DOI: 10.1002/net.70024

ARTICLE TYPE

Partial-outsourcing strategy for the vehicle routing problem with stochastic demands

Lin Zhu[1] | **Yossiri Adulyasak**[2] | **Louis-Martin Rousseau**[3]

[1]Logistics Research Center, Shanghai Maritime University, Shanghai, China

[2]GERAD and Department of Logistics and Operations Management, HEC Montréal, Quebec, Canada

[3]CIRRELT and Department of Mathematical and Industrial Engineering, Polytechnique Montréal, Quebec, Canada

Correspondence
Corresponding author Lin Zhu.
Email: zhulin.sife@gmail.com

Abstract

This paper studies a combined delivery strategy involving a private vehicle and external carriers under stochastic customer demands. The routing problem focuses on a single private vehicle, while external carriers are allowed to determine their own routes independently and are compensated with a fixed price per unit demand served. A strategy incorporating routing re-optimization is proposed, along with a new recourse mechanism that leverages outsourcing through external carriers. To enable routing re-optimization, a novel approximate linear programming (ALP) approach is introduced. This offers a new pathway for addressing vehicle routing problems under stochastic demand considerations. The ALP approach is adapted to the specific structure of routing under stochastic demands, leading to the development of a decomposition-based ALP solution framework. This adaptation arises from changes in the decision sequence of routing and restocking at each step of the Markov decision process (MDP), which differs from previous formulations of vehicle routing under stochastic demands. Additionally, further adaptations are made to facilitate the computation of the proposed strategy by exploring the relationships among variables and constraints specific to the problem context, as well as by developing a constraint sampling procedure designed to mimic the near-optimal heuristic policy. Our numerical results show that the proposed outsourcing-based policy yields notable operating-cost savings, with an average improvement of 4.06% over the traditional recourse strategy in midpoint-depot instances. Moreover, in small instances where the optimal policy within the traditional partial re-optimization framework can be computed, the proposed price-directed policy still provides cost advantages over this re-optimization scheme, demonstrating the value of our ALP-based framework.

KEYWORDS
private fleet and common carrier, crowd-shipping, stochastic vehicle routing, re-optimization, Markov decision process, approximate linear programming, outsourcing

1 | INTRODUCTION

With the rapid growth of e-commerce, companies are experiencing a surge in order volumes and face increasing pressure to meet customer demands in a timely and cost-effective manner. However, narrow profit margins make it impractical to improve logistics efficiency solely by expanding company-owned delivery capacity. To ensure timely fulfillment, many companies supplement their in-house fleets with external, or social, capacity—either occasionally or on a dedicated basis. Leveraging external carriers helps reduce the need to maintain a large private fleet, particularly in the face of fluctuating demand. At the same time, retaining a core private fleet remains essential to ensure delivery quality and reliability. This has led to a hybrid delivery model that combines the advantages of company-owned vehicles and external carriers. This model is especially relevant for groceries, electronics, pharmaceutical products, and meal deliveries [30]. In some cases, external carriers take on the primary delivery role, with the private fleet providing backup support [e.g., 6, 51]. In others, the private fleet leads, while external carriers assist with last-mile deliveries [e.g., 5, 12]. This paper focuses on the latter scenario, where external carriers supplement the private fleet.

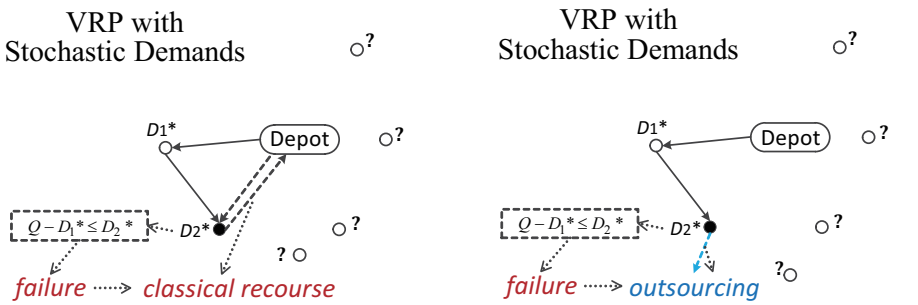
This delivery service undoubtedly faces various sources of uncertainty. Such uncertainty may arise from factors such as customer demand, customer presence, travel time, or service duration [42]. In this paper, we focus on the uncertainty associated with customer demand. A relevant example can be found in the meal delivery setting. For instance, a restaurant (acting as a depot) may serve nearby stalls (customers). These stalls function as fulfillment centers for continuously arriving orders. The company-owned vehicle is dispatched from the restaurant to deliver meals to the stalls. Consequently, the quantity delivered by the vehicle depends on the cumulative demand that has materialized by the time of its arrival. Meanwhile, the vehicle may fail to fulfill the observed demand upon its first visit, resulting in partial unmet demand. In such cases, it may need to return to the depot for replenishment or resort to using an outsourced (or crowd-sourced) vehicle, rather than making an additional round trip with its own vehicle before continuing service. In this paper, the dynamism of customer demand is not addressed. Instead, it assumes that deliveries are divided into discrete periods. The vehicle satisfies the demand observed at the time of its first arrival, while any demand that arrives afterward is deferred to the next delivery period. In this way, the dynamic nature of customer demand is managed.

This problem description corresponds to the situation in the vehicle routing problem with stochastic demands (VRPSD), where customer demands are uncertain and are only revealed upon the vehicle's arrival. In such cases, vehicles may encounter routing failures due to insufficient capacity. To address these disruptions, recourse strategies are required to restore route feasibility [24]. Additionally, it can be advantageous to introduce external carriers to support the company-owned vehicles, particularly when customer demand exhibits tidal characteristics. During peak hours, the company-owned vehicle may lack sufficient capacity to serve all customers, and frequent replenishment trips can be inefficient due to stringent delivery time constraints. If external carriers are available near the depot, a portion of the demand can be allocated to them to ensure timely service.

Resorting to external carriers to fulfill partial demands can enhance logistics efficiency, albeit at the expense of additional costs. External carriers are typically compensated for their services using various pricing schemes, including a fixed cost per customer [e.g., 10], payments based on the distance traveled [e.g., 5], and fixed payments per unit of demand served [e.g., 7, 12]. This paper adopts the last scheme, where a fixed price is paid for each unit of demand fulfilled by the external carrier. This compensation scheme is commonly used in practice, as evidenced by platforms like Amazon Flex [51] and is also analyzed in [7]. This pricing mechanism is particularly suitable for urban delivery settings, where couriers frequently manage multiple orders simultaneously within compact geographic areas. Given that they often traverse congested zones before or after completing assigned tasks, calculating fair compensation based on distance traveled becomes impractical. Such a fixed-rate pricing model is especially prevalent on Chinese e-commerce platforms such as Meituan [35] and Ele.me [18].

This paper studies the routing problem for a company-owned vehicle under stochastic customer demands, with support from other (external) carriers. The company-owned vehicle is required to dynamically update its route based on its current location, residual capacity, and the set of unvisited customers—in other words, to perform routing re-optimization. This study can fall within the scope of the vehicle routing problem with stochastic demands (VRPSD). As highlighted in previous studies [e.g., 24], implementing routing re-optimization is challenging, even in the single-vehicle case. Relevant research remains scarce. Therefore, this paper begins by focusing on the case of a single company-owned (private) vehicle. Additionally, it is assumed that other carriers are responsible for their own routing costs. The company does not concern itself with the actual routes taken by the carriers, as long as the assigned tasks are completed within the specified time limits. The other carriers are compensated based on the number of units assigned, assuming a fixed price per unit. Furthermore, it is assumed that an adequate number of external carriers are positioned around the depot, ensuring efficient responses to assignments—a pattern frequently observed in Chinese e-commerce platforms.

Given the problem described above, we propose a recourse strategy called the *partial-outsourcing strategy*. This strategy involves routing a private vehicle while outsourcing a portion of customer demand to other carriers. Specifically, the vehicle departs from the depot and ultimately returns, following a route determined by the vehicle's current state at each stage. At each decision stage, the vehicle selects the next customer to serve and decides whether to perform preventive restocking. Upon arrival, it fulfills the customer's demand within its available capacity. If the vehicle lacks sufficient capacity, the unmet demand is outsourced to other carriers at a penalty cost, which is charged at a fixed price per unit of demand. This strategy aims to minimize the total cost, which consists of the vehicle's travel expenses and the penalty cost incurred by outsourcing to other carriers. Our partial-outsourcing strategy differs from traditional recourse strategies for the VRPSD, as illustrated in Figure 1. In traditional recourse strategies, when a failure occurs, the vehicle performs a replenishment trip to the depot. The unmet demand is then fulfilled by the vehicle upon its return, which typically refers to the detour-to-depot (DTD) operation scheme, as shown in Fig. 1(a). In contrast, when a failure occurs, our strategy does not require the vehicle to take a detour to the depot solely for the purpose of recovering route feasibility. Instead, the unmet demand is outsourced to other carriers, as shown in Fig. 1(b).



(a) classical recourse action (b) recourse action in partial-outsourcing strategy

FIGURE 1 Comparison of recourse actions in partial-outsourcing to traditional strategies

1Q is the vehicle capacity. D_i^* $(i = 1, 2, \ldots)$ denotes the observed demand of customer i at the vehicle's arrival. "?" represents the unknown demands of customers that are yet to be visited and observed.

To formulate the strategy, a Markov decision process (MDP) model is developed. In this model, the sequence of routing and restocking decisions is altered, which enables the identification of a decomposition scheme for the value functions. Based on this, a decomposition-based value function approximation approach is proposed. This approximation framework leverages the problem's inherent structure, thereby enhancing the effectiveness of the solution methodology. Given that solving the MDP inevitably encounters the curse of dimensionality, an approximate linear programming (ALP) method is adapted to address this challenge. While most previous works on the VRPSD with re-optimization have adopted approximate dynamic programming (ADP) methods [e.g., 38, 48], our approach is based on the ALP, with the aim of opening a new pathway toward solving the problem more effectively and potentially moving closer to optimality.

The ALP method begins by translating the MDP formulation into a linear programming (LP) model, where the decision variables represent value functions. To enhance tractability, these value functions are approximated using affine functions, resulting in a formulation known as approximate linear program (ALP). This approach can yield bounds on the value functions, which are then incorporated into the traditional Bellman recursion to approximate the value functions and derive a price-directed policy [e.g., 16, 50]. Our ALP method is further developed by identifying and leveraging structural properties of the value functions through decomposition. To reduce computational complexity, interdependencies among the value functions are analyzed, and the resulting insights are utilized to reduce the number of variables in the ALP formulation, while limiting the number of affine functions used for approximation. Moreover, because the ALP formulation involves intractable constraints, a constraint sampling procedure is introduced to enhance tractability and enable efficient solution of the ALP model.

As illustrated in works related to the ALP approach [e.g., 41], two essential elements in implementing an ALP approach are: (i) the choice of basis functions, and (ii) the state-relevance distribution, which determines the importance of states and associated actions. Both are critical in minimizing the approximation error of the value function. In this paper, these two aspects are addressed in a standard manner [41]. The basis functions are fixed and established based on domain knowledge. The state-relevance distribution, which guides the selection of critical constraints in constraint sampling, is specified using the states visited under a baseline policy. In our paper, the baseline policy is mimicked by a set of selected heuristic policies. Our paper follows the conventional protocol for these components, which may impose limitations on the effectiveness of the ALP method in solving the problem. Nevertheless, the potential of ALP is evident in the experimental results. In fact, more effective handling of these two aspects can lead to significantly improved performance. Recent work in the MS/OR (management science/operations research) community has shown increasing interest in advancing ALP methods, particularly through the integration of machine learning techniques to improve basis selection and state-relevance distributions [e.g., 32, 41], as well as other algorithmic enhancements [e.g., 36, 54, 37]. With these developments, a viable and promising solution method for solving routing problems near-optimally appears to be on the horizon.

To demonstrate the effectiveness of the proposed method, computational studies are conducted from multiple perspectives: (i) against a traditional recourse strategy without outsourcing, (ii) in comparison with several high-quality benchmark approaches, (iii) under demand distributions with different degrees of variability, and (iv) by varying the outsourcing price. These comparisons offer a comprehensive evaluation of the proposed approach from different angles in addressing the problem at hand. For comparison (i), when evaluated against the traditional recourse strategy, the proposed approach demonstrates a clear cost-saving

advantage, likely attributable to more context-sensitive restocking decisions and driven by a restructured decision sequence. In comparison (ii), the results indicate that the proposed method has the potential to surpass the other methods, while keeping computational time manageable and comparable to that of the most time-consuming heuristic. For comparison (iii), experiments using demand distributions with low and high levels of variability yield results generally consistent with the findings of [20]. This suggests that the effectiveness of the proposed method is largely unaffected by demand variability. The observed differences are minor, indicating that either type of demand distribution can be appropriately used to evaluate the method's performance. Lastly, for comparison (iv), varying the outsourcing prices within the suggested ranges reveals corresponding trends across different price settings for midpoint and corner depot scenarios. The variations in these trends further suggest that a dedicated study focusing on the pricing problem for the problem at hand should be pursued in the future.

The primary contributions of this paper are summarized as follows.

- A vehicle routing problem under uncertain demands is addressed, where a private vehicle is assisted by external carriers. A partial-outsourcing strategy is proposed to address this problem, incorporating a recourse action that leverages outsourced delivery services.
- To enable the computation of this strategy, an ALP approach is introduced and specifically tailored to the structural characteristics of the routing problem under stochastic demands, leading to the development of a decomposition-based ALP solution framework.
- Additional adaptations are made to enhance computational efficiency, including the exploitation of relationships among problem-specific variables and constraints, as well as the design of a constraint sampling procedure.
- Finally, comprehensive numerical experiments are conducted, with the results analyzed to draw conclusions and provide insights into the performance and effectiveness of the proposed method.

The remainder of the paper is organized as follows. Section 2 reviews relevant literature. Section 3 presents the partial-outsourcing strategy, formulated using an MDP formulation. Section 4 introduces the ALP solution framework. By exploring the problem structure, a decomposition-based solution framework is developed. Section 5 specifies the approximation for the cost of recourse actions. In Section 6, a price-directed policy is derived based on the approximation of value functions, in which routing and restocking decisions are elaborated given observed routing states. Section 7 discusses the experimental results.

2 | LITERATURE REVIEW

This paper investigates the routing problem for a private vehicle assisted by external carriers in the context of stochastic customer demands. The study is closely related to combined delivery systems that integrate external carriers together with a private fleet, as well as to VRPSD. Given the focus on route design, the literature review on combined delivery systems is further narrowed to studies that address routing aspects. Furthermore, since the problem is solved using ALP, the literature on ALP methods is also reviewed, along with the relevant solution techniques employed in this study.

Combined delivery of private and crowd vehicles

Recent studies have shown a growing research interest in vehicle routing problems involving the combined use of private and crowd-sourced vehicles for delivery, driven by a logistics paradigm shift under the sharing economy. This line of research can generally be classified into two categories based on which type of vehicle plays the leading role. In the first category, private vehicles serve as the primary delivery agents, with external carriers providing supplementary support [e.g., 5, 14, 12, 15, 7]. In the second category, external carriers take the primary responsibility, while private vehicles act as backups [e.g., 23, 6, 34, 51]. Our research falls into the former category. In this category, Archetti et al. [5] were among the first to formalize the problem by extending the classical vehicle routing problem to incorporate in-store customers as crowd-shippers. They examined a deterministic routing setting in which each crowd-shipper was assigned to deliver to a single customer and was compensated according to the distance deviated from their original route. Dahle et al. [14] extended the model by allowing a common carrier to handle multiple delivery tasks and designing vehicle routes for each vehicle under time window constraints. They considered a deterministic routing problem. Dayarian and Savelsbergh [15] considered stochastic information regarding the arrival of in-store customers and online orders. They studied the routing problem on the same-day delivery setting where in-store customers were treated as potential crowd-shippers. Dabia et al. [12] and Baller et al. [7] studied the vehicle routing problem with private

fleet and common carriers, and proposed exact algorithms to solve it. Both studies were conducted under the deterministic settings. In this category of research, most studies focus on deterministic settings, with only Dayarian and Savelsbergh [15] addressing the stochastic nature of crowd-shippers' and customer orders' arrivals.

In addition to the work by Dayarian and Savelsbergh [15], other research also addresses uncertainty settings, though relevant studies remain rare. Torres et al. [51] and Dahle et al. [13] examined the stochasticity of crowd-shippers' availability. Gdowska et al. [23] incorporated the probability that a crowd-shipper may reject an assigned delivery task. Overall, existing works focus on uncertainty related to the supply of crowd vehicles and the arrival of customer orders, while uncertainty arising from customer demand has not been addressed. Moreover, this paper considers the routing problem only for private vehicles, assuming that crowd-sourced vehicles operate their routes independently. The setting was also adopted in [7], where crowd carriers were compensated based on a fixed price per unit of demand.

Recourse strategies and re-optimization approaches

The problem investigated in this paper falls within the scope of the VRPSD. To address demand uncertainty, routing re-optimization is considered, and a recourse strategy incorporating outsourcing is developed. Accordingly, the literature review includes existing research on recourse actions for the VRPSD, as well as approaches for implementing routing re-optimization. For a general overview of the VRPSD, the works by Gendreau et al. [24] and Florio et al. [21] can be referred to.

Various recourse strategies have been developed since the introduction of the classical DTD operating scheme. Under the DTD scheme, the vehicle returns to the depot for replenishment if a stockout occurs at a customer location. Most early recourse strategies were based on this approach [e.g., 29]. Another widely studied recourse action is preventive restocking, where the vehicle may proactively return to the depot before its inventory is depleted [e.g., 53, 33]. Besides, Novoa et al. [39] introduced an extended recourse strategy by disallowing partial deliveries and proposed two alternative recourse actions. In recent years, several new recourse strategies have emerged. Salavati-Khoshghalb et al. [43] presented a rule-based recourse policy, where a preventive restocking trip is triggered when the remaining vehicle capacity falls below a predefined customer-specific threshold. In a follow-up study, Salavati-Khoshghalb et al. [44] developed a hybrid policy that quantifies the risk of failure using a distance-based measure. More recently, Florio et al. [20] proposed the switch policy, which incorporates preventive restocking and allows the swapping of visiting orders between two adjacent customers. Existing recourse strategies have not considered outsourcing as a potential recourse action, possibly because most were developed prior to the rise of the sharing economy.

Several studies have addressed the VRPSD with routing re-optimization, primarily through approaches based on the ADP. Secomandi [45] proposed a neuro-dynamic programming algorithm that approximates the value functions of system states using linear combinations of pre-selected features. Later, Secomandi [46] introduced a one-step rollout algorithm that incrementally improves a base routing sequence. After that, Novoa and Storer [38] enhanced this approach by developing a two-step rollout algorithm. Then, Secomandi and Margot [48] proposed a partial re-optimization method, in which, given an a priori route, re-optimization is applied within customer blocks along the predetermined route. In the context of multi-vehicle routing, Goodson et al. [25, 26] studied the VRPSD with duration limits and introduced rollout-based policies. Zhu et al. [56] developed a paired cooperative re-optimization method, where customer assignments between two vehicles are dynamically updated, and routing is adjusted using the partial re-optimization procedure. More recently, Ulmer et al. [52] considered a variant of the VRP with stochastic customer requests and proposed an offline-online ADP framework to address it. In summary, the ALP method has not yet been applied to solve the VRPSD.

Approximate linear programming and related solution techniques

ALP methods have been employed in various problem contexts, including inventory routing [e.g., 1, 2], revenue management [e.g., 3, 31, 49], scheduling problems [e.g., 4, 8], knapsack problems [e.g., 9], and the traveling salesman problems (TSP) [e.g., 50]. However, ALP has not yet been adapted to address variants of the vehicle routing problem (VRP). Existing works that apply an ALP approach to routing problems are limited and primarily focus on the TSP, where capacity constraints or delivery demand requirements are not considered [e.g., 50, 22]. Since the use of affine functions to approximate value functions is a standard procedure in the ALP approach, and the lower bounds obtained by solving an approximate linear programming formulation are theoretically guaranteed, we do not elaborate on these steps here. Detailed discussions can be found in references such as [16] and [50].

Our solution framework differs from the standard ALP approach primarily through the development of a decomposition-based method. This decomposition leverages the problem-specific structure of the routing value function. A key novelty lies in the reordering of the decision sequence—specifically, altering the order of routing and restocking decisions in the VRPSD.

To the best of our knowledge, this change in decision order has not been previously explored [e.g., 48, 38]. Experimental results also demonstrate the benefits of this modified sequence. Further, the development of the decomposition-based value function approximation enables the introduction of tailored solution techniques. These include analyzing the interrelationships among variables and constraints to uncover opportunities for computational simplification. Notably, such techniques are highly problem-specific. As the ALP approach has not been adapted to the VRPs, the solution techniques developed here are specifically designed to address the unique challenges of this problem.

To tackle the unmanageable constraints in an ALP formulation, various approaches have been proposed, most notably constraint generation [e.g., 2, 22] and constraint sampling [e.g., 17, 19]. The development of techniques for addressing intractable constraints in the ALP framework has also been discussed in recent research [e.g., 36, 32]. In this paper, we adopt the constraint sampling approach, building on the idea of [2]. To implement this, a multi-sampling framework that mimics the baseline policy for constraint sampling is proposed. Since the constraint-handling procedure in the ALP approach is considered only a minor component of the overall solution methodology—and is not the primary focus of this work—a standard implementation is adopted for this step. Nevertheless, as highlighted in recent literature [e.g., 41], it has been suggested that a more in-depth investigation into this procedure could substantially enhance the performance of ALP methods. Accordingly, this aspect is identified as a promising direction for future algorithmic improvements.

3 | PARTIAL-OUTSOURCING STRATEGY

In this section, the partial-outsourcing strategy is formulated using MDP. The notation and assumptions for the routing problem are introduced, and then the value functions and optimal actions under the strategy with outsourcing are defined. In the end, the difference between our formulation and those used in previous works is clarified, and the optimality equation in our formulation is generalized.

Notation and assumptions

The notation used in this paper is generally in line with other VRPSD literature [e.g., 45, 38, 48]. The strategy can be formulated as a finite-horizon discrete-time Markov decision process. Considering a complete network, customers are denoted by node set $\mathcal{N} = \{1, \ldots, N\}$, and 0 denotes the depot. A vehicle with capacity Q ($Q \in \mathbb{N}^+$) is dispatched from a depot to visit customers, satisfies their demands and eventually returns to the depot. Distance d_{lj} between any two nodes l and j (l, $j \in \mathcal{N} \cup 0$), computed by Euclidean distance, is assumed to be known, symmetric, and satisfy the triangle inequality: $d_{lj} \leq d_{li} + d_{ij}$, with i an additional node. Demand quantity for customer l ($l \in \mathcal{N}$), $\tilde{\xi}_l$, is a random variable characterized by a probability distribution $p_l(e) = \Pr(\tilde{\xi}_l = e)$ ($e = 0, 1, \ldots, E \leq Q$) and $p_l(e) = 0$ ($e = E+1, \ldots, Q$), where E is a nonnegative integer. Customer demand $\tilde{\xi}_l$ is independent of the vehicle routing/replenishment policy, and its realization ξ_l can only be observed when the vehicle arrives at the customer. The total depot capacity is assumed to be at least $N \cdot E$, so all customer demands can be fully satisfied. A summary of notation is provided in Appendix C.

In our formulation, split deliveries are allowed. When a failure occurs at customer l, the vehicle delivers its existing load q ($q < \xi_l$) to the customer, and the remaining unmet demand is outsourced with an expense of $b \cdot (\xi_l - q)$, where b is the unit price for outsourcing.

Value functions

The strategy is formulated as an MDP with stages in set $\Omega = \{N, N-1, \ldots, 0\}$, with stage $k \in \Omega$ corresponding to the number of unvisited customers. Each stage $k \in \Omega \backslash \{N\}$ starts when the vehicle finishes serving the current customer. The corresponding state is denoted by $s_k = (l, q, \mathcal{R}_k(l))$, representing the vehicle departing from current location l ($l \in \mathcal{N}$) with available capacity q ($q \in \mathcal{Q} = \{0, 1, \ldots, Q\}$) and set of remaining unvisited customers $\mathcal{R}_k(l)$ ($\mathcal{R}_k(l) \subseteq \mathcal{N}$). Ψ denotes the state space for the process, and it is composed of

$$\Psi = \left\{ s_N = (0, Q, \mathcal{N}) \right\} \cup \left\{ s_k = (l, q, \mathcal{R}_k(l)) \,\middle|\, k \in \Omega \backslash \{N\},\ l \in \mathcal{N},\ q \in \mathcal{Q},\ \mathcal{R}_k(l) \subset \mathcal{N} \right\}. \tag{1}$$

For state $s_k = (l, q, \mathcal{R}_k(l))$ at stage $k \in \Omega \backslash \{N, 0\}$, two decisions must be made. First, it must be decided which customer $j \in \mathcal{R}_k(l)$ to visit next. Second, it must be decided whether the vehicle will go directly to that customer (a case labeled $D(j)$) or return to the depot to restock before proceeding to that customer (a case labeled $R(j)$). At the beginning stage N, the only available decision is which customer to visit first. For the final stage 0, the only available action is to return to the depot without replenishing.

Let $V_k(l,q,\mathcal{R}_k(l))$ denote the optimal expected cost-to-go from state $s_k=(l,q,\mathcal{R}_k(l))\in\Psi$. The cost-to-go values at the final stage are

$$V_0(l,q,\emptyset)=d_{l0},\qquad \forall l\in\mathcal{N},\ q\in\mathcal{Q}. \tag{2}$$

For state $s_k=(l,q,\mathcal{R}_k(l))$ at stage $k\in\Omega\backslash\{N,0\}$, the optimal policy satisfies the following Bellman equations,

$$V_k(l,q,\mathcal{R}_k(l))=\min_{j\in\mathcal{R}_k(l)}\left\{\min\left\{V_k^{D(j)}(l,q,\mathcal{R}_k(l)),V_k^{R(j)}(l,q,\mathcal{R}_k(l))\right\}\right\},\quad \forall s_k\in\Psi. \tag{3}$$

where $V_k^{D(j)}(l,q,\mathcal{R}_k(l))$ and $V_k^{R(j)}(l,q,\mathcal{R}_k(l))$ are the cost-to-go values associated with stage k and state $(l,q,\mathcal{R}_k(l))$, corresponding to visiting next customer j directly and by first replenishing at the depot, respectively. $\min\{V_k^{D(j)}(l,q,\mathcal{R}_k(l)),V_k^{R(j)}(l,q,\mathcal{R}_k(l))\}$ ensures customer j ($j\in\mathcal{R}_k(l)$) is reached in the most efficient way. The optimal next customer j^* ($j^*\in\mathcal{R}_k(l)$) corresponds to the one with the minimum cost-to-go value. The cost-to-go values for the two cases can be written as follows.

$$\begin{aligned}
V_k^{D(j)}(l,q,\mathcal{R}_k(l)) &= d_{lj}+B_j(q)+\sum_{e\le q}p_j(e)\cdot V_{k-1}(j,q-e,\mathcal{R}_{k-1}(j;l))+V_{k-1}(j,0,\mathcal{R}_{k-1}(j;l))\cdot\sum_{e>q}p_j(e),\\
V_k^{R(j)}(l,q,\mathcal{R}_k(l)) &= d_{l0}+d_{0j}+\sum_{e}p_j(e)\cdot V_{k-1}(j,Q-e,\mathcal{R}_{k-1}(j;l)),\qquad \forall j\in\mathcal{R}_k(l),\ s_k\in\Psi,
\end{aligned} \tag{4}$$

where $\mathcal{R}_{k-1}(j;l)=\mathcal{R}_k(l)\backslash\{j\}$, and $B_j(q)=b\cdot\sum_{e>q}p_j(e)\cdot(e-q)$ calculates the expected outsourcing cost if residual capacity q is not sufficient to meet customer j's demand. $B_j(q)$ equals to 0 when $q\ge E$. The vehicle may encounter two situations when visiting customer j directly. When residual capacity q is sufficient to satisfy the demand of customer j (i.e., $e\le q$), the residual capacity is updated after completing the service of customer j, and the remaining capacity equals $q-e$. Otherwise, when the demand exceeds the residual capacity (i.e., $e>q$), the vehicle depletes its inventory and leaves unmet demand $e-q$ to be outsourced.

At beginning stage N, for unique starting state $s_N=(0,Q,\mathcal{N})$, the optimal value function is

$$V_N(0,Q,\mathcal{N})=\min_{j\in\mathcal{N}}\left\{d_{0j}+\sum_{e}p_j(e)\cdot V_{N-1}(j,Q-e,\mathcal{N}\backslash\{j\})\right\}. \tag{5}$$

Optimal actions

The optimal action at final stage 0 is to return to the depot from the final customer, whereas beginning stage N includes a choice of which customer j to visit in such a way that

$$j_N(0,Q,\mathcal{N})=\arg\min_{j\in\mathcal{N}}\left\{d_{0j}+\sum_{e}p_j(e)\cdot V_{N-1}(j,Q-e,\mathcal{N}\backslash\{j\})\right\}. \tag{6}$$

Variable $u_{j,l,\mathcal{R}_k(l)}(q)$ is then introduced to denote the replenishment decision at state $(l,q,\mathcal{R}_k(l))$ concerning routing customer j next. The optimal replenish decision $u_{j,l,\mathcal{R}_k(l)}(q)$ is determined by

$$u_{j,l,\mathcal{R}_k(l)}(q)=\begin{cases}1,\ \text{if } V_k^{R(j)}(l,q,\mathcal{R}_k(l))\le V_k^{D(j)}(l,q,\mathcal{R}_k(l))\\ 0,\ \text{if } V_k^{R(j)}(l,q,\mathcal{R}_k(l))> V_k^{D(j)}(l,q,\mathcal{R}_k(l)).\end{cases}\quad \forall j\in\mathcal{R}_k(l),\ \text{for } s_k\in\Psi. \tag{7}$$

By defining $u_{j,l,\mathcal{R}_k(l)}(q)$, equation (4) can be rewritten as

$$\begin{aligned}
V_k^{u_{j,l,\mathcal{R}_k(l)}(q)}(l,q,\mathcal{R}_k(l)) &= d_{lj}+B_j(q)+(\Delta_{lj}-B_j(q))\cdot u_{j,l,\mathcal{R}_k(l)}(q)\\
&\quad+\mathbb{E}\left[V_{k-1}(j,q',\mathcal{R}_{k-1}(j;l))\,\middle|\,q,\ u_{j,l,\mathcal{R}_k(l)}(q)\right].\quad \forall j\in\mathcal{R}_k(l),\ s_k\in\Psi.
\end{aligned} \tag{8}$$

where $\Delta_{lj}=d_{l0}+d_{0j}-d_{lj}$ represents the extra cost for a preventive return to the depot when l and j are consecutive customers in the delivery route. q represents the initial residual capacity at current customer l. q' denotes the residual capacity after satisfying the demand of next customer j. $\mathbb{E}\left[V_{k-1}(j,q',\mathcal{R}_{k-1}(j;l))\,\middle|\,q,\ u_{j,l,\mathcal{R}_k(l)}(q)\right]$ is the expected future cost, given initial

residual capacity q and taking the replenish decision given by equation (7)

$$\mathbb{E}\left[V_{k-1}(j,q',\mathcal{R}_{k-1}(j;l)) \,\middle|\, q,\, u_{j,l,\mathcal{R}_k(l)}(q)\right] = \begin{cases} \sum\limits_{e\le q} p_j(e)\cdot V_{k-1}(j,q-e,\mathcal{R}_{k-1}(j;l)) + V_{k-1}(j,0,\mathcal{R}_{k-1}(j;l))\cdot \sum\limits_{e>q} p_j(e), & \text{if } u_{j,l,\mathcal{R}_k(l)}(q)=0, \\ \sum\limits_{e} p_j(e)\cdot V_{k-1}(j,Q-e,\mathcal{R}_{k-1}(j;l)), & \text{if } u_{j,l,\mathcal{R}_k(l)}(q)=1. \end{cases} \tag{9}$$

Based on the definition of $V_k^{u_{j,l,\mathcal{R}_k(l)}(q)}(l,q,\mathcal{R}_k(l))$, for state $s_k = (l,q,\mathcal{R}_k(l))$ at stage $k \in \Omega\backslash\{N,0\}$, the best next customer location is determined by

$$j_k(l,q,\mathcal{R}_k(l)) = \arg\min_{j\in\mathcal{R}_k(l)} \left\{ V_k^{u_{j,l,\mathcal{R}_k(l)}(q)}(l,q,\mathcal{R}_k(l)) \right\}. \tag{10}$$

Comparison of dynamic programming equations for VRPSD with outsourcing, and traditional VRPSD

In traditional VRPSD [see, e.g., 53, 46, 48], the cost-to-go value for each state s_k can be expressed as

$$V_k(l,q,\mathcal{R}_k(l)) = \min\left\{ V_k^{D(j_k^D(s_k))}(l,q,\mathcal{R}_k(l)),\, V_k^{R(j_k^R(s_k))}(l,q,\mathcal{R}_k(l)) \right\}, \quad \forall\, s_k \in \Psi, \tag{11}$$

where $j_k^D(s_k)$ and $j_k^R(s_k)$ are the best following customer locations for the case of proceeding to the next customer directly and the case of first replenishing at the depot, respectively. The best routing options are considered first in their formulations, and then replenishment decisions are made. It contrasts to equation (3), where replenishment decisions are made first, after which the best routing option is decided. By changing the decision sequence (i.e., $u \to j$, instead of $j \to u$), optimality equations (3) and (4) can be jointly expressed as

$$V_k(l,q,\mathcal{R}_k(l)) = \min_{j\in\mathcal{R}_k(l)} \left\{ d_{lj} + B_j(q) + (\Delta_{lj} - B_j(q))\cdot u_{j,l,\mathcal{R}_k(l)}(q) + \mathbb{E}\left[V_{k-1}(j,q',\mathcal{R}_{k-1}(j;l)) \,\middle|\, q,\, u_{j,l,\mathcal{R}_k(l)}(q)\right] \right\} \quad \forall\, j\in\mathcal{R}_k(l),\, s_k\in\Psi. \tag{12}$$

Note that we change the decision sequence for the sake of our approximation scheme. While value functions with or without restocking differ in their expressions as shown in equation (4), they can be unified as in equation (12). This unification facilitates value function decomposition and aids in deriving equation (16) in the next section. It is important to note that modifying the decision sequence can potentially affect the problem's structure and, consequently, influence the optimal action. However, reformulating a problem by altering the decision sequence can sometimes improve tractability, leading to different (or more efficiently computed) optimal actions. Section 4 explores the decomposition of value functions, based on the reformulation of the value function in equation (12), and introduces the solution framework for addressing the MDP discussed in this section.

4 | DECOMPOSITION-BASED APPROXIMATE LINEAR PROGRAM FRAMEWORK

Solving the MDP formulation inevitably leads to the curse of dimensionality. In our model, the state space has a cardinality of $1+N(Q+1)2^{N-1}$, making it intractable as the number of customers increases. One way to address this challenge is through the ALP approach. In this approach, the MDP is reformulated as a LP problem, where the decision variables represent the value functions. To further simplify the solution, affine functions are used to approximate these value functions, significantly reducing the number of decision variables in the LP. Additionally, techniques such as constraint sampling and constraint generation are employed to manage the LP's intractable number of constraints. Finally, a price-directed policy is derived based on the approximated value functions.

Applying the ALP approach directly can provide a solution to our problem. However, as noted in [50], the solution quality can be poor. Their experience with routing re-optimization under stochastic arc costs, solved using the ALP framework, shows that the approach performs well when returns to the depot are prohibited. However, its performance deteriorates when routing with recourse (i.e., allowing returns to the depot) is introduced. They argued that the problem structure of routing with recourse becomes complex, making direct value function approximation invalid and necessitating an alternative approach that better exploits the problem's structure in this context. In the following, we analyze the structure of the value functions in our formulation and propose a decomposition-based approach for value function approximation.

Decomposition-based value function approximation

Equation (12) unified the expression of value functions with and without restocking under routing re-optimization with stochastic demands. Using this formulation, restocking decisions are embedded within the value functions, leaving only routing decisions explicitly considered. This structure allows the value function to be decomposed into two components: one associated with routing decisions, and the other with replenishment trips and outsourcing resulting triggered by routing failures. A similar decomposition of the value function into two parts is observed in a priori optimization for solving the VRPSD. In this context, the problem is often modeled using stochastic programming with recourse (SPR), where a key structural property emerges in the objective function of SPR formulations. Specifically, the objective function typically consists of two components: the cost of routing along fixed routes and the penalty cost for replenishment trips [e.g., 33, 43].

In our formulation, the value functions are decomposed into two components: the cost associated with routing decisions and the cost incurred due to replenishment trips, as shown in equation (13). This decomposition offers a structured approach to modeling the problem, facilitating more efficient optimization and analysis. By distinguishing these two components, our formulation aligns with existing stochastic programming methods while providing a clearer interpretation of the underlying cost structure.

$$V_k(l,q,\mathcal{R}_k(l)) = \min_{[J,(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)]} \left\{ \left[d_{lJ} + v_{k-1}(J,\mathcal{R}_{k-1}(J;l)) \Big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)} \right] + f^J_k(l,q,\mathcal{R}_k(l)) \Big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)} \right\}, \quad J \in \mathcal{R}_k(l), \quad \forall\, s_k \in \Psi. \tag{13}$$

The first component $[d_{lJ} + v_{k-1}(J,\mathcal{R}_{k-1}(J;l))\Big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)}]$ represents the travel cost between customers, given the routing sequence $[l,J,(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)]$. The realized route starts from location l and includes the remaining customers in $\mathcal{R}_k(l)$, visiting customer $J \in \mathcal{R}_k(l)$ first, followed by customers $j^J_{k'}$ ($j^J_{k'} \in \mathcal{R}_{k-1}(J;l)\backslash\{j^J_{k-1},\ldots,j^J_{k'+1}\}$) at subsequent stages $k' = \{k-1,k-2,\ldots,1\}$. The term $v_{k-1}(J,\mathcal{R}_{k-1}(J;l))\Big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)}$ corresponds to the travel cost associated with the partial route starting from customer J and including the remaining customers in $\mathcal{R}_{k-1}(j;l)$. The second component $f^J_k(l,q,\mathcal{R}_k(l))\Big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)}$ accounts for the penalty cost, which includes additional travel distance for replenishment as well as potential outsourcing costs. This component represents the expected future penalty cost given the current state $(l,q,\mathcal{R}_k(l))$ and the decision to route customer J next. Both components depend on the future routing policy, represented by $[J,(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)]$. The superscript J in $j^J_{k'}$ ($k' = \{k-1,k-2,\ldots,1\}$) indicates that future routing decisions should be made by fixing J as the first of the next customers to visit. Equation (13) aims to determine the optimal routing policy $[J,(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)]$ for state s_k in order to minimize expected future costs.

The decomposition-based value function approximation is derived from equation (13). The value function is approximated by decomposing it into two components and then estimating each separately. The following relation holds for the second component in (13)

$$f^J_k(l,q,\mathcal{R}_k(l))\Big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)} \geq \min_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)} \left\{ f^J_k(l,q,\mathcal{R}_k(l))\Big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)} \right\} = \min\left\{ f^J_k(l,q,\mathcal{R}_k(l)) \right\}, \quad \forall\, s_k \in \Psi,\ J \in \mathcal{R}_k(l), \tag{14}$$

The right-hand side of the equation follows from the fact that $\min_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)} \left\{ f^J_k(l,q,\mathcal{R}_k(l))\Big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)} \right\}$ represents the minimal penalty cost among all possible routing sequences. Let $L^J_{s_k}$ denote the lower bound of the future expected penalty cost for state s_k when routing customer J next. Assuming that lower bounds $L^J_{s_k}$ ($\forall\, s_k \in \Psi$) are available, these bounds are then used to approximate the second component $f^J_k(l,q,\mathcal{R}_k(l))\Big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)}$. Let $\tilde{V}_k(l,q,\mathcal{R}_k(l))$ ($s_k \in \Psi$) represents the approximated value functions. The value functions are approximated as

$$\tilde{V}_k(l,q,\mathcal{R}_k(l)) = \min_{[J,(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)]} \left\{ d_{lJ} + v_{k-1}(J,\mathcal{R}_{k-1}(J;l))\Big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)} + L^J_{s_k} \right\} \tag{15.1}$$

$$= \min_{J\in\mathcal{R}_k(l)} \left\{ d_{lJ} + \min_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)} \left\{ v_{k-1}(J,\mathcal{R}_{k-1}(J;l))\Big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)} \right\} + L^J_{s_k} \right\} \tag{15.2}$$

$$= \min_{J\in\mathcal{R}_k(l)} \left\{ d_{lJ} + \min\left\{ v_{k-1}(J,\mathcal{R}_{k-1}(J;l)) \right\} + L^J_{s_k} \right\}, \quad \forall\, s_k \in \Psi, \tag{15.3}$$

equation (15.3) holds for the same reason as in equation (14). In fact, $\min\left\{v_{k-1}(J,\mathcal{R}_{k-1}(J;l))\right\}$ implies a deterministic TSP process. Following the definition of $v_{k-1}(J,\mathcal{R}_{k-1}(J;l))$, $\min\{v_{k-1}(J,\mathcal{R}_{k-1}(J;l))\}$ determines a route that includes remaining customers in $\mathcal{R}_{k-1}(J;l)$ and ends at the depot with the minimal traveling cost. For a deterministic TSP problem, tight lower bounds can be fast generated by extant methods [27]. Let $l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)}$ ($l \in \mathcal{N}, J \in \mathcal{R}_k(l), \left|\mathcal{R}_k(l)\right| = k$) represent the lower bounds for approximating $\min\left\{v_{k-1}(J,\mathcal{R}_{k-1}(J;l))\right\}$. Value functions can be further estimated by

$$\tilde{V}_k(l,q,\mathcal{R}_k(l)) = \min_{J\in\mathcal{R}_k(l)}\left\{d_{lJ} + l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)} + L_{s_k}^J\right\}, \quad \forall\, s_k \in \Psi. \tag{16}$$

In our method, value functions are approximated in two steps. In the first step, the expected penalty costs $L_{s_k}^J$ are estimated for possible states $s_k \in \Psi$ and their associated next customers $J \in \mathcal{R}_k(l)$. In the second step, the approximated travel costs for potential TSP routes $\left\{l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)} | \forall J \in \mathcal{R}_k(l)\right\}$ are computed, given the realized state s_k and each potential next customer $J \in \mathcal{R}_k(l)$. Based on these two steps, the optimal routing decision at state s_k is determined according to

$$j_k(l,q,\mathcal{R}_k(l)) = \arg\min_{J\in\mathcal{R}_k(l)}\left\{d_{lJ} + l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)} + L_{s_k}^J\right\}, \quad \text{given a realized state } s_k. \tag{17}$$

In other words, $L_{s_k}^J$ are calculated a priori, whereas $l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)}$ are generated as needed. Specifically, at state $s_k = (l,q,\mathcal{R}_k(l))$, values $L_{s_k}^J$ ($J \in \mathcal{R}_k(l)$) are available, so only $l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)}$ ($J \in \mathcal{R}_k(l)$, $\mathcal{R}_{k-1}(J;l) = \mathcal{R}_k(l)\backslash\{J\}$) need to be computed. The following LP formulation can be utilized to generate $l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)}$ for each potential next customer $J \in \mathcal{R}_k(l)$, and this LP is solvable in polynomial time [40, 28].

$$l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)} = \min_x \left\{ \sum_{i'\in\mathcal{R}_{k-1}(J;l)} d_{Ji'}x_{Ji'} + \sum_{i'\in\mathcal{R}_{k-1}(J;l)}\sum_{j'\in\mathcal{R}_{k-1}(J;l)\cup\{0\}\backslash i'} d_{i'j'}x_{i'j'} \right\} \tag{18.1}$$

$$\text{s.t.} \sum_{i'\in\mathcal{R}_{k-1}(J;l)} x_{Ji'} = 1, \tag{18.2}$$

$$\sum_{i'\in\mathcal{R}_{k-1}(J;l)} x_{i'0} = 1, \tag{18.3}$$

$$\sum_{j'\in\mathcal{R}_{k-1}(J;l)\backslash i'} x_{i'j'} = 1, \quad i' \in \mathcal{R}_{k-1}(J;l), \tag{18.4}$$

$$\sum_{j'\in\mathcal{R}_{k-1}(J;l)\backslash i'} x_{j'i'} = 1, \quad i' \in \mathcal{R}_{k-1}(J;l), \tag{18.5}$$

$$\sum_{i'\in U}\sum_{j'\in\mathcal{R}_{k-1}(J;l)\cup\{0\}\backslash U} x_{i'j'} \geq 1, \quad \forall\, \emptyset \neq U \subseteq \mathcal{R}_{k-1}(J;l)\cup\{0\}, \tag{18.6}$$

$$x \geq 0, \quad x \in \mathbb{R}. \tag{18.7}$$

The LP formulation (18) is used to determine lower bound $l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)}$ ($J \in \mathcal{R}_k(l)$). In Section 5, the approximation of expected penalty costs, $\left\{L_{s_k}^J \,\middle|\, \forall\, s_k \in \Psi,\, J \in \mathcal{R}_k(l)\right\}$, are determined.

Overview of our solution framework

With the value function decomposition scheme presented in equation (16), our solution framework can be generalized, as illustrated in Figure 2.

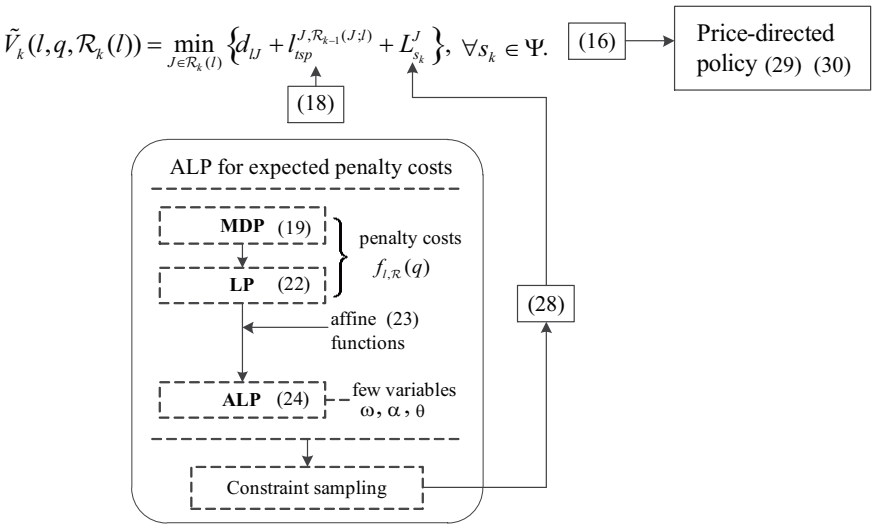


FIGURE 2 Decomposition-based approximate linear programming solution framework

As shown in Figure 2, our solution framework is developed based on value function decomposition, as described in equation (16). The value function is approximated by estimating two components separately: $l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)}$ and $L_{s_k}^J$. The first component, $l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)}$, obtained by formulation (18), estimates the traveling cost between customers by determining the minimal-cost route when visiting customer J next. The second component, $L_{s_k}^J$, obtained by equations (28), approximates the penalty cost for replenishment and outsourcing by formulating it as a MDP (19) and solving it using the ALP method. Following the ALP method, the MDP formulation (19) is first transformed into the LP formulation (22). Then, the ALP formulation (24) is derived by introducing affine functions (23). The ALP formulation (24) contains relatively few variables but an intractable number of constraints, which are handled using constraint sampling. Several techniques to facilitate the formulation and solution of the ALP are elaborated in Section 5. Additionally, a multi-sampling framework is proposed to implement constraint sampling. Finally, with the approximated value functions, a price-directed policy is generated based on equations (29) and (30).

5 | LOWER BOUND OF THE EXPECTED PENALTY

The value functions are approximated according to equation (16), with $l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)}$ estimated by (18). Thus, only $L_{s_k}^J$ remains to be determined. Section 5.1 establishes the optimality equation for expected penalty costs and subsequently reformulates the MDP using its linear programming counterpart. Section 5.2 introduces the ALP formulation for penalty costs by defining affine functions. Section 5.3 proposes a constraint sampling approach for solving the ALP formulation.

5.1 | Expected penalty cost

For any state $s_k = (l, q, \mathcal{R}_k(l)) \in \Psi$ and potential next customer $J \in \mathcal{R}_k(l)$, $L_{s_k}^J$ represents the lower bound of the expected penalty cost $f_k^J(l, q, \mathcal{R}_k(l))$. As explained in equation (13), $f_k^J(l, q, \mathcal{R}_k(l))$ denotes the expected future penalty cost given the current state $(l, q, \mathcal{R}_k(l))$ and the decision to route customer J next. This cost includes the additional travel distance for

replenishment and potential outsourcing expenses. It satisfies the following optimality equation

$$f_k^J(l,q,\mathcal{R}_k(l)) = \min \begin{cases} \Delta_{lJ} + \sum\limits_e p_J(e)\cdot f_{k-1}(J,Q-e,\mathcal{R}_{k-1}(J;l)), & u_{J,l,\mathcal{R}_k(l)}(q)=1, \\ \sum\limits_{e\le q} p_J(e)\cdot f_{k-1}(J,q-e,\mathcal{R}_{k-1}(J;l)) + b\cdot\sum\limits_{e>q} p_J(e)\cdot(e-q) + \sum\limits_{e>q} p_J(e)\cdot f_{k-1}(J,0,\mathcal{R}_{k-1}(J;l)), & u_{J,l,\mathcal{R}_k(l)}(q)=0, \end{cases}$$
$$\forall\, s_k=(l,q,\mathcal{R}_k(l))\in\Psi,\ J\in\mathcal{N}\backslash l,\ k\in\Omega\backslash\{N,0\}, \tag{19.1}$$

and the optimal value function at beginning stage N is

$$f_N^J(0,Q,\mathcal{N}) = \min_{j\in\mathcal{N}}\left\{\sum_e p_j(e)\cdot f_{N-1}(j,Q-e,\mathcal{N}\backslash\{j\})\right\}. \tag{19.2}$$

The MDP formulation (19) is derived from formulation (3)-(5), by only considering the penalty costs caused by replenishment trips and outsourcing expenses. The replenishment decision $u_{J,l,\mathcal{R}_k(l)}(q)$ for each state s_k aims to minimize the future expected penalty costs, either by replenishing before arriving at customer J (i.e., $u_{J,l,\mathcal{R}_k(l)}(q)=1$) or not (i.e., $u_{J,l,\mathcal{R}_k(l)}(q)=0$). If decision $u_{J,l,\mathcal{R}_k(l)}(q)=1$ is made, the vehicle arrives at customer J with full capacity Q and incurs an additional travel cost Δ_{lJ} where $\Delta_{lJ}=d_{l0}+d_{0J}-d_{lJ}$. Conversely, if $u_{J,l,\mathcal{R}_k(l)}(q)=0$, the vehicle proceeds to customer J without replenishing, leaving its capacity empty and facing a probability $\sum\limits_{e>q} p_J(e)$ of failing to service customer J. Note that $f_k(l,q,\mathcal{R}_k(l)) = \min\limits_{J\in\mathcal{R}_k(l)}\{f_k^J(l,q,\mathcal{R}_k(l))\}$, which indicates that value function $f_k(l,q,\mathcal{R}_k(l))$ corresponds to the minimal value among all possible value functions when fixing customer J as the next visit. In the following, $f_k(l,q,\mathcal{R}_k(l))$ ($f_k^J(l,q,\mathcal{R}_k(l))$) is substituted with $f_{l,\mathcal{R}}(q)$ ($f_{l,\mathcal{R}}^j(q)$) for ease of notation.

The penalty costs $f_{l,\mathcal{R}}^j(q)$ exhibits the properties outlined in Propositions 1 and 2. These properties are utilized to define the affine functions in Section 5.2, to select the constraints in Section 5.3, and to determine the outsourcing price b in Section 7.

Proposition 1. (Monotonicity of penalty cost on residual capacity q) *for given customer l, customer j, and unvisited set $\mathcal{R}$, penalty cost $f_{l,\mathcal{R}}^j(q)$ is non-increasing in residual capacity q.*

Proposition 2. (Possible threshold-type replenishment) *for particular customer l^*, customer j^* and unvisited set $\mathcal{R}^*$ (not all), the optimal choice between replenishing and moving directly to the next customer is of threshold type in residual capacity q.*

The proofs are shown in Appendix B.□

The MDP formulation (19) can be rewritten as the following LP model

$$\max \quad f_{0,\mathcal{N}}^j(Q) \tag{20.1}$$
$$\text{s.t.} \quad f_{0,\mathcal{N}}^j(Q) \le \sum_e p_j(e)\cdot f_{j,\mathcal{N}\backslash j}(Q-e), \quad \forall j\in\mathcal{N}, \tag{20.2}$$
$$f_{l,\mathcal{R}}^j(q) \le \Delta_{lj} + \sum_e p_j(e)\cdot f_{j,\mathcal{R}\backslash l}(Q-e), \quad \forall\, l\in\mathcal{N}, j\in\mathcal{R}\subseteq\mathcal{N}\backslash l,\ q\in\mathcal{Q}^{fe}_{|\mathcal{R}|}\ (|\mathcal{R}|\in\{N-1,N-2,\ldots,2\}), \tag{20.3}$$
$$f_{l,\mathcal{R}}^j(q) \le \sum_{e\le q} p_j(e)\cdot f_{j,\mathcal{R}\backslash l}(q-e) + b\cdot\sum_{e>q} p_j(e)\cdot(e-q) + \sum_{e>q} p_j(e)\cdot f_{j,\mathcal{R}\backslash l}(0),$$
$$\forall\, l\in\mathcal{N}, j\in\mathcal{R}\subseteq\mathcal{N}\backslash l,\ q\in\mathcal{Q}^{fe}_{|\mathcal{R}|}\ (|\mathcal{R}|\in\{N-1,N-2,\ldots,2\}), \tag{20.4}$$
$$f_{l,\{j\}}^j(q) \le \Delta_{lj}, \quad \forall\, l\in\mathcal{N}, j\in\mathcal{N}\backslash l, \mathcal{R}=\{j\},\ q\in\mathcal{Q}_1^{fe}, \tag{20.5}$$
$$f_{l,\{j\}}^j(q) \le b\cdot\sum_{e>q} p_j(e)\cdot(e-q), \quad \forall\, l\in\mathcal{N}, j\in\mathcal{N}\backslash l,\ \mathcal{R}=\{j\},\ q\in\mathcal{Q}_1^{fe}, \tag{20.6}$$
$$f_{l,\mathcal{R}}^j(q), f_{l,\mathcal{R}}(q), f_{0,\mathcal{N}}^j(Q), f_{0,\mathcal{N}}(Q)\in\mathbb{R}, \quad \forall\, l\in\mathcal{N}, j\in\mathcal{R}\subseteq\mathcal{N}\backslash l,\ q\in\mathcal{Q}^{fe}_{|\mathcal{R}|}\ (|\mathcal{R}|\in\{N-1,N-2,\ldots,1\}). \tag{20.7}$$

Objective (20.1) and constraint (20.2) jointly express equation (19.2). Constraints (20.3) and (20.4) correspondingly translate equation (19.1), capturing the transition of expected penalty costs from stage $N-1$ to 2. Constraints (20.5) and (20.6) define the conditions at stage 1, ensuring that any expected penalty cost at final stage 0 is zero, i.e., $f_{l,\emptyset}(q)=0$ for all $l\in\mathcal{N}$ and

$q \in \mathcal{Q}_0^{fe}$. Additionally, the feasible range of residual capacity q at each stage k ($k = |\mathcal{R}| = N-1, \ldots, 1$) in constraints (20.3)-(20.6), denoted as $\mathcal{Q}_k^{fe}$ (or $\mathcal{Q}_{|\mathcal{R}|}^{fe}$), is given by $[(Q-(N-k)\cdot E)^+, Q-e_{\min}]$, where $e_{\min}$ represents the minimal demand amount ξ can take. In LP formulation (20), the variables are $f_{l,\mathcal{R}}^j(q)$ and $f_{l,\mathcal{R}}(q)$. Under objective (20.1), the optimal solution $f_{l,\mathcal{R}}^j(q)^*$ ($f_{l,\mathcal{R}}(q)^*$) provides the largest lower bound for the penalty cost. This lower bound can then be used to approximate the expected penalty cost.

5.2 | Affine approximation for lower bound

Solving formulation (20) can be computationally inefficient due to the large number of variables. The variables $f_{l,\mathcal{R}}^j(q)$ and $f_{l,\mathcal{R}}(q)$ scale as $O(N^2 \cdot 2^{N-1} \cdot Q)$, which follows from the calculation: $N^2 \cdot (\mathrm{C}_{N-1}^1 + \mathrm{C}_{N-1}^2 + , \ldots, + \mathrm{C}_{N-1}^{N-1}) \cdot (Q+1) + 1$. Since the summation of binomial coefficients results in 2^{N-1}, the total number of variables grows exponentially with N, making computation challenging. To mitigate this, variable deduction is necessary. A key observation is that $f_{l,\mathcal{R}}(q) = \min_{j\in\mathcal{R}} \left\{ f_{l,\mathcal{R}}^j(q) \right\}$. This allows us to eliminate redundant variables by enforcing constraint (21) in formulation (20). By leveraging this relationship, the number of independent variables is reduced, easing the computational burden.

$$f_{l,\mathcal{R}}(q) \le f_{l,\mathcal{R}}^j(q), \quad \forall j \in \mathcal{R} \subseteq \mathcal{N}\backslash l \quad (l \in 0 \cup \mathcal{N},\, q \in \mathcal{Q}_k^{fe},\, k \in \{N, N-1, \ldots, 1\}) \tag{21}$$

Formulation (20) is then reformulated as follows:

$$\max \quad f_{0,\mathcal{N}}(Q) \tag{22.1}$$

$$\text{s.t.} \quad f_{0,\mathcal{N}}(Q) \le \sum_e p_j(e) \cdot f_{j,\mathcal{N}\backslash j}(Q-e), \quad \forall j \in \mathcal{N}, \tag{22.2}$$

$$f_{l,\mathcal{R}}(q) \le \Delta_{lj} + \sum_e p_j(e) \cdot f_{j,\mathcal{R}\backslash l}(Q-e), \quad \forall\, l \in \mathcal{N}, j \in \mathcal{R} \subseteq \mathcal{N}\backslash l,$$
$$q \in \mathcal{Q}_{|\mathcal{R}|}^{fe}\ (|\mathcal{R}| \in \{N-1, N-2, \ldots, 2\}), \tag{22.3}$$

$$f_{l,\mathcal{R}}(q) \le \sum_{e\le q} p_j(e) \cdot f_{j,\mathcal{R}\backslash j}(q-e) + b \cdot \sum_{e>q} p_j(e) \cdot (e-q) + \sum_{e>q} p_j(e) \cdot f_{j,\mathcal{R}\backslash j}(0),$$
$$\forall\, l \in \mathcal{N}, j \in \mathcal{R} \subseteq \mathcal{N}\backslash l,\, q \in \mathcal{Q}_{|\mathcal{R}|}^{fe}\ (|\mathcal{R}| \in \{N-1, N-2, \ldots, 2\}), \tag{22.4}$$

$$f_{l,\{j\}}(q) \le \Delta_{lj}, \quad \forall\, l \in \mathcal{N}, j \in \mathcal{N}\backslash l,\, \mathcal{R} = \{j\},\, q \in \mathcal{Q}_1^{fe}, \tag{22.5}$$

$$f_{l,\{j\}}(q) \le b \cdot \sum_{e>q} p_j(e) \cdot (e-q), \quad \forall\, l \in \mathcal{N}, j \in \mathcal{N}\backslash l, \mathcal{R} = \{j\},\, q \in \mathcal{Q}_1^{fe}, \tag{22.6}$$

$$f_{l,\mathcal{R}}(q), f_{0,\mathcal{N}}(Q) \in \mathbb{R}, \quad \forall\, l \in \mathcal{N},\, q \in \mathcal{Q}_{|\mathcal{R}|}^{fe}\ (|\mathcal{R}| \in \{N-1, N-2, \ldots, 1\}). \tag{22.7}$$

The number of variables is decreased, retaining only $f_{l,\mathcal{R}}(q)$ in the formulation. Further variable reduction is achieved by identifying a set of basis functions and substituting their affine forms for the original variables. To approximate the expected penalty costs, we derive affine function representations as given in (23.1)-(23.3). These affine functions are specifically designed to align with the structure of our problem and are adapted from existing research [e.g., 50, 49].

$$f_{0,\mathcal{N}}(Q) \approx \theta_{0,0,Q}, \tag{23.1}$$

$$f_{l,\mathcal{R}}(q) \approx \theta_{l,0,q} + \sum_{j\in\mathcal{R}} (\alpha_{l,j,q} \cdot \Delta_{lj} + \omega_{l,j} \cdot q), \quad l \in \mathcal{N}, j \in \mathcal{R} \subseteq \mathcal{N}\backslash l,\, |\mathcal{R}| \ge 2,\, q \in \mathcal{Q}_{|\mathcal{R}|}^{fe}, \tag{23.2}$$

$$f_{l,\{j\}}(q) \approx \theta_{l,0,q} + \alpha_{l,j,q} \cdot \Delta_{lj} + \omega_{l,j} \cdot q, \quad l \in \mathcal{N}, j \in \mathcal{N}\backslash l,\, \mathcal{R} = \{j\}. \tag{23.3}$$

where $\theta \in \mathbb{R}^{N\cdot Q+N+1}$, $\alpha \in \mathbb{R}^{(N^2-N)\cdot(Q+1)}$ and $\omega \in \mathbb{R}^{N^2-N}$. The affine functions (23.1)-(23.3) approximate the expected penalty cost $f_{l,\mathcal{R}}(q)$. First, penalty costs arise due to restocking actions or outsourcing. To capture these costs, terms $\alpha_{l,j,q} \cdot \Delta_{l,j}$ and $\omega_{l,j} \cdot q$ are introduced, respectively. Term $\alpha_{l,j,q} \cdot \Delta_{lj}$ accounts for the additional travel cost Δ_{lj} ($j \in \mathcal{R} \subseteq \mathcal{N}\backslash l$) incured when restocking, while $w_{l,j} \cdot q$ represents the outsourcing cost, which depends on the available residual capacity q. Additionally, constant θ is introduced to adjust the approximation of the penalty cost. Second, the expected penalty cost is determined by the system states ($\forall\, s_k = (l, q, \mathcal{R})$) so, parameters θ, α and ω are defined concerning the states. Note that ω depends only on current location l and

remaining unvisited customer $j \in \mathcal{R}$, since the impact of residual capacity q is already incorporated as a multiplicative factor in $\omega_{l,j} \cdot q$. Term $\omega \cdot q$ captures the monotonicity of the penalty cost with respect to residual capacity q, with the proof provided in Appendix B. Finally, expression (23.2) reflects how the penalty cost varies when selecting different next customers $j \in \mathcal{R}$, particularly when multiple customers remain unvisited ($|\mathcal{R}| \geq 2$). With the affine functions in (23.1)-(23.3), formulation (22) is reformulated as

$$\max \quad \theta_{0,0,Q} \tag{24.1}$$

$$\text{s.t.} \quad \theta_{0,0,Q} \leq \sum_e p_j(e) \cdot \theta_{j,0,Q-e} + \sum_{t \in \mathcal{N} \setminus j} \Delta_{jt} \cdot \Big(\sum_e p_j(e) \cdot \alpha_{j,t,Q-e}\Big) + \sum_{t \in \mathcal{N} \setminus j} \omega_{jt} \cdot \Big(\sum_e p_j(e) \cdot (Q-e)\Big), \quad \forall j \in \mathcal{N}, \tag{24.2}$$

$$\theta_{l,0,q} + \sum_{t \in \mathcal{R}} (\Delta_{lt} \cdot \alpha_{l,t,q} + \omega_{lt} \cdot q) \leq \Delta_{lj} + \sum_e p_j(e) \cdot \theta_{j,0,Q-e} + \sum_{t \in \mathcal{R} \setminus j} \Delta_{jt} \cdot \Big(\sum_e p_j(e) \cdot \alpha_{j,t,Q-e}\Big) + \sum_{t \in \mathcal{R} \setminus j} \omega_{jt} \cdot \Big(\sum_e p_j(e) \cdot (Q-e)\Big), \quad \forall l \in \mathcal{N}, j \in \mathcal{R} \subseteq \mathcal{N} \setminus l,\ q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}\ (|\mathcal{R}| \in \{N-1, N-2, \ldots, 2\}), \tag{24.3}$$

$$\theta_{l,0,q} + \sum_{t \in \mathcal{R}} (\Delta_{lt} \cdot \alpha_{l,t,q} + \omega_{lt} \cdot q) \leq \Big(\sum_{e \leq q} p_j(e) \cdot \theta_{j,0,q-e} + \sum_{e > q} p_j(e) \cdot \theta_{j,0,0}\Big) + b \cdot \sum_{e > q} p_j(e) \cdot (e-q) + \sum_{t \in \mathcal{R} \setminus j} \Delta_{jt} \cdot \Big(\sum_{e \leq q} p_j(e) \cdot \alpha_{j,t,q-e} + \sum_{e > q} p_j(e) \cdot \alpha_{j,t,0}\Big) + \sum_{t \in \mathcal{R} \setminus j} \omega_{jt} \cdot \Big(\sum_{e \leq q} p_j(e) \cdot (q-e)\Big), \quad \forall l \in \mathcal{N}, j \in \mathcal{R} \subseteq \mathcal{N} \setminus l,\ q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}\ (|\mathcal{R}| \in \{N-1, N-2 \ldots, 2\}), \tag{24.4}$$

$$\theta_{l,0,q} + \Delta_{lj} \cdot \alpha_{l,j,q} + \omega_{lj} \cdot q \leq \Delta_{lj}, \quad \forall l \in \mathcal{N}, j \in \mathcal{N} \setminus l,\ q \in \mathcal{Q}^{fe}_1, \tag{24.5}$$

$$\theta_{l,0,q} + \Delta_{lj} \cdot \alpha_{l,j,q} + \omega_{lj} \cdot q \leq b \cdot \sum_{e > q} p_j(e) \cdot (e-q), \quad \forall l \mathcal{N}, j \in \mathcal{N} \setminus l,\ q \in \mathcal{Q}^{fe}_1, \tag{24.6}$$

$$\theta,\ \alpha,\ \omega \in \mathbb{R}. \tag{24.7}$$

In formulation (24), variables include θ, α and ω, scaling as $O(N^2 \cdot Q)$, which is calculated by $N^2 \cdot (Q+1) + N^2 - N + 1$. This represents a dramatic reduction in the number of variables compared to the original formulation (22). While the ALP formulation (24) now has fewer variables, it still involves an intractable number of constraints. To address this challenge, the following subsection introduces a constraint sampling approach to manage the large number of constraints effectively.

5.3 | Constraint sampling

Formulation (24) reduces variables to a manageable size. However, the number of constraints is still too large to solve. We employ a constraint sampling approach [17] to tackle this issue. Constraint sampling is a general method used to tackle LP formulations with few variables and an intractable number of constraints. It approximates the solution to the ALP. Our constraint sampling framework selects promising constraints, and a solution based on the reduced formulation is obtained. Specifically, a promising constraint set is formed based on selected state-action pairs, and each pair is obtained by sample learning from a heuristic policy. Only a subset of constraints is included in the formulation, considering that some constraints are inactive or have a minor impact on the feasible region [17].

Our method is developed based on the general framework for constraint sampling. The general method relies on the existence of an optimal policy, which is usually unknown. We propose a multi-policy sampling framework to mimic the optimal policy. A similar idea appears in Novoa and Storer [38]. The constraint space relevant to the ideal policy is mimicked based on the constraints sampled by a set of heuristic policies. The local optimum obtained by a single heuristic policy can thus be escaped by exploring a more extensive solution space discovered via policy diversification.

In our multi-policy sampling framework, a set of heuristic policies (denoted Pls) is prepared and listed in Appendix A. Each policy is found using a heuristic algorithm to learn about each sample. The state-action pairs are thus generated. Specifically, for sample $\{\xi\}^{sam}$, if applying policy pl ($pl \in$ Pls), a sequence of states and actions is obtained in the form

$$\big(s_N,\ a_{s_N|pl};\ s_{N-1},\ a_{s_{N-1}|pl}; \ldots;\ s_0,\ a_{s_0|pl}\big)^{\{\xi\}^{sam}}, \tag{25}$$

where $\{\xi\}^{sam}$ denotes a sample of realized customer demand $\{\xi_l \mid l \in \mathcal{N}\}$. $a_{s_{klpl}} = (j, u_{j,l,\mathcal{R}}(q))$ specifies the outcome of applying distinct heuristic policy pl, potentially indicating a different routing decision j and restocking decision $u_{j,l,\mathcal{R}}(q)$ to be taken given realized state s_k. State $s_k = (l, q, \mathcal{R})$ transits to state $s_{k-1} = (j, [q + (Q-q)\cdot u_{j,l,\mathcal{R}}(q) - \xi_j]^+, \mathcal{R}\backslash j)$ depending on action $a_{s_{klpl}} = (j, u_{j,l,\mathcal{R}}(q))$ and realized demand ξ_j, where $[\cdot]^+$ indicates non-negative residual capacity. Therefore, as each heuristic policy learns each sample, a set of states s and associated actions $a_{s|pl}$ are obtained. The sequence can be written as a set of pairs of states and actions $(s, a_{s|pl})$ ($s = \{s_N, s_{N-1}, \ldots, s_0\}$). The pool of state-action pairs is finally formed by combining all sets of state-action pairs obtained during the learning, denoted as $\mathrm{P_{s-a}} = \bigcup_{\substack{pl\in \mathrm{Pls} \\ s' \leftarrow s, a_{s|pl}}} (s, a_{s|pl})$.

Each state-action pair corresponds to a specific group of constraints. For example, for state-action pair (s_k, a_{s_k}):= $(l, q, \mathcal{R}, j, u_{j,l,\mathcal{R}}(q)))$, if $|\mathcal{R}| \geq 2$ and $\mathcal{R} \neq \mathcal{N}$, then constraints (26) are applied.

$$\begin{cases} \theta_{l,0,q} + \sum\limits_{t\in\mathcal{R}} (\Delta_{lt}\cdot\alpha_{l,t,q} + \omega_{lt}\cdot q) \leq \Delta_{lj} + \sum\limits_{e} p_j(e)\cdot\theta_{j,0,Q-e} + \sum\limits_{t\in\mathcal{R}\backslash j} \Delta_{jt}\cdot(\sum\limits_{e} p_j(e)\cdot\alpha_{j,t,Q-e}) + \\ \qquad\qquad \sum\limits_{t\in\mathcal{R}\backslash j} \omega_{jt}\cdot(\sum\limits_{e} p_j(e)\cdot(Q-e)), \qquad u_{j,l,\mathcal{R}}(q) = 1, \\ \theta_{l,0,q} + \sum\limits_{t\in\mathcal{R}} (\Delta_{lt}\cdot\alpha_{l,t,q} + \omega_{lt}\cdot q) \leq (\sum\limits_{e\leq q} p_j(e)\cdot\theta_{j,0,q-e} + \sum\limits_{e>q} p_j(e)\cdot\theta_{j,0,0}) + b\cdot\sum\limits_{e>q} p_j(e)\cdot(e-q) + \\ \qquad \sum\limits_{t\in\mathcal{R}\backslash j} \Delta_{jt}\cdot(\sum\limits_{e\leq q} p_j(e)\cdot\alpha_{j,t,Q-e} + \sum\limits_{e>q} p_j(e)\cdot\alpha_{j,t,0}) + \sum\limits_{t\in\mathcal{R}\backslash j} \omega_{jt}\cdot(\sum\limits_{e\leq q} p_j(e)\cdot(q-e)), \qquad u_{j,l,\mathcal{R}}(q) = 0, \end{cases}$$
$$q \in \mathcal{Q}^{fe}_{|\mathcal{R}|},\ 2 \leq |\mathcal{R}| \leq N \quad (26)$$

Constraints (26) indicate that selecting customer j as the next destination is considered a promising action when the vehicle is at customer j and the set of unvisited customers is $\mathcal{R}$ ($j \in \mathcal{R}$). All feasible residual capacities $q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}$ are considered, as long as whether a restocking action is performed ($u_{j,l,\mathcal{R}}(q) = 1$) or not ($u_{j,l,\mathcal{R}}(q) = 0$). Since some residual capacities may not be observed during sampling, all feasible values of q are included. Additionally, constraints for both restocking and non-restocking scenarios are incorporated to capture the threshold-based nature of restocking under varying realized residual capacities. Similarly, when the state action pair is $(l, q, \{j\}, j, u_{j,l,\{j\}})$, i.e., at stage 1, when only one customer remains unvisited ($\mathcal{R} = \{j\}$), the selected constraints are

$$\begin{cases} \theta_{l,0,q} + \Delta_{lj}\cdot\alpha_{l,j,q} + \omega_{lj}\cdot q \leq \Delta_{lj}, & u_{j,l,\mathcal{R}}(q) = 1, \\ \theta_{l,0,q} + \Delta_{lj}\cdot\alpha_{l,j,q} + \omega_{lj}\cdot q \leq b\cdot\sum\limits_{e>q} p_j(e)\cdot(e-q), & u_{j,l,\mathcal{R}}(q) = 0, \end{cases} \quad q \in \mathcal{Q}^{fe}_1 \quad (27)$$

For state-action pair $(0,\ q \equiv Q,\ j,\ u_{j,0,\mathcal{N}}(q) \equiv 0)$ ($j \in \mathcal{N}$), at beginning stage N, the selected constraint follows the same form as in inequality (24.2). Consequently, the promising constraint set is established. All promising constraints are selected based on promising state-action pairs $(s_k, a_{s_k|m}) \in \mathrm{P_{s-a}}$, according to inequalities (26), (27), and (24.2).

By solving ALP formulation (24) with the constraints sampled by state-action pairs $(s_k, a_{s_k|m}) \in \mathrm{P_{s-a}}$, the values of parameters θ, α and ω are approximated. The lower bounds of expected penalty costs $L^J_{s_k}$ ($J \in \mathcal{R}$, $s_k = (l, q, \mathcal{R}) \in \Psi$) are subsequently obtained based on equations (28.1)-(28.3).

$$L^J_{s_N} \approx \sum_e p_J(e)\cdot\theta_{J,0,Q-e} + \sum_{t\in\mathcal{N}\backslash J} \Delta_{Jt}\cdot(\sum_e p_J(e)\cdot\alpha_{J,t,Q-e}) + \sum_{t\in\mathcal{N}\backslash J} \omega_{Jt}\cdot(\sum_e p_J(e)\cdot(Q-e)),$$
$$s_N = (0, Q, \mathcal{N}),\ j \in \mathcal{N}, \quad (28.1)$$

$$L^J_{s_k} = \min\left\{L^{R(J)}_{s_k}, L^{D(J)}_{s_k}\right\} \approx$$
$$\min\begin{cases} \Delta_{lJ} + \sum\limits_e p_J(e)\cdot\theta_{J,0,Q-e} + \sum\limits_{t\in\mathcal{R}\backslash j} \Delta_{Jt}\cdot(\sum\limits_e p_J(e)\cdot\alpha_{J,t,Q-e}) + \sum\limits_{t\in\mathcal{R}\backslash J} \omega_{Jt}\cdot(\sum\limits_e p_J(e)\cdot(Q-e)), & u_{J,l,\mathcal{R}}(q) = 1, \\ (\sum\limits_{e\leq q} p_J(e)\cdot\theta_{J,0,q-e} + \sum\limits_{e>q} p_J(e)\cdot\theta_{J,0,0}) + \sum\limits_{t\in\mathcal{R}\backslash J} \Delta_{Jt}\cdot(\sum\limits_{e\leq q} p_J(e)\cdot\alpha_{J,t,Q-e} + \sum\limits_{e>q}\cdot p_J(e)\cdot\alpha_{J,t,0}) + \\ \qquad \sum\limits_{t\in\mathcal{R}\backslash J} \omega_{Jt}\cdot(\sum\limits_{e\leq q} p_J(e)\cdot(q-e)) + b\cdot\sum\limits_{e>q} p_J(e)\cdot(e-q), & u_{J,l,\mathcal{R}}(q) = 0, \end{cases}$$
$$\forall\, s_k = (l, q, \mathcal{R}) \in \Psi,\ J \in \mathcal{N}\backslash l,\ l \in \mathcal{N},\ q \in \mathcal{Q}^{fe}_k,\ 2 \leq k \leq N-1, \quad (28.2)$$

$$L^J_{s_1} \approx \min\left\{\Delta_{lJ}, b\cdot\sum_{e>q} p_J(e)\cdot(e-q)\right\}, \quad \forall\, s_0 = (l, q, \{J\}) \in \Psi,\ J \in \mathcal{N}\backslash l,\ l \in \mathcal{N}, q \in \mathcal{Q}^{fe}_1. \quad (28.3)$$

Equations (28.1)-(28.3) approximate the lower bounds for each state s_k ($s_k \in \Psi$, $s_k \in \mathrm{P_{s-a}}$) along with each possible next customer J ($J \in \mathcal{R}$). The derivation of (28.1)-(28.3) is based on (23.1)-(23.3) and (19.1). Values θ, α and ω are substituted into (23.1)-(23.3) to approximate each $f_{l,\mathcal{R}}(q)$ ($\forall\, l \in \mathcal{N}$, $\mathcal{R} \subseteq \mathcal{N}\backslash l$, $q \in Q^{fe}_{|\mathcal{R}|}$, $s \in \mathrm{P_{s-a}}$). Consequently, lower bound $L^J_{s_k}$ of $f^J_{l,\mathcal{R}}(q)$ is obtained by substituting the approximation of $f_{l,\mathcal{R}}(q)$ into (19.1). Finally, value function $V_k(l,q,\mathcal{R})$ ($s_k \in \Psi$, $s_k \in \mathrm{P_{s-a}}$) is approximated by substituting $L^J_{s_k}$ into (16).

6 | PRICE-DIRECTED POLICY

With approximated value functions $\tilde{V}_k(l,q,\mathcal{R}_k(l))$, given by (16), the price-directed (PD) policy can be derived. This policy determines next customer $j_k(l,q,\mathcal{R})$ and restocking decision $u_{j,l,\mathcal{R}}(q)$ for state $s_k = (l,q,\mathcal{R})$ ($s_k \in \Psi$, $k \in \Omega\backslash\{N,0\}$, $s_k \in \mathrm{P_{s-a}}$) based on (29.1)-(29.2).

$$u_{J,l,\mathcal{R}}(q) = \begin{cases} 1, \text{ if } L^{R(J)}_{s_k} \le L^{D(J)}_{s_k} \\ 0, \text{ if } L^{D(J)}_{s_k} > L^{R(J)}_{s_k} \end{cases}, \quad \forall\, J \in \mathcal{R},\, J \in \mathrm{P_{s-a}} \tag{29.1}$$

$$j_k(l,q,\mathcal{R}) = \arg \min_{J \in \mathcal{R}, J \in \mathrm{P_{s-a}}} \left\{ d_{lJ} + l^{J,\mathcal{R}\backslash J}_{tsp} + L^J_{s_k} \right\}, \quad \text{where } L^J_{s_k} = \min \left\{ L^{D(J)}_{s_k}, L^{R(J)}_{s_k} \right\} \tag{29.2}$$

At stage $k \in \{N-1, \ldots, 1\}$, for state $s_k = (l,q,\mathcal{R})$, (29.1) is first applied to determine the restocking decision for each potential next customer $J \in \mathcal{R}$. The optimal next customer $j_k(l,q,\mathcal{R})$ is then selected based on (29.2), and restocking decision $u_{j,l,\mathcal{R}}(q)$ is made accordingly, depending on the chosen next customer $j_k(l,q,\mathcal{R})$. Values $L^{R(J)}_{s_k}$ and $L^{D(J)}_{s_k}$ are computed as per (28.2) or (28.3), depending on whether restocking occurs at customer J ($u_{J,l,\mathcal{R}}(q) = 1$) or not ($u_{J,l,\mathcal{R}}(q) = 0$). Additionally, at the beginning of the routing, (30) determines the first customer to visit, $j_N(0,Q,\mathcal{N})$, when the vehicle departs from depot 0 with full capacity Q and proceeds to the first customer directly (i.e., $u_{j,0,\mathcal{N}}(Q) \equiv 0$). In this initial step, $L^J_{s_N}$ is obtained from (28.1). Finally, at final stage 0, the vehicle proceeds directly to the depot, meaning $j_0(l,q,\emptyset) = 0$ and no restocking occurs ($u_{0,l,\emptyset}(q) = 0$).

$$j_N(0,Q,\mathcal{N}) = \arg \min_{J \in \mathcal{N}, J \in \mathrm{P_{s-a}}} \left\{ d_{0J} + l^{J,\mathcal{N}\backslash J}_{tsp} + L^J_{s_N} \right\} \tag{30}$$

Note that observed states s and decisions for the next customers j are restricted by the promising state-action space $\mathrm{P_{s-a}}$ to ensure the tractability of the ALP formulation. Theoretically, our price-directed policy can obtain the near-optimal policy of the underlying MDP if the state-action space $\mathrm{P_{s-a}}$ is well-selected.

7 | COMPUTATIONAL STUDY

This section evaluates the performance of the proposed policy (PD). First, the problem settings are described through instance generation, followed by the setting of the outsourcing price. Next, the parameters of the proposed policy are tuned. After these preparations, computational studies are conducted to assess the policy's effectiveness from four aspects: (i) comparison with the policy that uses the traditional recourse action, (ii) comparison with other high-quality policies, (iii) evaluation under demand distributions with different degrees of variability, and (iv) analysis of the policy's performance under various outsourcing prices. The comparison with the traditional recourse strategy is conducted first to demonstrate the effectiveness of outsourcing as the recourse action. Subsequently, other policies are adapted to incorporate outsourcing, enabling a performance comparison across different policies. The experiments were conducted on a personal computer with an Intel Core 3.2 GHz processor and 16 GB RAM, using Gurobi as the LP solver.

7.1 | Instance generation and settings

Instance generation

Instances are generated following the instance generation scheme used in the VRPSD, as described in [48, 55]. Customer locations are randomly placed within a 1,000 × 1,000 grid, where each grid unit represents 1 meter, 0.5 meters, 5 meters, or other scales, depending on the region the depot serves. The depot is located at either (0, 0) or (500, 500), referred to as the

corner and *midpoint*, respectively. Distances between customers and between customers and the depot are calculated using Euclidean metrics. Customer demand follows a discrete uniform distribution with values from $\{1, 2, \ldots, 5\}$, with an average demand of 3 per customer. The problem size varies from 10 to 40 customers in increments of 5. Instances with fewer than 40 customers are emphasized to compare with other benchmark approaches and to assess computational difficulty in solving larger problems. This aligns with the instance sizes used by Toriello et al. [50]. The expected filling rate (or load factor) is computed as $\bar{f} = \sum_{l=1}^{N} E(\tilde{\xi}_l)/Q$, where $\bar{f}$ takes 1.9, 2.5 or 3.4 to represent different failure frequencies. Vehicle capacity Q is determined by rounding $3N/\bar{f}$ to the nearest integer. 20 instances are generated for each combination of settings. Additionally, customer demand is modeled using normal distribution in addition to the uniform distribution, to analyze the impact of demand variability on solution quality.

Setting of outsourcing price b

Outsourcing price b is determined based on two criteria. First, as explained in Appendix B.2, price b should incentivize restocking decisions when the vehicle's residual capacity is depleted (i.e., $q = 0$), preventing empty cruising and ensuring efficient deliveries with the assistance of other carriers. Second, outsourcing should offer a cost advantage over non-outsourcing, meaning that price b should be set to demonstrate the benefits of outsourcing compared to the traditional recourse strategy, where deliveries are handled either through collaboration between the vehicle and other carriers or solely by the vehicle.

The two criteria are given by $b \geq \frac{\max \Delta_{lj}}{\sum_e p_j(e) \cdot e}$ and $b \leq \frac{\min 2d_{0j}}{\sum_e p_j(e) \cdot e}$, with a detailed explanation in Appendix D. By setting outsourcing price b within range $[\frac{\max \Delta_{lj}}{\sum_e p_j(e) \cdot e}, \frac{\min 2d_{0j}}{\sum_e p_j(e) \cdot e}]$, the vehicle can restock upon failure while also potentially reducing costs through outsourcing. Note that range of price b is derived based on the cost structure specific to our problem context and assuming that other carriers undertake delivery tasks regardless of the outsourcing price. In the following, experiments are conducted by setting price b basically to $\frac{\max \Delta_{lj}}{\sum_e p_j(e) \cdot e}$, particularly for comparisons (i)–(iii). A sensitivity analysis is then performed for price b within the specified range.

Settings of PD policy

The PD policy is developed based on the multi-policy sampling framework. In practice, tiny adjustments are made to elicit better performance. Specifically, an action is taken if it can be obtained from the state-action pool. Namely, the next customer is only selected from those regarded as promising by the policy set. We also arbitrarily adjust the composition of policies in the set. For each instance, the best combination of policies is chosen to find the solution to our PD policy. The candidate policies are described in Appendix A. In our implementation, the states and relevant actions are generated by implementing each candidate heuristic to learn about each sample. A sample is a set of realized customer demands ($\{\xi\}^{sam}$, as defined in Section 5.3). To determine state-action pool P_{s-a}, 500 samples are generated for learning by each candidate heuristic. We observe numerical stability with 200 samples where there is no significant deviation in terms of the solution quality compared to the solutions obtained using 500 samples or even more. This observation is also in line with the findings reported in Secomandi [46], where 200 samples are considered appropriate. Nevertheless, we use 500 samples as our approach could scale to this number of samples without any noticeable performance drop.

7.2 | Solution quality of PD policy

Performances are evaluated in terms of solution quality and computational time against the traditional recourse policy and several benchmark approaches from the literature. In particular, the partial re-optimization method, implemented as PH(10) following [48], serves as the primary baseline for performance evaluation. This method is applied both as the policy under the traditional recourse action, denoted by PR^{trd}, and as the adapted policy with outsourcing, denoted by PR^{ous}. Additional benchmark policies include the one-step rollout algorithm (ORA) [46], the two-step rollout algorithm (TRA) [38], and the rollout algorithm (RA) [47], all adapted to address the VRPSD with outsourcing. The comparison with PR^{trd} assesses the benefit of incorporating outsourcing as the recourse action. Comparisons with ORA and TRA further illustrate the impact of different re-optimization strategies, while RA serves as a benchmark to evaluate the advantages of dynamic routing. In this study, PH(10) is used to implement PR^{trd} (respectively, PR^{ous}), balancing computational effort and solution quality. However, alternative implementations of PR^{trd} (respectively, PR^{ous}), such as SH(·) and PH(8), as discussed in [48], are also viable. Since our approach is based on the multi-policy sampling framework, the impact of changing the heuristic policy remains consistent. This framework integrates the solution spaces of multiple heuristic policies, ensuring that the inclusion of superior policies

ultimately leads to an enhanced overall policy. Furthermore, to comprehensively assess our approach, we conduct comparisons from the multiple perspectives: (i) against the traditional recourse policy without outsourcing, (ii) against several high-quality benchmark policies, (iii) under demand distributions with varying degrees of variability, and (iv) by altering the outsourcing price. These comparisons provide a multi-faceted evaluation of our approach's effectiveness.

7.2.1 | Comparison with traditional recourse strategy

Under the traditional recourse strategy, preventive restocking is implemented to avoid routing failures, and a detour trip to the depot for replenishment is performed if a failure occurs, as outlined in [48]. In contrast, our outsourcing strategy differs in that outsourcing is used to fulfill unmet demand when a failure arises. The comparison between these two strategies is conducted by evaluating total costs across different problem settings. In the experiments, PD and PR^{trd} represent our policy and the traditional policy, respectively. The adapted policy with outsourcing, denoted by PR^{ous}, is included as an additional benchmark. A direct comparison between PR^{trd} and PR^{ous} further illustrates the effect of changing the recourse action from traditional to outsourcing-based. The results reported in each row of Tables 1 and 2 present the average performance for each setting. Table 1 corresponds to scenarios where the depot is located at the midpoint, while Table 2 presents results for corner-located depots. A problem number marked with an asterisk (*) indicates that the proposed approach (PD) yields the best performance in that particular setting.

TABLE 1 Comparison between PD policy and traditional recourse policy (midpoint depot)

problem set No.	customers & capacity (N,Q)	method PD	method PR^{trd}	method PR^{ous}	$\gamma_{(PD:PR^{trd})}$	$\gamma_{(PR^{ous}:PR^{trd})}$	method PD		method PR^{trd}		b
							n_f	n_{res}	n_f	n_{res}	
*1	(5,8)	2741.21	2835.92	2743.25	-3.34%	-3.27%	0.75	1.1	0.7	0.95	81.73
*3	(10,16)	3036.52	3167.83	3036.52	-4.15%	-4.15%	0.6	1	0.3	1.2	53.51
*5	(15,24)	3973.66	4245.82	3973.66	-6.41%	-6.41%	0.53	0.67	0.25	1.35	55.78
*7	(20,24)	4396.17	4843.89	4396.17	-9.24%	-9.24%	4.6	1	0.65	1.45	18.08
9	(25,30)	5543.47	5473.97	5323.66	1.27%	-2.75%	0.6	2.3	0.40	1.9	81.22
*11	(30,36)	5774.68	6376.70	5774.68	-9.44%	-9.44%	3.4	2	0.2	2	16.12
*13	(35,31)	5704.40	5892.35	5647.74	-3.19%	-4.15%	0.4	3	0.4	2.6	178.13
15	(40,35)	7222.17	7079.67	7157.31	2.01%	1.10%	0.1	2.13	1	2.2	206.51
ave.	/	4799.04	4989.52	4756.62	-4.06%	-4.79%	1.36	1.65	0.49	1.71	86.39

TABLE 2 Comparison between PD policy and traditional recourse policy (corner depot)

problem set No.	customers & capacity (N,Q)	method PD	method PR^{trd}	method PR^{ous}	$\gamma_{(PD:PR^{trd})}$	$\gamma_{(PR^{ous}:PR^{trd})}$	method PD		method PR^{trd}		b
							n_f	n_{res}	n_f	n_{res}	
*2	(5,8)	4928.80	4949.69	4944.36	-0.42%	-0.11%	1	0.8	0.65	0.65	334.40
*4	(10,16)	4544.89	4558.38	4454.57	-0.30%	-2.28%	0.65	1.05	0.65	0.6	224.47
*6	(15,24)	5789.25	5881.41	5811.18	-1.57%	-1.19%	0.3	1.15	0.55	0.8	487.46
8	(20,24)	6886.04	6785.71	6846.61	1.48%	0.90%	0.5	1.85	0.6	1.35	219.01
*10	(25,30)	7844.87	7899.97	7744.82	-0.70%	-1.96%	0.75	1.9	0.35	1.8	245.82
12	(30,36)	8785.66	8454.09	8436.81	3.92%	-0.20%	0.25	1.95	0.4	1.65	269.75
*14	(35,31)	9188.57	9248.51	8889.82	-0.65%	-3.88%	0.4	3.2	1	2	421.40
*16	(40,35)	10738.64	11113.15	10738.64	-3.37%	-3.37%	0.53	5	0.62	1.35	455.12
ave.	/	7338.34	7361.36	7233.35	-0.20%	-1.51%	0.55	2.11	0.60	1.28	332.18

Tables 1 and 2 present a comparison between the PD policy and the traditional recourse policy for midpoint and corner depot scenarios. The term $\gamma_{(PD:PR^{trd})}$ (respectively, $\gamma_{(PR^{ous}:PR^{trd})}$) denotes the improvement rate of PD policy over PR^{trd} policy (respectively, PR^{ous} policy over PR^{trd} policy), calculated as $\gamma_{(PD:PR^{trd})}=\frac{V_N^{PD}(0,\mathcal{Q},\mathcal{N})-V_N^{PR^{trd}}(0,\mathcal{Q},\mathcal{N})}{V_N^{PR^{trd}}(0,\mathcal{Q},\mathcal{N})}$ and $\gamma_{(PR^{ous}:PR^{trd})}=\frac{V_N^{PR^{ous}}(0,\mathcal{Q},\mathcal{N})-V_N^{PR^{trd}}(0,\mathcal{Q},\mathcal{N})}{V_N^{PR^{trd}}(0,\mathcal{Q},\mathcal{N})}$. n_f and n_{res} represent the average number of failures and restocking events, respectively. The mean values in the last row are used to indicate overall performance from a general perspective. As shown in the tables, the PD policy generally outperforms the traditional recourse policy PR^{trd} in both the midpoint and corner depot scenarios. A similar superiority is observed when comparing the PR^{ous} policy to the PR^{trd} policy, highlighting the effectiveness of replacing traditional recourse actions with outsourcing. Due to the multi-policy sampling procedure, the PD policy integrates the solution space of PR^{ous} and other constituent policies, thereby inheriting their improved performance relative to the traditional PR^{trd} policy. Moreover, the tables confirm the appropriateness of the outsourcing price b, which plays a key role in enhancing the effectiveness of the PD policy.

Comparing the results in Tables 1 and 2, the PD policy demonstrates a more pronounced advantage over the traditional recourse policy in the midpoint depot scenario than in the corner depot scenario, with average cost improvements of -4.06% and -0.2%, respectively. Furthermore, in the corner depot scenario, the PD policy results in more frequent restocking trips than the PR^{trd} policy —2.11 times on average versus 1.28—whereas in the midpoint depot scenario, the PD policy involves fewer restocking trips, averaging 1.65 times compared to 1.71. This pattern is mainly attributed to the higher outsourcing price b in the corner depot scenario—averaging 332.18 compared to 86.39 in the midpoint depot scenario—which stems from the longer average travel distances required for external carriers to reach customers. The elevated outsourcing price is intended to attract a sufficient number of external carriers despite the increased distance. Consequently, to minimize total costs, the private vehicle relies more on restocking than on outsourcing in the corner depot scenario. This is further evidenced by the higher ratio $\frac{n_{res}}{n_f}$, which equals $\frac{2.11}{0.55}$ in the corner scenario compared to $\frac{1.65}{1.36}$ in the midpoint scenario for the PD policy. In addition, since n_f also represents the average number of outsourcing instances, the tables reveal that outsourcing occurs more frequently in the midpoint depot scenario than in the corner depot scenario—1.36 times versus 0.55 times on average—due to the significantly lower outsourcing price in the former. This also confirms that the outsourcing price is reasonably set to accommodate different routing scenarios and reduce costs accordingly.

7.2.2 | Comparison with other high-quality policies

To evaluate solution quality, improvement rates $\gamma^{PR^{ous}}$, γ^{RA}, γ^{ORA}, and γ^{TRA} are introduced. These metrics represent the percentage improvements of the proposed policy PD over the benchmark approaches: $\gamma^{PR^{ous}}=\frac{V_N^{PD}(0,\mathcal{Q},\mathcal{N})-V_N^{PR^{ous}}(0,\mathcal{Q},\mathcal{N})}{V_N^{PR^{ous}}(0,\mathcal{Q},\mathcal{N})}$ compares PD with PR^{ous}, $\gamma^{RA}=\frac{V_N^{PD}(0,\mathcal{Q},\mathcal{N})-V_N^{RA}(0,\mathcal{Q},\mathcal{N})}{V_N^{RA}(0,\mathcal{Q},\mathcal{N})}$ compares PD with RA, $\gamma^{ORA}=\frac{V_N^{PD}(0,\mathcal{Q},\mathcal{N})-V_N^{ORA}(0,\mathcal{Q},\mathcal{N})}{V_N^{ORA}(0,\mathcal{Q},\mathcal{N})}$ compares PD with ORA, and $\gamma^{TRA}=\frac{V_N^{PD}(0,\mathcal{Q},\mathcal{N})-V_N^{TRA}(0,\mathcal{Q},\mathcal{N})}{V_N^{TRA}(0,\mathcal{Q},\mathcal{N})}$ compares PD with TRA. These improvement rates are computed based on the actual costs obtained from implementing policies PD, PR^{ous}, RA, ORA, and TRA across various problem settings.

TABLE 3 Total costs based on different approaches (midpoint depot)

problem set No.	customers & capacity (N,Q)	method PD	method PR^{ous}	method TRA	method ORA	method RA	rate $\gamma^{PR^{ous}}$	rate γ^{TRA}	rate γ^{ORA}	rate γ^{RA}
*1	(5,8)	2741.21	2743.25	2782.57	2772.93	2772.93	-0.07%	-1.49%	-1.14%	-1.14%
*3	(10,16)	3036.52	3036.52	3195.74	3210.07	3210.07	0.00%	-4.98%	-5.41%	-5.41%
*5	(15,24)	3973.66	3973.66	4248.38	4310.17	4310.17	0.00%	-6.47%	-7.81%	-7.81%
*7	(20,24)	4396.17	4396.17	4620.66	4739.61	4588.26	0.00%	-4.86%	-7.25%	-4.19%
9	(25,30)	5543.47	5323.66	5667.88	5689.24	5605.19	4.13%	-2.19%	-2.56%	-1.10%
*11	(30,36)	5774.68	5774.68	6541.33	6487.25	6487.25	0.00%	-11.72%	-10.98%	-10.98%
13	(35,31)	5704.40	5647.74	6002.20	6041.87	6111.47	1.00%	-4.96%	-5.59%	-6.66%
15	(40,35)	7222.17	7157.31	7285.36	7277.32	7277.32	0.91%	-0.87%	-0.76%	-0.76%

TABLE 4 Total costs based on different approaches (corner depot)

problem set No.	customers & capacity (N,Q)	method PD	method PR^{ous}	method TRA	method ORA	method RA	rate $\gamma^{PR^{ous}}$	rate γ^{TRA}	rate γ^{ORA}	rate γ^{RA}
*2	(5,8)	4928.80	4944.36	5187.68	4993.20	4994.97	-0.31%	-4.99%	-1.29%	-1.32%
4	(10,16)	4544.89	4454.57	4793.03	4774.27	4712.28	2.03%	-5.18%	-4.80%	-3.55%
*6	(15,24)	5789.25	5811.18	5725.10	5842.54	5842.54	-0.38%	1.12%	-0.91%	-0.91%
8	(20,24)	6886.04	6846.61	7553.71	7826.78	7695.17	0.58%	-8.84%	-12.02%	-10.51%
10	(25,30)	7844.87	7744.82	7726.64	7734.12	7793.28	1.29%	1.53%	1.43%	0.66%
12	(30,36)	8785.66	8436.81	8855.31	8949.21	8949.21	4.13%	-0.79%	-1.83%	-1.83%
14	(35,31)	9188.57	8889.82	9108.04	9191.13	9191.13	3.36%	0.88%	-0.03%	-0.03%
*16	(40,35)	10738.64	10738.64	10890.01	10830.70	10795.86	0.00%	-1.39%	-0.85%	-0.53%

As indicated in Tables 3 and 4, the PD policy consistently outperforms several benchmark policies across most problem settings. Furthermore, it demonstrates a small performance gap—within 5%—relative to the primary benchmark, the PR^{ous} policy. Notably, in specific problem settings such as 1, 2, and 6, the PD policy exhibits superior performance over the PR policy. This result is not unexpected, given that our solution framework is based on multi-policy sampling. As described in Appendix A, the selected policies are composed from a diverse set of heuristics, each capable of identifying promising regions of the solution space. Moreover, several of these policies are adapted versions of their original forms, further enhanced through value function decomposition (also detailed in Appendix A). These composite policies facilitate broader exploration of the solution space, thereby increasing the likelihood of discovering higher-quality solutions. A particularly noteworthy case arises in problem settings 1 and 2 with 5 customers, where our PD policy achieves the best results among all evaluated policies. Under the PR^{ous} policy framework, implementing PH(10) could yield a high-quality or near-optimal PR^{ous} (re-optimization) policy especially for instances with a small number of customers[†]. However, in our experiments, the PD policy yields even lower total costs in these instances. We attribute this improvement to the integration of policies generated under different decision sequences (see Appendix A), which effectively expands the accessible solution space within a given sequence and leads to superior outcomes.

In summary, the multi-policy sampling procedure integrates the solution spaces of multiple policies, enhancing exploration and increasing the potential to outperform any individual policy. In addition, the PD policy generally outperforms the two dynamic routing policies (ORA and TRA) as well as the fixed routing policy (RA), demonstrating its superior performance. Only a few problem settings with inferior performance are observed—such as problem setting 6—which may partially explain why the PD policy can outperform the primary benchmark, the PR^{ous} policy. In fact, this also suggests that improved selection of

[†] For instances with 5 customers, the PH(10) approach is equivalent to dynamic programming. As illustrated in [48], each block executes a full dynamic programming procedure, ensuring that the optimal re-optimization solution within that block is obtained exactly. Since the block size in PH(10) is 10—sufficient to cover all 5 customers—the optimal re-optimization solution can be fully achieved using PH(10). For further details, please refer to [48].

constraints during the constraint sampling procedure could enhance the performance of the ALP approach, potentially leading to solutions that are much closer to optimal. Overall, the results presented in the tables indicate that the PD policy is both viable and promising for addressing routing problems with stochastic demands. The computational time required to solve the problem under different settings is discussed below.

TABLE 5 CPU times of different approaches in seconds (midpoint depot)

problem set No.	customers & capacity (N,Q)	method PD				method PR^{ous}	method TRA	method ORA	method RA	time(PD)/ time(PR^{ous})
		prep.	alp.	imple.	total					
1	(5,8)	12.32	3.34	0.07	15.72	11.22	6.26	1.29	1.03	140.11%
3	(10,16)	8419.98	49.78	0.14	8469.90	8419.26	323.32	40.22	33.90	100.60%
5	(15,24)	20412.24	43.77	0.43	20456.44	20411.55	1533.24	216.16	194.30	100.22%
7	(20,24)	36435.41	39.91	0.08	36475.41	36437.17	2284.69	475.21	446.45	100.10%
9	(25,30)	60379.85	676.04	0.39	61056.28	60383.66	4630.82	317.76	260.68	101.11%
11	(30,36)	176981.58	242.45	0.51	177224.54	176988.51	8524.99	2458.64	2361.45	100.13%
13	(35,31)	196215.97	561.35	0.68	196778.01	196401.85	33199.56	3805.35	2954.90	100.19%
15	(40,35)	458787.72	954.63	0.35	459742.70	458850.68	82559.44	5732.28	4278.53	100.19%

TABLE 6 CPU times of different approaches in seconds (corner depot)

problem set No.	customers & capacity (N,Q)	method PD				method PR^{ous}	method TRA	method ORA	method RA	time(PD)/ time(PR^{ous})
		prep.	alp.	imple.	total					
2	(5,8)	11.64	4.05	0.07	15.76	11.24	6.38	1.38	1.17	140.21%
4	(10,16)	8652.06	91.86	0.50	8744.42	8652.38	314.97	40.20	34.22	101.06%
6	(15,24)	19906.02	35.37	0.10	19941.49	19907.41	1371.64	115.94	90.69	100.17%
8	(20,24)	40282.03	67.12	0.26	40349.42	40287.50	2722.06	640.48	594.77	100.15%
10	(25,30)	60553.16	978.78	0.13	61532.06	60557.26	4554.14	326.12	261.19	101.61%
12	(30,36)	110713.50	1103.38	0.22	111817.10	110721.72	10379.55	2742.78	2627.08	100.99%
14	(35,31)	209853.34	607.83	0.28	210461.45	209963.38	34520.63	4200.87	3125.39	100.24%
16	(40,35)	489653.56	938.13	0.38	490592.07	489735.71	91008.32	5842.91	4389.39	100.17%

The total time for the PD policy consists of three components: the pre-compute time (prep.), the time for solving the ALP formulation (alp.), and the implementation time (imple.). The prep. time primarily refers to the longest offline training time among the heuristic policies selected in the multi-policy sampling procedure. This time is recorded based on the parallel computation times of the various heuristic policies involved in the procedure (see Appendix A). The alp. time includes both the time required to generate constraints for the ALP formulation and the time to solve it using the Gurobi solver. As shown in the tables, alp. times are significantly smaller than the prep. times. Therefore, imposing a time limit on solving the ALP formulation has little effect on reducing the overall computational time, and thus no such limit is enforced. The imple. time refers to the average time required to apply the PD policy to solve a specific problem instance. If the prep. and alp. times are considered as offline computation, then the imple. time can be viewed as online computation. As a consequence, the PD policy can also potentially be suitable for real-time decision-making since part of its computation time can be done offline.

As shown in Tables 5 and 6, the computational effort required to obtain a solution using the PD policy is approximately equal to that of the PR^{ous} policy. In contrast, the computation times for the TRA, ORA, and RA policies are significantly smaller

than those for the PD and PR^{ous} policies. By combining the performance results from Tables 3 and 4, it is evident that better performance often comes at the cost of increased computational complexity, which aligns with intuition.

7.2.3 | Solution quality under demand distributions with different variations

To account for varying customer demand patterns, experiments were also conducted under normally distributed demand in addition to uniformly distributed demand. These different distributions allow for analyzing the impact of demand variability on solution quality, while maintaining a consistent demand range and mean across all cases. Specifically, the uniform distribution—with a variance-to-mean ratio of 0.67—represents a low level of demand variability and was used in the experiments discussed above. In contrast, the normal distribution—with a variance-to-mean ratio of 3.66—captures a high level of variability.

TABLE 7 PD policy performance under demand distributions with different variations (midpoint depot)

problem set No.	customers & capacity (N,Q)	uniform (low variability)					normal (high variability)				
		method PD	$\gamma^{PR^{ous}}$	$\gamma_{(PD:PR^{trd})}$	n_{res}	n_f	method PD	$\gamma^{PR^{ous}}$	$\gamma_{(PD:PR^{trd})}$	n_{res}	n_f
1	(5,8)	2741.21	-0.07%	-3.34%	1.1	0.75	2735.22	0.65%	-3.53%	0.9	0.95
3	(10,16)	3036.52	0.00%	-4.15%	1	0.6	3024.75	0.00%	-2.64%	1	0.5
5	(15,24)	3973.66	0.00%	-6.41%	0.67	0.53	3996.02	0.00%	-3.56%	1.00	0.45
7	(20,24)	4396.17	0.00%	-9.24%	1	4.6	4425.09	0.17%	-6.02%	1	5
9	(25,30)	5543.47	4.13%	1.27%	2.3	0.6	5251.08	0.00%	-0.62%	2.1	0
11	(30,36)	5774.68	0.00%	-9.44%	2	3.4	5740.49	0.00%	-6.51%	2	2.2
13	(35,31)	5704.40	1.00%	-3.19%	3	0.4	5759.84	0.66%	0.69%	3	1
15	(40,35)	7157.31	0.91%	2.01%	2.13	0.1	7102.75	0.83%	1.95%	3	0.2
ave.	/	4790.93	0.75%	-4.06%	1.65	1.36	4754.41	0.29%	-2.53%	1.75	1.29

TABLE 8 PD policy performance under demand distributions with different variations (corner depot)

problem set No.	customers & capacity (N,Q)	uniform (low variability)					normal (high variability)				
		method PD	$\gamma^{PR^{ous}}$	$\gamma_{(PD:PR^{trd})}$	n_{res}	n_f	method PD	$\gamma^{PR^{ous}}$	$\gamma_{(PD:PR^{trd})}$	n_{res}	n_f
2	(5,8)	4928.80	-0.31%	-0.42%	0.8	1	4842.55	-0.86%	-2.97%	0.95	0.85
4	(10,16)	4544.89	2.03%	-0.30%	1.05	0.65	4518.59	2.10%	2.26%	1.05	0.65
6	(15,24)	5789.25	-0.38%	-1.57%	1.15	0.3	5723.40	-0.18%	-2.10%	1.05	0.35
8	(20,24)	6886.04	0.58%	1.48%	1.85	0.5	6759.99	0.56%	0.57%	2	0.3
10	(25,30)	7844.87	1.29%	-0.70%	1.9	0.75	7689.66	-1.08%	-1.43%	2	0.2
12	(30,36)	8785.66	4.13%	3.92%	1.95	0.25	8592.16	-0.05%	-0.05%	2.6	0
14	(35,31)	9188.57	3.36%	-0.65%	3.2	0.4	9031.45	2.28%	-0.30%	3.5	0.3
16	(40,35)	10738.64	0.00%	-3.37%	5	0.53	10563.89	0.00%	-0.83%	5	0.23
ave.	/	7338.34	1.34%	-0.20%	2.11	0.55	7215.21	0.35%	-0.61%	2.27	0.36

In Tables 7 and 8, $\gamma^{PR^{ous}}$ denotes the improvement rate of policy PD over policy PR^{ous}, and $\gamma_{(PD:PR^{trd})}$ represents the improvement rate of policy PD over policy PR^{trd}. As shown in the tables, policy PD generally performs better under higher demand variability in both the midpoint and corner depot scenarios, with average total costs of 4750.41 vs. 4790.93 in the midpoint scenario and

7215.21 vs. 7338.34 in the corner scenario. This demonstrates the consistent adaptability of the proposed approach to increased demand variability. Specifically, higher restocking rates are observed under high demand variability—averaging 1.75 vs. 1.65 in the midpoint scenario and 2.27 vs. 2.11 in the corner scenario—which lead to fewer failures (averaging 1.29 vs. 1.36 in the midpoint scenario and 0.36 vs. 0.55 in the corner scenario). These findings further highlight the robustness and adaptability of our approach in handling high demand variability.

In fact, as shown in Tables 7 and 8, policy PD exhibits no apparent difference in performance across different levels of demand variability, as indicated by improvement rate $\gamma^{\mathrm{PR^{ous}}}$, which remains within an absolute margin of 1.35% on average across various problem settings. A similar observation was also reported in [20]. In their computational study, the authors introduced three probability distributions to generate demand variability at different levels and concluded that cost savings arise primarily from adopting a superior recourse scheme, rather than from changes in demand variability. This suggests that any demand distribution can reliably represent the performance of the proposed approach. Additionally, they observed that savings are generally higher when demand variability is low, particularly in the corner depot scenario. In contrast, our policy performs better under high demand variability. This divergence can be attributed to a key difference in model design—namely, the explicit inclusion of outsourcing prices. In our model, effective pricing decisions for outsourcing play an essential role in reducing overall costs, especially when demand is more uncertain and customer locations are farther from the depot.

7.2.4 | Sensitivity analysis of outsourcing price

The outsourcing price fluctuates to assess its impact on total costs and decision-making. As explained in the experimental setup and in Appendix D, which define the outsourcing price and its range, the experiment is conducted by varying the price within this range. Specifically, b^0 represents the base price used in the previous experiments, while $b^{\max}$ denotes the maximum price within the range. Additional prices are tested at increments of $0.3 * (b^{\max} - b^0)$ between b^0 and $b^{\max}$, meaning that $b^1 = b^0 + 0.3 * (b^{\max} - b^0)$ and $b^2 = b^0 + 0.6 * (b^{\max} - b^0)$ are used in the experiment. To evaluate the impact of outsourcing price on decision-making, the ratio of restocking to outsourcing is analyzed. Additionally, the total costs under different price settings are examined to assess the price's effect on overall costs. The experiment is conducted under the uniform demand distribution.

TABLE 9 Sensitivity analysis of outsourcing price (midpoint depot)

problem set No.	customers & capacity (N,Q)	b^0	$b^{\max}$	b^0		b^1		b^2		$b^{\max}$	
				PD	n_{res}/n_f	PD	n_{res}/n_f	PD	n_{res}/n_f	PD	n_{res}/n_f
1	(5,8)	81.73	91.43	2741.21	1.47	2746.89	1.47	2750.52	1.47	2755.37	1.47
3	(10,16)	53.51	73.41	3036.52	1.67	3043.69	1.67	3050.86	1.67	3043.69	1.67
5	(15,24)	55.78	150.55	3973.66	1.26	4027.28	1.25	4043.37	2.20	4071.46	2.20
7	(20,24)	18.08	78.32	4396.17	0.22	4603.69	0.21	4679.79	0.79	4741.54	2.06
9	(25,30)	81.22	225.18	5543.47	3.83	5364.69	5.50	5428.47	5.22	5663.80	11.75
11	(30,36)	16.12	29.47	5774.68	0.59	5826.51	0.63	5876.04	0.70	5917.17	0.70
13	(35,31)	178.13	224.75	5704.40	7.5	5759.41	8.00	5773.39	8.00	5808.03	8.00
15	(40,35)	206.51	255.19	7157.31	21.30	7203.83	23.00	7256.71	23.00	7348.24	25.00
ave.	/	86.39	141.04	4790.93	4.73	4822.00	5.22	4857.39	5.38	4918.66	6.61

TABLE 10 Sensitivity analysis of outsourcing price (corner depot)

problem set No.	customers & capacity (N,Q)	b^0	$b^{\max}$	b^0		b^1		b^2		$b^{\max}$	
				PD	n_{res}/n_f	PD	n_{res}/n_f	PD	n_{res}/n_f	PD	n_{res}/n_f
2	(5,8)	342.66	361.96	4928.80	0.80	4908.85	2.67	4927.45	2.67	4913.24	2.67
4	(10,16)	224.47	455.58	4544.89	1.62	4523.84	3.29	4543.23	2.33	4528.66	3.14
6	(15,24)	487.46	548.75	5789.25	3.83	5824.05	3.43	5836.92	3.29	5854.08	4.60
8	(20,24)	219.01	479.16	6886.04	3.70	6844.48	7.80	6873.47	20.00	6871.26	20.00
10	(25,30)	245.82	480.17	7844.87	2.53	7840.73	8.00	7868.85	10.00	8070.29	13.67
12	(30,36)	269.75	538.20	8785.66	7.8	8789.99	8.70	8819.98	10.25	8912.72	10.5
14	(35,31)	421.40	462.82	9188.57	8	9189.51	11.17	9217.85	13.60	9314.78	17.5
16	(40,35)	455.12	496.26	10738.64	9.43	10745.53	16.65	10765.43	18.43	10849.89	26.50
ave.	/	333.21	477.86	7338.34	4.71	7333.37	7.71	7356.65	10.07	7414.37	12.32

Generally, as shown in Tables 9 and 10, the overall cost increases with the outsourcing price in the midpoint and corner depot scenarios. This trend is particularly evident in the last row of the tables, which reports the average costs. These results complement the findings in Tables 7 and 8, highlighting the critical role of outsourcing price in achieving cost savings. With adaptive outsourcing prices and more frequent triggering of restocking actions (reflected by increased n_{res}/n_f), cost savings are achieved as the outsourcing price increases. In addition, the observed variations in the trends may be attributed to the determination of the outsourcing price range $[b^0, b^{\max}]$, which is generally defined as $[\frac{\max \Delta lj}{\sum_e p_j(e)\cdot e}, \frac{\min 2d_{0j}}{\sum_e p_j(e)\cdot e}]$. However, in some problem settings, the lower bound $\frac{\max \Delta lj}{\sum_e p_j(e)\cdot e}$ exceeds the upper bound $\frac{\min 2d_{0j}}{\sum_e p_j(e)\cdot e}$, causing the range to collapse. In such cases, artificial adjustments must be introduced, which may lead to less consistent or less observable trends. This highlights that the determination of the outsourcing price is a critical issue that warrants further investigation. A more accurate and robust pricing strategy could enable more effective recourse policies. In this study, initial efforts have been made to address the outsourcing price, as reflected in the parameter setting and detailed in Appendix D. However, further work is needed to explore pricing strategies more comprehensively. This may lead to a separate study specifically focused on the pricing problem within the context of the current problem setting, which lies beyond the scope of this paper.

8 | CONCLUSIONS

The paper investigates the routing problem for a private vehicle supported by auxiliary carriers under stochastic customer demands. It proposes a strategy that leverages outsourcing to maintain route feasibility, motivated by the rise of the sharing economy in the shipping industry. The strategy dynamically updates the vehicle's routing and restocking decisions based on the observed state at each customer arrival along the route. To formulate the strategy, a Markov decision process (MDP) is developed, with a key difference of altering the sequence of routing and restocking decisions at each stage. This change enables the identification of the problem's inherent structure, facilitating the development of a decomposition-based value function approximation approach. An approximate linear programming (ALP) approach based on value function decomposition is thus proposed to compute the strategy, providing a novel and viable solution pathway for addressing stochastic demands in vehicle routing problems. Several techniques tailored to the problem context are introduced to ease computation. In particular, a constraint sampling procedure, termed the multi-policy sampling framework, is developed to manage the intractable constraints in solving the ALP formulation. The multi-policy sampling framework employs a set of selected heuristic policies to mimic an ideal policy, thereby identifying promising constraints in the ALP formulation and enabling the computation of near-optimal solutions.

A comprehensive computational study is conducted to evaluate the effectiveness of the proposed approach from multiple perspectives. Evaluations against the traditional recourse strategy clearly demonstrate the cost-saving advantages of the proposed method. Comparisons with other high-quality solution methods indicate that the proposed method has the potential to generate solutions close to the optimum within a reasonable computational time. Tests under various demand distributions with differing levels of variability reveal that the impact of demand variability on cost savings is relatively minor, and the experimental results

based on either demand distribution can reliably represent the performance of the proposed approach. Furthermore, analysis under fluctuating outsourcing prices suggests a general trend: overall costs tend to increase as prices rise in the midpoint and corner depot scenarios. This analysis also highlights the need for dedicated research focused on the pricing problem for further investigation.

Research avenues exist to enhance the proposed ALP method. In particular, addressing two essential elements—the choice of basis functions and the state-relevance distribution—can significantly improve the solution methodology. These two elements are critical for minimizing the error in value function approximation [41], yet they remain highly challenging to address. It is encouraging to see growing attention to ALP methods in recent years, along with notable advances such as those in [32, 41, 36, 54, 37], where machine learning techniques and algorithmic enhancements from computational sciences and operations research are integrated to improve ALP performance. Future efforts could focus on incorporating these recent developments into our solution framework to further strengthen its effectiveness. With these advancements, a viable and promising solution method for solving routing problems near-optimally appears to be on the horizon.

ACKNOWLEDGMENTS

This work was supported by the China Scholarship Council under Grant 201908310007, for which the authors express their sincere gratitude. They also thank the three anonymous referees and the editors for their insightful comments and suggestions, which significantly improved this paper.

FINANCIAL DISCLOSURE

None reported.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

REFERENCES

1. D. Adelman, *Price-directed replenishment of subsets: Methodology and its application to inventory routing*, Manufacturing & Service Operations Management **5** (2003), no. 4, 348–371.
2. D. Adelman, *A price-directed approach to stochastic inventory/routing*, Operations Research **52** (2004), no. 4, 499–514.
3. D. Adelman, *Dynamic bid prices in revenue management*, Operations Research **55** (2007), no. 4, 647–661.
4. D. Adelman and C. Barz, *A price-directed heuristic for the economic lot scheduling problem*, IIE Transactions **46** (2014), no. 12, 1343–1356.
5. C. Archetti, M. Savelsbergh, and M. G. Speranza, *The vehicle routing problem with occasional drivers*, European Journal of Operational Research **254** (2016), no. 2, 472–480.
6. A. M. Arslan, N. Agatz, L. Kroon, and R. Zuidwijk, *Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers*, Transportation Science **53** (2019), no. 1, 222–235.
7. A. C. Baller, S. Dabia, W. E. Dullaert, and D. Vigo, *The vehicle routing problem with partial outsourcing*, Transportation Science **54** (2020), no. 4, 1034–1052.
8. C. Barz and K. Rajaram, *Elective patient admission and scheduling under multiple resource constraints*, Production and Operations Management **24** (2015), no. 12, 1907–1930.
9. D. Blado and A. Toriello, *Relaxation analysis for the dynamic knapsack problem with stochastic item sizes*, SIAM Journal on Optimization **29** (2019), no. 1, 1–30.
10. C.-W. Chu, *A heuristic algorithm for the truckload and less-than-truckload problem*, European Journal of Operational Research **165** (2005), no. 3, 657–667.
11. H. Crowder and M. W. Padberg, *Solving large-scale symmetric travelling salesman problems to optimality*, Management Science **26** (1980), no. 5, 495–509.
12. S. Dabia, D. Lai, and D. Vigo, *An exact algorithm for a rich vehicle routing problem with private fleet and common carrier*, Transportation Science **53** (2019), no. 4, 986–1000.
13. L. Dahle, H. Andersson, and M. Christiansen, *The vehicle routing problem with dynamic occasional drivers*, *International conference on computational logistics*, Springer, 2017, 49–63.
14. L. Dahle, H. Andersson, M. Christiansen, and M. G. Speranza, *The pickup and delivery problem with time windows and occasional drivers*, Computers & Operations Research **109** (2019), 122–133.
15. I. Dayarian and M. Savelsbergh, *Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders*, Production and Operations Management **29** (2020), no. 9, 2153–2174.
16. D. P. De Farias and B. Van Roy, *The linear programming approach to approximate dynamic programming*, Operations Research **51** (2003), no. 6, 850–865.
17. D. P. De Farias and B. Van Roy, *On constraint sampling in the linear programming approach to approximate dynamic programming*, Mathematics of Operations Research **29** (2004), no. 3, 462–478.
18. Ele.me, *Ele.me delivery service*, https://www.ele.me/ (2025). Accessed: 2025-04-08.
19. V. F. Farias and B. Van Roy, *Tetris: A study of randomized constraint sampling*, *Probabilistic and randomized methods for design under uncertainty*, Springer, 2006. 189–201.
20. A. M. Florio, D. Feillet, M. Poggi, and T. Vidal, *Vehicle routing with stochastic demands and partial reoptimization*, Transportation Science **56** (2022), no. 5, 1393–1408.

21. A. M. Florio, M. Gendreau, R. F. Hartl, S. Minner, and T. Vidal, *Recent advances in vehicle routing with stochastic demands: Bayesian learning for correlated demands and elementary branch-price-and-cut*, European Journal of Operational Research **306** (2023), no. 3, 1081–1093.
22. R. Fukasawa, A. S. Barboza, and A. Toriello, *On the strength of approximate linear programming relaxations for the traveling salesman problem*, Available online: www2. isye. gatech. edu/~ atoriello3/bcpalp. pdf (accessed on 30 June 2025) (2016).
23. K. Gdowska, A. Viana, and J. P. Pedroso, *Stochastic last-mile delivery with crowdshipping*, Transportation Research Procedia **30** (2018), 90–100.
24. M. Gendreau, O. Jabali, and W. Rei, *50th anniversary invited article–future research directions in stochastic vehicle routing*, Transportation Science **50** (2016), no. 4, 1163–1173.
25. J. C. Goodson, J. W. Ohlmann, and B. W. Thomas, *Rollout policies for dynamic solutions to the multivehicle routing problem with stochastic demand and duration limits*, Operations Research **61** (2013), no. 1, 138–154.
26. J. C. Goodson, B. W. Thomas, and J. W. Ohlmann, *Restocking-based rollout policies for the vehicle routing problem with stochastic demand and duration limits*, Transportation Science **50** (2016), no. 2, 591–607.
27. M. Grötschel and O. Holland, *Solution of large-scale symmetric travelling salesman problems*, Mathematical Programming **51** (1991), no. 1, 141–202.
28. J. Hao and J. B. Orlin, *A faster algorithm for finding the minimum cut in a directed graph*, Journal of Algorithms **17** (1994), no. 3, 424–446.
29. O. Jabali, W. Rei, M. Gendreau, and G. Laporte, *Partial-route inequalities for the multi-vehicle routing problem with stochastic demands*, Discrete Applied Mathematics **177** (2014), 121–136.
30. M. Joerss, F. Neuhaus, and J. Schröder, *How customer demands are reshaping last-mile delivery*, https://www.mckinsey.com/industries/logistics/our-insights/how-customer-demands-are-reshaping-last-mile-delivery (2016). (accessed April 7, 2025).
31. S. Kunnumkal and H. Topaloglu, *Computing time-dependent bid prices in network revenue management problems*, Transportation Science **44** (2010), no. 1, 38–62.
32. Q. Lin, S. Nadarajah, and N. Soheili, *Revisiting approximate linear programming: Constraint-violation learning with applications to inventory control and energy storage*, Management Science **66** (2020), no. 4, 1544–1562.
33. F. V. Louveaux and J.-J. Salazar-González, *Exact approach for the vehicle routing problem with stochastic demands and preventive returns*, Transportation Science **52** (2018), no. 6, 1463–1478.
34. G. Macrina, L. D. P. Pugliese, F. Guerriero, and G. Laporte, *Crowd-shipping with time windows and transshipment nodes*, Computers & Operations Research **113** (2020), 104806.
35. Meituan, *Meituan delivery platform*, https://waimaie.meituan.com/ (2025). Accessed: 2025-04-08.
36. S. Nadarajah and A. A. Cire, *Network-based approximate linear programming for discrete optimization*, Operations Research **68** (2020), no. 6, 1767–1786.
37. S. Nadarajah and A. A. Cire, *Self-adapting network relaxations for weakly coupled markov decision processes*, Management Science **71** (2025), no. 2, 1779–1802.
38. C. Novoa and R. Storer, *An approximate dynamic programming approach for the vehicle routing problem with stochastic demands*, European Journal of Operational Research **196** (2009), no. 2, 509–515.
39. C. Novoa, R. Berger, J. Linderoth, and R. Storer, *A set-partitioning-based model for the stochastic vehicle routing problem*, 06T-008, Lehigh University, 2006. Available at https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=21c731ac7c4dc67a0f6538fa2c7a47333ad77aa4.
40. M. Padberg and G. Rinaldi, *An efficient algorithm for the minimum capacity cut problem*, Mathematical Programming **47** (1990), no. 1, 19–36.
41. P. Pakiman, S. Nadarajah, N. Soheili, and Q. Lin, *Self-guided approximate linear programs: randomized multi-shot approximation of discounted cost markov decision processes*, Management science **71** (2025), no. 4, 3384–3404.
42. H. N. Psaraftis, M. Wen, and C. A. Kontovas, *Dynamic vehicle routing problems: Three decades and counting*, Networks **67** (2016), no. 1, 3–31.
43. M. Salavati-Khoshghalb, M. Gendreau, O. Jabali, and W. Rei, *A rule-based recourse for the vehicle routing problem with stochastic demands*, Transportation Science **53** (2019a), no. 5, 1334–1353.
44. M. Salavati-Khoshghalb, M. Gendreau, O. Jabali, and W. Rei, *A hybrid recourse policy for the vehicle routing problem with stochastic demands*, EURO Journal on Transportation and Logistics **8** (2019b), no. 3, 269–298.
45. N. Secomandi, *Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands*, Computers & Operations Research **27** (2000), no. 11-12, 1201–1225.
46. N. Secomandi, *A rollout policy for the vehicle routing problem with stochastic demands*, Operations Research **49** (2001), no. 5, 796–802.
47. N. Secomandi, *Analysis of a rollout approach to sequencing problems with stochastic routing applications*, Journal of Heuristics **9** (2003), no. 4, 321–352.
48. N. Secomandi and F. Margot, *Reoptimization approaches for the vehicle-routing problem with stochastic demands*, Operations Research **57** (2009), no. 1, 214–230.
49. C. Tong and H. Topaloglu, *On the approximate linear programming approach for network revenue management problems*, INFORMS Journal on Computing **26** (2014), no. 1, 121–134.
50. A. Toriello, W. B. Haskell, and M. Poremba, *A dynamic traveling salesman problem with stochastic arc costs*, Operations Research **62** (2014), no. 5, 1107–1125.
51. F. Torres, M. Gendreau, and W. Rei, *Vehicle routing with stochastic supply of crowd vehicles and time windows*, Transportation Science **56** (2022), no. 3, 631–653.
52. M. W. Ulmer, J. C. Goodson, D. C. Mattfeld, and M. Hennig, *Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests*, Transportation Science **53** (2019), no. 1, 185–202.
53. W.-H. Yang, K. Mathur, and R. H. Ballou, *Stochastic vehicle routing problem with restocking*, Transportation Science **34** (2000), no. 1, 99–112.
54. R. Zhang, S. Samiedaluie, and D. Zhang, *Product-based approximate linear programs for network revenue management*, Operations Research **70** (2022), no. 5, 2837–2850.
55. L. Zhu and J.-B. Sheu, *Failure-specific cooperative recourse strategy for simultaneous pickup and delivery problem with stochastic demands*, European Journal of Operational Research **271** (2018), no. 3, 896–912.
56. L. Zhu, L.-M. Rousseau, W. Rei, and B. Li, *Paired cooperative reoptimization strategy for the vehicle routing problem with stochastic demands*, Computers & Operations Research **50** (2014), 1–13.

□

APPENDIX

A POLICY SET IN MULTI-POLICY SAMPLING FRAMEWORK

A set of policies is prepared to generate the price-directed policy. These policies determine the state-action pairs and the sampled constraints. The policy set comprises the following candidates: two PR^{ous}-type policies, two ORA-type policies, two TRA-type policies, a category of a priori optimization policies, a myopic policy, and the PR^{trd} policy under traditional recourse action.

In the candidate policies, the value functions are either computed originally as their methods indicate or based on the decomposition framework as in equation (16). For example, partial re-optimization (PR^{ous}) [48] is applied as one heuristic policy. Another PR^{ous}-type policy is generated by implementing the partial re-optimization framework in the formulation of penalty cost, and the value functions are then obtained based on (16). So, based on different value function evaluations (i.e., with or without decomposition), an original policy and its variant based on decomposition are generated for each heuristic policy. We introduce partial re-optimization (PR^{ous}) [48], one-step rollout algorithm (ORA) [46], two-step rollout algorithm (TRA) [38] and rollout algorithm (RA) [47] to generate policies in the policy set. The variants based on the decomposition framework are generated accordingly.

Among the candidates, some policies belong to the a priori optimization method category. The fixed routing sequence is implemented, and only restocking decisions are made during the execution of the policy. We diversify the generation of the a priori route using the rollout static method [47], a variant of the rollout static method (i.e., based on decomposition as in equation (16)) and the TSP method [11, 40, 28].

We also diversify the candidate choice by introducing a myopic policy. Under the myopic paradigm, routing and restocking decisions are made by only considering the immediate cost of the current state. For example, assume the current state is $s_k = (l, q, \mathcal{R})$. The restocking decision is first made for each potential routing choice, i.e., $u_{j,l,\mathcal{R}}(q) = 1$, if $d_{l0} + d_{0j} \leq d_{lj} + b \cdot \sum_{e>q} p_j(e) \cdot (e-q)$, otherwise, $u_{j,l,\mathcal{R}}(q) = 0$, and let $c_{ime}(j)$ denote the immediate cost if traveling to customer j ($\forall j \in \mathcal{R}$). Then, the routing decision is made based on $\arg\min_{j\in\mathcal{R}}\{c_{ime}(j)\}$, and the restocking decision is determined accordingly.

Finally, the policy PR^{trd}, under the traditional recourse action, is included as a candidate policy. This policy is distinct from the two PR^{ous}-type policies in that its recourse relies on the classical DTD action rather than outsourcing. More importantly, this policy is generated under the original decision sequence (i.e., $j \to u_j$), in contrast to the two PR^{ous}-type policies, which are based on the reversed decision sequence (i.e., $u_j \to j$). By integrating policies derived from different decision sequences, the solution space is further enriched, potentially leading to improved outcomes.

Overall, eleven candidate heuristic policies are included in our setting, called par-reopt (PR^{ous}), par-reopt-de (PR^{ous}-de), rollout-dynamic (ORA), rollout-dynamic-de (ORA-de), two-rollout-dynamic (TRA), two-rollout-dynamic-de (TRA-de), rollout-static (RA), rollout-static-de (RA-de), TSP, myopic policies, and par-reopt-trd (PR^{trd}), where '-de' represents the policy variant based on decomposition. In practice, some of them may be selected to form the policy set, depending on the performance of the resulting price-directed policy. Policy candidates can also be hand-selected. Different combinations of policies can be used to sample constraints, with the goal of obtaining a better price-directed policy.

B PROPERTIES OF PENALTY COSTS

B.1 Monotonicity of penalty cost on residual capacity q

Proof. Penalty cost $f^j_{l,\mathcal{R}}(q)$ is defined as in equation (19.1). Proving non-increasing in q is to testify $f^j_{l,\mathcal{R}}(q_2) \leq f^j_{l,\mathcal{R}}(q_1)$ given $0 \leq q_1 \leq q_2 \leq Q$. Four situations need to be considered, when $u_{j,l,\mathcal{R}}(q_1)$ and $u_{j,l,\mathcal{R}}(q_2)$ take different values ($u_{j,l,\mathcal{R}}(q) \in \{0,1\}$).
Situation (1): If $u_{j,l,\mathcal{R}}(q_1) = 1$ (case R) and $u_{j,l,\mathcal{R}}(q_2) = 1$ (case R), then, $f^{j(R)}_{l,\mathcal{R}}(q_2) = f^{j(R)}_{l,\mathcal{R}}(q_1)$;
Situation (2): If $u_{j,l,\mathcal{R}}(q_2) = 0$ (case D) and $u_{j,l,\mathcal{R}}(q_1) = 1$, then, $f^{j(D)}_{l,\mathcal{R}}(q_2) \leq f^{j(R)}_{l,\mathcal{R}}(q_2) = f^{j(R)}_{l,\mathcal{R}}(q_1)$;

Situation (3): If $u_{j,l,\mathcal{R}}(q_1) = 0$ and $u_{j,l,\mathcal{R}}(q_2) = 0$, then,

$$\begin{aligned} f^{j(D)}_{l,\mathcal{R}}(q_2) - f^{j(D)}_{l,\mathcal{R}}(q_1) &= \sum_{e\le q_2} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(q_2-e) + b\cdot\sum_{e>q_2} p_j(e)\cdot(e-q_2) + \sum_{e>q_2} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(0) - \\ &\quad \sum_{e\le q_1} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(q_1-e) - b\cdot\sum_{e>q_1} p_j(e)\cdot(e-q_1) - \sum_{e>q_1} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(0) \\ &= b\cdot\sum_{e>q_2} p_j(e)\cdot(q_1-q_2) - \sum_{e\le q_1} p_j(e)\cdot(f_{j,\mathcal{R}\setminus j}(q_1-e) - f_{j,\mathcal{R}\setminus j}(q_2-e)) - \\ &\quad b\cdot\sum_{e>q_1}^{q_2} p_j(e)\cdot(e-q_1) - \sum_{e>q_1}^{q_2} p_j(e)\cdot(f_{j,\mathcal{R}\setminus j}(0) - f_{j,\mathcal{R}\setminus j}(q_2-e)) \\ &\le 0; \end{aligned}$$

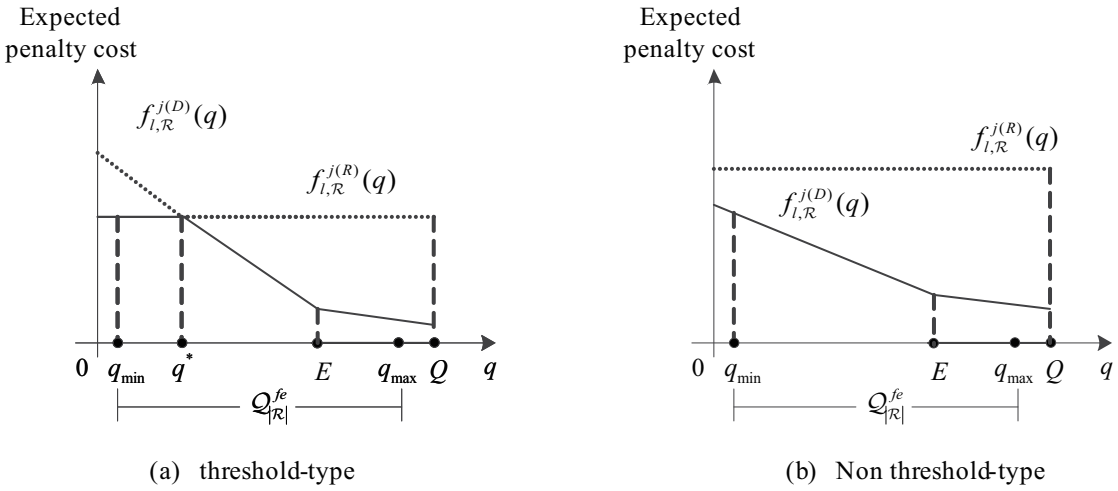
If $f_{l,\mathcal{R}}(\cdot)$ is non-increasing in q, then the same property must hold for $f_{j,\mathcal{R}\setminus j}(\cdot)$. Consequently, we have $f_{j,\mathcal{R}\setminus j}(q_1-e) \ge f_{j,\mathcal{R}\setminus j}(q_2-e)$ for $q_1 - e \le q_2 - e$, and $f_{j,\mathcal{R}\setminus j}(0) \ge f_{j,\mathcal{R}\setminus j}(q_2-e)$ for $0 \le q_2 - e$. Thus, terms $-\sum_{e\le q_1} p_j(e)\cdot(f_{j,\mathcal{R}\setminus j}(q_1-e) - f_{j,\mathcal{R}\setminus j}(q_2-e))$ and $-\sum_{e>q_1}^{q_2} p_j(e)\cdot(f_{j,\mathcal{R}\setminus j}(0) - f_{j,\mathcal{R}\setminus j}(q_2-e))$ are non-positive. Additionally, since $b\cdot\sum_{e>q_2} p_j(e)\cdot(q_1-q_2)$ and $-b\cdot\sum_{e>q_1}^{q_2} p_j(e)\cdot(e-q_1)$ are both negative, it follows that $f^{j(D)}_{l,\mathcal{R}}(q_2) - f^{j(D)}_{l,\mathcal{R}}(q_1) \le 0$;
Situation (4): If $u_{j,l,\mathcal{R}}(q_2) = 1$ and $u_{j,l,\mathcal{R}}(q_1) = 0$, then,

$$\begin{aligned} &f^{j(D)}_{l,\mathcal{R}}(q_1) \le f^{j(R)}_{l,\mathcal{R}}(q_1) = f^{j(R)}_{l,\mathcal{R}}(q_2), \text{ and} \\ &f^{j(D)}_{l,\mathcal{R}}(q_1) = \sum_{e\le q_1} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(q_1-e) + b\cdot\sum_{e>q_1} p_j(e)\cdot(e-q_1) + \sum_{e>q_1} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(0) \\ &\quad \ge \sum_{e\le q_1} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(q_2-e) + b\cdot\sum_{e>q_2} p_j(e)\cdot(e-q_1) + \sum_{e>q_2} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(0) + \sum_{e>q_1}^{q_2} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(0) \\ &\quad = \sum_{e\le q_2} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(q_2-e) - \sum_{e>q_1}^{q_2} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(q_2-e) + \sum_{e>q_1}^{q_2} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(0) + b\cdot\sum_{e>q_2} p_j(e)\cdot(e-q_1) + \sum_{e>q_2} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(0) \\ &\quad \ge \sum_{e\le q_2} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(q_2-e) + b\cdot\sum_{e>q_2} p_j(e)\cdot(e-q_2) + \sum_{e>q_2} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(0) \\ &\quad = f^{j(D)}_{l,\mathcal{R}}(q_2) \ge f^{j(R)}_{l,\mathcal{R}}(q_2) \\ &\therefore f^{j(D)}_{l,\mathcal{R}}(q_1) \ge f^{j(R)}_{l,\mathcal{R}}(q_2). \square \end{aligned}$$

If $f_{l,\mathcal{R}}(\cdot)$ is non-increasing in q, then it follows that $f_{j,\mathcal{R}\setminus j}(q_1-e) \ge f_{j,\mathcal{R}\setminus j}(q_2-e)$ for $q_1 - e \le q_2 - e$. Therefore, $\sum_{e\le q_1} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(q_1-e) \ge \sum_{e\le q_1} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(q_2-e)$. Moreover, since $f_{j,\mathcal{R}\setminus j}(q_2-e) \le f_{j,\mathcal{R}\setminus j}(0)$ for $q_2 - e \ge 0$, it follows that $-\sum_{e>q_1}^{q_2} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(q_2-e) + \sum_{e>q_1}^{q_2} p_j(e)\cdot f_{j,\mathcal{R}\setminus j}(0) \ge 0$. Thus, the inequality holds.

B.2 Possible threshold-type replenishment

For particular customer l^* and unvisited set $\mathcal{R}^*$ (not all), the optimal decision between replenishing and moving directly to the next customer is of threshold type in residual capacity $q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}$, where $\mathcal{Q}^{fe}_{|\mathcal{R}|}$ (i.e. $[q_{\min}, q_{\max}]$) denotes the feasible range of residual capacity q and differs for the number of unvisited customers ($|\mathcal{R}|$). Please refer the definition of $\mathcal{Q}^{fe}_{|\mathcal{R}|}$ in formulation (20), where $q_{\min}$ equals to $(Q-(N-|\mathcal{R}|)\cdot E)^+$ and $q_{\max}$ equals to $Q - e_{\min}$.



(a) threshold-type (b) Non threshold-type

FIGURE B1 Possible threshold-type replenishment

As Proposition 1, penalty cost $f_{l,\mathcal{R}}(\cdot)$ is non-increasing in residual capacity q, two situations reflected by Fig. B1(a) and Fig. B1(b) are shown, which indicates situations $f_{l,\mathcal{R}}^{j(D)}(q_{\min}) > f_{l,\mathcal{R}}^{j(R)}(q_{\min})$ and $f_{l,\mathcal{R}}^{j(D)}(q_{\min}) \leq f_{l,\mathcal{R}}^{j(R)}(q_{\min})$, respectively. If $f_{l,\mathcal{R}}^{j(D)}(q_{\min}) > f_{l,\mathcal{R}}^{j(R)}(q_{\min})$, the decision for replenishing or proceeding to the next customer directly is of threshold-type in residual capacity $q \in \mathcal{Q}_{|\mathcal{R}|}^{fe}$, as shown in Fig. B1(a), Otherwise, the optimal decision is always to move directly to the next customer (case D), whatever the residual capacity q is, as shown in Fig. B1(b).

Outsourcing price b influences whether situation (a) or (b) occurs. If the price is sufficiently low, visiting the next customer directly is always preferable. However, outsourcing is typically priced in a way that incentivizes restocking as the preferred option when the vehicle's residual capacity is depleted. Price b is adjusted to accommodate this scenario. Additionally, note that the expected penalty cost function can be piece-wise linear due to the threshold-based decision structure and the fact that the maximum customer demand E is lower than vehicle capacity Q.

C NOTATION

TABLE C1 Notation

l, j, J, i	Customers
$\tilde{\xi}_l$	Demand of customer l
e	Realized amount for random variable $\tilde{\xi}_l$
$p_l(e)$	Probability when variable $\tilde{\xi}_l$ takes e
q, Q	Residual capacity and capacity limit of the vehicle
d_{lj}	Traveling distance between customers l and j
Δ_{lj}	Extra traveling distance for a preventive return to the depot
b	Unit price for outsourcing
$\mathcal{R}_k(l)$	Set of unvisited customers from current customer l on stage k
s_k	State variable, equaling to $(l, q, \mathcal{R}_k(l))$, representing the vehicle departs from customer l with residual capacity q and set of unvisited customers $\mathcal{R}_k(l)$
$V_k(l, q, \mathcal{R}_k(l))$	Cost-to-go value at state $(l, q, \mathcal{R}_k(l))$
$j_k(l, q, \mathcal{R}_k(l))$	Optimal routing decision, customer j, given state $(l, q, \mathcal{R}_k(l))$
$u_{j,l,\mathcal{R}_k(l)}(q)$	Restocking decision at state $(l, q, \mathcal{R}_k(l))$ if routing customer j next
$f_k^j(l, q, \mathcal{R}_k(l))$	Expected penalty cost for routing customer j next at state $(l, q, \mathcal{R}_k(l))$, $f_{l,\mathcal{R}}^j(q)$ for ease of notation
$f_k(l, q, \mathcal{R}_k(l))$	Expected penalty cost at state $(l, q, \mathcal{R}_k(l))$, $f_{l,\mathcal{R}}(q)$ for ease of notation
$L_{s_k}^j$	Lower bound of $f_k^j(l, q, \mathcal{R}_k(l))$
$v_{k-1}(j, \mathcal{R}_{k-1}(j; l))$	Traveling cost by following a partial route starting from customer j and visiting customers in set $\mathcal{R}_{k-1}(j; l)$ subsequently
$L_{tsp}^{j,\mathcal{R}_{k-1}(j;l)}$	Lower bound of $v_{k-1}(j, \mathcal{R}_{k-1}(j; l))$
θ, α, ω	Basis within affine functions to approximate value functions

D RANGE OF OUTSOURCING PRICE

Outsourcing price b follows Criterion 1, which encourages restocking decisions when the vehicle's residual capacity is empty (i.e., $q = 0$). Specifically, restocking is preferred when condition $f^{R(j)}(l, 0, \mathcal{R}) \leq f^{D(j)}(l, 0, \mathcal{R})$ holds, where

$$\begin{aligned} f_k^{R(j)}(l, 0, \mathcal{R}) &= \Delta_{lj} + \sum_e p_j(e) \cdot f_{k-1}(j, Q, \mathcal{R} \setminus j) \\ f_k^{D(j)}(l, 0, \mathcal{R}) &= b \sum_e p_j(e) \cdot e + \sum_e p_j(e) \cdot f_{k-1}(j, 0, \mathcal{R} \setminus j). \end{aligned} \tag{A1}$$

These equations are obtained by substituting $q = 0$ into equation (19.1). Since penalty cost function $f(\cdot)$ is non-increasing in q, it follows that $f_{k-1}(j, Q, \mathcal{R} \setminus j) \leq f_{k-1}(j, 0, \mathcal{R} \setminus j)$. Therefore, $\sum_e p_j(e) \cdot f_{k-1}(j, Q, \mathcal{R} \setminus j) \leq \sum_e p_j(e) \cdot f_{k-1}(j, 0, \mathcal{R} \setminus j)$. If $\Delta_{lj} \leq b \sum_e p_j(e) \cdot e$, then condition $f^{R(j)}(l, 0, \mathcal{R}) \leq f^{D(j)}(l, 0, \mathcal{R})$ holds, which implies that $b \geq \frac{\Delta_{lj}}{\sum_e p_j(e) \cdot e}$. Furthermore, if price b reaches at least $\frac{\max \Delta_{lj}}{\sum_e p_j(e) \cdot e}$, then the requirement is fully satisfied.

Outsourcing price b satisfies Criterion 2, partially ensuring the cost savings of the outsourcing strategy compared to the traditional restocking strategy. Under the traditional recourse strategy, the vehicle fulfills unmet demand by making a replenishment trip to the depot. In contrast, our strategy satisfies unmet demand through external carriers, incurring an additional outsourcing

cost. The corresponding cost functions can be expressed by

$$\begin{aligned} V_k(l,q,\mathcal{R}) &= d_{lj} + \sum_{e\le q} p_j(e)\cdot V_{k-1}(j,q-e,\mathcal{R}\setminus j) + \sum_{e>q} p_j(e)\cdot [V_{k-1}(j,Q+q-e,\mathcal{R}\setminus j) + 2d_{oj}] \\ &= d_{lj} + \sum_{e\le q} p_j(e)\cdot V_{k-1}(j,q-e,\mathcal{R}\setminus j) + \sum_{e>q} p_j(e)\cdot V_{k-1}(j,Q+q-e,\mathcal{R}\setminus j) + \sum_{e>q} p_j(e)\cdot 2d_{oj} \\ V'_k(l,q,\mathcal{R}) &= d_{lj} + \sum_{e\le q} p_j(e)\cdot V'_{k-1}(j,q-e,\mathcal{R}\setminus j) + \sum_{e>q} p_j(e)\cdot V'_{k-1}(j,0,\mathcal{R}\setminus j) + b\sum_{e>q} p_j(e)\cdot(e-q). \end{aligned} \tag{A2}$$

From (A2), cost savings for outsourcing, compared to the traditional recourse scheme, can be achieved only if $b\sum_{e>q} p_j(e)\cdot(e-q) \le \sum_{e>q} p_j(e)\cdot 2d_{oj}$, considering that the advantage in residual capacity may already be established, as $Q+q-e\ge 0$ after implementing the recourse actions under different strategies. This condition implies that $b \le \frac{\sum_{e>q} p_j(e)\cdot 2d_{0j}}{\sum_{e>q} p_j(e)\cdot(e-q)}$ must hold. The fraction $\frac{\sum_{e>q} p_j(e)}{\sum_{e>q} p_j(e)\cdot(e-q)}$ is non-decreasing in q, as seen from its continuous form $\frac{\frac{1}{E}\int_q^Q dx}{\frac{1}{E}\int_q^Q (x-q)dx}$, which is non-decreasing in q under the assumption that demand e follows the uniform distribution. Consequently, $\frac{\sum_{e>q} p_j(e)\cdot 2d_{0j}}{\sum_{e>q} p_j(e)\cdot(e-q)}$ attains its minimum value when $q=0$. This leads to condition $b \le \frac{\sum_e p_j(e)\cdot 2d_{0j}}{\sum_e p_j(e)\cdot e} = \frac{2d_{0j}}{\sum_e p_j(e)\cdot e}$, ensuring cost savings when the outsourcing strategy is employed. Additionally, to further quantify the boundary for b, we establish the condition that price b should not exceed $\frac{\min 2d_{0j}}{\sum_e p_j(e)\cdot e}$.

In summary, outsourcing price b falls within the range $[\frac{\max \Delta_{lj}}{\sum_e p_j(e)\cdot e}, \frac{\min 2d_{0j}}{\sum_e p_j(e)\cdot e}]$, where Δ_{lj} and d_{0j} are determined based on a feasible TSP trip in our setting. Setting b within this range basically ensures that restocking can occur when the vehicle's residual capacity is depleted while also potentially yielding cost savings under the outsourcing strategy compared to the traditional recourse strategy. Note that $\frac{\max \Delta_{lj}}{\sum_e p_j(e)\cdot e}$ is not always less than $\frac{\min 2d_{0j}}{\sum_e p_j(e)\cdot e}$. In such cases, $\frac{\bar{\Delta}_{lj}}{\sum_e p_j(e)\cdot e}$ and $\frac{2\bar{d}_{0j}}{\sum_e p_j(e)\cdot e}$ are used as the replacements, where $\bar{\Delta}_{lj}$ and $\bar{d}_{0j}$ represent the average values. $\frac{\bar{\Delta}_{lj}}{\sum_e p_j(e)\cdot e}$ is chosen to ensure that restocking can take place upon failure on average, while $\frac{2\bar{d}_{0j}}{\sum_e p_j(e)\cdot e}$ ensures the cost advantage of the outsourcing strategy on average.