theta) = \mathbb{E}_t \Big[\log \pi_\theta(a_t|s_t)\hat{A}_t - \psi\, \text{KL}[\pi_{\text{old}}(\cdot|s_t), \pi_\theta(\cdot|s_t)]\Big] \\ - \lambda\, \mathbb{E}_{s_t,\delta\sim\Delta}\left[\text{KL}\left(\pi_\theta(\cdot|s_t), \pi_\theta(\cdot|s_t+\delta)\right)\right] \end{aligned}$$

where: $\hat{A}_t = \hat{V}_t - V_\phi(s_t)$ is the advantage function estimated using the critic $V_\phi(s_t)$; δ is a sampled perturbation from the uncertainty set Δ; $\text{KL}(\cdot,\cdot)$ denotes the KL divergence; λ is a hyperparameter controlling the robustness-accuracy trade-off. The robustness regularizer (the last term) penalizes the policy if its output distribution is overly sensitive to small changes in the input state. It effectively enforces a Lipschitz-like continuity in the learned policy, reducing behavioral variance in response to minor environmental shifts. In doing so, the learned policy becomes both robust to input noise and more temporally stable in dynamic environments. The actor and critic networks are updated alternately using stochastic gradient ascent and descent, respectively.

Neural Network Designs: We use a $1D$ convolutional layer to generate continuous customer state embeddings for customer-related features such as locations, delivery demand, and time window. To represent the courier's status, including its position and remaining capacity, we employ an LSTM (Long Short-Term Memory) module that captures temporal dependencies and outputs a memory embedding. These state and memory embeddings are then fused through an attention mechanism, which computes the selection probabilities over all possible next customer nodes. To ensure feasibility, we apply a masking operation that prevents selection of already-served customers. This architecture enables the model to capture both spatial and temporal patterns essential for robust route planning in a dynamic environment.

Pseudo-code Description: The proposed algorithm is shown in Algorithm 1. In line 1, the PPO coefficient ψ, the perturbation set Δ, and the learning rate η are given. The actor network parameters θ and the critic network parameters ϕ are initialized. At time $t = 0$, in lines 3 and 4, we initialize an empty executed route, observe the initial state s_0, and plan an initial route a_0. At each time point t, if customer-driven updates, such as demand changes, time window adjustments, new delivery requests, and cancellations, occur, in line 7, we firstly concatenate the actually traversed segments of current action a_t into the executed route $\bar{a}$. Then we observe the new state s_{t+1} and plan a new route a_{t+1} as shown in line 8. If the only changes in the state are due to the completion of scheduled deliveries, we retain the current route without replanning, as indicated in line 10. Finally, we complete the executed route, and then compute the episode-level reward as shown in lines 13 and 14. Lines 15 and 16 are to update the actor and the critic. This mechanism promotes route stability and reduces unnecessary changes, improving the practicality and interpretability of the resulting delivery plans.

5. Experiments

The objective of our experiments is to validate the following hypotheses for R^{3}S^{2}Route: (1) The integration of a smoothness-aware component into the reward function improves route smoothness. (2) The application of a KL-divergence-based robust policy regularizer enhances route stability. (3) R^{3}S^{2}Route maintains low travel cost and time-window violation rates while significantly improving route smoothness and stability.

5.1. Experimental Setup

We compare our model with the following baselines:

- Tabu Search (TS) (Li & Li, 2020): This is a metaheuristic optimization algorithm that guides local search beyond local optima by using memory-based strategies. It employs a tabu list to avoid cycling and encourages exploration of the solution space, making it effective for solving complex combinatorial problems.
- RL-SVRP (Iklassov et al., 2023): This is a Reinforcement Learning framework for solving vehicle routing problems with stochastic customer demands and travel costs.

Performance is measured using multiple evaluation metrics:

- Distance r_d: This metric measures the total distance traveled by a courier to complete all assigned delivery tasks. A shorter total distance indicates greater operational efficiency.
- Smoothness r_s: This metric quantifies the average angle between route segments. Higher smoothness indicate less sharp turns and zigzag patterns.
- Stability $stab$: This metric calculates the Jaccard similarity between the sets of directed edges in the routes before and after customer-driven updates. Higher stability indicates greater consistency between routes in response to dynamic changes.
- Time-window violation rate r_v: This metric counts the proportion of deliveries violating time-window constraints. A lower time-window violation rate indicates improved adherence to delivery time constraints.

We construct synthetic datasets to emulate dynamic last-mile delivery scenarios within a predefined two-dimensional service area. Customer locations are sampled from a continuous uniform distribution $\mathcal{U}(0,1)^2$, representing their geographic coordinates. Each customer is assigned a service demand drawn from a discrete uniform distribution $\mathcal{U}\{1,2,\ldots,10\}$, and a time window $[T_s^i, T_e^i]$, where $T_s^i \sim \mathcal{U}\{0,1,\ldots,7\}$ and $T_e^i = T_s^i + 1$.

To simulate operational uncertainty and evolving customer behavior, we develop a dynamic delivery simulation environment in TensorFlow. At regular intervals during the simulation, a single customer-driven event is introduced. Each event is sampled from one of the following types: (1) modifying the demand of existing customers, (2) modifying the time window of existing customers, (3) introducing new delivery requests, or (4) canceling existing requests. For demand or time window modifications, 50% of the customers are randomly selected, and their values are resampled using the original distributions. For adding new requests, 10% of the total number of customers are created using the same generation process as in the initial dataset. These are treated as entirely new customers, independent of the existing set. For cancellations, 10% of existing requests are randomly selected and removed by setting their demand to zero and masking them from the routing process. This design enables controlled injection of dynamic events while preserving the stochastic and real-time characteristics of last-mile delivery.

We implement our method and baselines with Tensorflow 1.4.0 in Python 3.6 environment and train it with 503GiB memory and AMD EPYC 9254 24-Core Processor (CPU). Our proposed R^{3}S^{2}Route builds upon the baseline RL-SVRP (Iklassov et al., 2023). We augment it by integrating additional objectives for route smoothness and time-window violation rate, along with a robust policy regularizer. The coefficients of the reward function α, β, and γ for efficiency, route smoothness, and time-window violation rate are set to 10, 3, and 10, respectively. The coefficient for the robust policy regularizer λ is set to 10.

5.2. Overall Performance

As shown in Table 1 and Figure 3, the proposed R^{3}S^{2}Route framework demonstrates superior performance in balancing courier-centric and efficiency-centric objectives compared to TS. Specifically, R^{3}S^{2}Route reduces the total travel distance by 53.67%, increases route smoothness by 31.61%, improves route stability following customer-driven updates by 6.67%, and lowers the time-window violation rate by 3.64%.

Furthermore, when compared to RL-SVRP, R^{3}S^{2}Route achieves higher route smoothness and stability while maintaining a lower time-window violation rate. Specifically, it improves route smoothness by 59.68%, increases route stability following customer-driven updates by 14.29%, and reduces the time-window violation rate by 5.36%. Although R^{3}S^{2}Route incurs a slightly higher travel distance relative to RL-SVRP, this is an expected outcome given its objective to strike a balanced trade-off among multiple performance criteria.

Method	$r_d \downarrow$	$r_s \uparrow$	$stab \uparrow$	$r_v \downarrow$
TS	8.85 ± 0.32	39.36 ± 0.42	0.60 ± 0.10	0.55 ± 0.01
RL-SVRP	$\mathbf{0.65 \pm 0.25}$	32.44 ± 0.68	0.56 ± 0.10	0.56 ± 0.00
R^{3}S^{2}Route	4.10 ± 0.38	$\mathbf{51.80 \pm 1.54}$	$\mathbf{0.64 \pm 0.08}$	$\mathbf{0.53 \pm 0.02}$

Table 1. Overall Performance

5.3. Ablation Study

5.3.1. The Effect of the Smoothness Objective

As illustrated in Table 2 and Figure 4, the inclusion of the smoothness objective in the R^{3}S^{2}Route framework significantly enhances route quality, particularly in terms of route smoothness and stability. Compared to the variant without the smoothness objective (i.e., w/o SM), R^{3}S^{2}Route achieves a 59.53% increase in route smoothness, an 18.52% improvement in route stability following customer-driven updates, and a 5.36% reduction in the time-window violation rate. Although R^{3}S^{2}Route incurs a higher total travel distance than its smoothness-excluded counterpart, this out-

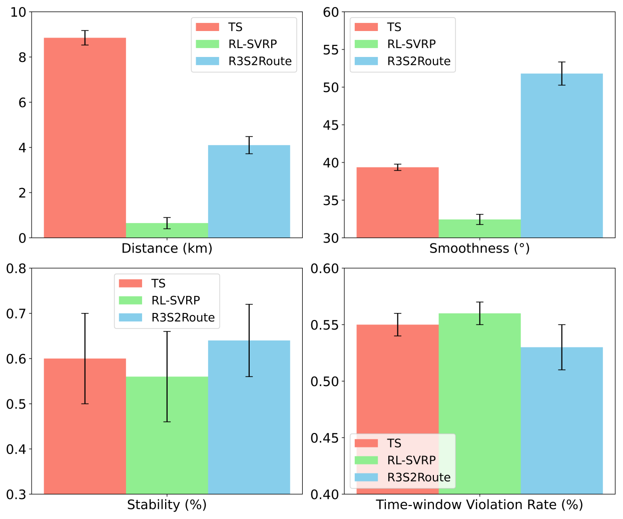


Figure 3. Overall Performance

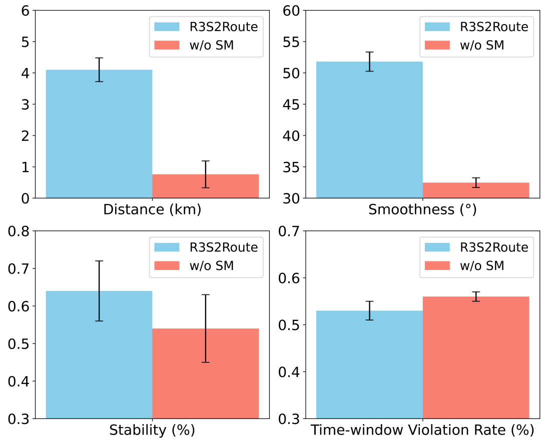


Figure 4. The Effect of the Smoothness Objective

come aligns with the design goal of balancing multiple objectives while prioritizing smoother and more stable routing solutions.

Method	$r_d \downarrow$	$r_s \uparrow$	$stab \uparrow$	$r_v \downarrow$
R^{3}S^{2}Route	4.10 ± 0.38	$\mathbf{51.80 \pm 1.54}$	$\mathbf{0.64 \pm 0.08}$	$\mathbf{0.53 \pm 0.02}$
w/o SM	$\mathbf{0.76 \pm 0.43}$	32.47 ± 0.77	0.54 ± 0.09	0.56 ± 0.01

Table 2. The Effect of the Smoothness Objective

5.3.2. The Effect of the Robust Policy Regularizer

As shown in Table 3 and Figure 5, the robust policy regularizer in R^{3}S^{2}Route plays a crucial role in enhancing the adaptability and consistency of routing decisions in a dynamic environment. Compared to the variant without the robust policy regularizer (w/o RO), R^{3}S^{2}Route achieves a 21.15% reduction in total travel distance and a substantial 60.00% increase in route stability following customer-driven updates. Although the improvement in route smoothness is modest (2.17%), the robust policy regularizer contributes primarily to preserving structural consistency across route updates. The slightly higher time-window violation rate observed in R^{3}S^{2}Route is an anticipated trade-off, reflecting the framework's emphasis on maintaining stable and efficient routes under uncertainty, rather than narrowly optimizing for punctuality alone.

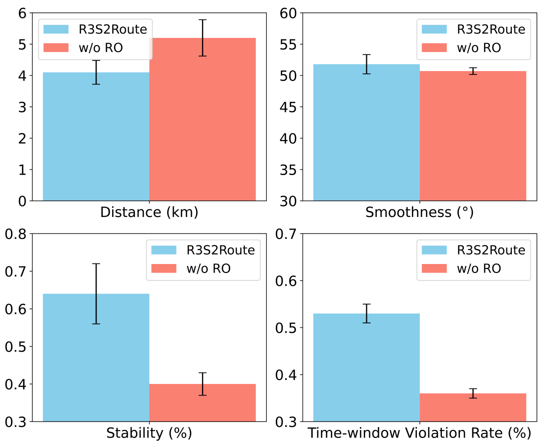


Figure 5. The Effect of the Robust Policy Regularizer

Method	$r_d \downarrow$	$r_s \uparrow$	$stab \uparrow$	$r_v \downarrow$
R^{3}S^{2}Route	$\mathbf{4.10 \pm 0.38}$	$\mathbf{51.80 \pm 1.54}$	$\mathbf{0.64 \pm 0.08}$	0.53 ± 0.02
w/o RO	5.20 ± 0.58	50.70 ± 0.54	0.40 ± 0.03	$\mathbf{0.36 \pm 0.01}$

Table 3. The Effect of the Robust Policy Regularizer

6. Conclusion

In this paper, we propose R^{3}S^{2}Route, a robust reinforcement learning framework for smooth and stable route planning in dynamic last-mile delivery scenarios. By incorporating a smoothness-aware objective and a KL-divergence–based robust policy regularizer into the learning process, R^{3}S^{2}Route optimizes courier-centric factors with traditional efficiency metrics. Our formulation captures the challenges of real-time customer-driven updates and demonstrates how robust policies can improve route consistency and practical deployability. Experimental results show that R^{3}S^{2}Route outperforms baseline methods in both operational efficiency and route quality. This work highlights the importance of integrating robustness and courier-centric design in intelligent transportation systems. Future work will explore multi-agent extensions and real-world deployment with richer operational constraints such as traffic and parcel types.

References

Abdullah, M. A., Ren, H., Ammar, H. B., Milenkovic, V., Luo, R., Zhang, M., and Wang, J. Wasserstein robust reinforcement learning, 2019. URL https://arxiv.org/abs/1907.13196.

Ara, S., Mostafa Kamal Akib, M. M., Shariar Rahman Oion, M., Rahman Shohel, M. H., Noman Faysal Ridoy, M., Kabita, F. A., and Shahiduzzaman, M. Vehicle routing problem solving using reinforcement learning. In *2023 26th International Conference on Computer and Information Technology (ICCIT)*, pp. 1–6, 2023. doi: 10.1109/ICCIT60459.2023.10441644.

Bello, I., Pham, H., Le, Q. V., Norouzi, M., and Bengio, S. Neural combinatorial optimization with reinforcement learning, 2017. URL https://arxiv.org/abs/1611.09940.

Boysen, N., Fedtke, S., and Schwerdfeger, S. Last-mile delivery concepts: a survey from an operational research perspective. *OR Spectrum*, 43(1):1–58, Mar 2021. ISSN 1436-6304. doi: 10.1007/s00291-020-00607-8. URL https://doi.org/10.1007/s00291-020-00607-8.

Breedam, A. V. Comparing descent heuristics and metaheuristics for the vehicle routing problem. *Computers Operations Research*, 28 (4):289–315, 2001. ISSN 0305-0548. doi: https://doi.org/10.1016/S0305-0548(99)00101-X. URL https://www.sciencedirect.com/science/article/pii/S030505489900101X.

Cai, J., Zhang, X., Lin, Q., Dong, L., Chen, W., and Ming, Z. Deep reinforcement learning for solving the vehicle routing problem in practical logistics. In *2024 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, 2024. doi: 10.1109/CEC60901.2024.10612190.

Foa, S., Coppola, C., Grani, G., and Palagi, L. Solving the vehicle routing problem with deep reinforcement learning, 2022. URL https://arxiv.org/abs/2208.00202.

Gauvin, C., Desaulniers, G., and Gendreau, M. A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands. *Computers Operations Research*, 50:141–153, 2014. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2014.03.028. URL https://www.sciencedirect.com/science/article/pii/S0305054814000835.

Iklassov, Z., Sobirov, I., Solozabal, R., and Takac, M. Reinforcement learning for solving stochastic vehicle routing problem, 2023. URL https://arxiv.org/abs/2311.07708.

Kumar, S. N. and Panneerselvam, R. A survey on the vehicle routing problem and its variants. *Intelligent Information Management*, 04No.03:9, 2012.

Li, G. and Li, J. An improved tabu search algorithm for the stochastic vehicle routing problem with soft time windows. *IEEE Access*, 8:158115–158124, 2020. doi: 10.1109/ACCESS.2020.3020093.

Li, S., Wu, Y., Cui, X., Dong, H., Fang, F., and Russell, S. Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33 (01):4213–4220, Jul. 2019. doi: 10.1609/aaai.v33i01.33014213. URL https://ojs.aaai.org/index.php/AAAI/article/view/4327.

Lin, F., Fang, X., and Gao, Z. Distributionally robust optimization: A review on theory and applications, 2022. ISSN 2155-3289. URL https://www.aimsciences.org/article/id/5a228ac4-f49b-4499-aab6-0656d63eb577.

Lysgaard, J., Letchford, A. N., and Eglese, R. W. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2): 423–445, Jun 2004. ISSN 1436-4646. doi: 10.1007/s10107-003-0481-8. URL https://doi.org/10.1007/s10107-003-0481-8.

Macioszek, E. First and last mile delivery – problems and issues. In Sierpiński, G. (ed.), *Advanced Solutions of Transport Systems for Growing Mobility*, pp. 147–154, Cham, 2018. Springer International Publishing. ISBN 978-3-319-62316-0.

Matl, P., Hartl, R. F., and Vidal, T. Workload equity in vehicle routing: The impact of alternative workload resources, 2018. URL https://arxiv.org/abs/1803.01795.

Moos, J., Hansel, K., Abdulsamad, H., Stark, S., Clever, D., and Peters, J. Robust reinforcement learning: A review of foundations and recent advances. *Machine Learning and Knowledge Extraction*, 4(1):276–315, 2022. ISSN 2504-4990. doi: 10.3390/make4010013. URL https://www.mdpi.com/2504-4990/4/1/13.

Nazari, M., Oroojlooy, A., Snyder, L. V., and Takáč, M. Reinforcement learning for solving the vehicle routing problem, 2018. URL https://arxiv.org/abs/1802.04240.

Pan, W. and Liu, S. Q. Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. *Applied Intelligence*, 53(1):405–422, Jan 2023. ISSN 1573-7497. doi: 10.1007/s10489-022-03456-w. URL https://doi.org/10.1007/s10489-022-03456-w.

Pinto, L., Davidson, J., Sukthankar, R., and Gupta, A. Robust adversarial reinforcement learning. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2817–2826. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/pinto17a.html.

Pisinger, D. and Ropke, S. A general heuristic for vehicle routing problems. *Computers Operations Research*, 34(8):2403–2435, 2007. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2005.09.012. URL https://www.sciencedirect.com/science/article/pii/S0305054805003023.

Tien, Z. C. and Qi-lee, J. Enhancing vehicle routing problem solutions through deep reinforcement learning and graph neural networks. *International Journal of Enterprise Modelling*, 16(3):125–135, Sep. 2022. doi: 10.35335/emod.v16i3.64. URL https://ieia.ristek.or.id/index.php/ieia/article/view/64.

Wanganoo, L. and Patil, A. Preparing for the smart cities: Iot enabled last-mile delivery. In *2020 Advances in Science and Engineering Technology International Conferences (ASET)*, pp. 1–6, 2020. doi: 10.1109/ASET48392.2020.9118197.

Zhang, H., Chen, H., Xiao, C., Li, B., Liu, M., Boning, D., and Hsieh, C.-J. Robust deep reinforcement learning against adversarial perturbations on state observations, 2021. URL https://arxiv.org/abs/2003.08938.