Les Cahiers du GERAD ISSN: 0711–2440

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

L. Zhu, Y. Adulyasak, L.-M. Rousseau

G–2024–70

October 2024

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2024
– Bibliothèque et Archives Canada, 2024

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2024
– Library and Archives Canada, 2024

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

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

Lin Zhu [a]

Yossiri Adulyasak [b]

Louis-Martin Rousseau [c]

[a] *Logistics Research Center, Shanghai Maritime University, Shanghai 201306, China*

[b] *GERAD & Department of Logistics and Operations Management, HEC Montréal, Montréal (Qc), Canada, H3T 1J4*

[c] *CIRRELT & Department of Mathematical and Industrial Engineering, Polytechnique Montréal, Montréal (Qc), Canada, H3T 2A7*

zhulin.sife@gmail.com

October 2024
Les Cahiers du GERAD
G–2024–70

Abstract : This paper presents a partial outsourcing strategy for the vehicle routing problem with stochastic demands (VRPSD), and routing reoptimization is considered for the single-vehicle case. In the VRPSD, a vehicle may arrive at a customer's location with insufficient capacity to meet the customer demand, which results in a route failure and requires a subsequent recourse action. We propose a recourse action that utilizes outsourcing to handle unmet demand, deferring from the classical recourse action. It is formulated as a Markov decision process (MDP), and an approach based on an approximate linear programming (ALP) scheme is proposed to solve it. To effectively deal with the curse of dimensionality, several algorithmic enhancements that take advantage of the problem's structure are proposed. These include lower bounding methods based on affine functions, decomposition-based value function approximations, and constraint sampling. Our approach is compared against other approaches, and the results show that our approach generally yields high-quality solutions.

Keywords : Stochastic vehicle routing, re-optimization, Markov decision process, approximate linear programming, recourse strategy, outsourcing

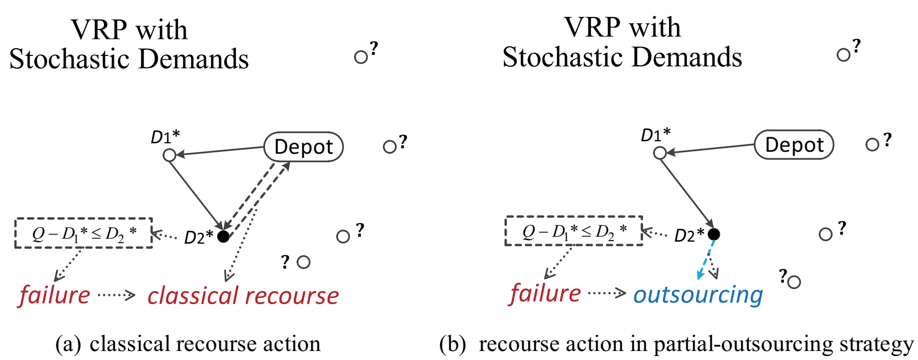
Acknowledgements: This work was supported by the China Scholarship Council under Grant 201908310007. This support is gratefully acknowledged.

1 Introduction

The vehicle routing problem with stochastic demands (VRPSD) is essential in the family of vehicle routing problems [15]. In the VRPSD, vehicles are dispatched to satisfy customer demands while respecting vehicle capacity. Customer demands are random variables with known probability distributions. The VRPSD has many real-world applications, such as local deposit collection from bank branches, package collection, waste collection, residential gas propane refilling, and recent electric vehicle recharging [36]. In these applications, customer demands fluctuate during the routing, and the actual demands are observed upon the vehicles' arrival at the customers.

In the VRPSD, a vehicle may reach a customer location with insufficient residual capacity to fulfill the demand for collections. This leads to a route failure, in which case a recourse action is necessary [41]. The classical recourse action requires the vehicle to perform a replenishment trip at the depot when a failure occurs, which refers to the detour-to-depot (DTD) operating scheme, as shown in Fig. 1(a). The DTD scheme is computationally appealing and straightforward to implement. However, substantial effort can be made to conduct the detour trip for a small unmet demand.

This paper considers an alternative recourse action for the VRPSD, in which unmet demands are outsourced to other carriers when failures occur, as shown in Fig. 1(b). Specifically, vehicle routes are planned with the help of outsourced carriers. The operator is only concerned with its route optimization, leaving the routing choices of other carriers at ease and compensating them with running errands per unit load. This problem setting can be found in third-party logistics companies (3PL), where the outsourcing cost can be charged per unit load depending on the contracts negotiated [e.g., in 35].



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

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

[1] Q 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.

Routing optimization for the single-vehicle case is considered in this paper. We develop a partial-outsourcing strategy to solve the problem. Given observed customer demands, our strategy enables the re-optimization of the operating scheme, by which dynamic routing and proactive restocking decisions are made. Due to the complexity of realizing dynamic routing, determining a re-optimization strategy remains highly challenging, and research in this area needs to be addressed [e.g., 28, 34]. The decomposition-based value function approximation is proposed to tackle the computational difficulty. As the reoptimization strategy for the VRPSD corresponds to a Markov decision process (MDP) [32, 34, 38], one must deal with the curse of dimensionality of both the expanded state and action spaces when considering the outsourcing action. Thus, it is not trivial to solve this extended model in a tractable fashion. As clearly demonstrated in the previous literature of VRPSD [e.g. 38], it is important to exploit structural properties and devise an effective reformulation scheme that allows computational tractability.

An approximate linear programming (ALP) method is developed to compute our strategy. ALP methods have been employed in other problem contexts, including inventory routing [1, 2], revenue management [3, 22, 37], scheduling problems [4, 5], and the traveling salesperson problem [38]. However, the method has not been adapted to tackle this variant of VRP with outsourcing. With an ALP method, the Markov decision process (MDP) formulation is first transformed into its linear programming (LP) counterpart, in which the variables are value functions. For tractability, affine functions are then introduced to approximate the variables. The ALP method can yield the bounds on the value functions. The bounds are then used to approximate value functions within the traditional Bellman recursion to derive a price-directed policy [e.g., 9, 38].

The primal contribution of the paper is concluded as follows. A partial-outsourcing strategy is proposed for the VRPSD. An alternative recourse action is developed by leveraging outsourced delivery services. An ALP method is adapted to determine a policy for the VRPSD and is developed by exploiting the structure of the problem. Several algorithm enhancement techniques are introduced, including decomposition-based value function approximations, lower bounding procedures based on affine functions, and constraint sampling. Finally, we compare our approach with other competitive solution methods. The experimental results demonstrate that our approach yields high-quality solutions.

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 presents a solution framework that incorporates an outsourcing recourse strategy for the VRPSD through a re-optimization paradigm. In the following section, routing strategies with different recourse actions for the VRPSD are reviewed. Then, solution methods for solving the VRPSD with re-optimization are presented. Here, we briefly review the relevant research to the VRPSD with recourse decisions. For a more general review of the VRPSD, we refer to the works by Gendreau et al. [15], Florio et al. [13], and Florio et al. [14].

Different recourse strategies have been developed since the classical DTD operating scheme was proposed. The DTD scheme requires the vehicle to make a replenishment trip if a failure occurs at a customer. Most recourse strategies were proposed based on this operation scheme [e.g., in 11, 21]. For another widely studied recourse action, preventive restocking is considered. By this recourse action, the vehicle may execute a restocking trip to the depot before the inventory is depleted [e.g., in 24, 40]. Besides, Novoa et al. [25] introduced an extended recourse strategy and proposed two recourse actions by disallowing partial deliveries. In recent years, new recourse actions have been developed. Salavati-Khoshghalb et al. [29] presented a rule-based recourse policy in which a preventive recourse trip is made when the remaining vehicle capacity falls below a preset customer-specific threshold. Salavati-Khoshghalb et al. [30] developed a hybrid recourse policy by defining the risk of failure with a distance-based measure. Florio et al. [12] proposed the switch policy. Under this policy, preventive restocking is considered, and the visiting orders of two adjacent customers can be swapped.

There are several studies that consider the VRPSD with re-optimization. Secomandi [31] proposed a neuro-dynamic programming (ADP) algorithm that approximates the value functions for states as linear functions of pre-selected features. Secomandi [32] proposed a one-step rollout algorithm in which a base routing sequence is sequentially improved, then, Novoa and Storer [26] further improved the algorithm and developed a two-step rollout algorithm. Secomandi and Margot [34] proposed a partial re-optimization method. By assuming a priori route, partial re-optimization is performed in each block of customers along the considered route. For the multi-vehicle setting, Goodson et al. [17] and Goodson et al. [18] studied the VRPSD with duration limits and proposed rollout policies. Zhu et al. [41] developed a paired cooperative re-optimization method in which customers are dynamically assigned between two vehicles, and routing is implemented based on the partial re-optimization procedure. Recently, Ulmer et al. [39] considered a variant of the VRP with stochastic customer requests and proposed an offline-online approximate dynamic programming approach.

In summary, outsourcing was not considered a recourse action before, and the ALP method has yet to be introduced to solve the VRPSD. Florio et al. [13] generalized another modeling perspective for VRPSD, i.e., chance-constrained VRPSD. We do not review the relevant research because it is a different modeling paradigm in which the capacity constraint can be violated probabilistically. In the following section, the MDP formulation for our outsourcing strategy is introduced.

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., 7, 26, 31, 34]. 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 ($l, j \in \mathcal{N} \cup 0$) 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

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

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

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

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,

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

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

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

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

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

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., 32, 34, 40], the cost-to-go value for each state s_k can be expressed as

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

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. Note that we change the decision sequence (i.e., $u \to j$, instead of $j \to u$) for ease of formulation for our approximation scheme.

Our partial-outsourcing strategy is formulated as in Section 3. Correspondingly, optimality equations (3) and (4) can be jointly expressed as

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

The ALP solution framework to solve (12) is described in the subsequent section.

4 Decomposition-based approximate linear program framework

Solving the MDP formulation will inevitably lead to the curse of dimensionality. In our formulation, the cardinality of the state space is $1+N(Q+1)2^{N-1}$, which is intractable as the number of customers grows. One effective approach to solving this problem is the approximate linear programming framework. The MDP formulation is transformed into a linear programming (LP) formulation. Then, the lower bounds of the value functions are generated in a minimization problem by approximating the value function with affine functions. The MDP model (2)–(5) can be formally rewritten as an LP

$$
\begin{aligned}
\max \quad & V_N(0,Q,\mathcal{N}) && \text{(13a)}\\
s.t. \quad & V_N(0,Q,\mathcal{N}) \le d_{0j} + \sum_e p_j(e)\cdot V_{N-1}\big(j, Q-e, \mathcal{N}\backslash\{j\}\big), && j \in \mathcal{N}, \quad \text{(13b)}\\
& V_k\big(l,q,\mathcal{R}_k(l)\big) \le d_{lj} + B_j(q) + \sum_{e\le q} p_j(e)\cdot V_{k-1}\big(j, q-e, \mathcal{R}_{k-1}(j;l)\big) &&\\
& \qquad + V_{k-1}\big(j, 0, \mathcal{R}_{k-1}(j;l)\big)\cdot \sum_{e>q} p_j(e), &&\\
& V_k\big(l,q,\mathcal{R}_k(l)\big) \le d_{l0} + d_{0j} + \sum_e p_j(e)\cdot V_{k-1}\big(j, Q-e, \mathcal{R}_{k-1}(j;l)\big), && \forall k\in\Omega\backslash\{N,0\}, \forall l\in\mathcal{N},\\
& && q\in\mathcal{Q}, \mathcal{R}_{k-1}(j;l)\subseteq \mathcal{N}\backslash\{l,j\},\\
& && j\in\mathcal{R}_k(l), \quad \text{(13c)}\\
& V_0(l,q,\phi) \le d_{l0}, && \forall l\in\mathcal{N}, q\in\mathcal{Q}. \quad \text{(13d)}
\end{aligned}
$$

One potential approach to reduce the complexity of the model is to leverage affine functions to approximate value functions. Toriello et al. [38] studied a traveling salesman problem with stochastic arc costs and identified the challenge of applying ALP in a routing problem. They shared the insight that the approximation of value functions can be poor if directly replaced by simple affine functions and pointed out that strong affine functions can be derived based on exploring the structure of the routing problem first [38].

In the subsequent section, the structure of the value functions in our formulation is analyzed, and an approximation reformulation scheme is proposed.

Decomposition-based value function approximation

Re-optimization and a priori optimization are two major approaches for VRPSD [15]. A priori optimization is often modeled by stochastic programming with recourse (SPR). A fundamental structural characteristic is revealed in the objective function of the SPR formulations, as the objective function is typically composed of two parts [e.g., 23, 24, 29]

Similarly, value functions within our problem can also be decomposed into two parts.

$$
\begin{aligned}
&V_k(l,q,\mathcal{R}_k(l)) =\\
&\min_{[J,(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)]} \left\{ \left[d_{lJ} + v_{k-1}(J,\mathcal{R}_{k-1}(J;l)) \Big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)} \right] + f^J_k(l,q,\mathcal{R}_k(l)) \Big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)} \right\},\\
&\qquad J\in\mathcal{R}_k(l), \quad \forall s_k\in\Psi. \quad (14)
\end{aligned}
$$

The first component $[d_{lJ} + v_{k-1}(J,\mathcal{R}_{k-1}(J;l)) \big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)}]$ is relevant to the travel cost between customers, given 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 remaining customers in $\mathcal{R}_k(l)$, by visiting customer $J \in \mathcal{R}_k(l)$ first and 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 $k' = \{k-1, k-2, \ldots, 1\}$. $v_{k-1}(J,\mathcal{R}_{k-1}(J;l)) \big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)}$ refers to the travel cost related to the partial route starting from customer J and including the remaining customers in $\mathcal{R}_{k-1}(j;l)$. The second component $f^J_k(l,q,\mathcal{R}_k(l)) \big|_{(j^J_{k-1},j^J_{k-2},\ldots,j^J_1)}$ is related to the penalty cost, including any additional distance traveled for replenishment, as well as potential outsourcing cost. The second component also indicates the expected future penalty cost given current state $(l, q, \mathcal{R}_k(l))$ and the decision for routing customer J next. Both components rely on the future routing policy, represented by $[J, (j^J_{k-1}, j^J_{k-2}, \ldots, j^J_1)]$, where superscript J in $j^J_{k'}$ ($k' = \{k-1, k-2, \ldots, 1\}$) indicates that future routing policy should be made by fixing J as the next customer to be visited. Equation (14) aims at determining the optimal routing policy $[J, (j^J_{k-1}, j^J_{k-2}, \ldots, j^J_1)]$ for state s_k, to minimize expected future costs.

The decomposition-based value function approximation is derived from Equation (14). An original value function is approximated by decomposing it into two components and then estimating each separately. The following relation holds for the second component in Equation (14)

$$f_k^J\left(l,q,\mathcal{R}_k(l)\right)\Big|_{(j_{k-1}^J,j_{k-2}^J,\ldots,j_1^J)} \geq \min_{(j_{k-1}^J,j_{k-2}^J,\ldots,j_1^J)}\left\{f_k^J\left(l,q,\mathcal{R}_k(l)\right)\Big|_{(j_{k-1}^J,j_{k-2}^J,\ldots,j_1^J)}\right\} = \min\left\{f_k^J\left(l,q,\mathcal{R}_k(l)\right)\right\}, \quad \forall s_k\in\Psi, J\in\mathcal{R}_k(l), \tag{15}$$

where the right-hand side of the equation is obtained as term $\min\left\{f_k^J\left(l,q,\mathcal{R}_k(l)\right)\Big|_{(j_{k-1}^J,j_{k-2}^J,\ldots,j_1^J)}\right\}$ indicates the minimal penalty cost among all realized routing sequences. Let $L_{s_k}^J$ denote the lower bound of the future expected penalty cost for state s_k along with routing customer J next. Assuming that better lower bounds $L_{s_k}^J$ ($\forall s_k\in\Psi,\ J\in\mathcal{R}_k(l)$) are given, $L_{s_k}^J$ are then used to approximate the second component $f_k^J\big(l,q,\mathcal{R}_k(l)\big)\Big|_{(j_{k-1}^J,j_{k-2}^J,\ldots,j_1^J)}$. Let $\tilde{V}_k\big(l,q,\mathcal{R}_k(l)\big)$ ($s_k\in\Psi$) denote the approximated value functions. The value functions are approximated as

$$\tilde{V}_k\big(l,q,\mathcal{R}_k(l)\big) = \min_{[J,(j_{k-1}^J,j_{k-2}^J,\ldots,j_1^J)]}\left\{d_{lJ}+v_{k-1}(J,\mathcal{R}_{k-1}(J;l))\Big|_{(j_{k-1}^J,j_{k-2}^J,\ldots,j_1^J)}+L_{s_k}^J\right\} \tag{16a}$$

$$= \min_{J\in\mathcal{R}_k(l)}\left\{d_{lJ}+\min_{(j_{k-1}^J,j_{k-2}^J,\ldots,j_1^J)}\left\{v_{k-1}(J,\mathcal{R}_{k-1}(J;l))\Big|_{(j_{k-1}^J,j_{k-2}^J,\ldots,j_1^J)}\right\}+L_{s_k}^J\right\} \tag{16b}$$

$$= \min_{J\in\mathcal{R}_k(l)}\left\{d_{lJ}+\min\left\{v_{k-1}(J,\mathcal{R}_{k-1}(J;l))\right\}+L_{s_k}^J\right\}, \quad \forall s_k\in\Psi, \tag{16c}$$

where Equation (16c) holds for the same reason as in equation (15). In fact, $\min\left\{v_{k-1}\big(J,\mathcal{R}_{k-1}(J;l)\big)\right\}$ implies a deterministic TSP process. Following the definition of $v_{k-1}\big(J,\mathcal{R}_{k-1}(J;l)\big)$, $\min\{v_{k-1}\big(J,\mathcal{R}_{k-1}(J;l)\big)\}$ 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 [8, 19]. Let $l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)}$ ($l\in\mathcal{N},\ J\in\mathcal{R}_k(l),\ |\mathcal{R}_k(l)|=k$) represent the lower bounds for approximating $\min\left\{v_{k-1}\big(J,\mathcal{R}_{k-1}(J;l)\big)\right\}$. Value functions can be further estimated by

$$\tilde{V}_k\big(l,q,\mathcal{R}_k(l)\big) = \min_{J\in\mathcal{R}_k(l)}\left\{d_{lJ}+l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)}+L_{s_k}^J\right\}, \quad \forall s_k\in\Psi. \tag{17}$$

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

$$j_k\big(l,q,\mathcal{R}_k(l)\big) = \arg\min_{J\in\mathcal{R}_k(l)}\left\{d_{lJ}+l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)}+L_{s_k}^J\right\}, \quad \text{given a realized state} s_k. \tag{18}$$

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. More specifically, at state $s_k=\big(l,q,\mathcal{R}_k(l)\big)$, values $L_{s_k}^J$ ($J\in\mathcal{R}_k(l)$) are already 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 routing decision $J\in\mathcal{R}_k(l)$, and this LP is solvable in polynomial time [8, 20, 27].

$$l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)} = \min_x\left\{\sum_{i'\in\mathcal{R}_{k-1}(J;l)} d_{Ji'}x_{Ji'}+\sum_{i'\in\mathcal{R}_{k-1}(J;l)}\ \sum_{j'\in\mathcal{R}_{k-1}(J;l)\cup\{0\}\backslash i'} d_{i'j'}x_{i'j'}\right\} \tag{19a}$$

$$s.t. \quad \sum_{i'\in\mathcal{R}_{k-1}(J;l)} x_{Ji'}=1, \tag{19b}$$

$$\sum_{i'\in\mathcal{R}_{k-1}(J;l)} x_{i'0}=1, \tag{19c}$$

$$\sum_{j' \in \mathcal{R}_{k-1}(J;l) \backslash i'} x_{i'j'} = 1, \qquad i' \in \mathcal{R}_{k-1}(J;l), \tag{19d}$$

$$\sum_{j' \in \mathcal{R}_{k-1}(J;l) \backslash i'} x_{j'i'} = 1, \qquad i' \in \mathcal{R}_{k-1}(J;l), \tag{19e}$$

$$\sum_{i' \in U} \sum_{j' \in \mathcal{R}_{k-1}(J;l) \cup \{0\} \backslash U} x_{i'j'} \geq 1, \qquad \forall \phi \neq U \subseteq \mathcal{R}_{k-1}(J;l) \cup \{0\}, \tag{19f}$$

$$x \geq 0, \quad x \in \mathbb{R}. \tag{19g}$$

The LP formulation (19) is used to determine lower bounds $l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)}$ $(J \in \mathcal{R}_k(l))$. In the next section, the approximation of expected penalty cost, $\{L_{s_k}^J \,|\forall s_k \in \Psi, J \in \mathcal{R}_k(l)\}$, is calculated.

5 Lower bound of the expected penalty

The expected cost-to-go value for each state s_k $(s_k \in \Psi)$ is approximated according to (17). d_{lJ} is the distance between current location l and the potential next destination $J \in \mathcal{R}_k(l)$. $l_{tsp}^{J,\mathcal{R}_{k-1}(J;l)}$ is the lower bound of the corresponding TSP problem, which can be obtained from formulation (19). Therefore, to approximate the cost-to-go value, only the lower bound of expected penalty cost $L_{s_k}^J$ is required.

Section 5.1 determines the optimality equation for expected penalty costs, and then the MDP formulation is reformulated using the linear program counterpart. Section 5.2 provides the ALP formulation for penalty costs by defining the affine functions. Section 5.3 proposes a method for solving the ALP formulation.

5.1 Expected penalty cost

$L_{s_k}^J$ $(\forall s_k = (l, q, \mathcal{R}_k(l)) \in \Psi,\, J \in \mathcal{R}_k(l))$ are the lower bounds of expected penalty costs $f_k^J(l, q, \mathcal{R}_k(l))$. The expected penalty cost of visiting customer J next for each state s_k satisfies the following optimality equation

$$f_k^J(l, q, \mathcal{R}_k(l)) = \min \begin{cases} \Delta_{lJ} + \sum_e p_J(e) \cdot f_{k-1}(J, Q-e, \mathcal{R}_{k-1}(J;l)), & u_{J,l,\mathcal{R}_k(l)}(q) = 1, \\ \sum_{e \leq q} p_J(e) \cdot f_{k-1}(J, q-e, \mathcal{R}_{k-1}(J;l)) + b \cdot \sum_{e>q} p_J(e) \cdot (e-q) + \sum_{e>q} p_J(e) \cdot f_{k-1}(J, 0, \mathcal{R}_{k-1}(J;l)), & u_{J,l,\mathcal{R}_k(l)}(q) = 0, \end{cases}$$
$$\forall s_k = (l, q, \mathcal{R}_k(l)) \in \Psi, J \in \mathcal{N} \backslash l, k \in \Omega \backslash \{N, 0\}, \tag{20}$$

where replenishment decision $u_{J,l,\mathcal{R}_k(l)}(q)$ for each state s_k is made to minimize the future expected penalty costs, either with $(u_{J,l,\mathcal{R}_k(l)}(q) = 1)$ or without $(u_{J,l,\mathcal{R}_k(l)}(q) = 0)$ replenishing before arriving at customer J. If decision $u_{J,l,\mathcal{R}_k(l)}(q) = 1$ is made, the vehicle arrives at customer J with full capacity Q and with the expense of additional travel cost Δ_{lJ} $(\Delta_{lJ} = d_{l0} + d_{0J} - d_{lJ})$. Otherwise, decision $u_{J,l,\mathcal{R}_k(l)}(q) = 0$ dictates that the vehicle proceeds to customer J, with probability $\sum_{e>q} p_J(e)$ of failing the service of customer J and leaving empty vehicle capacity.

Note that $f_{k-1}(J, q', \mathcal{R}_{k-1}(J;l))$ equals to $\min_{j' \in \mathcal{R}_{k-1}(J;l)} \left\{ f_{k-1}^{j'}(J, q', \mathcal{R}_{k-1}(J;l)) \right\}$, indicating that $f_{k-1}(J, q', \mathcal{R}_{k-1}(J;l))$ is the minimal expected penalty cost among all next routing decisions j' in set $\mathcal{R}_{k-1}(j';l)$. In the following, $f_k(l, q, \mathcal{R}_k(l))$ $(f_k^j(l, q, \mathcal{R}_k(l)))$ is substitued with $f_{l,\mathcal{R}}(q)$ $(f_{l,\mathcal{R}}^j(q))$ for ease of notation.

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.

Equation (20) is expressed by the following LP formulation (21)

$$
\begin{aligned}
\max \quad & f_{0,\mathcal{N}}(Q) && && \text{(21a)}\\
s.t. \quad & f^j_{0,\mathcal{N}}(Q) \le \sum_e p_j(e)\cdot f_{j,\mathcal{N}\setminus j}(Q-e), && \forall j \in \mathcal{N}, && \text{(21b)}\\
& f^j_{l,\mathcal{R}}(q) \le \Delta_{lj} + \sum_e p_j(e)\cdot f_{j,\mathcal{R}\setminus l}(Q-e), && \forall l \in \mathcal{N}, j \in \mathcal{R} \subseteq \mathcal{N}\setminus l, \\
& && q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}(|\mathcal{R}| \in \{N-1, N-2, \ldots, 2\}), && \text{(21c)}\\
& f^j_{l,\mathcal{R}}(q) \le \sum_{e\le q} p_j(e)\cdot f_{j,\mathcal{R}\setminus l}(q-e) \\
& \qquad + b\cdot \sum_{e>q} p_j(e)\cdot(e-q) + \sum_{e>q} p_j(e)\cdot f_{j,\mathcal{R}\setminus l}(0), && \forall l \in \mathcal{N}, j \in \mathcal{R} \subseteq \mathcal{N}\setminus l, \\
& && q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}(|\mathcal{R}| \in \{N-1, N-2, \ldots, 2\}), && \text{(21d)}\\
& f^j_{l,\{j\}}(q) \le \Delta_{lj}, && \forall l \in \mathcal{N}, j \in \mathcal{N}\setminus l, \mathcal{R} = \{j\}, q \in \mathcal{Q}^{fe}_1, && \text{(21e)}\\
& f^j_{l,\{j\}}(q) \le b\cdot \sum_{e>q} p_j(e)\cdot(e-q), && \forall l \in \mathcal{N}, j \in \mathcal{N}\setminus l, \mathcal{R} = \{j\}, q \in \mathcal{Q}^{fe}_1, && \text{(21f)}\\
& f^j_{l,\mathcal{R}}(q), f_{l,\mathcal{R}}(q), f^j_{0,\mathcal{N}}(Q), f_{0,\mathcal{N}}(Q) \in \mathbb{R}, && \forall l \in \mathcal{N}, j \in \mathcal{R} \subseteq \mathcal{N}\setminus l, \\
& && q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}(|\mathcal{R}| \in \{N-1, N-2, \ldots, 1\}). && \text{(21g)}
\end{aligned}
$$

Constraints (21b) indicate that the vehicle departs from the depot with full capacity Q at initial stage N, and its only option is to proceed to the first customer directly. Constraints (21c) and (21d) translate Equation (20) correspondingly and capture the transition of expected penalty costs from stage $N-1$ to 2. Constraints (21e) and (21f) state the situation at stage 1 when any expected penalty cost at final stage 0, $f_{l,\phi}(q)$ ($\forall l \in \mathcal{N}$, $q \in \mathcal{Q}^{fe}_0$), is zero. In addition, $q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}$ ($|\mathcal{R}| = N-1, \ldots, 1$) in constraints (21c)–(21f) indicates the feasible range of residual capacity q at each stage k ($k = |\mathcal{R}| = N-1, \ldots, 1$), and $\mathcal{Q}^{fe}_k$ ($\mathcal{Q}^{fe}_{|\mathcal{R}|}$) is $[(Q-(N-k)\cdot E)^+, Q-e_{\min}]$ for each stage k, where $e_{\min}$ is the minimal amount that demand ξ takes.

In LP formulation (21), the variables are $f^j_{l,\mathcal{R}}(q)$ and $f_{l,\mathcal{R}}(q)$ ($l \in \mathcal{N} \cup 0$, $j \in \{\mathcal{N} \cup 0\}\setminus l$, $\mathcal{R} \subseteq \{\mathcal{N} \cup 0\}\setminus l$, $q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}$). Under objective (21a), the optimal solution $f^j_{l,\mathcal{R}}(q)^*$ ($f_{l,\mathcal{R}}(q)^*$) is the largest lower bound of each expected penalty cost $f^j_{l,\mathcal{R}}(q)$ ($f_{l,\mathcal{R}}(q)$).

5.2 Affine approximation for lower bound

Solving formulation (21) can be inefficient due to its many variables. Variables $f^j_{l,\mathcal{R}}(q)$ and $f_{l,\mathcal{R}}(q)$ amount to the scale of $O(N^2\cdot 2^{N-1}\cdot Q)$ (calculated by $N^2\cdot(\mathrm{C}^1_{N-1}+\mathrm{C}^2_{N-1}+,\ldots,+\mathrm{C}^{N-1}_{N-1})\cdot(Q+1)+1$). Thus, a variable reduction method is applied to reduce the dimensionality of the model.

Variable reduction is firstly achieved by noting that $f_{l,\mathcal{R}}(q) = \min_{j\in\mathcal{R}}\left\{f^j_{l,\mathcal{R}}(q)\right\}$, meaning that constraints (22) are added in formulation (21).

$$
f_{l,\mathcal{R}}(q) \le f^j_{l,\mathcal{R}}(q), \qquad \forall j \in \mathcal{R} \subseteq N\setminus l \quad \left(l \in 0 \cup \mathcal{N}, q \in \mathcal{Q}^{fe}_k, k \in \{N, N-1, \ldots, 1\}\right) \tag{22}
$$

Formulation (23) is then obtained

$$
\begin{aligned}
\max \quad & f_{0,\mathcal{N}}(Q) && && \text{(23a)}\\
s.t. \quad & f_{0,\mathcal{N}}(q) \le \sum_e p_j(e)\cdot f_{j,\mathcal{N}\setminus j}(Q-e), && \forall j \in \mathcal{N}, && \text{(23b)}\\
& f_{l,\mathcal{R}}(q) \le \Delta_{lj} + \sum_e p_j(e)\cdot f_{j,\mathcal{R}\setminus l}(Q-e), && \forall l \in \mathcal{N}, j \in \mathcal{R} \subseteq \mathcal{N}\setminus l, \\
& && q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}\big(|\mathcal{R}| \in \{N-1, N-2, \ldots, 2\}\big), && \text{(23c)}
\end{aligned}
$$

$$f_{l,\mathcal{R}}(q) \leq \sum_{e \leq q} p_j(e) \cdot f_{j,\mathcal{R}\backslash j}(q-e) + b \cdot \sum_{e>q} p_j(e) \cdot (e-q) + \sum_{e>q} p_j(e) \cdot f_{j,\mathcal{R}\backslash j}(0), \quad \forall l \in \mathcal{N}, j \in \mathcal{R} \subseteq \mathcal{N}\backslash l, q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}\big(|\mathcal{R}| \in \{N-1, N-2, \ldots, 2\}\big), \tag{23d}$$

$$f_{l,\{j\}}(q) \leq \Delta_{lj}, \quad \forall l \in \mathcal{N}, j \in \mathcal{N}\backslash l, \mathcal{R} = \{j\}, q \in \mathcal{Q}^{fe}_1, \tag{23e}$$

$$f_{l,\{j\}}(q) \leq b \cdot \sum_{e>q} p_j(e) \cdot (e-q), \quad \forall l \in \mathcal{N}, j \in \mathcal{N}\backslash l, \mathcal{R} = \{j\}, q \in \mathcal{Q}^{fe}_1, \tag{23f}$$

$$f_{l,\mathcal{R}}(q), f_{0,\mathcal{N}}(Q) \in \mathbb{R}, \quad \forall l \in \mathcal{N}, q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}\big(|\mathcal{R}| \in \{N-1, N-2, \ldots, 1\}\big). \tag{23g}$$

The number of variables is decreased, with only variables $f_{l,\mathcal{R}}(q)$ included in the formulation. Variable size can be further reduced by identifying a set of bases and substituting their affine form for the original variables in the formulation. We derive the affine function forms as in (24a)–(24c) to approximate the expected penalty costs. The affine functions are tailored to our problem structure. The affine functions are adapted from the forms in extant research [e.g., in 37, 38].

$$f_{0,\mathcal{N}}(Q) \approx \theta_{0,Q,0}, \tag{24a}$$

$$f_{l,\mathcal{R}}(q) \approx \theta_{l,q,0} + \sum_{j \in \mathcal{R}} (\alpha_{l,q,j} \cdot \Delta_{l,j} + \omega_{l,j} \cdot q), \quad l \in \mathcal{N}, j \in \mathcal{R} \subseteq \mathcal{N}\backslash l, |\mathcal{R}| \geq 2, q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}, \tag{24b}$$

$$f_{l,\{j\}}(q) \approx \theta_{l,q,0} + \alpha_{l,q,j} \cdot \Delta_{lj} + \omega_{l,j} \cdot q, \quad l \in \mathcal{N}, j \in \mathcal{N}\backslash l, \mathcal{R} = \{j\}. \tag{24c}$$

where $\theta \in \mathbb{R}^{N \cdot Q+N+1}$, $\alpha \in \mathbb{R}^{(N^2-N)\cdot(Q+1)}$ and $\omega \in \mathbb{R}^{N^2-N}$. Affine functions (24a)–(24c) approximate expected penalty cost $f_{l,\mathcal{R}}(q)$. Firstly, penalty costs occur because restocking actions are taken or outsourcing occurs, so terms $\alpha_{l,q,j} \cdot \Delta_{l,j}$ and $\omega_{l,j} \cdot q$ are introduced, respectively. $\alpha_{l,q,j} \cdot \Delta_{lj}$ implies that restocking contributes to an additional travel cost Δ_{lj} ($j \in \mathcal{R} \subseteq \mathcal{N}\backslash l$), and $w_{l,j} \cdot q$ indicates that the outsourcing cost is relevant to available residual capacity q, whereas constant θ adjusts the approximation of the penalty cost. Secondly, expected penalty costs are determined based on the states ($\forall s_k = \big(l, q, \mathcal{R}\big)$), so parameters θ, α and ω are defined concerning the states. Note that ω is only relevant to current location l and remaining unvisited customer j ($j \in \mathcal{R}$), considering that residual capacity q is already reflected as a multiplier factor in term $\omega_{l,j} \cdot q$. Term $\omega \cdot q$ reflects the monotonicity of penalty cost given residual capacity q, where the proof is indicated in Appendix A. Lastly, (24b) reflects that the penalty cost is attributed to visiting different next possible customer $j \in \mathcal{R}$ when more than one customer is included in the remaining unvisited set ($|\mathcal{R}| \geq 2$). With this approximation (24a)–(24c), formulation (23) becomes

$$\max \quad \theta_{0,0,Q} \tag{25a}$$

$$\text{s.t.} \quad \theta_{0,0,Q} \leq \sum_{e} p_j(e) \cdot \theta_{j,0,Q-e} + \sum_{t \in \mathcal{N}\backslash j} \Delta_{jt} \cdot \Big(\sum_{e} p_j(e) \cdot \alpha_{j,t,Q-e}\Big) + \sum_{t \in \mathcal{N}\backslash j} \omega_{jt} \cdot \Big(\sum_{e} p_j(e) \cdot (Q-e)\Big), \quad \forall j \in \mathcal{N}, \tag{25b}$$

$$\theta_{l,0,q} + \sum_{t \in \mathcal{R}} (\Delta_{lt} \cdot \alpha_{l,t,q} + \omega_{lt} \cdot q) \leq \Delta_{lj} + \sum_{e} p_j(e) \cdot \theta_{j,0,Q-e} + \sum_{t \in \mathcal{R}\backslash j} \Delta_{jt} \cdot \Big(\sum_{e} p_j(e) \cdot \alpha_{j,t,Q-e}\Big) + \sum_{t \in \mathcal{R}\backslash j} \omega_{jt} \cdot \Big(\sum_{e} p_j(e) \cdot (Q-e)\Big), \quad \forall l \in \mathcal{N}, j \in \mathcal{R} \subseteq \mathcal{N}\backslash l, q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}\big(|\mathcal{R}| \in \{N-1, N-2, \ldots, 2\}\big), \tag{25c}$$

$$\theta_{l,0,q} + \sum_{t\in\mathcal{R}} (\Delta_{lt} \cdot \alpha_{l,t,q} + \omega_{lt} \cdot q) \le \Big(\sum_{e\le q} p_j(e) \cdot \theta_{j,0,q-e} + \sum_{e>q} p_j(e) \cdot \theta_{j,0,0}\Big) + b \cdot \sum_{e>q} p_j(e)\cdot(e-q) + \sum_{t\in\mathcal{R}\setminus j} \Delta_{jt} \cdot \Big(\sum_{e\le q} p_j(e)\cdot \alpha_{j,t,q-e} + \sum_{e>q} p_j(e) \cdot \alpha_{j,t,0}\Big) + \sum_{t\in\mathcal{R}\setminus j} \omega_{jt} \cdot \Big(\sum_{e\le q} p_j(e)\cdot(q-e)\Big) \quad \forall l \in \mathcal{N}, j \in \mathcal{R} \subseteq \mathcal{N}\setminus l, \; q \in \mathcal{Q}^{fe}_{|\mathcal{R}|} \big(|\mathcal{R}| \in \{N-1, N-2\ldots,2\}\big), \tag{25d}$$

$$\theta_{l,0,q} + \Delta_{lj} \cdot \alpha_{l,j,q} + \omega_{lj}\cdot q \le \Delta_{lj}, \quad \forall l \in \mathcal{N}, j \in \mathcal{N}\setminus l, q \in \mathcal{Q}^{fe}_1, \tag{25e}$$

$$\theta_{l,0,q} + \Delta_{lj} \cdot \alpha_{l,j,q} + \omega_{lj}\cdot q \le b \cdot \sum_{e>q} p_j(e)\cdot(e-q), \quad \forall l\mathcal{N}, j \in \mathcal{N}\setminus l, q \in \mathcal{Q}^{fe}_1, \tag{25f}$$

$$\theta, \alpha, \omega \in \mathbb{R}. \tag{25g}$$

In formulation (25), variables include θ, α and ω, reaching the scale of $O(N^2 \cdot Q)$ (calculated by $N^2 \cdot (Q+1) + N^2 - N + 1$). The variable size is dramatically reduced concerning its original size in formulation (23a)–(23g). In the following subsection, the method to solve ALP formulation (25) is introduced.

5.3 Constraint sampling

Formulation (25) reduces variables to a manageable size. However, the number of constraints is still too large to solve. We employ a constraint sampling approach [10] 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 [10].

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 [26]. 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, we prepare a set of heuristic policies 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, a sequence of states and actions is obtained in the form

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

where $\{\xi\}^{sam}$ denotes a sample of realized customer demand $\{\xi_l \,|l \in \mathcal{N}\}$. $a_{s_{k|pl}} = (j, u_{j,l,\mathcal{R}}(q))$ 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 $s_k = \big(l, q, \mathcal{R}\big)$ transits to state $s_{k-1} = (j, [q + (Q-q)\cdot u_{j,l,\mathcal{R}}(q) - \xi_j]^+, \mathcal{R}\setminus j)$ depending on action $a_{s_{k|pl}} = (j, u_{j,l,\mathcal{R}}(q))$ 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 $\mathrm{P_{s-a}} = \sum_{\substack{pl\in \mathrm{Pls} \\ s' \leftarrow s, a_{s_k|pl}}} (s, a_{s|pl})$.

Each state-action pair determines a group of constraints. For example, state-action pair $(s_k, a_{s_k}) = \big(l, q, \mathcal{R}, j, u_{j,l,\mathcal{R}}(q)\big)$ (if $|\mathcal{R}| \geq 2$, $\mathcal{R} \neq \mathcal{N}$) implies that constraints (26) are selected.

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

Constraints (26) imply that visiting customer j next is regarded as a promising action when the vehicle is located at customer l and given unvisited customer set $\mathcal{R}$ ($j \in \mathcal{R}$). Note that all feasible residual capacities $q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}$ are included, and whether the restocking action taken ($u_{j,l,\mathcal{R}}(q) = 1$) or not ($u_{j,l,\mathcal{R}}(q) = 0$) is taken into account. All feasible residual capacities $q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}$ are considered since some residual capacities may not be observed on sampling. Also, constraints indicating with and without restocking are included to reveal the threshold-type restocking nature given different realized residual capacities. Similarly, when the state-action pair is $\big(l, q, \{j\}, j, u_{j,l,\{j\}}(q)\big)$ ($l \in \mathcal{N}$, $j \in \mathcal{N}\setminus l$, $\mathcal{R} = \{j\}$), the selection of constraints is discussed identically, indicated by (27).

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

As for a 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 is naturally the same form as in (25b). Therefore, the promising constraint set is formed. All promising constraints are selected based on promising state-action pairs $(s_k, a_{s_k|m}) \in \mathrm{P}_{\mathrm{s-a}}$, according to (26), (27) and (25b).

By solving ALP formulation (25) with the constraints sampled by state-action pairs $(s_k, a_{s_k|m}) \in \mathrm{P}_{\mathrm{s-a}}$, the values of parameters θ, α and ω are approximated. The lower bounds of expected penalty costs $L^J_{s_k}$ ($J \in \mathcal{R}$, $s_k = \big(l, q, \mathcal{R}\big) \in \Psi$) are subsequently obtained based on (28a)–(30).

$$\begin{aligned} L^J_{s_N} \approx & \sum_{e} p_J(e) \cdot \theta_{J,0,Q-e} + \sum_{t\in\mathcal{N}\setminus J} \Delta_{Jt} \cdot \Big(\sum_{e} p_J(e) \cdot \alpha_{J,t,Q-e}\Big) \\ & + \sum_{t\in\mathcal{N}\setminus J} \omega_{Jt} \cdot \Big(\sum_{e} p_J(e) \cdot (Q-e)\Big), \qquad s_N = (0, Q, \mathcal{N}), j \in \mathcal{N}, \end{aligned} \tag{28a}$$

$$L^J_{s_k} = \min\Big\{L^{R(J)}_{s_k}, L^{D(J)}_{s_k}\Big\} \approx \min \begin{cases} \Delta_{lJ} + \sum_{e} p_J(e) \cdot \theta_{J,0,Q-e} + \sum_{t\in\mathcal{R}\setminus j} \Delta_{Jt} \cdot \big(\sum_{e} p_J(e) \cdot \alpha_{J,t,Q-e}\big) \\ \qquad + \sum_{t\in\mathcal{R}\setminus J} \omega_{Jt} \cdot \big(\sum_{e} p_J(e) \cdot (Q-e)\big), \quad u_{J,l,\mathcal{R}}(q) = 1, \\ \big(\sum_{e\leq q} p_J(e) \cdot \theta_{J,0,q-e} + \sum_{e>q} p_J(e) \cdot \theta_{J,0,0}\big) + \sum_{t\in\mathcal{R}\setminus J} \Delta_{Jk} \\ \qquad \cdot \big(\sum_{e\leq q} p_J(e) \cdot \alpha_{J,t,Q-e} + \sum_{e>q} \cdot p_J(e) \cdot \alpha_{J,t,0}\big) \\ \qquad + \sum_{t\in\mathcal{R}\setminus J} \omega_{Jt} \cdot \big(\sum_{e\leq q} p_J(e) \cdot (q-e)\big) + b \cdot \sum_{e>q} p_J(e) \cdot (e-q), \quad u_{J,l,\mathcal{R}}(q) = 0, \\ \qquad \forall s_k = (l, q, \mathcal{R}) \in \Psi, \quad J \in \mathcal{N}\setminus l, \quad l \in \mathcal{N}, \quad q \in \mathcal{Q}^{fe}_k, \quad 2 \leq k \leq N-1, \end{cases} \tag{28b}$$

$$L_{s_1}^J \approx \min\left\{\Delta_{lJ}, b \cdot \sum_{e>q} p_J(e)\cdot(e-q)\right\}, \qquad \forall s_0 = (l, q, \{J\}) \in \Psi, J \in \mathcal{N}\backslash l, l \in \mathcal{N}, q \in \mathcal{Q}_1^{fe}. \tag{28c}$$

(28a)–(30) approximates lower bounds for each state s_k ($s_k \in \Psi$, $s_k \in \mathrm{P_{s-a}}$) along with each routing choice J ($J \in \mathcal{R}$). (28a)–(30) is derived based on (24a)–(24c) and (20). Values θ, α and ω are substituted into (24a)–(24c), to approximate each $f_{l,\mathcal{R}}(q)$ ($\forall l \in \mathcal{N}, \mathcal{R} \subseteq \mathcal{N}\backslash l, q \in Q_{|\mathcal{R}|}^{fe}, s \in \mathrm{P_{s-a}}$). Then, $L_{s_k}^J$, the lower bound of $f_{l,\mathcal{R}}^J(q)$, is obtained by substituting the approximation of $f_{l,\mathcal{R}}(q)$ into (20). Finally, expected cost-to-go values $V_k\big(l, q, \mathcal{R}\big)$ ($s_k \in \Psi$, $s_k \in \mathrm{P_{s-a}}$) are approximated by substituting $L_{s_k}^J$ into (17).

6 Price-directed policy

ALP methods can lead to price-directed policies [2, 4, 38]. If solving a minimum problem, any feasible solution of an ALP formulation generates lower bounds of value functions. The proof can be found in relevant articles [e.g., in 3, 38]. In a price-directed policy, lower bounds are utilized to approximate the value functions and then to guide decision-making for each realized state.

We develop an ALP method to approximate expected penalty costs. Based on (17), lower bounds for the value functions are generated. Our price-directed policy determines next routing location $j_k\big(l, q, \mathcal{R}\big)$ and makes restocking decision $u_{j,l,\mathcal{R}}(q)$ for state $s_k = \big(l, q, \mathcal{R}\big)$ ($s_k \in \Psi$, $k \in \Omega\backslash\{N, 0\}$, $s_k \in \mathrm{P_{s-a}}$) based on (29a)–(29b).

$$u_{J,l,\mathcal{R}}(q) = \begin{cases} 1, & \text{if} L_{s_k}^{R(J)} \le L_{s_k}^{D(J)} \\ 0, & \text{if} L_{s_k}^{D(J)} > L_{s_k}^{R(J)} \end{cases}, \quad \forall J \in \mathcal{R}, J \in \mathrm{P_{s-a}} \tag{29a}$$

$$j_k\big(l, q, \mathcal{R}\big) = \arg\min_{J\in\mathcal{R}, J\in \mathrm{P_{s-a}}} \left\{d_{lJ} + l_{tsp}^{J,\mathcal{R}\backslash J} + L_{s_k}^J\right\}, \quad \text{where } L_{s_k}^J = \min\left\{L_{s_k}^{D(J)}, L_{s_k}^{R(J)}\right\} \tag{29b}$$

For state $s_k = \big(l, q, \mathcal{R}\big)$ at stage $k \in \{N-1, \dots, 1\}$, (29a) is applied first to make the restocking decision for each routing choice $J \in \mathcal{R}$. Then, the optimal routing option $j_k\big(l, q, \mathcal{R}\big)$ is selected based on (29b), and restocking decision $u_{j,l,\mathcal{R}}(q)$ is made accordingly based on routing choice $j_k\big(l, q, \mathcal{R}\big)$. $L_{s_k}^{R(J)}$ and $L_{s_k}^{D(J)}$ are calculated as in (28b) or (30) for different states, when restocking occurs at customer J ($u_{J,l,\mathcal{R}}(q) = 1$) or not ($u_{J,l,\mathcal{R}}(q) = 0$), respectively. Additionally, (30) determines routing decision ($j_N(0, Q, \mathcal{N})$) at the beginning, when the vehicle departs from depot 0 with full capacity Q and goes directly to the first customer (i.e., $u_{j,0,\mathcal{N}}(Q) \equiv 0$), where $L_{s_N}^J$ is obtained from (28a).

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

Lastly, at final stage 0, the vehicle proceeds directly to the depot (i.e., $j_0\big(l, q, \phi\big) = 0$ and $u_{0,l,\phi}(q) = 0$).

Note that observed states s and routing decisions j are restricted by the promising state-action space $\mathrm{P_{s-a}}$ for the tractability of the corresponding ALP formulation. Theoretically, our price-directed policy can infinitely approach the optimal policy if the state-action space $\mathrm{P_{s-a}}$ is well-selected.

7 Computational study

This paper proposes a partial-outsourcing strategy and an ALP method is developed to compute it. The ALP method is employed to approximate value functions which are then used within the Bellman equation to derive a price-directed policy (PD). The solution quality of PD policy is demonstrated using a computationally effective a posteriori bound, and the method is further compared with other dynamic and static routing policies. The experimental results show the solution quality and computational cost of our method.

Instance generation

Instances are generated following the instance generation procedures used in Gendreau et al. [16]. Customer locations are randomly generated in 1,000 by 1,000 square. The depot is located at coordinates (0, 0) or (500, 500), labeled as *corner* and *midpoint*, respectively. Customer demand corresponds to a discrete uniform random variable with support $\{1, 2, \ldots, 5\}$, and the mean demand of any customer in any instance is 3. The problem size ranges from 10 to 40 customers in increments of 5. We consider instances with 40 customers or fewer for comparison with other benchmark approaches and for the computational difficulty of solving larger ones. This is in line with the sizes of instances considered in Toriello et al. [38]. The expected filling rate is determined by $\bar{f} = \sum_{l=1}^{N} E(\tilde{\xi}_l)\,/Q$, and takes 1.9, 2.5 or 3.4 to show different failure frequencies. Vehicle capacity Q is computed by rounding $3N/\bar{f}$ to the nearest integer.

Outsourcing price b is set to $\bar{\Delta}_{lj}\,/\sum_e (p_j(e)\cdot e)$, when the outsourcing cost $b\cdot\sum_e(p_j(e)\cdot e)$ equals to $\bar{\Delta}_{lj}$, where $\bar{\Delta}_{lj}$ is the average cost of a restocking trip. Price b reveals a threshold. If a higher price is incurred, restocking may save costs. Otherwise, it is better to outsource. $\bar{\Delta}_{lj}$ is the average restocking cost between consecutive customers on a feasible route; herein, a TSP route is used. Also, price b can fluctuate to reflect the network characteristics. 20 instances are generated for each combination of settings.

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 $\mathrm{P_{s-a}}$, 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 reported in Secomandi [32] and Bertsimas [6], 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.

Solution quality of PD policy

Performances are compared in terms of solution quality and solution time against the benchmark approaches in the literature, namely, the partial re-optimization (PR) [34], the one-step rollout algorithm (ORA) [32], the two-step rollout algorithm (TRA) [26], and the rollout algorithm (RA) [33]. The benchmark approaches are all adapted to generate solutions for the VRPSD with outsourcing. PR, realized by PH(10) [refers to 34], creates an effective a posteriori bound (denoted as B^{PR}). The comparisons with ORA and TRA exhibit the performance of various re-optimization methods. RA is introduced as a baseline approach to indicate the benefit of routing dynamically.

To indicate the solution quality, relative gap ε^{PR} and improvement rates γ^{RA}, γ^{ORA} and γ^{TRA} are introduced. ε^{PR} is used to evaluate the difference between the results obtained by PD policy and the posteriori bound ($\varepsilon^{PR} = \frac{V_N^{PD}(0,Q,\mathcal{N})-B^{PR}}{B^{PR}}$). γ^{RA}, γ^{ORA} and γ^{TRA} are used to reveal the improvement percentages of policies PD vs. RA ($\gamma^{RA} = \frac{V_N^{PD}(0,Q,\mathcal{N})-V_N^{RA}(0,Q,\mathcal{N})}{V_N^{RA}(0,Q,\mathcal{N})}$), policies PD vs. ORA ($\gamma^{ORA} = \frac{V_N^{PD}(0,Q,\mathcal{N})-V_N^{ORA}(0,Q,\mathcal{N})}{V_N^{ORA}(0,Q,\mathcal{N})}$), and policies PD vs. TRA ($\gamma^{TRA} = \frac{V_N^{PD}(0,Q,\mathcal{N})-V_N^{TRA}(0,Q,\mathcal{N})}{V_N^{TRA}(0,Q,\mathcal{N})}$), respectively. By implementing policies PD, PR, RA, ORA and TRA, the actual costs are observed for each problem setting. 20 instances are generated for each problem setting. The results in each line of Tables 1 and 2 reveal the averages over the 20 instances in a set. Tables 1 and 2 indicate the cases where the depot is located at the midpoint and in the corner, respectively. The problem number with an asterisk(*) indicates the problem setting when our approach performs the best.

Table 1: Total costs based on different approaches (midpoint depot)

problem No.	customers & capacity (N,Q)	method PD	bound B^{PR}	method TRA	method ORA	method RA	gap ε^{PR}	rate γ^{TRA}	rate γ^{ORA}	rate γ^{RA}
*1	(10,16)	4065.21	4066.06	4522.68	4455.09	4372.1	-0.02%	-10.12%	-8.75%	-7.02%
*3	(15,24)	4188.86	4196.78	4094.76	4247.62	4247.62	-0.19%	2.30%	-1.38%	-1.38%
5	(20,24)	4343.21	4003.77	4363.64	4598.18	4481.84	8.48%	-0.47%	-5.55%	-3.09%
7	(25,30)	5587.74	5471.54	5646.72	5583.55	5583.55	2.12%	-1.04%	0.08%	0.08%
9	(30,36)	5378.2	5271.42	5631.77	5663.46	5426.35	2.03%	-4.50%	-5.04%	-0.89%
11	(35,31)	5722.17	5646.82	6077.37	6112.4	6112.41	1.33%	-5.84%	-6.38%	-6.38%
13	(40,35)	6107.5	5922.32	6456.65	6317.32	6317.32	3.13%	-5.40%	-3.32%	-3.32%

Table 2: Total costs based on different approaches (corner depot)

problem No.	customers & capacity (N,Q)	method PD	bound B^{PR}	method TRA	method ORA	method RA	gap ε^{PR}	rate γ^{TRA}	rate γ^{ORA}	rate γ^{RA}
2	(10,16)	5908.02	5863.57	6628.63	7174.42	7309.71	0.76%	-10.87%	-17.65%	-19.18%
4	(15,24)	6256.74	6149.11	6366	6157.94	6181.17	1.75%	-1.72%	1.60%	1.22%
*6	(20,24)	7018.32	7106.39	7925.02	7947.54	7962.74	-1.24%	-11.44%	-11.69%	-11.86%
8	(25,30)	7798.34	7426.21	8158.41	8310.55	8348.96	5.01%	-4.41%	-6.16%	-6.60%
*10	(30,36)	7849.41	7991.7	9690.45	9752.51	9326.71	-1.78%	-19.00%	-19.51%	-15.84%
12	(35,31)	9560.66	9442.15	9811.08	9566.41	9592.3	1.26%	-2.55%	-0.06%	-0.33%
14	(40,35)	10547.54	9954.79	10812.76	10929.05	10936.33	5.95%	-2.45%	-3.49%	-3.56%

Our method exhibits good performance. The solution quality is demonstrated using posterior bound B^{PR}. A smaller gap of less than 10% is observed for our ALP approach versus the posterior bound B^{PR}. A similar finding was observed by Torriello et al. [38] in their study of a TSP problem. The gap between their ALP approach and the posteriori bound was within 20%. Their bound is comparable to bound B^{PR}, which is determined by adapting the PR method to their problem context. Additionally, our policy (PD) generally outperforms the two dynamic routing policies (ORA and TRA) and the fixed routing policy (RA).

The results demonstrate the potential of our algorithm in solving VRPSD. The competitive performances of our policy versus PR are shown in problem settings 10, 6, 3, and 1. Our algorithm is realized by incorporating enhancement techniques, including decomposition-based value function approximations, lower bounding procedures based on affine functions, and constraint sampling. It should be noted that a simple and intuitive constraint sampling is in use in our framework. This constraint sampling approach exploits a local solution space restricted by current heuristic policies, which might impede the performance of our method. A better version of constraint sampling is to exploit the entire space. The encouraging results from problem settings 10 and 6 make us optimistic about the potential of our solution framework for solving VRPSD.

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. The computational time for solving the problem with different settings is discussed in the following.

Computational costs

The total time for PD policy comprises two parts, i.e., the pre-compute time (indicated by prep.) and the time for implementation (indicated by imple.). The pre-compute time mainly consists of the time required to prepare constraints (i.e., the formation of the state-action pool) and the time required to choose the best subset of constraints (see Appendix A). As shown in Tables 3 and 4, it takes an almost equal amount of computational effort to obtain the solution by PD policy as it does to obtain the posteriori bound by PR policy.

Table 3: CPU times of different approaches in seconds (midpoint depot)

problem No.	customers & capacity (N,Q)	method PD			bound B^{PR}	method TRA	method ORA	method RA	time(PD)/ time(B^{PR})
		prep.	imple.	total					
1	(10,16)	1657.22	5.34	1662.56	1612.07	209.83	7.94	3.28	103.13%
3	(15,24)	4501.26	10.81	4512.07	4480.33	1063.96	23.76	5.38	100.71%
5	(20,24)	8254.44	25.78	8280.22	8187.7	1827.99	52.06	19.05	101.13%
7	(25,30)	13197.41	37.82	13235.23	12655.04	4550.32	95.08	27.11	104.58%
9	(30,36)	36873.15	229.15	37102.3	33943.47	8277.56	166.59	49.18	109.31%
11	(35,31)	32546.75	170.37	32717.12	31928.81	8008.07	286.81	143.74	102.47%
13	(40,35)	37249.45	176	37425.45	35754.07	11883.46	393.28	184.03	104.67%

Table 4: CPU times of different approaches in seconds (corner depot)

problem No.	customers & capacity (N,Q)	method PD			bound B^{PR}	method TRA	method ORA	method RA	time(PD)/ time(B^{PR})
		prep.	imple.	total					
2	(10,16)	5677.84	6.29	5684.13	5619.02	235.3	23.56	18.84	101.16%
4	(15,24)	4571.56	28.09	4599.65	4472.1	1050.55	22.44	4.23	102.85%
6	(20,24)	9028.58	29.83	9058.41	8705.63	2114.29	60.86	27.69	104.05%
8	(25,30)	13081.9	59.18	13141.08	12425.25	4452.79	105.11	34.95	105.76%
10	(30,36)	37971.87	76.65	38048.52	35572.98	8127.48	174.37	52	106.96%
12	(35,31)	23957.89	143.37	24101.26	23291.4	9938.88	306.63	162.09	102.86%
14	(40,35)	50101.47	348.81	50450.28	48165.15	14865.93	579.09	367.47	104.74%

Solution quality of PD policy in time limits

Computational efficiency is further improved by limiting runtimes for solving the ALP formulation (25) to see if the performance of our algorithm noticeably degrades. The performance of PD policy is captured at different runtimes (at 1 minute, 5 minutes, 10 minutes, 30 minutes, and 1 hour), and comparisons are drawn with posteriori bound B^{PR}. Tables 5 shows the resulting gaps and indicates that our algorithm can yield good-quality solutions within limited runtimes. The relative gaps are reduced along with the increased time limits and achieve within 15% after a runtime of 30 minutes.

Sensitivity analysis of outsourcing price

In the sensitivity analysis, outsourcing price b fluctuates; the mean ratios of outsourcing to restocking are computed. In Table 6, b^0 refers to the initial price, taking value $\bar{\Delta}_{lj} / \sum_e (p_j(e) \cdot e)$. The price increases ($\beta \cdot b^0, \beta = 1.2, 1.5, 1.9$) or decreases ($\beta \cdot b^0, \beta = 0.8, 0.5, 0.1$) to reflect the influence of the outsourcing price on decision-making. As shown in Table 6, outsourcing frequency generally increases as the outsourcing price drops, which is consistent with intuition. The ratios sometimes do not change following the general trend, indicating the decisions are ultimately made by $\min\left\{V_k^{D(j)}\big(l, q, \mathcal{R}_k(l)\big), V_k^{R(j)}\big(l, q, \mathcal{R}_k(l)\big)\right\}$ based on Equation (4). Uniform demand is considered in the analysis, and outsourcing price is altered.

Table 5: Relative gaps ε^{PR} at different runtimes

Problem set No.	1 min	5 min	10 min	30 min	1 hour
1	-0.02%	-0.02%	-0.02%	-0.02%	-0.02%
2	1.03%	1.03%	1.03%	1.03%	1.03%
3	7.18%	7.18%	7.18%	7.18%	7.18%
4	5.44%	5.44%	5.44%	5.44%	5.44%
5	8.48%	8.48%	8.48%	8.48%	8.48%
6	8.04%	8.04%	-0.26%	-0.26%	-0.26%
7	3.83%	2.92%	2.92%	2.92%	2.92%
8	15.20%	15.20%	13.35%	13.35%	10.02%
9	2.63%	2.63%	2.63%	2.63%	2.63%
10	11.41%	4.14%	4.14%	4.14%	4.14%
11	1.34%	1.34%	1.34%	1.34%	1.34%
12	1.26%	1.26%	1.26%	1.26%	1.26%
13	3.51%	3.51%	3.51%	3.51%	3.51%
14	17.37%	17.37%	7.43%	5.95%	5.95%

Table 6: Sensitivity analysis of outsourcing price b

Problem set No.	$0.1{\cdot}b^0$	$0.5{\cdot}b^0$	$0.8{\cdot}b^0$	b^0	$1.2{\cdot}b^0$	$1.5{\cdot}b^0$	$1.8{\cdot}b^0$
1	Inf	1.38	1.15	1.15	1.05	0.63	0.78
2	Inf	1.10	0.65	0.33	0.33	0.33	0.23
3	Inf	0.38	0.28	0.28	0.28	0.13	0.03
4	Inf	0.50	0.40	0.10	0.10	0.13	0.10
5	Inf	Inf	Inf	Inf	Inf	Inf	Inf
6	1.56	0.68	0.64	0.61	0.58	0.58	0.64
7	0.71	0.16	0.09	0.08	0.04	0.02	0.04
8	Inf	0.93	0.33	0.30	0.28	0.25	0.15
9	0.58	0.50	0.20	0.15	0.12	0.09	0.08
10	5.83	0.36	0.20	0.23	0.23	0.10	0.17
11	9.62	4.67	4.53	3.78	4.62	4.57	3.59
12	1.79	1.30	1.37	1.18	1.07	0.35	0.31
13	6.55	5.41	2.38	2.93	1.53	1.56	0.92
14	14.03	0.55	0.13	0.09	0.09	0.07	0.01

8 Conclusions

The paper studies the VRPSD with re-optimization concerning the single-vehicle situation. A recourse policy in which unmet demands are outsourced to other carriers is proposed. The partial-outsourcing strategy is developed. The strategy is formulated with an MDP formulation, and an ALP method is developed to solve it. Lower bounds of value functions are generated based on the proposed decomposition-based ALP framework and are used to guide decision-making to obtain a price-directed policy.

The multi-policy sampling framework is developed to select constraints for ALP formulation. Based on the proposed constraint sampling, the sub-optimal solution spaces discovered by different heuristic policies are considered using an integrated approach. The experimental results show that our approach generally yields competitive solutions to the VRPSD.

The proposed approach can be implemented to tackle the scenarios in a reoptimization scheme when the dispatcher optimally makes the adaptive routing decision while considering the trade-off between the routing and outsourcing (or penalty) costs. It is worth mentioning that our approach is developed under a specific application background, in which outsourcing (or penalty) cost is amounted by the demand outsourced. Further efforts can be made to enhance the approach's applicability by considering penalties for the unpunctuality of delivery and outsourcing carriers' spatial and temporal availability.

Future efforts can be spent on refining the solution methodology, to improve its ability to solve larger problems. The ALP approach is theoretically appealing, providing a computationally tractable solution framework. In this solution framework, an LP formulation is solved, and then, the approximations of value functions are made. The decisions can be determined in real-time based on the approximated value functions that have been predetermined. To further improve the approach in solving larger problems and scaling its ability to approximate the value functions, one should devise a heuristic that can potentially be based on a reduced feasible space or implemented in a rolling fashion (i.e., using a restricted set of common parameters and their associated constraints, or adding them progressively).

Future efforts can also be spent on enhancing the approach with machine learning techniques. One possible way to enhance our approach using machine learning techniques is to enhance the ALP method with constraint selection. Besides, it is also interesting to introduce outsourcing recourse strategies to other routing problems, such as two-echelon logistics systems and inventory routing, to which problem contexts the outsourcing recourse strategies can be adapted.

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-type policies, two ORA-type policies, two TRA-type policies, a category of a priori optimization policies, and a myopic policy.

In the candidate policies, the value functions are either computed originally as their methods indicate or based on the decomposition framework as in (17). For example, partial re-optimization (PR) [34] is applied as one heuristic policy. Another PR-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 (17). 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) [34], one-step rollout algorithm (ORA) [32], two-step rollout algorithm (TRA) [26] and rollout algorithm (RA) [33] 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 [33], a variant of the rollout static method (i.e., based on decomposition as in (17)) and the TSP method [8, 20, 27].

In addition, 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.

Overall, ten candidate heuristic policies are included in our setting, called par-reopt, par-reopt-de, rollout-dynamic, rollout-dynamic-de, two-rollout-dynamic, two-rollout-dynamic-de, rollout-static, rollout-static-de, TSP and myopic policies, 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 (20). 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,

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

Situation (4) If $u_{j,l,\mathcal{R}}(q_2) = 1$ and $u_{j,l,\mathcal{R}}(q_1) = 0$, then,

$$
f^{j(D)}_{l,\mathcal{R}}(q_1) \leq f^{j(R)}_{l,\mathcal{R}}(q_1) = f^{j(R)}_{l,\mathcal{R}}(q_2),
$$

and

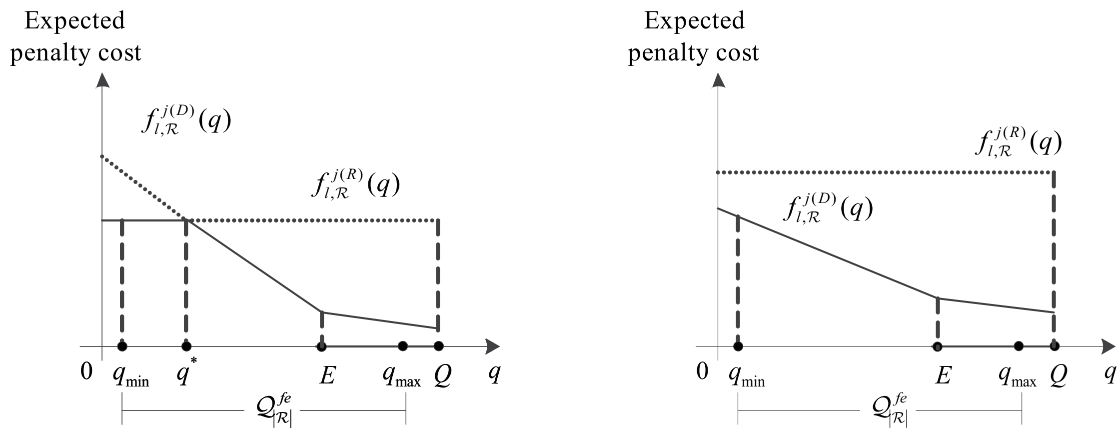
$$
\begin{aligned}
f^{j(D)}_{l,\mathcal{R}}(q_1) &= \sum_{e \leq q_1} p_j(e) \cdot f_{j,\mathcal{R}\setminus j}(q_1 - e) + b \cdot \sum_{e > q_1} p_j(e) \cdot (e - q_1) + \sum_{e > q_1} p_j(e) \cdot f_{j,\mathcal{R}\setminus j}(0) \\
&= \sum_{e \leq q_2} p_j(e) \cdot f_{j,\mathcal{R}\setminus j}(q_1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

□

B.2 Possible threshold-type replenishment

For particular customer l^* 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 $q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}$.

Because of the monotonicity of the expected penalty cost function, if $f^{j(D)}_{l,\mathcal{R}}(q_{\min}) > f^{j(R)}_{l,\mathcal{R}}(q_{\min})$, the decision for replenishing or proceeding to the next customer directly is of threshold-type in available capacity $q \in \mathcal{Q}^{fe}_{|\mathcal{R}|}$, as shown in Fig. B1(a). ($q_{\min} \in \mathcal{Q}^{fe}_{|\mathcal{R}|}$ is the minimum feasible residual capacity when $|\mathcal{R}|$ number of customers remain unvisited.) 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).



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

Figure B1: Possible threshold-type replenishment

Outsourcing price b appears to influence whether situation (a) or (b) happens. Visiting the next customer directly is always welcomed if the price is low enough. However, outsourcing is normally priced so that restocking is preferred in some circumstances and can be avoided in others. We should note that the expected penalty cost function can be piece-wise linear because of the threshold type, and the max customer demand is lower than vehicle capacity $E < Q$.

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 $\big(l, q, \mathcal{R}_k(l)\big)$, representing the vehicle departs from customer l with residual capacity q and set of unvisited customers $\mathcal{R}_k(l)$
$V_k\big(l, q, \mathcal{R}_k(l)\big)$	Cost-to-go value at state $\big(l, q, \mathcal{R}_k(l)\big)$
$j_k\big(l, q, \mathcal{R}_k(l)\big)$	Optimal routing decision, customer j, given state $\big(l, q, \mathcal{R}_k(l)\big)$
$u_{j,l,\mathcal{R}_k(l)}(q)$	Restocking decision at state $\big(l, q, \mathcal{R}_k(l)\big)$ if routing customer j next
$f_k^j\big(l, q, \mathcal{R}_k(l)\big)$	Expected penalty cost for routing customer j next at state $\big(l, q, \mathcal{R}_k(l)\big)$, $f_{l,\mathcal{R}}^j(q)$ for ease of notation
$f_k\big(l, q, \mathcal{R}_k(l)\big)$	Expected penalty cost at state $\big(l, q, \mathcal{R}_k(l)\big)$, $f_{l,\mathcal{R}}(q)$ for ease of notation
$L_{s_k}^j$	Lower bound of $f_k^j\big(l, q, \mathcal{R}_k(l)\big)$
$v_{k-1}(j, \mathcal{R}_{k-1}(j; l))$	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

References

[1] Daniel Adelman. Price-directed replenishment of subsets: Methodology and its application to inventory routing. Manufacturing & Service Operations Management, 5(4):348–371, 2003.

[2] Daniel Adelman. A price-directed approach to stochastic inventory/routing. Operations Research, 52(4):499–514, 2004.

[3] Daniel Adelman. Dynamic bid prices in revenue management. Operations Research, 55(4):647–661, 2007.

[4] Daniel Adelman and Christiane Barz. A price-directed heuristic for the economic lot scheduling problem. IIE Transactions, 46(12):1343–1356, 2014.

[5] Christiane Barz and Kumar Rajaram. Elective patient admission and scheduling under multiple resource constraints. Production and Operations Management, 24(12):1907–1930, 2015.

[6] Dimitris Bertsimas, Philippe Chervi, and Michael Peterson. Computational approaches to stochastic vehicle routing problems. Transportation science, 29(4):342–352, 1995.

[7] Dimitris J Bertsimas. A vehicle routing problem with stochastic demand. Operations Research, 40(3):574–585, 1992.

[8] Harlan Crowder and Manfred W Padberg. Solving large-scale symmetric travelling salesman problems to optimality. Management Science, 26(5):495–509, 1980.

[9] Daniela Pucci De Farias and Benjamin Van Roy. The linear programming approach to approximate dynamic programming. Operations research, 51(6):850–865, 2003.

[10] Daniela Pucci De Farias and Benjamin Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. Mathematics of operations research, 29(3):462–478, 2004.

[11] Moshe Dror, Gilbert Laporte, and Pierre Trudeau. Vehicle routing with stochastic demands: Properties and solution frameworks. Transportation science, 23(3):166–176, 1989.

[12] Alexandre M Florio, Dominique Feillet, Marcus Poggi, and Thibaut Vidal. Vehicle routing with stochastic demands and partial reoptimization. Transportation Science, 56(5):1393–1408, 2022.

[13] Alexandre M. Florio, Michel Gendreau, Richard F. Hartl, Stefan Minner, and Thibaut 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(3):1081–1093, 2023.

[14] Alexandre M Florio, Richard F Hartl, Stefan Minner, and Juan-José Salazar-González. A branch-and-price algorithm for the vehicle routing problem with stochastic demands and probabilistic duration constraints. Transportation Science, 55(1):122–138, 2021.

[15] M. Gendreau, O. Jabali, and W. Rei. 50th anniversary invited article–future research directions in stochastic vehicle routing. Transportation Science, 50(4):1163–1173, 2016.

[16] Michel Gendreau, Gilbert Laporte, and René Séguin. An exact algorithm for the vehicle routing problem with stochastic demands and customers. Transportation science, 29(2):143–155, 1995.

[17] Justin C Goodson, Jeffrey W Ohlmann, and Barrett W Thomas. Rollout policies for dynamic solutions to the multivehicle routing problem with stochastic demand and duration limits. Operations Research, 61(1):138–154, 2013.

[18] Justin C Goodson, Barrett W Thomas, and Jeffrey W Ohlmann. Restocking-based rollout policies for the vehicle routing problem with stochastic demand and duration limits. Transportation Science, 50(2):591–607, 2016.

[19] Martin Grötschel and Olaf Holland. Solution of large-scale symmetric travelling salesman problems. Mathematical Programming, 51(1):141–202, 1991.

[20] JX Hao and James B Orlin. A faster algorithm for finding the minimum cut in a directed graph. Journal of Algorithms, 17(3):424–446, 1994.

[21] Ola Jabali, Walter Rei, Michel Gendreau, and Gilbert Laporte. Partial-route inequalities for the multi-vehicle routing problem with stochastic demands. Discrete Applied Mathematics, 177:121–136, 2014.

[22] Sumit Kunnumkal and Huseyin Topaloglu. Computing time-dependent bid prices in network revenue management problems. Transportation Science, 44(1):38–62, 2010.

[23] Gilbert Laporte and François V Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. Operations research letters, 13(3):133–142, 1993.

[24] François V Louveaux and Juan-José Salazar-González. Exact approach for the vehicle routing problem with stochastic demands and preventive returns. Transportation Science, 52(6):1463–1478, 2018.

[25] Clara Novoa, Rosemary Berger, Jeffrey Linderoth, and Robert Storer. A set-partitioning-based model for the stochastic vehicle routing problem. Technical Report 06T-008, Lehigh University, 2006. available at https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=21c731ac7c4dc67a0f6538fa2c7a47333ad77aa4.

[26] Clara Novoa and Robert Storer. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. European journal of operational research, 196(2):509–515, 2009.

[27] Manfred Padberg and Giovanni Rinaldi. An efficient algorithm for the minimum capacity cut problem. Mathematical Programming, 47(1):19–36, 1990.

[28] Harilaos N Psaraftis. Dynamic vehicle routing: Status and prospects. Annals of operations research, 61(1):143–164, 1995.

[29] Majid Salavati-Khoshghalb, Michel Gendreau, Ola Jabali, and Walter Rei. A rule-based recourse for the vehicle routing problem with stochastic demands. Transportation Science, 53(5):1334–1353, 2019a.

[30] Majid Salavati-Khoshghalb, Michel Gendreau, Ola Jabali, and Walter Rei. A hybrid recourse policy for the vehicle routing problem with stochastic demands. EURO Journal on Transportation and Logistics, 8(3):269–298, 2019b.

[31] Nicola Secomandi. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. Computers & Operations Research, 27(11-12):1201–1225, 2000.

[32] Nicola Secomandi. A rollout policy for the vehicle routing problem with stochastic demands. Operations Research, 49(5):796–802, 2001.

[33] Nicola Secomandi. Analysis of a rollout approach to sequencing problems with stochastic routing applications. Journal of Heuristics, 9(4):321–352, 2003.

[34] Nicola Secomandi and François Margot. Reoptimization approaches for the vehicle-routing problem with stochastic demands. Operations Research, 57(1):214–230, 2009.

[35] Sparkshipping Co. Fulfillment center pricing–how much does outsourcing fulfillment cost. https://www.sparkshipping.com/blog/outsourcing-fulfillment-costs, 2023. (accessed November 22 2023).

[36] The Wall Street Journal. Not have an ev charging service that comes to you? https://www.wsj.com/articles/ev-charging-service-mobile-432572f2, 2023. (accessed November 22 2023).

[37] Chaoxu Tong and Huseyin Topaloglu. On the approximate linear programming approach for network revenue management problems. INFORMS Journal on Computing, 26(1):121–134, 2014.

[38] Alejandro Toriello, William B. Haskell, and Michael Poremba. A dynamic traveling salesman problem with stochastic arc costs. Operations Research, 62(5):1107–1125, 2014.

[39] Marlin W Ulmer, Justin C Goodson, Dirk C Mattfeld, and Marco Hennig. Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. Transportation Science, 53(1):185–202, 2019.

[40] Wen-Huei Yang, Kamlesh Mathur, and Ronald H. Ballou. Stochastic vehicle routing problem with restocking. Transportation Science, 34(1):99–112, 2000.

[41] Lin Zhu, Louis-Martin Rousseau, Walter Rei, and Bo Li. Paired cooperative reoptimization strategy for the vehicle routing problem with stochastic demands. Computers & Operations Research, 50:1–13, 2014.