Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

The Stochastic Two-Echelon Production Routing Problem with Adaptive Routing

Ali Kermani | **Jean-François Cordeau** | **Raf Jans**

Department of Logistics and Operations Management, HEC Montréal, Montréal, Canada

Correspondence
Ali Kermani, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal H3T 2A7, Canada.
Email: ali.kermani@hec.ca

Funding Information
This research was supported by the Natural Sciences and Engineering Research Council of Canada; HEC Montréal Research Chair in Supply Chain Operations Planning

Abstract

We address a stochastic two-echelon production routing problem with adaptive routing under demand uncertainty. This problem involves a production plant with production and inventory capacities where goods are produced and transported to a set of warehouses using a homogeneous fleet of vehicles. These products are then stored and subsequently delivered to customers. The products distributed from warehouses to customers can satisfy current demand or be stored for future use. We also consider the flexibility to modify customer-to-warehouse assignments for each period and scenario. We introduce a two-stage stochastic formulation to address this problem. In the first stage, the setup plan, production quantities, and routing plan to deliver goods from the production plant to the warehouses are decided. The second stage incorporates the routing decisions to deliver goods from warehouses to customers based on realized demand, with penalty costs applied for each unit of unmet demand. The objective is to minimize the total costs of production, inventory holding, and transportation in the first echelon, as well as the expected inventory holding, transportation, and penalty costs for unsatisfied demand in the second echelon. To reduce the complexity of the problem, we decompose it into two subproblems. The first subproblem addresses the first-stage decisions using an approximation of the demand from the second stage, while the second subproblem focuses on solving the second-stage problem. We propose a hybrid heuristic approach to tackle these subproblems: a mixed-integer programming model is employed for the first stage, and the second stage is handled by an Iterated Local Search metaheuristic. Computational experiments and sensitivity analyses are conducted to demonstrate the benefits of solving this problem and to assess the performance of the algorithm. The proposed algorithm successfully generated feasible solutions for instances with up to five warehouses and two vehicles in the first stage, and three vehicles per warehouse in the second stage. These solutions can typically not be improved, or only to a small extent, by a branch-and-cut algorithm which is run for 2 hours. The results also indicate that solving the stochastic problem yields, on average, a substantial improvement compared to solving a deterministic problem where the random variables are replaced by their expectation.

KEYWORDS
Two-Echelon Production Routing Problem, Stochastic Programming, Metaheuristics, Mixed-integer Programming

1 | INTRODUCTION

Integrating different problems of supply chain planning can significantly enhance the efficiency of the entire network. However, the complexity of such integrated problems has traditionally led researchers and practitioners to address these components sequentially to maintain tractability (Hrabec et al., 2022). Recent advances in optimization techniques and computational capabilities have enabled the study of more challenging integrated problems (Absi et al., 2018). One example is the Production Routing Problem (PRP), which integrates production planning, inventory management, and transportation decisions into a single optimization framework. This integration promotes better coordination among supply chain components, offering substantial opportunities for cost reduction and efficiency gains (Adulyasak et al., 2015b).

 © 2023 Copyright Holder Name

Following its introduction by Chandra (1993) and Chandra and Fisher (1994), the PRP has been widely studied. In its classical form, the PRP assumes a production facility that manufactures a single product, which must be distributed to customers to meet their demands over a finite planning horizon. The main decisions in this setting involve determining the setup plans of the production facility, production quantities, vehicle routing plans, delivery quantities to customers in each period, and inventory levels at both the plant and customer locations. These decisions are made in an integrated manner to minimize the total production, inventory, and transportation costs while ensuring that customer demands are satisfied throughout the planning horizon.

Despite the complexity of the problem, several studies have attempted to develop and improve exact algorithms to solve small and medium-sized PRP instances. Bard and Nananukul (2010) propose a Branch-and-Price algorithm to solve the problem. Archetti et al. (2011) propose a Branch-and-Cut (BC) algorithm to solve the PRP with a single vehicle. Several studies have improved the BC algorithm and introduced valid inequalities to enhance the efficiency of this algorithm for the PRP (Adulyasak et al., 2014a; Schenekemberg et al., 2021, 2023).

Most studies, however, focus on developing heuristic and matheuristic methods to leverage the problem structure and obtain good solutions for larger practical instances. Among these, several studies decompose the problem into more tractable subproblems and use iterative algorithms to obtain high-quality solutions for the original problem (Archetti et al., 2011; Absi et al., 2015; Solyalı and Süral, 2017; Russell, 2017; Manousakis et al., 2022; Vadseth et al., 2023; Ben Ahmed et al., 2023). Other studies consider metaheuristic algorithms or hybrid versions of these algorithms within an iterative approach to solve the problem (Bard and Nananukul, 2009; Armentano et al., 2011; Adulyasak et al., 2014b; Rodrigues et al., 2023).

Different variants of the PRP, such as the PRP with perishable products (Neves-Moreira et al., 2019; Alvarez et al., 2022b), multi-product PRP (Rodrigues et al., 2023; Li et al., 2019; Qiu et al., 2019), and PRP with transshipment (Avci and Topaloglu Yildiz, 2020; Alvarez et al., 2022b), have been explored in the literature. Chitsaz et al. (2019, 2020) propose a different variant called the assembly routing problem, where instead of considering multiple customers, they address a setting with multiple suppliers and a manufacturing facility. Alvarez et al. (2022a) study a variant of the PRP with multiple plants and products and incorporate four forms of consistency, namely driver, source, product, and plant consistency. Other factors, such as environmental effects and greenhouse gas emissions, are also addressed in various studies (Kumar et al., 2016; Qiu et al., 2017, 2018).

In real-world supply chains, demand uncertainty often leads to disruptions and unexpected costs. To mitigate these risks, several studies have introduced demand uncertainty into the PRP, enhancing the resiliency of supply chains. Adulyasak et al. (2015a) propose a two-stage and a multi-stage stochastic formulation for the PRP with demand uncertainty, where setup and routing decisions are made in the first stage and production and delivery quantities are decided in the second stage. Agra et al. (2018) investigate the PRP under demand uncertainty while considering backlogs. Zhang et al. (2018) consider the PRP with simultaneous pickup and delivery with remanufacturing possibilities. Mousavi et al. (2022) study the PRP with perishable products under demand uncertainty. Wang et al. (2021) examine the PRP in an uncertain environment with potential uncertainties in demand and costs. Kermani et al. (2024) present a two-stage stochastic problem for the PRP with demand uncertainty and adaptive routing. They highlight that this adaptive structure improves cost efficiency by increasing flexibility in responding to demand uncertainty. Kermani et al. (2025) investigate different types of service levels in the stochastic PRP with adaptive routing where, instead of penalty costs for unmet demands, a minimum demand satisfaction based on different service levels is imposed.

While the PRP represents a critical step toward integrating supply chain components, its traditional form is limited to single-layer distribution networks. However, today's supply chains have evolved into increasingly complex networks, often involving multiple layers that require coordinated decision-making across different echelons. For instance, the inclusion of intermediate layers, specifically, warehouses located between production plants and customers, is increasingly common. This configuration, particularly common in large urban areas, helps reduce congestion and environmental impacts by limiting the use of large trucks within cities (Perboli et al., 2011). In such systems, large vehicles transport goods from manufacturing plants to warehouses, from which smaller, low-emission vehicles handle last-mile deliveries to customers (Qiu et al., 2021).

While both the lot-sizing problem (LSP) and vehicle routing problem (VRP) literature have tackled aspects of multi-echelon supply chains individually, integrated models that consider both production and transportation across multiple layers remain scarce. For example, Gruson et al. (2019) propose a three-level lot-sizing and replenishment problem (3LSPD) involving a manufacturing facility, warehouses, and retailers. However, their model focuses solely on production and replenishment decisions, omitting routing considerations between the different levels. In contrast, the two-echelon VRP (2E-VRP) introduced by Perboli et al. (2011) considers two transportation layers, one from a central depot to distribution centers, and another from these centers to customers. Yet, this model does not account for production decisions.

Only a few studies attempt to bridge this gap by integrating production, inventory, and routing decisions across multiple echelons. Schenekemberg et al. (2021) present a two-echelon PRP inspired by the petrochemical industry, encompassing

suppliers, production plants, and customers. Gruson et al. (2024) extend the 3LSPD to include distribution, using direct shipments from production plants to warehouses while considering routing decisions for the warehouse-to-customer level. Qiu et al. (2021) propose a two-echelon PRP with cross-docking satellites (2E-PRPCS), where routing decisions apply to both echelons. Despite these contributions, a significant gap remains in integrating supply chain planning under demand uncertainty within realistic, multi-layered networks. Furthermore, the scale of problems addressed in deterministic models tends to be small compared to real-world applications, underscoring the need for efficient algorithms capable of solving larger instances.

To address these challenges, we propose a two-stage stochastic formulation for a two-echelon production routing problem. In this problem, a single manufacturing facility produces a product and transports it to a set of distribution facilities. The first-stage decisions include the production setup, production quantities, and transportation of goods from the plant to the warehouses. These decisions are made before demand realization while considering expected recourse costs. The second stage, performed after demand scenarios are realized, manages inventory at both the warehouse and customer levels and determines the distribution of goods from warehouses to customers. This flexible routing approach, where routes are selected post-demand realization, aligns with the concept of adaptive routing in stochastic PRP frameworks (Kermani et al., 2024).

To the best of our knowledge, this is the first study to incorporate demand uncertainty in a two-echelon production routing problem with adaptive routing in the second echelon. We refer to this model as the Stochastic Two-Echelon Production Routing Problem with Adaptive Routing (S2EPRP-AR). In the proposed S2EPRP-AR, shortages are allowed, with penalty costs incurred for unmet demand, providing additional flexibility for managing disruptions. Moreover, our model introduces flexible customer-to-warehouse assignments, allowing allocations to vary across periods and scenarios. Nonetheless, each customer remains assigned to one warehouse per period and scenario to maintain operational feasibility.

To solve this complex problem, we adopt a decomposition approach, breaking the model into two subproblems to enhance computational tractability. The first subproblem focuses on the first-stage decisions including production setups, production quantities, and deliveries from the plant to the warehouses, using approximate aggregate demand based on customer-to-warehouse assignments across periods and scenarios. Once these plans are established, the second subproblem handles the distribution of products from warehouses to customers, leveraging the warehouse inventory determined in the first stage. Notably, this second subproblem is scenario-decomposable, allowing for the independent solution of each scenario, thereby improving computational efficiency.

We develop a hybrid heuristic algorithm to solve these subproblems. The first-stage subproblem is formulated as a Mixed-Integer Programming (MIP) model and solved using a general-purpose solver. For the second stage, we employ an Iterated Local Search (ILS) metaheuristic to address the scenario-dependent subproblems. Specifically, we solve a multi-depot Inventory Routing Problem (IRP) for each scenario using ILS, where local search refines the solution while perturbation mechanisms help escape local optima. These subproblems are solved iteratively, with customer-to-warehouse assignments updated in every iteration, progressively improving the overall solution. It is worth noting that we also adapt this algorithm to solve the deterministic two-echelon PRP, enabling it to handle larger instances more effectively.

The remainder of the paper is organized as follows: Section 2 presents the definition and mathematical formulation of the S2EPRP-AR, detailing the assumptions and constraints. Section 3 describes the solution algorithm developed to address the problem. Computational experiments and results are presented in Section 4. Finally, Section 5 concludes the paper with a summary of findings and future research directions.

2 | PROBLEM FORMULATION

We define the S2EPRP-AR on an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \mathcal{N}_o \cup \mathcal{N}_w \cup \mathcal{N}_c$, with $\mathcal{N}_o = \{0\}$ representing the plant node; $\mathcal{N}_w = \{1, \ldots, N_w\}$ denoting the set of warehouse nodes, where N_w is the number of warehouses; and $\mathcal{N}_c = \{N_w+1, \ldots, N_w+N_c\}$ the set of customer nodes, where N_c is the number of customers. The set of edges is defined as $\mathcal{E} = \mathcal{E}_{pw} \cup \mathcal{E}_{wc}$, where $\mathcal{E}_{pw} = \{(i,j) : i,j \in \mathcal{N}_o \cup \mathcal{N}_w,\ i < j\}$ represents the edges between the plant and warehouses (first echelon), and $\mathcal{E}_{wc} = \{(i,j) : i \in \mathcal{N}_w, j \in \mathcal{N}_c \vee i,j \in \mathcal{N}_c, i < j\}$ represents the edges related to the second echelon (warehouses to customers and customers to customers). We also define the subset $\tilde{\mathcal{E}}_{wc}(\eta)$ to represent edges $(i,j) \in \mathcal{E}_{wc}$, where i,j belong to η and $\eta \subseteq \mathcal{N}_w \cup \mathcal{N}_c$. For all edges incident to node $i \in \mathcal{N}_w \cup \mathcal{N}_c$, we define the set $\tilde{\epsilon}_{wc}\{i\}$.

The problem is defined over a finite and discrete planning horizon of T periods, denoted by $\mathcal{T} = \{1, \ldots, T\}$. The set $\mathcal{K}_p = \{1, \ldots, K_p\}$ represents the vehicles available at the plant, where K_p homogeneous vehicles with capacity Q^p are responsible for transporting products from the plant to the warehouses. These vehicles must start and end their routes at the plant in

each period. Additionally, each warehouse $w \in \mathcal{N}_w$ is assigned a set of homogeneous vehicles with capacity Q^w, defined as $\mathcal{K}_w = \left\{\sum_{l=1}^{w-1} K_l + 1, \ldots, \sum_{l=1}^{w} K_l\right\}$, where K_l is the number of vehicles at warehouse l.

Figure 1 illustrates an example of the S2EPRP-AR structure with three warehouses and ten customers. It demonstrates the flexibility of customer-to-warehouse assignments and highlights that first-echelon routes remain fixed for a given period across all scenarios. In contrast, second-echelon routes can vary from period to period and across different scenarios.

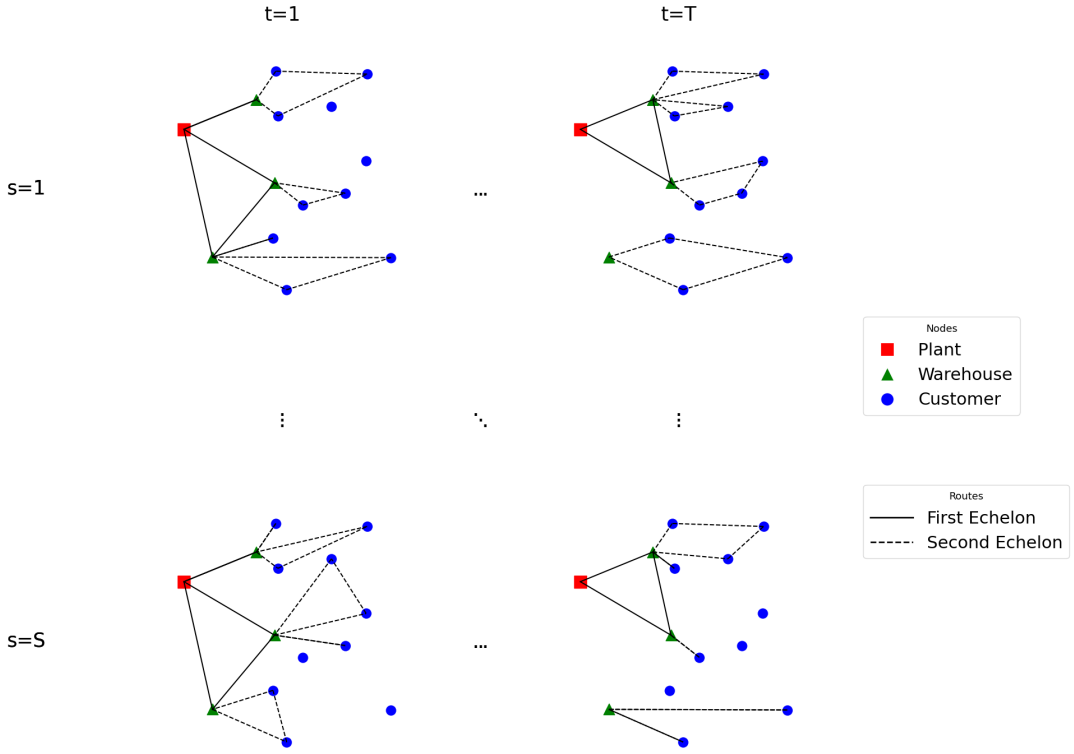


FIGURE 1 Stochastic two-echelon production routing problem with adaptive routing in the second echelon

The production plant has a production capacity of C in each period. A fixed setup cost F is incurred in any period where production takes place, along with an additional cost u per unit produced. Each node $i \in \mathcal{N}$ has an inventory capacity L_i and an initial inventory level I_{i0}. A unit holding cost h_i is applied to every item stored at node $i \in \mathcal{N}$. We assume that shortages are allowed in the model, and each unit of unmet demand for customer i incurs a penalty cost α_i.

Customer demand is modeled as a random variable with a known probability distribution, estimated from historical data. To account for demand uncertainty, we consider a finite set of scenarios $\mathcal{S} = \{1, \ldots, S\}$, where S is the number of scenarios. Each scenario $s \in \mathcal{S}$ has an associated probability $\xi_s > 0$, with $\sum_{s\in\mathcal{S}} \xi_s = 1$. The parameter d_{it}^s represents the demand of customer $i \in \mathcal{N}_c$ in period $t \in \mathcal{T}$ under scenario s.

We define the set $\mathcal{R} = \{1, \ldots, R\}$ to represent all possible routes in the first echelon. To construct this set, we assume to have the optimal route with associated cost $\tilde{c}_r$ for every non-empty subset of warehouse nodes, leading to $R = 2^{N_w} - 1$. This definition allows us to model first-echelon transportation constraints based on route selection. We also define a_{wr} as a binary parameter equal to 1 if warehouse w is present in route r, and 0 otherwise. The upper bound on delivery quantities to warehouse w is given by $\mathcal{M}_w = \min\{\mathcal{Q}^p, L_w\}$. In the second echelon, we define a cost c_{ij} for visiting edge $(i, j) \in \mathcal{E}_{wc}$. The delivery quantities to customer i in period t and scenario s are bounded by $\mathcal{M}'^s_{it} = \min\{\mathcal{Q}^w, L_i, \sum_{l=t}^{T} d_{il}^s\}$.

The first-stage decisions of the model are the production setup decisions, represented by the binary variables y_t, which takes value 1 if production occurs in period t, and 0 otherwise. The production quantity in period t is denoted by p_t, and the inventory level at the plant by I_{0t}. The variable q_{wrt} indicates the quantity delivered to warehouse w via route r in period t, and the binary variable o_{rt} is 0 if route r is selected in period t and 0 otherwise.

Second-stage decisions include the inventory levels at warehouses and customer locations, denoted by I_{it}. The amount of unmet demand for customer i in period t under scenario s is captured by l^s_{it}. The delivery quantity to customer i in period t and scenario s via vehicle $k \in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w$ is denoted by w^s_{ikt}. The binary variable z^s_{ikt} indicates whether or not node i is visited by vehicle k in period t under scenario s, and the integer variable x^s_{ijkt} represents the number of times edge $(i,j) \in \mathcal{E}_{wc}$ is traversed by vehicle k in period t under scenario s.

A summary of all sets, parameters, and decision variables used in the model is provided in A. Given these notations, the mathematical formulation of the S2EPRP-AR is presented below.

$$\text{Min} \sum_{t\in\mathcal{T}} \left(F y_t + u p_t + h_0 I_{0t} + \sum_{r\in\mathcal{R}} \tilde{c}_r o_{rt} + \sum_{s\in\mathcal{S}} \xi_s \Big(\sum_{w\in\mathcal{N}_w} I^s_{wt} + \sum_{w\in\mathcal{N}_w} \sum_{k\in\mathcal{K}_w} \sum_{(i,j)\in\mathcal{E}_{wc}} c_{ij} x^s_{ijkt} + \sum_{i\in\mathcal{N}_c} h_i I^s_{it} + \sum_{i\in\mathcal{N}_c} \alpha_i l^s_{it} \Big) \right) \tag{1}$$

s.t.

$$p_t \le C\, y_t \qquad \forall t \in \mathcal{T} \tag{2}$$

$$I_{0t} = I_{0,t-1} + p_t - \sum_{w\in\mathcal{N}_w} \sum_{r\in\mathcal{R}} q_{wrt} \qquad \forall t \in \mathcal{T} \tag{3}$$

$$I_{0t} \le L_0 \qquad \forall t \in \mathcal{T} \tag{4}$$

$$I^s_{wt} = I^s_{w,t-1} + \sum_{r\in\mathcal{R}} q_{wrt} - \sum_{i\in\mathcal{N}_c} \sum_{k\in\mathcal{K}_w} w^s_{ikt} \qquad \forall w \in \mathcal{N}_w, t \in \mathcal{T}, s \in \mathcal{S} \tag{5}$$

$$I^s_{wt} \le L_w \qquad \forall w \in \mathcal{N}_w, t \in \mathcal{T}, s \in \mathcal{S} \tag{6}$$

$$\sum_{r\in\mathcal{R}} a_{wr}\, o_{rt} \le 1 \qquad \forall w \in \mathcal{N}_w, t \in \mathcal{T} \tag{7}$$

$$\sum_{r\in\mathcal{R}} o_{rt} \le K_p \qquad \forall t \in \mathcal{T} \tag{8}$$

$$q_{wrt} \le \mathcal{M}_w\, a_{wr} \qquad \forall w \in \mathcal{N}_w, r \in \mathcal{R}, t \in \mathcal{T} \tag{9}$$

$$\sum_{w\in\mathcal{N}_w} q_{wrt} \le Q^p\, o_{rt} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \tag{10}$$

$$I^s_{it} = I^s_{i,t-1} + \sum_{k\in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w} w^s_{ikt} + l^s_{it} - d^s_{it} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \mathcal{S} \tag{11}$$

$$I^s_{it} + d^s_{it} \le L_i \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \mathcal{S} \tag{12}$$

$$w^s_{ikt} \le \mathcal{M}'^s_{it}\, z^s_{ikt} \qquad \forall i \in \mathcal{N}_c, k \in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S} \tag{13}$$

$$\sum_{i\in\mathcal{N}_c} w^s_{ikt} \le Q^w\, z^s_{wkt} \qquad \forall w \in \mathcal{N}_w, k \in \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S} \tag{14}$$

$$\sum_{k\in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w} z^s_{ikt} \le 1 \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \mathcal{S} \tag{15}$$

$$\sum_{(j,j')\in\tilde{e}(\{i\})} x^s_{jj'kt} = 2\, z^s_{ikt} \qquad \forall i \in \mathcal{N}_w \cup \mathcal{N}_c, k \in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S} \tag{16}$$

$$\sum_{(i,j)\in\tilde{\mathcal{E}}(\eta)} x^s_{ijkt} \le \sum_{i\in\eta} z^s_{ikt} - z^s_{ekt} \qquad \forall \eta \subseteq \mathcal{N}_c, |\eta| \ge 2, e \in \eta, k \in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S} \tag{17}$$

$$y_t \in \{0,1\} \qquad \forall t \in \mathcal{T} \tag{18}$$

$$p_t \ge 0 \qquad \forall t \in \mathcal{T} \tag{19}$$

$$q_{wrt} \ge 0 \qquad \forall w \in \mathcal{N}_w, r \in \mathcal{R}, t \in \mathcal{T} \tag{20}$$

$$I_{0t} \ge 0 \qquad \forall t \in \mathcal{T} \tag{21}$$

$$o_{rt} \in \{0,1\} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \tag{22}$$

$$I^s_{it} \ge 0 \qquad \forall i \in \mathcal{N}_w \cup \mathcal{N}_c, t \in \mathcal{T}, s \in \mathcal{S} \tag{23}$$

$$l^s_{it} \ge 0 \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \mathcal{S} \tag{24}$$

$$w^s_{ikt} \geq 0 \qquad \forall i \in \mathcal{N}_c, k \in \bigcup_{w \in \mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S} \quad (25)$$

$$z^s_{wkt} \in \{0,1\} \qquad \forall w \in \mathcal{N}_w, k \in \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S} \quad (26)$$

$$z^s_{ikt} \in \{0,1\} \qquad \forall i \in \mathcal{N}_c, k \in \bigcup_{w \in \mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S} \quad (27)$$

$$x^s_{ijkt} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{E}_{wc}, i \notin \mathcal{N}_w, k \in \bigcup_{w \in \mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S} \quad (28)$$

$$x^s_{ijkt} \in \{0,1,2\} \qquad \forall (i,j) \in \mathcal{E}_{wc}, i \in \mathcal{N}_w, k \in \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S}. \quad (29)$$

The objective function (1) minimizes the total setup costs, inventory holding costs at the plant, and transportation costs of delivering products from the plant to warehouses, as well as the expected inventory holding costs at the warehouses and customers, transportation costs of goods from warehouses to customers, and penalty costs incurred on every unit of unmet demand. Constraints (2) ensure that production is only possible if a setup takes place in period t. Constraints (3), (5), and (11) are the inventory balance constraints for the plant, warehouses, and customers, respectively. The inventory limit of each of these facilities is enforced by constraints (4), (6), and (12). Note that for constraints (12), the original form $I^s_{it} + d^s_{it} - l^s_{it} \leq L_i$ accounts for lost sales. While this makes the constraints less restrictive, in the proposed setting a shortage in one period will never be created to satisfy demand in a later period. This is because inventory holding costs are positive and lost sales incur a cost that is independent of the period. Therefore, we will not have both a strictly positive inventory level and lost sales in the same period, and the term l^s_{it} can be omitted without affecting the model's behavior.

For the first echelon of the problem, we assume that all routes are explicitly defined. Constraints (7) guarantee that warehouse w is visited by at most one route in each period. Constraints (8) limit the number of selected routes to the available vehicles in each period. Constraints (9) ensure that delivery to warehouse w is only possible if it is included in route r, while the sum of deliveries on route r is limited to the vehicle capacity by constraints (10). Constraints (13) enforce that delivery to customer i is only possible if the customer is visited by vehicle k from warehouse w. The capacity on the vehicles at the warehouses is imposed by constraints (14). Constraints (15) prevent split deliveries to customers. Constraints (16) are the degree constraints, and constraints (17) are the subtour elimination constraints (SECs). Finally, constraints (18 - 29) define the domains of the decision variables.

3 | HYBRID HEURISTIC ALGORITHM

Solving the S2EPRP-AR presents significant challenges due to its NP-hard nature, especially when incorporating uncertainty and adaptive routing. Even for medium-size instances of the deterministic version, exact algorithms often struggle to find feasible solutions. In such cases, Qiu et al. (2021) propose a matheuristic algorithm that first generates an incumbent solution, then applies BC to reduce the optimality gap and improve the upper bound (UB).

In response to these challenges, we propose a Hybrid Heuristic Algorithm (HHA) that leverages the problem's structure by decomposing it into smaller subproblems, which are solved iteratively to enhance the overall solution. The HHA consists of two main components: a MIP model for the first echelon, and an ILS metaheuristic to manage the second echelon.

The algorithm begins by constructing an initial solution, which is iteratively refined by solving a restricted S2EPRP-AR with fixed second-echelon routes to improve the first-echelon solutions. Next, ILS is applied to the second-echelon routes, followed by solving an LP to optimize the continuous variables based on the updated routes. The structure of the HHA is outlined in Algorithm 1. The quality of the initial solution is critical to the algorithm's performance, and the construction heuristic used for this purpose is detailed in Section 3.1. After obtaining the initial solution, a local search heuristic, specifically a multi-start Randomized Variable Neighborhood Descent (RVND), is applied to each scenario s to improve the solution (Line 5). This local search algorithm is similar to the one used as the local search component of the ILS, as explained in Section 3.2.1.

In subsequent iterations, the algorithm adjusts customer-to-warehouse assignments whenever stockouts occur. When a shortage is detected for a customer, it is removed from its current warehouse and delivery route and reassigned to a different warehouse's route to reduce the cost of unmet demand. The algorithm evaluates potential routes based on minimum insertion cost. If the insertion cost is lower than the current stockout penalty, and both warehouse and vehicle capacities allow the reassignment, the customer is integrated into the new route. This process is repeated for all customers facing shortages across every period and scenario (lines 11–13).

Algorithm 1 Hybrid Heuristic Algorithm

```
1: Input: Problem Parameters
2: Output: Incumbent solution for the S2EPRP-AR
3: Initialize incumbent solution: sol ← ∅;
4: sol_0 ← ConstructInitialSolution;   (Algorithm 2)
5: sol ← RVND(sol_0);   (Algorithm 4)
6: sol' ← sol
7: while stopping criteria not met do
8:    if no improvement in the global best OF for NIG iterations then
9:       sol' ← sol;
10:   end if
11:   for each period t ∈ T and scenario s ∈ S do
12:      Reassign customers experiencing stockouts to alternative warehouses;
13:   end for
14:   if no stockouts are present & no local improvements on the OF for NIL
   iterations then
15:      for each scenario s ∈ S do
16:         Select a random warehouse w ∈ N_w in a random period t ∈ T to remove;
17:         Reassign all customers of the removed warehouse to other feasible
   warehouses;
18:      end for
19:   end if
20:   Solve the restricted S2EPRP-AR using the updated customer assignments;
21:   Update sol' with the new solution;
22:   sol' ← ILS(sol');   (Algorithm 3)
23:   if objVal(sol') < objVal(sol) then
24:      Update incumbent solution: sol ← sol'
25:   end if
26: end while
27: return sol
```

When no shortages are present in an iteration, the algorithm diversifies the search by randomly removing a warehouse within a selected period. This removal forces the reassignment of its customers to other warehouses across all scenarios in that period. This strategy helps explore new solution spaces and prevents the search from getting trapped in local optima. The removal of a warehouse only occurs if the local best objective function value (OF) has not improved for *NIL* consecutive iterations. If the absence of improvement continues, a new warehouse is removed after every additional *NIL* iterations (lines 14–19). If no improvement in the global best OF is observed after *NIG* iterations, the algorithm resets the current solution to the incumbent solution. This reset allows for the exploration of new neighborhoods and helps avoid prolonged stagnation (lines 8–10).

At the end of each iteration, a restricted version of the S2EPRP-AR is solved in which the second-echelon routes are fixed, allowing the algorithm to update the first-echelon decisions. This is followed by applying Iterated Local Search (ILS) to refine the second-echelon routes. The ILS procedure improves the routes using local search operators and applies perturbations to escape local optima. If an improved solution is found, the current solution is updated; otherwise, the algorithm proceeds to the next iteration (lines 20–22). Additionally, if the current OF is better than the global best, the incumbent solution and the global best value are also updated. This process is repeated until one of the stopping criteria is met.

The algorithm terminates when no improvement is observed in the best solution for *Max_NIG* consecutive iterations, or the algorithm's runtime exceeds the predefined time limit *HHA_TimeLimit*.

3.1 | Initial Solution

To generate an initial solution that is both feasible and provides a strong starting point for the algorithm, we decompose the problem into two subproblems. The pseudocode for the construction heuristic algorithm is provided in Algorithm 2. The first subproblem is a stochastic multi-warehouse PRP (MW-PRP), which focuses exclusively on the plant and warehouse nodes. Since customer demand drives the problem, we introduce a virtual demand for each warehouse. This is achieved by assigning customers to their nearest warehouse and by setting the demand of each warehouse equal to the sum of the demands of its assigned customers. Thus, set $\mathcal{C}^w = \left\{ i \in \mathcal{N}_c \mid w = \arg\min_{w'} \{c_{w',i}\} \right\}$ denotes the set of customers that are assigned to $w \in \mathcal{N}_w$.

To prevent over-delivery, the initial inventory of each customer is subtracted from its demand during the calculation. Additionally, no inventory or delivery costs are considered for customers at this stage; however, an approximate penalty cost on each unit of unmet demand is considered based on the average cost of unmet (aggregate) demand for customers assigned to each warehouse. Consequently, the objective function of this subproblem includes fixed and variable production costs, transportation costs between the plant and warehouses, inventory holding costs at both the plant and warehouses, and approximate costs for unmet demand.

It is worth noting that we explicitly define the routes between the plant and warehouses in this subproblem. Although the number of potential routes increases exponentially with the number of warehouses, we assume that the number of warehouses remains limited, as is typical in real-world scenarios. This assumption enables us to explicitly define all possible routes, thereby simplifying the problem. To generate these routes, we solve a TSP for each non-empty subset of nodes, denoted as $\eta_{FE} \subseteq \mathcal{N}_w \backslash \{\emptyset\}$, based on the transportation costs of the edges in $\mathcal{E}_{pw}$. Hence, the total number of binary variables required for the first-echelon routes is $2^{N_w} - 1$.

The mathematical formulation for the MW-PRP is presented below:

$$\text{Min} \sum_{t\in\mathcal{T}} \left(Fy_t + up_t + h_0 I_{0t} + \sum_{r\in\mathcal{R}} \tilde{c}_r o_{rt} + \sum_{s\in\mathcal{S}} \xi_s \Big(\sum_{w\in\mathcal{N}_w} \left(I^s_{wt} + \hat{\alpha}_w \hat{l}^s_{wt} \right) \Big) \right) \tag{30}$$

s.t. (2) - (4), (6) - (10), (18) - (20), and (23)

$$I^s_{wt} = I^s_{w,t-1} + \sum_{r\in\mathcal{R}} q_{wrt} - \hat{d}^s_{wt} + \hat{l}^s_{wt} \qquad \forall w \in \mathcal{N}_w, t \in \mathcal{T}, s \in \mathcal{S} \tag{31}$$

$$I^s_{it} \geq 0 \qquad \forall i \in \mathcal{N}_o \cup \mathcal{N}_w, t \in \mathcal{T}, s \in \mathcal{S} \tag{32}$$

$$\hat{l}^s_{wt} \geq 0 \qquad \forall w \in \mathcal{N}_w, t \in \mathcal{T}, s \in \mathcal{S}, \tag{33}$$

where $\hat{d}^s_{wt} = \sum_{i\in\mathcal{C}^w} \left(d^s_{it} - (I_{i0} - \sum_{l=1}^{t-1} d^s_{it})^+ \right)$ for warehouse w, period t and scenario s and $\hat{\alpha_w} = \frac{\sum_{i\in\mathcal{C}^w} \alpha_i}{|\mathcal{C}^w|}$ for each warehouse. The variable $\hat{l}^s_{wt}$ is used to indicate the unmet demand of each warehouse in a period and scenario. The variable $\tilde{q}^s_{wt} = \sum_{r\in\mathcal{R}} q_{wrt}$ represents the delivery amount to warehouse w in period t and scenario s. It is used as a parameter in the second phase of the algorithm to compute the warehouse inventory for each scenario.

After determining the production plan and the routes for the first echelon, we proceed to construct a feasible solution for the second echelon. It is important to emphasize that the second-echelon problem is always feasible, as shortages in customer demand are allowed. To generate a feasible solution for the second echelon, we utilize the available inventory at each warehouse obtained in the previous phase while accounting for the capacity of the vehicles located at these warehouses. The solution is derived in three phases. In the first phase, we identify the customers to be visited and estimate the delivery quantities. The second phase involves determining the delivery routes for the selected customers. These two phases are modeled similarly to the approach in Alvarez et al. (2018) with some changes. We also introduce a third phase in our approach, where the continuous variables are optimized by solving an LP, rather than being determined during the construction heuristic.

We begin by ranking customers according to a ratio that compares the cost of unmet demand with the approximate delivery cost per unit. The latter includes the unit production cost, the approximate fixed production cost per unit based on production capacity, the estimated delivery cost of a single unit from the plant to the nearest warehouse based on Q^p, and the estimated delivery cost from the warehouse to the customer based on Q^w. For a given customer i, assuming that warehouse w' is the closest,

Algorithm 2 Construction Heuristic

Input: Problem Parameters, Set of customers initially assigned to each warehouse $\mathcal{C}^w$
Output: Feasible solution for the S2EPRP-AR
Initialize $sol \leftarrow \emptyset$
Initialize $sol_{\text{FE}} \leftarrow \emptyset$
Initialize $\hat{d}^s_{wt}$ and $\hat{\alpha}_w$
Solve MW-PRP to obtain production plan and first-echelon routes
Update sol_{FE}
Initialize $sol_{\text{SE}} \leftarrow \emptyset$
Determine $\tilde{q}^s_{wt}$
Construct *sorted_customers* as in (34)
for each scenario s **do**
  for each warehouse w **do**
    $look_ahead \leftarrow 1$
    while $look_ahead \leq \lceil \frac{T}{2} \rceil$ **do**
      Compute projected inventory $\tilde{I}^s_{it}$ for each customer $i \in \mathcal{C}^w$, period $t \in \mathcal{T}$, and scenario $s \in \mathcal{S}$ as in (35)
      for each period t **do**
        Initialize empty sets $\vartheta_1 \leftarrow \emptyset$, $\vartheta_2 \leftarrow \emptyset$
        for each customer $i \in$ *sorted_customers* **do**
          if customer i is projected to stockout in the current period **then**
            Add customer i to set ϑ_1
          else if customer i is projected to stockout in the next *look_ahead* periods **then**
            Add customer i to set ϑ_2
          end if
        end for
        Assign customers in ϑ_1 to routes, respecting $\tilde{q}^s_{wt}$ and vehicle capacities, using the nearest neighbor insertion heuristic
        Assign customers in ϑ_2 to routes, respecting $\tilde{q}^s_{wt}$ and vehicle capacities, using the minimum insertion cost method
        Update $\tilde{I}^s_{it}$ with the computed delivery quantities for each customer
      end for
      Update sol_{SE} with the second-echelon solution for this *look_ahead* value
      Increase *look_ahead* value by 1
    end while
  end for
end for
Combine sol_{FE} and sol_{SE} to update the overall solution sol
Fix all integer variables in sol
Solve the LP to improve production and delivery plan
Update sol with the improved solution
Return the best feasible solution sol

this relative shortage penalty ratio is computed as follows:

$$\frac{\alpha_i}{u + \frac{F}{C} + \frac{2c_{0,w'}}{Q^p} + \frac{2c_{w',i}}{Q^w}}. \tag{34}$$

Based on this ratio, we sort the customers in descending order of priority and define the set *sorted_customers*. For each customer i, we then calculate the partial inventory levels for each time period and scenario by using the customer's demand, initial inventory, and the deliveries made to the customer. The partial inventory for each time period can be calculated as follows:

$$\tilde{I}_{it}^{s} = I_{i0} - \sum_{l=1}^{t} d_{il}^{s} + \sum_{l=1}^{t} \tilde{w}_{il}^{s} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \mathcal{S}, \quad (35)$$

where $\tilde{w}_{it}^{s}$ is the delivery quantity to customer i, in period t and scenario s. In the first iteration of the algorithm, since no delivery is assigned to customers, we have:

$$\tilde{w}_{it}^{s} = 0 \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \mathcal{S}. \quad (36)$$

Assuming a warehouse w and scenario s, the heuristic is executed for each period $t \in \mathcal{T}$ to determine which customers are assigned to warehouse w. Two sets, ϑ_1 and ϑ_2, are defined. The first set contains customers with $\tilde{I}_{it}^{s} < 0$, who will face stockouts immediately in period t if not visited. Thus, these customers have a higher priority of being visited. The approximate delivery quantity to each customer in set ϑ_1 is calculated using:

$$\begin{cases} \tilde{w}_{it}^{s} = \min\{|(\tilde{I}_{it}^{s})^{-}|, Q^{w}, L_i - (\tilde{I}_{i,t-1}^{s})^{+}\}, & \forall i \in \vartheta_1,\ t \in \mathcal{T}\backslash\{T\},\ s \in \mathcal{S}, \\ \tilde{w}_{iT}^{s} = \min\{|(\tilde{I}_{iT}^{s})^{-}|, Q^{w}\}, & \forall i \in \vartheta_1,\ s \in \mathcal{S} \end{cases} \quad (37)$$

After assigning customers to ϑ_1, we proceed to assign customers to the second set, ϑ_2, which consists of those projected to experience stockouts in the upcoming *look_ahead* periods ($\tilde{I}_{i,t+look_ahead}^{s} < 0$). The approximate delivery quantity to each customer in set ϑ_2 is given by:

$$\tilde{w}_{it}^{s} = \min\{|(\tilde{I}_{i,t+look_ahead}^{s})^{-}|, Q^{w}, L_i - (\tilde{I}_{i,t-1}^{s})^{+}\}, \quad \forall i \in \vartheta_2,\ t \in \mathcal{T}\backslash\{T\},\ s \in \mathcal{S}. \quad (38)$$

When adding customers to ϑ_1 and ϑ_2, we also take *sorted_customers* into account, since shortages are allowed and we aim to prioritize those with a higher penalty ratio for unmet demand. Customers are inserted into each set in rank order, starting with those having the highest relative penalty ratio, thereby increasing their likelihood of being visited. Customers from ϑ_1 may also appear in ϑ_2, in which case they are removed from ϑ_2 and their approximate delivery is updated according to (38).

Following the definition of ϑ_1 and ϑ_2, we apply the standard nearest neighbor insertion heuristic (Bräysy and Gendreau, 2005) to add customers from ϑ_1. For customers in ϑ_2, we use the minimum insertion cost criterion to integrate them into the routes. During the assignment of customers to routes, we also consider the remaining available inventory and vehicle capacities to ensure feasibility. Accordingly, the values of $\tilde{w}_{il}^{s}$ are updated if the available inventory at the warehouse or vehicle capacities do not allow full delivery of the initially estimated quantities. We start by assigning customers from ϑ_1 and then move to ϑ_2, stopping when capacity or available inventory prevents further deliveries.

At the end of each iteration, after assigning deliveries to customers, customer inventories are updated using (35). Once routes are established for all periods within the current scenario and warehouse, the *look_ahead* value is increased by 1 and all deliveries are reset to zero to assess whether accounting for additional periods improves the objective function. This process continues until *look_ahead* = $\lceil \frac{T}{2} \rceil$, and the solution with the lowest objective value is selected. The same procedure is applied to all warehouses and scenarios, ultimately yielding a feasible solution for the entire problem.

In the third phase of this heuristic, we solve an LP for the entire S2EPRP-AR, where all integer variables are fixed, allowing for further improvements in continuous variables i.e., production and delivery quantities.

3.2 | Iterated Local Search

The ILS metaheuristic iteratively improves an initial feasible solution by alternating between local search (intensification) to move toward local optima, and a perturbation algorithm to escape those local optima (diversification). At the end of each iteration, if a better solution is found, the incumbent solution is updated. After generating the initial solution and reassigning customers to warehouses, the solution is passed to the ILS for iterative improvement. The pseudocode of the ILS is presented in algorithm 3.

Since the ILS focuses only on the second-echelon routes and the first-stage binary variables are fixed, the problem can be decomposed into an independent subproblem for each scenario. Thus, the ILS is applied separately to each scenario (lines 4-5).

Algorithm 3 Iterated Local Search (ILS)

```
1: Input: sol_0
2: Output: Incumbent solution for the S2EPRP-AR
3: sol ← RVND(sol_0)
4: for each scenario s do
5:    sol' ← sol_SE^(s)
6:    while Stopping criteria for ILS is not met do
7:       sol'' ← perturb(sol')
8:       sol'' ← RVND(sol'')
9:       if objVal(sol'') < objVal(sol') then
10:         sol' ← sol''
11:      end if
12:   end while
13:   Update sol_SE^(s) with sol'
14: end for
15: return sol
```

In each iteration, the solution for a given scenario s undergoes a perturbation, which helps escape the current local optimum (line 7). After the perturbation, the solution is improved using a multi-start RVND local search algorithm (line 8). If a better solution is found, the solution for that scenario is updated, and the next iteration begins (line 9-11). This process continues until a stopping criterion is met.

Although the ILS in this study shares components with those used for the IRP in Alvarez et al. (2018), it differs in key aspects. Previous algorithms relied solely on ILS to optimize both integer and continuous variables. In contrast, our approach solves an LP after each local search operator is executed, which leads to better overall results, especially when accounting for the uncertainty in the problem. The ILS algorithm terminates if the number of iteration reaches *ILS_TimeLimit* or the computation time reaches *ILS_ItertionLimit*.

3.2.1 | Local Search

As mentioned earlier, the local search algorithm used in this study (line 3 of Algorithm 3) is a multi-start RVND heuristic (Mladenović and Hansen, 1997), where local search operators are selected randomly and applied to the current solution. The multi-start framework allows the algorithm to explore multiple neighborhoods, as the local optima may differ depending on the sequence and selection of local search operators. This diversification enhances the ability of the algorithm to escape poor-quality local optima. If the application of an operator improves the objective function value, the set of operators is reset to its initial state. Otherwise, the operator is removed from the set. This process continues until all operators have been exhausted. The pseudocode for the RVND procedure is provided in Algorithm 4.

The local search operators used within RVND primarily focus on modifying the routes from a selected warehouse to its assigned customers. Some operators also allow changes in customer-to-warehouse assignments, depending on the current values of the first-stage variables.

The operators used in the RVND are as follows:

- **Or-Opt**(v), $v \in 1, 2, 3$: Moves v adjacent customers from their current position to another position within the same route.
- **Route-Shift**(v), $v \in 1, 2, 3$: Selects v adjacent customers from a route and moves them to another route, either of the same warehouse or a different warehouse in the same period.
- **Period-Shift**: A customer is removed from its current visited route in one period and randomly inserted into another route in a different period.
- **Swap**(v_1, v_2), $v_1, v_2 \in 1, 2$, $v_1 \geq v_2$: Exchanges v_1 customers from one route with v_2 customers from another route, either from the same warehouse or from different warehouses in the same period.
- **Remove**: A randomly selected customer is removed from a route.

Algorithm 4 Randomized Variable Neighborhood Descent (RVND)

```
Input: Solution for scenario s
Output: Improved solution for scenario s
sol* ← sol_0^(s)
O ← initialize the set of local search operators
sol' ← sol*
while |O| > 0 do
   Select a random local search operator o from O
   Perform o on sol'
   if o improves sol' then
      sol* ← sol'
      Reset O to the initial set of operators
   else
      Remove o from O
   end if
end while
return sol*
```

- **Insert**: A customer that is not visited in period t is inserted into a randomly selected route from warehouse w at the minimum insertion cost position.
- **Merge**: For a customer visited in more than one period, two of the visited periods are merged into one, selected randomly.
- **Transfer**: A customer visited in multiple periods is removed from its visited periods and inserted into unvisited periods for a randomly selected warehouse, based on minimum insertion cost.

After applying an operator, we solve an LP with a fixed production plan and fixed routes for both echelons across all scenarios. If the resulting objective function value improves, the incumbent solution for that scenario is updated.

3.2.2 | Perturbation

The perturbation step in ILS is crucial as it significantly enhances the diversification of the search. After applying a perturbation, the algorithm initiates the local search in the hope of exploring solutions beyond those identified in previous iterations. To achieve this, we define four types of perturbations for the proposed ILS, as follows:

- **Random Shift**: A customer is selected from a route and moved to another route, either from the same warehouse or from a different warehouse in the same period.
- **Random Insertion**: A customer that is not visited in period t is inserted into a randomly selected warehouse w at the minimum insertion cost position.
- **Random Removal**: A randomly selected customer is removed from a route.
- **Random Swap**: A randomly selected customer from one route is exchanged with another randomly selected customer from a route from the same or a different warehouse.

A maximum of *ILS_MaxPerturb* perturbations are allowed for each scenario, and the perturbation operators are selected randomly. After applying the perturbation, we solve the LP to evaluate the current value of the objective function before proceeding with the local search. The new solution is only accepted if it improves upon the previous solution after applying the local search.

4 | COMPUTATIONAL EXPERIMENTS

The algorithms developed in this study were implemented in C++, and IBM ILOG CPLEX version 22.1.1 was used as solver to handle LP and MIP subproblems. All computational experiments were carried out on machines equipped with Intel Xeon Gold 6148 2.4 GHz processors and 32 GB of memory.

Due to the novelty of the problem addressed in this study, benchmark instances available in the literature are very limited. To the best of our knowledge, this is the first study to tackle this specific variant of the two-echelon production routing problem with demand uncertainty.

As a result, we generated a new set of test instances specifically designed for the S2EPRP-AR. These instances are made publicly available at Instances Repository. To generate the new dataset we used the benchmark instances introduced by Archetti et al. (2011), which are widely used in the classical PRP literature. However, the original instances do not include several key elements required in our setting, such as multiple echelons and stochastic demand. We therefore adapted and extended them to fit the structure and requirements of our problem.

We first generated instances for the deterministic version of the problem. These instances vary in the number of warehouses (2 to 5), number of customers (ranging from 20 to 50, in increments of 10), and vehicle configurations: 1 vehicle at both the plant and each warehouse, 2 vehicles at the plant and 2 vehicles at each warehouse, and 2 vehicles at the plant and 3 vehicles at each warehouse. Additionally, we considered the four classes of instances defined in Archetti et al. (2011), namely: standard cost structure, high production cost, high transportation cost, and no inventory cost at the customer level. Warehouse locations were randomly selected from within the customer set using a clustering-based approach. In total, 192 deterministic instances were generated.

A similar approach was used to generate instances for the stochastic problem. For these instances, we considered 2 to 5 warehouses when the number of customers was in $\{20, 25, 30\}$, and 2 and 3 warehouses when having $\{10, 15, 35, 40, 50\}$ customers. The same vehicle configurations as in the deterministic case were used. In total, 66 stochastic instances were generated. A detailed explanation of the instance generation process is provided in the Online Supplements.

To incorporate demand uncertainty, we used Monte Carlo simulation to generate demand scenarios based on the average customer demands from the instances. In the standard setting, each instance includes 100 scenarios unless stated otherwise. For each customer i and period t, the demand in scenario s is sampled from a uniform distribution in the range $[\bar{d}_{it}(1-\delta), \bar{d}_{it}(1+\delta)]$, where $\bar{d}_{it}$ is the average demand and δ is the uncertainty level, ranging between 0 and 1. Each scenario $s \in \mathcal{S}$ is assigned an equal probability of $\xi_s = 1/S$.

Finally, the stochastic model includes a penalty cost α_i for each unit of unmet demand for customer i. This cost is calculated as the sum of the unit production cost, an estimate of the fixed cost per unit of production, and the approximate transportation cost of delivering a unit from the plant to the nearest warehouse to customer i, and then from the warehouse to the customer, each multiplied by two to reflect round trips. The total is then multiplied by a penalty factor σ. The penalty is computed as:

$$\alpha_i = \left\lceil \sigma \left(u + \frac{F}{C} + 2\frac{c_{w_i^*,i}}{Q^w} + 2\frac{c_{0,w_i^*}}{Q^p} \right) \right\rceil, \tag{39}$$

where w_i^* is the closest warehouse to customer i and σ is set to 50 as in Kermani et al. (2024), unless stated otherwise.

4.1 | Deterministic Problem

We employ the HHA algorithm to solve large instances of the deterministic 2EPRP. In the largest configuration, the model considers 5 warehouses, 50 customers, 2 vehicles at the plant and 3 vehicles at each warehouse, resulting in a total of 15 vehicles in the second echelon. Compared to the stochastic variant, the deterministic problem differs in two important aspects: there is no need to account for multiple scenarios, and shortages are no longer allowed. Therefore, the original formulation of the S2EPRP can be simplified. Specifically, the scenario index is removed from all variables and constraints, and the shortage variable l_{it}^s is eliminated. In addition, constraints (11) are replaced by the following deterministic version:

$$I_{it} = I_{i,t-1} + \sum_{k \in \bigcup_{w \in \mathcal{N}_w} \mathcal{K}w} w_{ikt} - d_{it} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}. \tag{40}$$

TABLE 1 Computational results for the deterministic problem with one vehicle at the plant and one vehicle per warehouse.

			HHA			BC				
N_w	N_c	# Ins	OF	Gap (%)	CPU (secs)	UB	LB	CPU (secs)	Gap (%)	Diff-UB∇ (%)
2	20	4	143,098	1.88	279.94	142,372	141,387	7,205.92	1.09	0.80
	30	4	190,141	2.07	293.35	189,758	187,911	6,440.05	1.72	0.35
	40	4	239,975	2.77	507.42	239,674	235,954	5,316.52	2.58	0.21
	50	4	282,133	3.29	564.94	281,961	276,997	6,919.66	3.25	0.04
3	20	4	145,713	5.87	291.38	145,526	140,343	5,374.16	5.70	0.20
	30	4	190,852	4.20	345.36	190,816	186,101	3,491.56	4.17	0.04
	40	4	241,380	5.49	364.53	241,271	233,795	5,033.68	5.41	0.09
	50	4	285,557	4.58	776.82	285,225	277,960	4,263.03	4.43	0.17
4	20	4	144,865	5.39	331.61	144,703	140,000	4,930.37	5.16	0.25
	30	4	192,886	5.05	373.03	192,869	186,922	5,234.71	5.04	0.01
	40	4	241,946	4.32	659.30	241,894	235,925	6,714.82	4.28	0.04
	50	4	286,906	5.14	1,086.24	286,895	278,272	7,209.52	5.14	0.01
5	20	4	143,044	4.32	297.82	142,966	139,186	4,431.63	4.28	0.05
	30	4	191,214	4.93	516.39	191,090	185,390	6,010.52	4.86	0.08
	40	4	243,590	5.32	707.10	243,492	235,723	6,821.45	5.23	0.09
	50	4	287,339	5.48	542.48	287,206	278,270	7,026.67	5.43	0.06
Average		64	215,665	4.38	496.11	215,482	210,008	5,776.52	4.23	0.15

The complete formulation of the deterministic problem is provided in B. To solve this problem, modifications are also made to the HHA algorithm to remove all iterations over scenarios.

The time limit for the HHA is fixed at 3600 seconds. The MW-PRP, which is solved only once to construct the initial solution, is assigned a time limit of 300 seconds with an optimality gap of 0.01%. The restricted 2EPRP, which is solved in every iteration of HHA, is given 60 seconds of computation time with a 1% optimality gap. This setting is adopted because reducing the optimality gap significantly increases the computational time, while it typically does not yield a proportional improvement in solution quality. All of these computations are performed using 10 threads.

The number of multi-starts for the ILS is set to 10. It is worth noting that parallel computing is incorporated into the deterministic version of the algorithm to enhance computational efficiency. Specifically, parallelism is applied within the multi-start RVND procedure, where 10 threads are used to execute multiple runs concurrently. The algorithm retains the best solution based on the objective function value. Although parallel computing is also employed in the stochastic version, its implementation is adapted to the scenario-based decomposition structure of that problem.

The ILS-specific parameters are set to *ILS_MaxPerturb* = 10 and *ILS_IterationLimit* = 30. A grid search was used to tune the parameters *Max_NIG* and *Max_NIL*, which were ultimately set to 25 and 5, respectively. The value of *Max_NIG* was selected from the candidate set $\{10, 15, 25, 50\}$, while *Max_NIL* was chosen from $3, 5, 10, 15$.

To assess the quality of the HHA solutions, we use a BC algorithm implemented in CPLEX to solve the 2EPRP, using the HHA solution as a warm start. Details of the BC algorithm are provided in D. The BC is run with a 7200 second time limit using 10 threads, and the optimality gap is set to 10^{-6}.

All 192 deterministic instances are defined over a planning horizon of 6 periods. To present the results, we include three tables, each corresponding to one of the aforementioned vehicle configurations. In each table, the number of warehouses (N_w) and the number of customers (N_c) are reported. Each row shows the average solution to four instances corresponding to the four problem classes of instance. Each table summarizes 64 instances in total.

Under the HHA section of the table, the column labeled **OF** shows the best objective function value found by the algorithm, **Gap** (%) presents the relative optimality gap with respect to the lower bound (LB) obtained from BC, and **CPU (secs)** indicates the computation time in seconds. Under the BC section, the **UB**, **LB**, and **CPU** time are reported. The **Gap** column shows the final optimality gap from BC. It is important to note that the UB from BC is always at least as good as that of HHA, since the HHA solution is provided as a warm start. The final column, labeled **Diff-UB∇**, shows the improvement in the UB after two hours of solving the problem with the BC algorithm.

TABLE 2 Computational results for the deterministic problem with two vehicles at the plant and two vehicles per warehouse.

N_w	N_c	# Ins	HHA			BC				Diff-UB∇
			OF	Gap (%)	CPU (secs)	UB	LB	CPU (secs)	Gap (%)	(%)
2	20	4	144,930	3.90	395.15	144,928	141,544	5,491.38	3.89	0.00
	30	4	194,312	5.18	384.13	194,289	188,270	4,087.31	5.15	0.03
	40	4	242,878	4.76	503.53	242,861	236,281	4,488.20	4.76	0.01
	50	4	286,078	5.78	667.73	286,067	276,840	3,346.37	5.77	0.01
3	20	4	145,911	7.18	449.93	145,735	139,032	3,988.31	7.00	0.20
	30	4	192,038	6.21	461.72	191,983	184,729	4,631.70	6.16	0.05
	40	4	242,137	6.51	816.61	242,101	233,115	7,069.14	6.47	0.04
	50	4	289,535	8.21	803.95	289,473	276,326	5,132.83	8.17	0.04
4	20	4	144,891	6.09	452.91	144,864	139,534	4,265.45	6.08	0.01
	30	4	193,437	6.59	790.12	193,423	185,842	5,112.01	6.58	0.02
	40	4	243,457	6.02	978.18	243,420	235,040	7,209.37	5.99	0.03
	50	4	290,204	9.12	1,021.84	290,174	275,819	5,829.16	9.09	0.03
5	20	4	144,097	4.76	644.42	144,019	139,808	4,648.82	4.72	0.05
	30	4	192,235	6.54	793.02	192,235	184,796	7,105.51	6.54	0.00
	40	4	242,657	7.42	1,045.21	242,591	232,629	7,111.09	7.36	0.06
	50	4	288,275	7.93	1,910.22	288,275	275,856	7,210.29	7.93	0.00
Average		64	217,317	6.39	757.42	217,277	209,091	5,420.43	6.35	0.04

The results of the first configuration, specifically one vehicle at the plant and one for each warehouse, are provided in Table 1. The average optimality gap is 4.38% with an average computation time of 496 seconds. The BC algorithm was able to improve the given UB by only 0.15% on average. This result highlights the performance of the HHA in finding high-quality solutions within a reasonable time. However, the optimality gap is also affected by the weak LB, showing the inability of the BC to close the gap even for the smallest instances.

Table 2 displays the results for the second configuration of instances. As expected, both the gap and computation time are higher, at 6.39% and 757 seconds, respectively. The improvement in the UB by BC is even lower at 0.04%, which shows how increasing the size of the problem has a drastic effect on the BC performance, while the HHA is still able to provide solutions in less than 15 minutes.

Finally, Table 3 illustrates the results of the largest instance configuration, showing an average gap of 7.48% and an average computation time of 1005 seconds. The Diff-UB∇ remains very low at 0.03%, confirming that while BC struggles to improve the solutions, the HHA continues to deliver good solutions within a reasonable time frame.

Table 4 presents the average costs associated with each component of the deterministic problem, broken down by instance class and by the number of vehicles per plant and warehouse. Under **First Echelon**, the average total cost of the first echelon, as well as the average production, plant inventory, and transportation costs, are presented, respectively. Under **Second Echelon**, the total cost of the second echelon is shown, followed by warehouse inventory, customer inventory, and second-echelon transportation costs. The second class of instances, characterized by high production costs, incurs the highest total costs across all configurations. This is largely driven by the high setup costs, which significantly influence the objective function. Note that the production cost (Prod) includes both fixed setup costs and variable production costs, and in this class, both components are substantially larger than in other classes.

4.2 | Deterministic Problem with Cross-docking Satellites

In this section, we solve the deterministic 2EPRP with cross-docking satellites (2EPRP-CS), similar to the problem introduced in the study by Qiu et al. (2021). We attempted to obtain the instances used by Qiu et al. (2021), however, despite our efforts, we were unable to access those instances. The main difference between the two variants is that, instead of warehouses that can store products for future distribution, the cross-docking satellites act as distribution hubs where storage is not possible. For simplicity, we use the same notation for satellites as for warehouses. Instead of a unit holding cost, a unit handling cost $\tilde{h}_w$ is defined for

TABLE 3 Computational results for the deterministic problem with two vehicles at the plant and three vehicles per warehouse.

N_w	N_c	# Ins	HHA			BC				Diff-UB∇
			OF	Gap (%)	CPU (secs)	UB	LB	CPU (secs)	Gap (%)	(%)
2	20	4	146,586	5.71	252.5	146,586	141,324	4,291.7	5.71	0.00
	30	4	194,790	5.78	389.9	194,582	187,979	4,674.9	5.62	0.17
	40	4	242,902	4.68	801.6	242,897	236,257	7,074.1	4.68	0.00
	50	4	290,089	9.12	1,115.8	290,058	275,234	5,395.5	9.11	0.01
3	20	4	146,865	8.40	490.5	146,839	138,923	4,925.1	8.39	0.01
	30	4	192,004	6.51	693.6	191,873	184,499	5,891.8	6.43	0.09
	40	4	243,081	7.88	835.5	243,054	231,981	6,718.3	7.86	0.02
	50	4	289,287	9.41	1,350.3	289,199	274,362	6,653.0	9.37	0.04
4	20	4	145,026	6.37	584.2	144,993	139,465	5,134.0	6.36	0.01
	30	4	193,764	6.88	889.8	193,747	185,627	6,080.4	6.86	0.02
	40	4	244,220	8.35	1,043.0	244,184	232,741	7,209.4	8.33	0.02
	50	4	290,915	10.70	1,922.0	290,915	270,424	7,202.3	10.70	0.00
5	20	4	143,953	4.77	871.6	143,950	139,735	5,829.4	4.77	0.00
	30	4	192,332	7.71	1,262.5	192,327	183,874	7,208.0	7.71	0.01
	40	4	241,414	7.47	1,674.8	241,403	231,678	7,205.9	7.47	0.00
	50	4	288,971	9.95	1,902.8	288,971	269,432	7,212.4	9.95	0.00
Average		64	217,887	7.48	1,005.0	217,849	207,721	6,169.1	7.46	0.03

TABLE 4 Breakdown of costs across different components of the problem

K_p	K_w	Class	Total	First Echelon				Second Echelon			
				FE Total	Prod	Inv Plt	Trn FE	SE Total	Inv Whs	Inv Cus	Trn SE
1	1	Class 1	89,669	58,748	52,425	2,882	3,442	30,921	1,076	19,651	10,194
		Class 2	559,995	529,292	522,375	3,172	3,745	30,703	902	19,610	10,191
		Class 3	143,869	70,821	52,988	1,694	16,140	73,048	2,290	20,518	50,240
		Class 4	68,395	57,533	52,425	1,805	3,303	10,862	864	0	9,998
Average			215,482	179,099	170,053	2,388	6,657	36,384	1,283	14,945	20,156
2	2	Class 1	90,744	59,640	52,425	2,950	4,265	31,104	1,056	19,455	10,592
		Class 2	560,930	529,488	522,375	3,051	4,062	31,443	1,117	19,533	10,792
		Class 3	147,654	71,987	52,800	1,645	17,542	75,668	2,993	20,579	52,095
		Class 4	69,781	58,544	52,425	2,119	4,000	11,237	869	0	10,368
Average			217,277	179,915	170,006	2,441	7,467	37,363	1,509	14,892	20,962
2	3	Class 1	90,899	59,437	52,238	3,048	4,152	31,462	1,131	19,541	10,790
		Class 2	561,230	529,584	522,375	2,993	4,216	31,646	1,328	19,454	10,864
		Class 3	149,386	72,096	52,800	1,685	17,611	77,290	3,373	20,685	53,233
		Class 4	69,879	58,310	52,425	1,910	3,975	11,568	1,071	0	10,497
Average			217,849	179,857	169,959	2,409	7,488	37,992	1,726	14,920	21,346

each satellite $w \in \mathcal{N}_w$. We compute the handling cost as in Qiu et al. (2021), where $\tilde{h}_w = \theta h_0$, with $\theta = 0.1$ and h_0 representing the unit holding cost at the plant. The mathematical formulation of the 2EPRP-CS is given in C.

To solve the problem, we adapt the algorithm used for the 2EPRP. The structure of the algorithm, as well as the parameter settings and instance configurations, remain the same, while the mathematical formulations are updated to address the 2EPRP-CS wherever needed. Tables 5, 6, and 7 present the results for the three possible vehicle configurations at the plant and warehouses, consistent with those reported in the previous section.

The HHA was able to solve all instances of the first vehicle configuration for the 2EPRP-CS. By comparing the results of the 2EPRP and 2EPRP-CS, one can observe that the OF is higher for the latter. This is mainly due to the inability to store products at the warehouse for future deliveries. With storage not permitted at cross-docking satellites, all deliveries to satellites must be forwarded to customers within the same period. Inventory costs are typically lower at warehouses than customers. Additionally,

TABLE 5 Computational results for the 2EPRP-CS with one vehicle at the plant and one vehicle per warehouse.

			HHA			BC				
N_w	N_c	# Ins	OF	Gap (%)	CPU (secs)	UB	LB	CPU (secs)	Gap (%)	Diff-UB∇ (%)
2	20	4	151,118	4.10	269.03	142,806	142,008	7,203.08	0.95	3.18
	30	4	191,005	2.07	315.36	190,749	188,684	5,601.15	1.81	0.27
	40	4	240,107	2.02	515.70	239,850	237,120	5,040.86	1.85	0.17
	50	4	285,942	4.68	553.23	281,181	277,549	5,996.36	2.48	2.30
3	20	4	146,330	6.12	501.42	145,901	140,938	5,589.63	5.49	0.65
	30	4	193,534	5.24	456.44	192,995	187,224	4,712.68	5.04	0.21
	40	4	242,705	5.75	500.37	242,671	234,868	5,441.87	5.74	0.01
	50	4	287,625	4.89	921.99	287,338	279,572	6,076.69	4.71	0.20
4	20	4	156,182	10.87	216.87	147,846	140,624	4,599.00	7.97	2.99
	30	4	195,382	6.07	762.54	195,376	187,905	5,457.74	6.07	0.00
	40	4	243,884	5.42	621.63	243,821	236,284	6,628.21	5.37	0.06
	50	4	290,609	6.60	969.71	290,498	279,484	6,060.21	6.51	0.09
5	20	4	154,501	10.90	159.12	147,324	139,510	5,256.21	8.26	2.70
	30	4	194,297	6.90	428.98	194,287	186,188	6,780.14	6.90	0.01
	40	4	244,455	5.58	530.24	244,387	236,340	7,148.48	5.54	0.05
	50	4	289,341	6.65	802.64	289,268	278,576	7,204.13	6.64	0.02
Average		64	219,189	5.86	532.83	217,269	210,805	5,924.78	5.08	0.81

TABLE 6 Computational results for the 2EPRP-CS with two vehicles at the plant and two vehicles per warehouse.

			HHA			BC				
N_w	N_c	# Ins	OF	Gap (%)	CPU (secs)	UB	LB	CPU (secs)	Gap (%)	Diff-UB∇ (%)
2	20	4	148,303	7.42	368.19	147,612	141,916	3,046.19	6.32	1.16
	30	4	195,811	5.67	480.88	195,613	188,876	4,099.09	5.48	0.21
	40	4	244,395	5.05	574.68	244,328	237,132	6,580.58	5.01	0.05
	50	4	288,480	6.72	729.94	288,455	277,487	5,658.91	6.69	0.03
3	20	4	155,115	10.34	877.13	144,836	138,623	4,141.77	6.49	4.24
	30	4	193,445	7.19	683.50	193,351	184,994	5,529.70	7.12	0.09
	40	4	-	-	-	-	-	-	-	-
	50	4	-	-	-	-	-	-	-	-
4	20	4	157,011	13.40	459.85	148,340	138,547	4,029.71	9.99	3.59
	30	4	195,212	8.60	1,113.31	195,212	184,877	6,426.97	8.60	0.00
	40	4	245,815	7.98	1,239.66	245,662	233,994	7,210.12	7.91	0.08
	50	4	293,211	10.62	2,009.55	293,205	275,505	7,211.43	10.62	0.00
5	20	4	155,453	13.31	314.79	148,345	137,978	4,322.77	11.54	1.79
	30	4	194,865	8.90	943.21	194,863	184,181	6,672.78	8.90	0.00
	40	4	244,405	9.15	1,554.20	244,393	231,765	7,204.96	9.15	0.00
	50	4	-	-	-	-	-	-	-	-
Average		52	208,579	8.79	872.99	206,478	196,606	5,548.85	7.98	0.86

the limited capacity of second-echelon vehicles may prevent full utilization of the first-echelon vehicles' capacity. As a result, more frequent deliveries are required in both echelons, leading to higher delivery costs and, consequently, a higher OF value.

Moreover, the optimality gap in this setting is slightly higher than in the standard 2EPRP, at 5.86%, indicating the increased complexity of the problem. Nonetheless, the Diff_UB∇ remains at 0.81%, which again demonstrates the strong performance of the HHA.

TABLE 7 Computational results for the 2EPRP-CS with two vehicles at the plant and three vehicles per warehouse.

N_w	N_c	# Ins	HILS			BKS				Diff-UB∇ (%)
			OF	Gap (%)	CPU (secs)	UB	LB	CPU (secs)	Gap (%)	
2	20	4	147,486	6.25	644.04	147,016	141,784	4,391.86	5.53	0.76
	30	4	195,705	5.96	547.66	195,693	188,429	5,357.98	5.94	0.02
	40	4	246,684	6.34	953.32	246,583	237,145	6,353.92	6.28	0.07
	50	4	291,802	8.35	1,503.41	291,801	277,196	6,149.98	8.35	0.00
3	20	4	144,917	7.11	1,700.52	144,872	138,428	5,829.22	7.05	0.07
	30	4	194,120	7.87	1,065.78	194,120	184,725	6,673.11	7.87	0.00
	40	4	-	-	-	-	-	-	-	-
	50	4	-	-	-	-	-	-	-	-
4	20	4	156,589	13.39	700.39	150,192	138,364	4,854.37	11.67	1.84
	30	4	195,930	9.66	1,115.02	195,930	184,422	6,390.98	9.66	0.00
	40	4	245,596	10.06	1,734.18	245,596	231,498	7,209.96	10.06	0.00
	50	4	293,978	13.12	2,073.00	293,953	268,143	7,154.80	13.11	0.00
5	20	4	154,917	12.51	836.33	148,751	137,814	6,354.49	10.81	1.80
	30	4	195,267	9.88	1,487.29	195,267	183,634	7,205.81	9.88	0.00
	40	4	244,940	10.39	2,186.08	244,940	230,318	7,204.38	10.39	0.00
	50	4	-	-	-	-	-	-	-	-
Average		52	208,302	9.30	1,272.85	207,286	195,531	6,240.83	8.97	0.35

For the other two vehicle configurations of the 2EPRP-CS, a significant observation is that some instances could not be solved. Specifically, the cases with 3 satellites and 40 or 50 customers, as well as those with 5 satellites and 50 customers, could not be handled by the HHA. This failure is attributed to insufficient vehicle capacity to meet customer demand, primarily due to the absence of storage capability at the satellites, which otherwise could have been used to buffer deliveries for future periods.

Interestingly, the BC algorithm also failed to find any feasible solution for these instances and did not terminate with an infeasibility status. For the remaining instances, we again observe a slightly higher OF value and optimality gap compared to the standard 2EPRP. Nevertheless, the Diff_UB∇ remains low at 0.86% and 0.35% for the second and third vehicle configurations, respectively.

It is also worth highlighting that the largest instance solved by Qiu et al. (2021) consisted of 2 satellites and 35 customers over 3 periods, with 2 vehicles in the first and 4 in the second echelon. In contrast, we solved instances as large as 5 warehouses and 40 customers over 6 periods, with 2 vehicles in the first echelon and 15 in the second echelon. The average optimality gap for this configuration was 10.39%, while the BC algorithm was unable to improve the UB provided by the HHA.

4.3 | Stochastic Problem

In this section, we present the results of the experiments for the stochastic version of the problem. All parameters are set similarly to those in the deterministic case. The key difference is that, since the second echelon of the problem is scenario-decomposable, we leverage this structure by solving scenarios in parallel using 40 CPU threads, enabling up to 40 scenarios to be processed simultaneously. For the experiments in this section, the number of scenarios S is set to 100, and the uncertainty level δ is set to 0.2, unless stated otherwise.

After solving the problem with the HHA, the resulting solution is used as a warm start for the BC algorithm. However, the LBs provided by the BC are generally weak; thus, we use the LB of the Wait-and-See (WS) problem as a valid LB to the recourse problem (RP). While this LB is theoretically weak as well, it consistently outperforms the one obtained from the BC. This highlights one of the main challenges in solving this problem, which is to find a strong LB.

Table 8 summarizes the main findings. The first five columns show the number of warehouses, the number of vehicles at the plant, the number of vehicles at each warehouse, the number of scenarios, and the number of instances for each configuration. Column **CPU-HHA** reports the average CPU time for the HHA, and column **CPU-BC** displays the computation time for the BC algorithm. We should highlight that building the initial solution from the given warm start may take anywhere from several

TABLE 8 Computational results for the stochastic problem

N_w	K_p	K_w	S	# Ins	CPU-HHA (secs)	CPU-BC (secs)	Gap (%)	Diff-UB∇ (%)	VSS^{LB} (%)	VSS^{UB} (%)	$EVPI^{UB}$ (%)
	1	1	100	8	3,587.8	7,868.5	14.8%	0.0%	17.9%	31.1%	14.8%
2	2	2	100	8	3,638.0	8,993.8	14.7%	0.0%	18.8%	33.1%	14.7%
	2	3	100	8	3,640.1	7,643.7	15.7%	0.0%	16.7%	33.2%	15.7%
	1	1	100	8	3,430.5	8,065.5	19.1%	0.0%	13.6%	32.6%	19.1%
3	2	2	100	8	3,587.3	7,996.4	19.3%	0.0%	11.8%	32.3%	19.3%
	2	3	100	8	3,688.4	7,871.7	21.9%	0.0%	11.5%	34.3%	21.9%
	1	1	100	3	3,656.0	7,963.5	23.0%	0.0%	9.9%	36.2%	23.0%
4	2	2	100	3	3,682.3	9,160.0	26.6%	0.0%	4.0%	34.6%	26.6%
	2	3	100	3	3,678.6	8,471.8	28.6%	0.0%	5.1%	34.5%	28.6%
	1	1	100	3	3,668.2	10,670.4	23.6%	0.0%	10.3%	39.3%	23.6%
5	2	2	100	3	3,688.6	10,670.4	22.8%	0.0%	4.4%	34.4%	22.8%
	2	3	100	3	3,680.0	16,631.7	24.6%	0.0%	4.2%	37.1%	24.6%
Average			100	66	3,617.2	8,563.6	19.4%	0.0%	12.7%	33.6%	19.4%

minutes to several hours in CPLEX, due to the large scale of the problem. The two-hour computation limit is applied only after the initial solution is built. Therefore, the reported CPU time often exceeds two hours, reflecting the total time required for both building the initial solution and solving the problem with BC. Column **Gap** provides the average relative optimality gap in %, based on the best UB and LB obtained for the problem. Column **Diff-UB∇** shows the average improvement in the UB of the HHA after solving the problem with BC.

To evaluate the value of the stochastic solution, we report the Value of the Stochastic Solution (VSS) and the Expected Value of Perfect Information (EVPI), both expressed as percentages. However, optimality is not guaranteed for the RP nor for the evaluation problems, i.e., the expected value of the expected value (EEV) problems and the WS problems. Thus, the VSS^{LB}, VSS^{UB}, and $EVPI^{UB}$ are reported as ranges around their true values, which we cannot compute exactly. These values are calculated as follows:

$$VSS^{LB} = \frac{EEV^{LB} - RP^{UB}}{EEV^{LB}} \tag{41}$$

$$VSS^{UB} = \frac{EEV^{UB} - RP^{LB}}{EEV^{UB}} \tag{42}$$

$$EVPI^{UB} = \frac{RP^{UB} - WS^{LB}}{RP^{UB}}, \tag{43}$$

where EEV^{LB} and EEV^{UB} are the LB and UB to the EEV problem, and WS^{LB} is the LB to the WS problem. Note that the value of WS^{LB} is similar to the problem's gap, as WS^{LB} provides the LB to the problem in all cases. The $EVPI^{LB}$ is also not reported, as calculating it using a similar approach would result in negative values, which are lower than its natural LB of zero.

As seen in Table 8, the average HHA computation time remains around 1 hour, indicating that the time limit was the primary stopping criterion. The BC algorithm, on the other hand, requires approximately 2.5 hours on average. Despite receiving the HHA solution as a warm start, the BC fails to improve the given solution in any of the instances, resulting in an average improvement of 0.0%. The average optimality gap is around 19.4%, mainly due to the lack of a strong LB.

The reported average VSS ranges from 12.7% to 33.6%, demonstrating the added value of solving the stochastic model instead of relying on expected-value decisions. While the LB of VSS tends to decrease for larger instances due to weaker LBs in the EEV problems, the UB remains above 30% in most cases. The $EVPI^{UB}$ is also reported at 19.4%, which is calculated similar to the LB of the problem using the WS^{LB}.

While this number is relatively large, we expect the true EVPI to be smaller

which corresponds to the LB of the problem since we used the WS solution to estimate it. While this number is relatively large, we expect the true EVPI to be smaller, as the optimality gap in the evaluation problems introduces overestimation.

Figure 2 displays the average of the best obtained UBs for the stochastic problems, compared to the UBs for the WS and EEV problems, across different numbers of warehouses. Only instances with a similar number of customers are used in this figure;

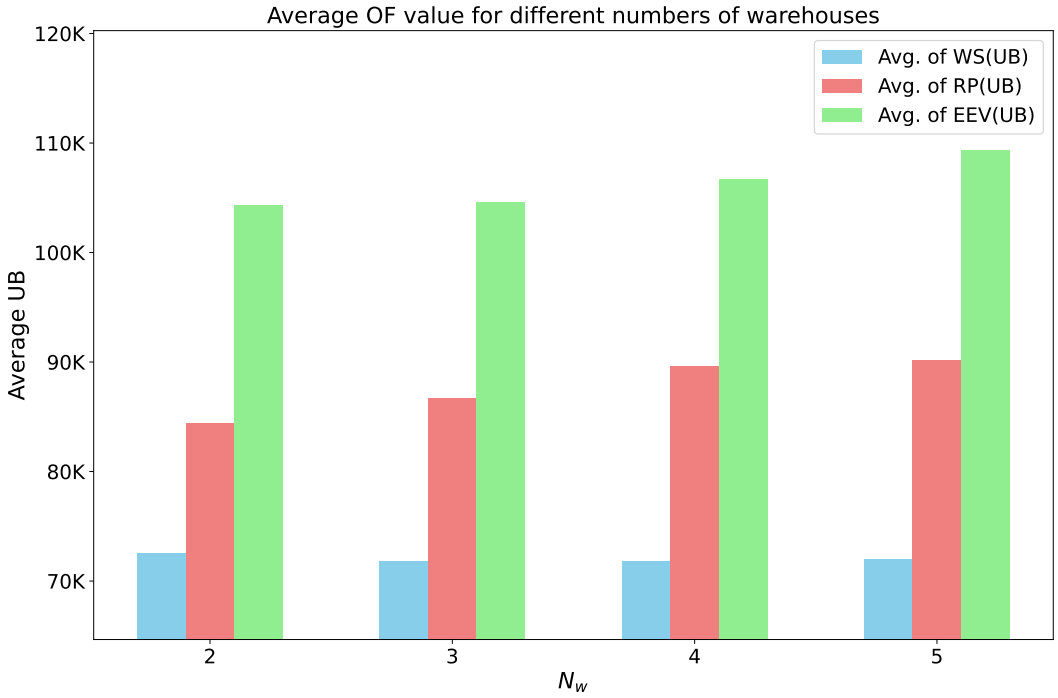


FIGURE 2 Objective function values for different numbers of warehouses

that is, only instances with 20, 25, or 30 customers are shown. Based on the figure, we observe that the objective function value slightly increases as the number of warehouses increases. This is mainly due to higher transportation costs resulting from having more vehicles with lower capacity in the second echelon of the problem.

To provide further insight into the role of customer-to-warehouse assignments, Table 9 reports related results. Column **NW** shows the average number of distinct warehouses that serve each customer across all periods and scenarios. A value close to 1 indicates that allowing flexibility in customer-to-warehouse assignments would have little impact on the solution. In contrast, values closer to $\mathcal{N}_w$ suggest that ignoring this flexibility would increase the OF value. The results show that NW is 2 when there are two warehouses, and it increases as the number of warehouses grows. This indicates that, for each customer, there usually is at least one period and scenario in which they are served by a warehouse other than their most frequently assigned warehouse.

We introduce an additional metric to capture the relative importance of warehouses to each customer. The importance metric is computed as the proportion of periods and scenarios ($T \times S$) in which a customer is served by each warehouse. For each customer, we first calculate the importance of all warehouses and rank them in descending order. Then the average of these values across all customers provide a measure of warehouse importance that is relative to each customer rather than tied to a specific warehouse ID.

The results indicate that, on average, when there are two warehouses, customers are served by their most frequent warehouse 55.1% of the time and by their second most frequent warehouse 5.0% of the time. Note that in some periods, certain customers are not visited by any warehouse. This needs to be accounted for to ensure that the percentages sum to 100%. Interestingly, the importance of the most frequent warehouse, as well as the percentage of periods in which customers are not visited, remains relatively stable as the number of warehouses increases. In contrast, the importance of the second most frequent warehouse declines, with its share redistributed among the other warehouses. This trend occurs because having more warehouses increases the likelihood that some of them are geographically closer to a specific customer, allowing the customer to be served by different warehouses in different periods or scenarios. Even if a warehouse is farther than the most frequent one, serving the customer from that warehouse in certain cases can help reduce overall costs by balancing capacity utilization and improving routing efficiency.

TABLE 9 Customer-to-warehouse assignment across periods and scenarios

$\mathcal{N}_w$	#Ins	NW	Most Frequent (%)	Second Most Frequent (%)	Third Most Frequent (%)	Fourth Most Frequent (%)	Fifth Most Frequent (%)	Not Visited (%)
2	9	2.00	55.1%	5.0%	-	-	-	39.9%
3	9	2.93	53.9%	6.0%	1.9%	-	-	38.2%
4	9	3.65	54.1%	4.9%	1.9%	0.6%	-	38.5%
5	9	4.19	55.9%	4.2%	1.2%	0.6%	0.3%	37.8%
Average	36	3.19	54.8%	5.1%	1.7%	0.6%	0.3%	38.6%

TABLE 10 Sensitivity analysis for different uncertainty ranges

δ	N_w	S	# Ins	CPU-HHA (secs)	CPU-BC (secs)	Gap (%)	Diff-UB∇ (%)	VSS^{LB} (%)	VSS^{UB} (%)	$EVPI^{UB}$ (%)
0.2	3	100	3	3,704.5	8,522.9	20.7%	0.0%	10.9%	32.9%	20.7%
	4	100	3	3,682.3	12,588.9.0	26.6%	0.0%	4.0%	34.6%	26.6%
0.4	3	100	3	3,671.1	8,623.1	22.1%	0.0%	31.1%	48.3%	22.1%
	4	100	3	3,678.0	8,785.1	26.7%	0.0%	25.6%	49.6%	26.7 %
0.6	3	100	3	3,661.0	8,688.8	25.9%	0.0%	42.9%	58.7%	25.9%
	4	100	3	3,697.7	8,750.9	32.7%	0.0%	35.0%	58.6%	32.7%
0.8	3	100	3	3,726.9	8,630.9	32.1%	0.0%	48.8%	68.3%	32.1%
	4	100	3	3,732.4	9,034.5	41.9%	0.0%	44.8%	66.5%	41.9%
Average		100	24	3,694.2	9252.7	28.6%	0.0%	29.8%	51.6%	28.6%

4.4 | Sensitivity Analysis

In this section, we provide a more detailed analysis of the stochastic problem by discussing how changing parameters such as the uncertainty level and the number of scenarios can affect the solution. To this end, we consider different ranges of uncertainty levels to observe how increasing stochasticity influences the problem. We also solve the problem under various numbers of scenarios to examine how different sample sizes impact the obtained solution.

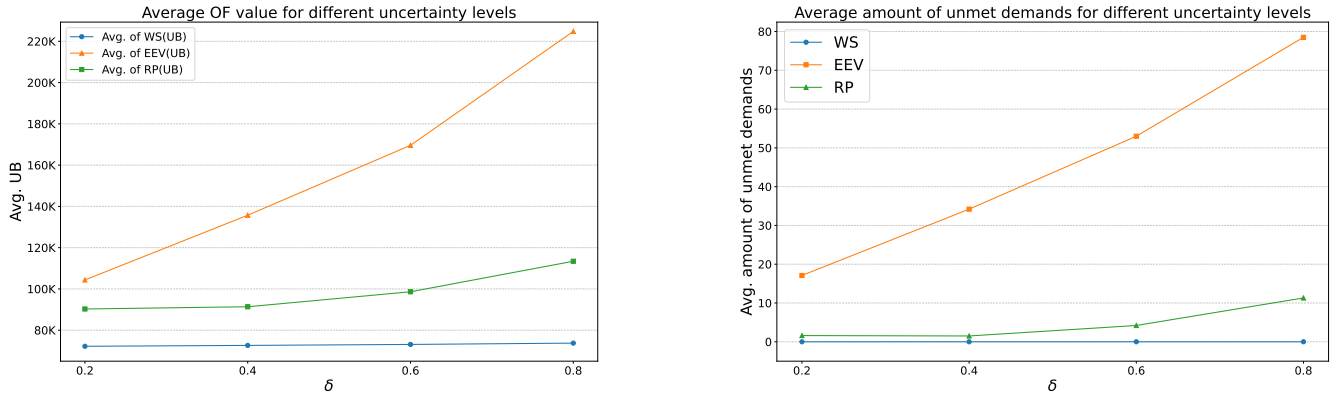
For the sensitivity analysis, we use a subset of the introduced instances: those with 3 or 4 warehouses, 20, 25, and 30 customers, 2 vehicles at the plant, and 2 vehicles at each warehouse. The main reason for conducting experiments on a subset of the instances is that solving the problems, especially the evaluation problems, is computationally expensive, which prevents us from running experiments on the entire dataset. Therefore, we selected instances that are representative of the broader set while still allowing us to extract insights regarding various parameter settings. The number of instances used in this part is 6, and the following sections provide further details on the sensitivity analysis.

4.4.1 | Uncertainty Level

In this section, we present the results of the problem under various uncertainty levels δ to demonstrate how different levels of uncertainty affect the results. The uncertainty levels are set to 0.2, 0.4, 0.6, and 0.8, and as mentioned before, instances with 3 and 4 warehouses and 20, 25, and 30 customers were used, which led to a total of 6 instances per each uncertainty level and 24 instances in total. Table 10 presents the results for this analysis. The optimality gap is larger for higher levels of uncertainty; however, this is mainly because WS^{LB} is used as the LB of the problem, and it decreases as uncertainty increases, which results in a larger calculated gap.

The optimality gap is larger for higher levels of uncertainty; however, this is mainly because WS^{LB} decreases as uncertainty grows, which leads to a larger calculated gap.

The results demonstrate higher VSS in environments with higher uncertainty, showing an average of 46.8% for the VSS^{LB} when $\delta = 0.8$. This shows the significance of considering uncertainty when demand volatility is high. Even in environments with moderate uncertainty, it is essential to take stochasticity into account, as the average VSS ranges between 28.4% and 49.0% with δ set to 0.4. This section shows the importance of incorporating stochasticity despite the difficulties of solving such



(a) Avg. UB

(b) Avg. amount of unmet demands

FIGURE 3 Average UB and amount of unmet demands for different uncertainty levels

a complex problem and the existing limitations in guaranteeing a near-optimal solution. The results show that even with the existing limitations, using HHA for solving the S2EPRP could result in significant cost reductions and improved performance.

Figure 3 provides more details on the OF value under different uncertainty levels for the RP as well as the WS and EEV. As shown in Figure 3a, the WS stays relatively stable even for higher uncertainty levels, which shows the total cost if perfect information were available and no uncertainty was present. On the other hand, the RP increases in more uncertain environments, while the highest impact is on the EEV, where the first-stage decisions are based on the mean value of demands. This highlights the significance of considering stochasticity in the problem, especially in environments with high volatility.

Figure 3b illustrates the number of units of unmet demand under different uncertainty levels, following a pattern similar to that of the OF values. One notable observation is that the number of units of unmet demand remains zero in all cases for the WS problem, as expected, since the demand is known in advance. For the RP, the average amount of unmet units stays close to zero in low-uncertainty environments and does not exceed 11.3 units even at the highest uncertainty level. In contrast, solving the deterministic problem using average demand can lead to significantly more unmet demand, ranging from 17.1 to 78.5 units on average, which can dramatically affect the overall costs.

4.4.2 | Number of Scenarios

In this section, we provide the sensitivity analysis of the stochastic problem under different numbers of scenarios. The aim is to assess how the solution is affected when the sample size changes. In addition to the standard 100-scenario case used throughout the paper, we also solve the problem with 20, 50, and 200 scenarios. While increasing the number of scenarios is generally expected to improve solution quality, it also drastically increases the problem's complexity, creating a computational limitation that may prevent the problem from being solved. Thus, the goal of this analysis is to investigate how the solution behaves under varying sample sizes.

For this part, we use the same data configurations as in the previous section, with a total of 24 instances. A summary of the results is provided in Table 11. While the BC algorithm is provided with the HHA solution as a warm start, for instances with 200 scenarios and 4 warehouses, CPLEX was unable to build the initial solution within the time limit. As a result, the BC algorithm was not even started for any instance in this configuration.

For the case with 20 scenarios, we observe a slight improvement in the OF after solving the problem with the BC algorithm. However, this improvement is small and highlights the ability of the HHA to obtain high-quality solutions for the problem. In terms of VSS^{LB}, there is an increasing trend in the values, which we believe results from a combination of two factors. First, using a higher number of scenarios yields a better sample set that more closely resembles real-world uncertainty. Second, the increase in VSS^{LB} may also be a consequence of the growing complexity of the problem as more scenarios are considered, which can lead to weaker UBs. Nonetheless, it is important to note that using 100 scenarios, which is the main setting throughout this study, represents a reasonable compromise between problem complexity and capturing existing uncertainty.

Figure 4 provides more details on the OF values for different numbers of scenarios. This figure illustrates how increasing the number of scenarios results in higher UBs for the RP, while also showing that the OF appears to converge after 100 scenarios.

TABLE 11 Sensitivity analysis for different numbers of scenarios

S	N_w	δ	# Ins	CPU-HHA (secs)	CPU-BC (secs)	Gap (%)	Diff-UB∇ (%)	VSS^{LB} (%)	VSS^{UB} (%)	$EVPI^{UB}$ (%)
20	3	0.2	3	2,304.8	7,205.3	17.9%	0.1%	9.6%	27.8%	17.9%
	4	0.2	3	2,900.3	7,206.7	19.7%	0.1%	4.8%	34.9%	19.7%
50	3	0.2	3	3,333.2	7,212.4	18.8%	0.0%	10.0%	30.7%	18.8%
	4	0.2	3	3,415.2	7,217.1	23.1%	0.0%	4.6%	37.7%	23.1%
100	3	0.2	3	3,704.5	8,522.9	20.7%	0.0%	10.9%	32.9%	20.7%
	4	0.2	3	3,682.3	12,588.9	26.6%	0.0%	4.0%	34.6%	26.6%
200	3	0.2	3	3,771.4	11,703.3	21.5%	0.0%	12.8%	27.5%	21.5%
	4	0.2	3	3,699.9	-	23.8%	-	7.7%	46.2%	23.8%
Average		0.2	24	3,351.4	8,503.3	21.5%	0.0%	16.6%	37.3%	21.5%

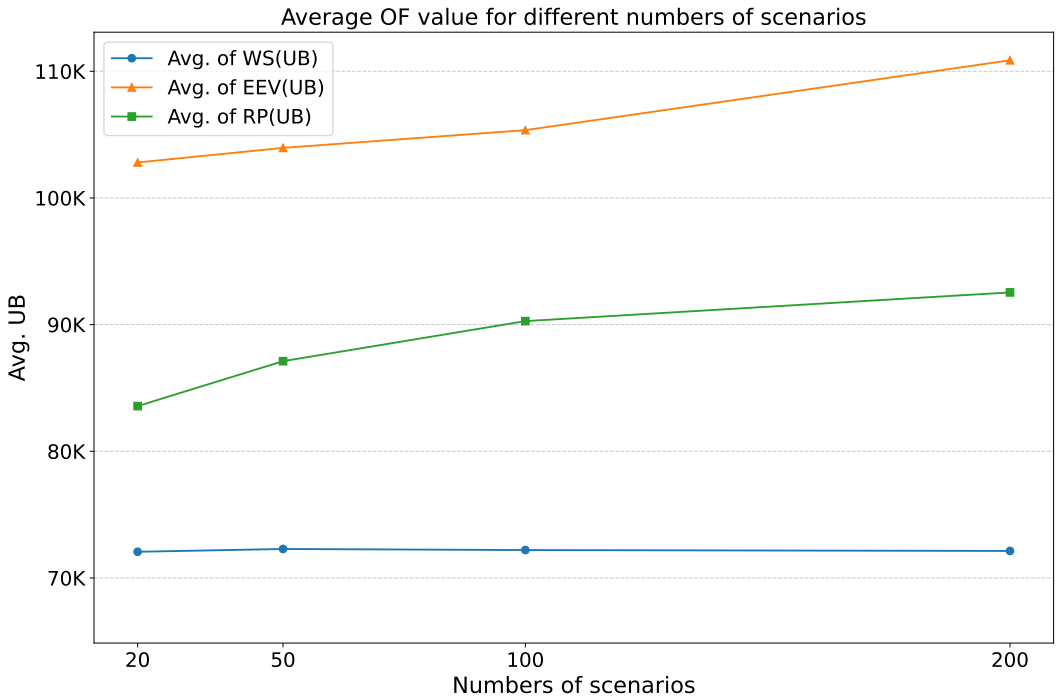


FIGURE 4 Objective function values for different numbers of scenarios

5 | CONCLUSION

This study addresses the stochastic two-echelon production routing problem. We present a two-stage stochastic formulation in which the first stage models the production decisions at the plant, as well as the deliveries to warehouses. The second stage captures the decisions related to customer inventories and deliveries from warehouses to customers. A fleet of homogeneous vehicles is considered at the plant, along with a separate set of homogeneous vehicles at each warehouse. These warehouse vehicles are assumed to have smaller capacities suitable for intra-city deliveries.

One important assumption of our study is the possibility of assigning customers to different warehouses in different periods or scenarios, which enhances overall efficiency. Additionally, we incorporate adaptive routing into the stochastic model, allowing the delivery routes to vary across scenarios. This flexibility contributes to cost savings but also significantly increases the problem's complexity.

To address this complexity, we developed a hybrid heuristic algorithm that decomposes the problem into two subproblems. The first subproblem handles first-stage decisions, including production planning and deliveries to warehouses, based on estimated demands of customers assigned to each warehouse. A MIP is solved for this subproblem in each iteration. The second subproblem concerns second-echelon routing and deliveries, which are solved using an ILS algorithm followed by an LP to determine the values of continuous variables. These two steps are solved iteratively, and in each iteration, customer-to-warehouse reassignments may occur if potential cost savings are identified. A perturbation phase is also incoporated to explore a wider region of the solution space.

While the main objective of this algorithm is to solve the S2EPRP-AR, it also performs well on large-scale deterministic instances. We first present computational experiments on the deterministic version to evaluate the algorithm's performance. The test instances include cases with up to 5 warehouses and 50 customers, with 2 vehicles in the first echelon and 15 vehicles in the second. The algorithm solved these instances in approximately 30 minutes on average, with an average optimality gap of 7.46%. A total of 192 instances were solved in the deterministic case.

For the stochastic version, we solved 66 instances. The BC algorithm, even with the HHA solution provided as a warm start, was unable to improve the objective function within a 2-hour time limit, demonstrating the difficulty of the problem. Despite this, the developed algorithm was able to solve all instances. An average optimality gap of 19.4% was observed. This relatively large gap is mainly due to the weak LBs obtained from the WS problems. Finding stronger LBs remains a key challenge in this study.

For future research, incorporating the supplier level could improve supply chain integration and enable more coordinated decision-making. Additionally, modeling uncertainty on both the supply and demand sides could result in a more reliable system.

AUTHOR CONTRIBUTIONS

This is an author contribution text. This is an author contribution text. This is an author contribution text. This is an author contribution text. This is an author contribution text.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (RGPIN-2019-05036, RGPIN-2019-05759 and RGPAS-2019-00100), and the HEC Montréal Research Chair in Supply Chain Operations Planning. The authors also acknowledge the support of Digital Research Alliance of Canada for providing computational resources for performing this research.

FINANCIAL DISCLOSURE

None reported.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

REFERENCES

Absi, N., Archetti, C., Dauzère-Pérès, S., Feillet, D., 2015. A two-phase iterative heuristic approach for the production routing problem. Transportation Science 49, 784–795.

Absi, N., Archetti, C., Dauzère-Pérès, S., Feillet, D., Speranza, M.G., 2018. Comparing sequential and integrated approaches for the production routing problem. European Journal of Operational Research 269, 633–646.

Adulyasak, Y., Cordeau, J.F., Jans, R., 2014a. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. INFORMS Journal on Computing 26, 103–120.

Adulyasak, Y., Cordeau, J.F., Jans, R., 2014b. Optimization-based adaptive large neighborhood search for the production routing problem. Transportation Science 48, 20–45.

Adulyasak, Y., Cordeau, J.F., Jans, R., 2015a. Benders decomposition for production routing under demand uncertainty. Operations Research 63, 851–867.

Adulyasak, Y., Cordeau, J.F., Jans, R., 2015b. The production routing problem: A review of formulations and solution algorithms. Computers & Operations Research 55, 141–152.

Agra, A., Requejo, C., Rodrigues, F., 2018. An adjustable sample average approximation algorithm for the stochastic production-inventory-routing problem. Networks 72, 5–24.

Alvarez, A., Cordeau, J., Jans, R., 2022a. The consistent production routing problem. Networks 80, 356–381.

Alvarez, A., Miranda, P., Rohmer, S., 2022b. Production routing for perishable products. Omega 111, 102667.

Alvarez, A., Munari, P., Morabito, R., 2018. Iterated local search and simulated annealing algorithms for the inventory routing problem. International Transactions in Operational Research 25, 1785–1809.

Applegate, D., Bixby, R., Chvátal, V., Cook, W., 2020. Concorde TSP solver. URL: https://www.math.uwaterloo.ca/tsp/concorde.html.

Archetti, C., Bertazzi, L., Paletta, G., Speranza, M.G., 2011. Analysis of the maximum level policy in a production-distribution system. Computers & Operations Research 38, 1731–1746.

Armentano, V.A., Shiguemoto, A.L., Løkketangen, A., 2011. Tabu search with path relinking for an integrated production–distribution problem. Computers & Operations Research 38, 1199–1209.

Avci, M., Topaloglu Yildiz, S., 2020. A mathematical programming-based heuristic for the production routing problem with transshipments. Computers & Operations Research 123, 105042.

Bard, J.F., Nananukul, N., 2009. The integrated production–inventory–distribution–routing problem. Journal of Scheduling 12, 257–280.

Bard, J.F., Nananukul, N., 2010. A branch-and-price algorithm for an integrated production and inventory routing problem. Computers & Operations Research 37, 2202–2217.

Ben Ahmed, M., Hvattum, L.M., Agra, A., 2023. The effect of different mathematical formulations on a matheuristic algorithm for the production routing problem. Computers Operations Research 155, 106232.

Bräysy, O., Gendreau, M., 2005. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. Transportation Science 39, 104–118.

Chandra, P., 1993. A dynamic distribution model with warehouse and customer replenishment requirements. Journal of the Operational Research Society 44.

Chandra, P., Fisher, M.L., 1994. Coordination of production and distribution planning. European Journal of Operational Research 72, 503–517.

Chitsaz, M., Cordeau, J.F., Jans, R., 2019. A unified decomposition matheuristic for assembly, production, and inventory routing. INFORMS Journal on Computing 31, 134–152.

Chitsaz, M., Cordeau, J.F., Jans, R., 2020. A branch-and-cut algorithm for an assembly routing problem. European Journal of Operational Research 282, 896–910.

Fischetti, M., Lodi, A., 2003. Local branching. Mathematical programming 98, 23–47.

Gruson, M., Bazrafshan, M., Cordeau, J.F., Jans, R., 2019. A comparison of formulations for a three-level lot sizing and replenishment problem with a distribution structure. Computers & Operations Research 111, 297–310.

Gruson, M., Cordeau, J.F., Jans, R., 2023. Split demand and deliveries in an integrated three-level lot sizing and replenishment problem. Computers & Operations Research , 106434.

Gruson, M., Cordeau, J.F., Jans, R., 2024. Split demand and deliveries in an integrated three-level lot sizing and replenishment problem. Computers & Operations Research 161, 106434.

Hrabec, D., Hvattum, L.M., Hoff, A., 2022. The value of integrated planning for production, inventory, and routing decisions: A systematic review and meta-analysis. International Journal of Production Economics 248, 108468.

Jans, R., 2009. Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints. INFORMS Journal on Computing 21, 123–136.

Kermani, A., Cordeau, J.F., Jans, R., 2024. A progressive hedging-based matheuristic for the stochastic production routing problem with adaptive routing. Computers & Operations Research 169, 106745.

Kermani, A., Cordeau, J.F., Jans, R., 2025. The stochastic production routing problem with adaptive routing and service level constraints .

Kumar, R.S., Kondapaneni, K., Dixit, V., Goswami, A., Thakur, L.S., Tiwari, M.K., 2016. Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach. Computers & Industrial Engineering 99, 29–40.

Li, Y., Chu, F., Chu, C., Zhu, Z., 2019. An efficient three-level heuristic for the large-scaled multi-product production routing problem with outsourcing. European Journal of Operational Research 272, 914–927.

Manousakis, E.G., Kasapidis, G.A., Kiranoudis, C.T., Zachariadis, E.E., 2022. An infeasible space exploring matheuristic for the production routing problem. European Journal of Operational Research 298, 478–495.

Miranda, P.L., Cordeau, J.F., Ferreira, D., Jans, R., Morabito, R., 2018. A decomposition heuristic for a rich production routing problem. Computers & Operations Research 98, 211–230.

Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Computers Operations Research 24, 1097–1100.

Mousavi, R., Bashiri, M., Nikzad, E., 2022. Stochastic production routing problem for perishable products: Modeling and a solution algorithm. Computers & Operations Research 142, 105725.

Neves-Moreira, F., Almada-Lobo, B., Cordeau, J.F., Guimarães, L., Jans, R., 2019. Solving a large multi-product production-routing problem with delivery time windows. Omega 86, 154–172.

Perboli, G., Tadei, R., Vigo, D., 2011. The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. Transportation Science 45, 364–380.

Qiu, Y., Ni, M., Wang, L., Li, Q., Fang, X., Pardalos, P.M., 2018. Production routing problems with reverse logistics and remanufacturing. Transportation Research Part E: Logistics and Transportation Review 111, 87–100.

Qiu, Y., Qiao, J., Pardalos, P.M., 2017. A branch-and-price algorithm for production routing problems with carbon cap-and-trade. Omega 68, 49–61.

Qiu, Y., Qiao, J., Pardalos, P.M., 2019. Optimal production, replenishment, delivery, routing and inventory management policies for products with perishable inventory. Omega 82, 193–204.

Qiu, Y., Zhou, D., Du, Y., Liu, J., Pardalos, P.M., Qiao, J., 2021. The two-echelon production routing problem with cross-docking satellites. Transportation Research Part E: Logistics and Transportation Review 147, 102210.

Rodrigues, L.F., Santos, M.O.D., Almada-Lobo, B., 2023. A memetic algorithm for the multi-product production routing problem. Computers Industrial Engineering 182, 109388.

Russell, R.A., 2017. Mathematical programming heuristics for the production routing problem. International Journal of Production Economics 193, 40–49.

Schenekemberg, C.M., Guimarães, T.A., Augusto, C.A., Coelho, L.C., 2023. A three-front parallel branch-and-cut algorithm for production and inventory routing problems. Transportation Science .

Schenekemberg, C.M., Scarpin, C.T., Pécora, J.E., Guimarães, T.A., Coelho, L.C., 2021. The two-echelon production-routing problem. European Journal of Operational Research 288, 436–449.

Solyalı, O., Süral, H., 2017. A multi-phase heuristic for the production routing problem. Computers & Operations Research 87, 114–124.

Tempelmeier, H., 2013. Stochastic Lot Sizing Problems. Springer. volume 192. p. 313–344.

Vadseth, S.T., Andersson, H., Stålhane, M., Chitsaz, M., 2023. A multi-start route improving matheuristic for the production routeing problem. International Journal of Production Research 61, 7608–7629.

Wang, C., Ni, Y., Yang, X., 2021. The production routing problem under uncertain environment. IEEE Access 9, 15375–15387.

Zhang, Y., Alshraideh, H., Diabat, A., 2018. A stochastic reverse logistics production routing model with environmental considerations. Annals of Operations Research 271, 1023–1044.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Kermani A., Cordeau J.-F., and Jans R. The Stochastic Two-Echelon Production Routing Problem with Adaptive Routing. *Networks.* 2021;00(00):1–18.

APPENDIX

A SUMMARY OF SETS, PARAMETERS, AND DECISION VARIABLES

Table A1 presents the descriptions of all sets and parameters, and Table A2 provides the descriptions of the decision variables used in the problem introduced in Section 2.

TABLE A1 Sets and parameters

Sets:	
$\mathcal{N}_o$	Plant node, $\mathcal{N}_o = \{0\}$.
$\mathcal{N}_w$	Set of warehouses, $\mathcal{N}_w = \{1, ..., N_w\}$.
$\mathcal{N}_c$	Set of customers, $\mathcal{N}_c = \{N_w + 1, ..., N_w + N_c\}$.
$\mathcal{N}$	Set of nodes, $\mathcal{N} = \mathcal{N}_o \cup \mathcal{N}_w \cup \mathcal{N}_c$.
$\mathcal{E}_{pw}$	Set of edges of the first echelon, $\mathcal{E}_{pw} = \{(i,j) : i,j \in \mathcal{N}_0 \cup \mathcal{N}_w, i < j\}$.
$\mathcal{E}_{wc}$	Set of edges of the second echelon, $\mathcal{E}_{wc} = \{(i,j) : i \in \mathcal{N}_w, j \in \mathcal{N}_c \vee i,j \in \mathcal{N}_c, i < j\}$.
$\mathcal{E}$	Set of edges, $\mathcal{E} = \mathcal{E}_{pw} \cup \mathcal{E}_{wc}$.
$\mathcal{R}$	Set of all possible routes in the first echelon, where $\mathcal{R} = \{1, \ldots, R\}$ and $R = 2^{N_w} - 1$.
$\mathcal{T}$	Set of time periods, $\mathcal{T} = \{1, ..., T\}$.
$\mathcal{K}_p$	Set of vehicles in the plant, $\mathcal{K}_p = \{1, ..., K_p\}$.
$\mathcal{K}_w$	Set of vehicles in warehouse $\mathcal{K}_w = \{\sum_{l=1}^{w-1} K_l + 1, ..., \sum_{l=1}^{w} K_w\}$.
$\mathcal{S}$	Set of scenarios, $\mathcal{S} = \{1, ..., S\}$.
$\tilde{\mathcal{E}}_{wc}(\eta)$	Subset of edges $(i,j) \in \mathcal{E}_{wc}$ such that $i,j \in \eta$ and $\eta \subseteq \mathcal{N}_w \cup \mathcal{N}_c$ is a given set of nodes.
$\tilde{\epsilon}_{wc}(\{i\})$	Subset of edges incident to node $i \in \mathcal{N}_w \cup \mathcal{N}_c$.
Parameters:	
F	Setup cost.
u	Unit production cost.
$\tilde{c}_r$	Optimal cost of route $r \in \mathcal{R}$.
c_{ij}	Cost of visiting edge $(i,j) \in \mathcal{E}_{wc}$.
h_i	Per unit holding cost at node $i \in \mathcal{N}$.
α_i	Unit penalty cost of unmet demand for customer $i \in \mathcal{N}_c$.
C	Production capacity.
$\mathcal{Q}^p$	Capacity of vehicles located at the plant.
$\mathcal{Q}^w$	Capacity of vehicles located at warehouses.
a_{wr}	Binary parameter, equal to 1 if warehouse w is visited in route r, 0 otherwise.
ξ_s	Probability of scenario $s \in \mathcal{S}$.
d_{it}^s	Demand of customer i, in period t, under scenario s.
I_{i0}	Initial inventory at node $i \in \mathcal{N}$.
L_i	Storage capacity at node $i \in \mathcal{N}$.
$\mathcal{M}_w$	Upper bound on delivery quantity to warehouse w. $\mathcal{M}_w = \min\{\mathcal{Q}^p, L_w\}, \forall w \in \mathcal{N}_w$.
$\mathcal{M}'^s_{it}$	Upper bound on delivery quantity to customer i in period t under scenario s. $\mathcal{M}'^s_{it} = \min\{\mathcal{Q}^w, L_i, \sum_{l=t}^{T} d_{il}^s\}, \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \mathcal{S}$.

TABLE A2 Decision variables

Decision variables:	
First-stage Decisions:	
y_t	Setup decision, equal to 1 if a setup takes place in period $t \in \mathcal{T}$, and 0 otherwise.
p_t	Production quantity in period $t \in \mathcal{T}$.
I_{0t}	Inventory of plant at the end of each period t.
q_{wrt}	Quantity of products delivered to warehouse $w \in \mathcal{N}_w$ using route $r \in \mathcal{R}$ in period t.
o_{rt}	Binary variable, equal to 1 if route r is selected in period t in the first echelon, 0 otherwise.
Second-stage Decisions:	
I^s_{it}	Inventory of node $i \in \mathcal{N}_w \cup \mathcal{N}_c$ at the end of each period t in scenario s.
l^s_{it}	Unsatisfied demand of customer i in period t and scenario s.
w^s_{ikt}	Quantity of products delivered to customer $i \in \mathcal{N}_c$ by vehicle $k \in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w$ in period t in scenario s.
z^s_{wkt}	Warehouse visiting decision, equal to 1 if node $w \in \mathcal{N}_w$ is visited by vehicle $k \in \mathcal{K}_w$ in period t in scenario s, 0 otherwise.
z^s_{ikt}	Customer visiting decision, equal to 1 if node $i \in \mathcal{N}_c$ is visited by vehicle $k \in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w$ in period t in scenario s, 0 otherwise.
x^s_{ijkt}	Number of times edge $(i,j) \in \mathcal{E}_{wc}$ is traveled by vehicle $k \in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w$ in period t and in scenario s.

B MATHEMATICAL FORMULATION FOR THE DETERMINISTIC 2EPRP

In this section, we present the mathematical formulation for the deterministic two-echelon PRP, where the demand is assumed to be known in advance. As a result, we no longer consider scenarios to represent possible customer demands, and the scenario index s is no longer needed in the model. All parameters and decision variables are similar to those introduced in Table A1 and Table A2, except that the scenario index is removed. In addition, for the deterministic case, shortages are no longer allowed, and the variable l^s_{it} is therefore excluded from the formulation. The objective function minimizes the total cost of production, storage, and transportation across all nodes and both echelons. The UB on deliveries to customers is also revised and should be replaced by $\mathcal{M}''_{it} = \min\left\{\mathcal{Q}^w, L_i, \sum_{l=t}^{T} d_{il}\right\} \quad \forall i \in \mathcal{N}_c, t \in \mathcal{T}$. The full mathematical formulation for the deterministic 2EPRP is given below.

$$\text{Min} \sum_{t\in\mathcal{T}} \left(F y_t + u p_t + h_0 I_{0t} + \sum_{r\in\mathcal{R}} \tilde{c}_r o_{rt} + \sum_{w\in\mathcal{N}_w} I_{wt} + \sum_{w\in\mathcal{N}_w} \sum_{k\in\mathcal{K}_w} \sum_{(i,j)\in\mathcal{E}_{wc}} c_{ij} x_{ijkt} + \sum_{i\in\mathcal{N}_c} h_i I_{it} \right) \tag{B1}$$

s.t.

$$p_t \le C\, y_t \qquad \forall t \in \mathcal{T} \tag{B2}$$

$$I_{0t} = I_{0,t-1} + p_t - \sum_{w\in\mathcal{N}_w} \sum_{r\in\mathcal{R}} q_{wrt} \qquad \forall t \in \mathcal{T} \tag{B3}$$

$$I_{0t} \le L_0 \qquad \forall t \in \mathcal{T} \tag{B4}$$

$$I_{wt} = I_{w,t-1} + \sum_{r\in\mathcal{R}} q_{wrt} - \sum_{i\in\mathcal{N}_c} \sum_{k\in\mathcal{K}_w} w_{ikt} \qquad \forall w \in \mathcal{N}_w, t \in \mathcal{T} \tag{B5}$$

$$I_{wt} \le L_w \qquad \forall w \in \mathcal{N}_w, t \in \mathcal{T} \tag{B6}$$

$$\sum_{r\in\mathcal{R}} a_{wr}\, o_{rt} \le 1 \qquad \forall w \in \mathcal{N}_w, t \in \mathcal{T} \tag{B7}$$

$$\sum_{r\in\mathcal{R}} o_{rt} \le K_p \qquad \forall t \in \mathcal{T} \tag{B8}$$

$$q_{wrt} \le \mathcal{M}_w\, a_{wr} \qquad \forall w \in \mathcal{N}_w, r \in \mathcal{R}, t \in \mathcal{T} \tag{B9}$$

$$\sum_{w\in\mathcal{N}_w} q_{wrt} \le \mathcal{Q}^p\, o_{rt} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \tag{B10}$$

$$I_{it} = I_{i,t-1} + \sum_{k\in\bigcup_{w\in\mathcal{N}_w}\mathcal{K}_w} w_{ikt} - d_{it} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \tag{B11}$$

$$I_{it} + d_{it} \leq L_i \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \quad \text{(B12)}$$

$$w_{ikt} \leq \mathcal{M}''_{it}\, z_{ikt} \qquad \forall i \in \mathcal{N}_c, k \in \bigcup_{w \in \mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T} \quad \text{(B13)}$$

$$\sum_{i \in \mathcal{N}_c} w_{ikt} \leq Q^w\, z_{wkt} \qquad \forall w \in \mathcal{N}_w, k \in \mathcal{K}_w, t \in \mathcal{T} \quad \text{(B14)}$$

$$\sum_{k \in \bigcup_{w \in \mathcal{N}_w} \mathcal{K}_w} z_{ikt} \leq 1 \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \quad \text{(B15)}$$

$$\sum_{(j,j') \in \tilde{\epsilon}_{wc}(\{i\})} x_{jj'kt} = 2\, z_{ikt} \qquad \forall i \in \mathcal{N}_w \cup \mathcal{N}_c, k \in \bigcup_{w \in \mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T} \quad \text{(B16)}$$

$$\sum_{(i,j) \in \tilde{\mathcal{E}}_{wc}(\eta)} x_{ijkt} \leq \sum_{i \in \eta} z_{ikt} - z_{ekt} \qquad \forall \eta \subseteq \mathcal{N}_c, |\eta| \geq 2, e \in \eta, k \in \bigcup_{w \in \mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T} \quad \text{(B17)}$$

$$y_t \in \{0,1\} \qquad \forall t \in \mathcal{T} \quad \text{(B18)}$$

$$p_t \geq 0 \qquad \forall t \in \mathcal{T} \quad \text{(B19)}$$

$$q_{wrt} \geq 0 \qquad \forall w \in \mathcal{N}_w, r \in \mathcal{R}, t \in \mathcal{T} \quad \text{(B20)}$$

$$I_{0t} \geq 0 \qquad \forall t \in \mathcal{T} \quad \text{(B21)}$$

$$o_{rt} \in \{0,1\} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \quad \text{(B22)}$$

$$I_{it} \geq 0 \qquad \forall i \in \mathcal{N}_w \cup \mathcal{N}_c, t \in \mathcal{T} \quad \text{(B23)}$$

$$w_{ikt} \geq 0 \qquad \forall i \in \mathcal{N}_c, k \in \bigcup_{w \in \mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T} \quad \text{(B24)}$$

$$z_{wkt} \in \{0,1\} \qquad \forall w \in \mathcal{N}_w, k \in \mathcal{K}_w, t \in \mathcal{T} \quad \text{(B25)}$$

$$z_{ikt} \in \{0,1\} \qquad \forall i \in \mathcal{N}_c, k \in \bigcup_{w \in \mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T} \quad \text{(B26)}$$

$$x_{ijkt} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{E}_{wc}, i \notin \mathcal{N}_w, k \in \bigcup_{w \in \mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T} \quad \text{(B27)}$$

$$x_{ijkt} \in \{0,1,2\} \qquad \forall (i,j) \in \mathcal{E}_{wc}, i \in \mathcal{N}_w, k \in \mathcal{K}_w, t \in \mathcal{T}. \quad \text{(B28)}$$

C MATHEMATICAL FORMULATION FOR THE DETERMINISTIC 2EPRP WITH CROSS-DOCKING SATELLITES

In this section, we present the mathematical formulation for the deterministic two-echelon PRP with cross-docking satellites. Similar to the standard deterministic variant, the demand is assumed to be known in advance. All parameters and decision variables are consistent with those in the 2EPRP. However, inventory holding is no longer allowed at the satellites, which are used solely for transferring goods from the first-echelon vehicles to the second-echelon vehicles for delivery to customers. As mentioned earlier, we use the same notation for satellites to maintain consistency; thus, satellites are denoted by $w \in \mathcal{N}_w$. We assume a unit handling cost $\tilde{h}_w \ \forall w \in \mathcal{N}_w$, which is calculated as $\tilde{h}_w = \theta h_0$. The full mathematical formulation for the deterministic 2EPRP with cross-docking satellites is as follows.

$$\text{Min} \sum_{t \in \mathcal{T}} \left(F y_t + u p_t + h_0 I_{0t} + \sum_{r \in \mathcal{R}} \tilde{c}_r o_{rt} + \sum_{w \in \mathcal{N}_w} \tilde{h}_w \sum_{i \in \mathcal{N}_c} \sum_{k \in \mathcal{K}_w} w_{ikt} + \sum_{w \in \mathcal{N}_w} \sum_{k \in \mathcal{K}_w} \sum_{(i,j) \in \mathcal{E}_{wc}} c_{ij} x_{ijkt} + \sum_{i \in \mathcal{N}_c} h_i I_{it} \right) \quad \text{(C29)}$$

s.t.

$$p_t \leq C\, y_t \qquad \forall t \in \mathcal{T} \quad \text{(C30)}$$

$$I_{0t} = I_{0,t-1} + p_t - \sum_{w \in \mathcal{N}_w} \sum_{r \in \mathcal{R}} q_{wrt} \qquad \forall t \in \mathcal{T} \quad \text{(C31)}$$

$$I_{0t} \leq L_0 \qquad \forall t \in \mathcal{T} \quad \text{(C32)}$$

$$\sum_{r \in \mathcal{R}} q_{wrt} = \sum_{i \in \mathcal{N}_c} \sum_{k \in \mathcal{K}_w} w_{ikt} \qquad \forall w \in \mathcal{N}_w, t \in \mathcal{T} \quad \text{(C33)}$$

$$\sum_{r\in\mathcal{R}} a_{wr}\, o_{rt} \le 1 \qquad \forall w \in \mathcal{N}_w, t \in \mathcal{T} \quad \text{(C34)}$$

$$\sum_{r\in\mathcal{R}} o_{rt} \le K_p \qquad \forall t \in \mathcal{T} \quad \text{(C35)}$$

$$q_{wrt} \le Q^p a_{wr} \qquad \forall w \in \mathcal{N}_w, r \in \mathcal{R}, t \in \mathcal{T} \quad \text{(C36)}$$

$$\sum_{w\in\mathcal{N}_w} q_{wrt} \le Q^p\, o_{rt} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \quad \text{(C37)}$$

$$I_{it} = I_{i,t-1} + \sum_{k\in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w} w_{ikt} - d_{it} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \quad \text{(C38)}$$

$$I_{it} + d_{it} \le L_i \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \quad \text{(C39)}$$

$$w_{ikt} \le \mathcal{M}''_{it}\, z_{ikt} \qquad \forall i \in \mathcal{N}_c, k \in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T} \quad \text{(C40)}$$

$$\sum_{i\in\mathcal{N}_c} w_{ikt} \le Q^w\, z_{wkt} \qquad \forall w \in \mathcal{N}_w, k \in \mathcal{K}_w, t \in \mathcal{T} \quad \text{(C41)}$$

$$\sum_{k\in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w} z_{ikt} \le 1 \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \quad \text{(C42)}$$

$$\sum_{(j,j')\in\tilde{\varepsilon}_{wc}(\{i\})} x_{jj'kt} = 2\, z_{ikt} \qquad \forall i \in \mathcal{N}_w \cup \mathcal{N}_c, k \in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T} \quad \text{(C43)}$$

$$\sum_{(i,j)\in\tilde{\mathcal{E}}_{wc}(\eta)} x_{ijkt} \le \sum_{i\in\eta} z_{ikt} - z_{ekt} \qquad \forall \eta \subseteq \mathcal{N}_c, |\eta| \ge 2, e \in \eta, k \in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T} \quad \text{(C44)}$$

$$y_t \in \{0,1\} \qquad \forall t \in \mathcal{T} \quad \text{(C45)}$$

$$p_t \ge 0 \qquad \forall t \in \mathcal{T} \quad \text{(C46)}$$

$$q_{wrt} \ge 0 \qquad \forall w \in \mathcal{N}_w, r \in \mathcal{R}, t \in \mathcal{T} \quad \text{(C47)}$$

$$I_{0t} \ge 0 \qquad \forall t \in \mathcal{T} \quad \text{(C48)}$$

$$o_{rt} \in \{0,1\} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \quad \text{(C49)}$$

$$I_{it} \ge 0 \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \quad \text{(C50)}$$

$$w_{ikt} \ge 0 \qquad \forall i \in \mathcal{N}_c, k \in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T} \quad \text{(C51)}$$

$$z_{wkt} \in \{0,1\} \qquad \forall w \in \mathcal{N}_w, k \in \mathcal{K}_w, t \in \mathcal{T} \quad \text{(C52)}$$

$$z_{ikt} \in \{0,1\} \qquad \forall i \in \mathcal{N}_c, k \in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T} \quad \text{(C53)}$$

$$x_{ijkt} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{E}_{wc}, i \notin \mathcal{N}_w, k \in \bigcup_{w\in\mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T} \quad \text{(C54)}$$

$$x_{ijkt} \in \{0,1,2\} \qquad \forall (i,j) \in \mathcal{E}_{wc}, i \in \mathcal{N}_w, k \in \mathcal{K}_w, t \in \mathcal{T}. \quad \text{(C55)}$$

D THE BRANCH-AND-CUT ALGORITHM

In this section, we provide the details of the branch-and-cut (BC) algorithm used to solve both the stochastic and deterministic versions of the problem. Since the BC structure is similar for both cases, we present the algorithm for the stochastic version. With minor modifications, the same approach can be applied to the deterministic version. The problem is solved with CPLEX, and a callback function is employed to generate subtour elimination constraints (SECs).

It is important to emphasize that, as mentioned earlier, the BC algorithm is primarily used to evaluate the quality of solutions obtained by the hybrid heuristic algorithm (HHA), rather than serving as the main solution method. In all cases, the solution produced by the HHA is provided to CPLEX as an initial feasible solution.

As the first-echelon routes are explicitly defined in the model, there is no need to generate SECs for that part of the problem. For each subset of warehouses, a traveling salesman problem (TSP) is solved to determine the optimal visiting sequence and corresponding cost. All possible combinations of warehouse visits are enumerated and added to the model. Since the number of warehouses is assumed to be small, the total number of possible combinations does not exceed $2^W - 1$ (for instances with

W warehouses), which can be processed in a few seconds, since $W \leq 5$. The Concorde algorithm is used to solve these TSP problems efficiently.

In contrast, the number of SECs required in the second echelon grows exponentially with the number of customers. For the stochastic case, these constraints must be generated for each scenario, which adds significant complexity. Therefore, constraints 17 are relaxed at the beginning and added to the model only when violated. To identify violated SECs, we solve a separation problem formulated as a minimum s–t cut for each scenario. We use the algorithm provided by the Concorde library, which is known for its efficiency. Once violations are detected, the corresponding constraints are added to exclude the infeasible solution, and the BC algorithm continues from there. SECs are checked only at the root node and for integer solutions, as this has proven to be more efficient in practice than checking at every node in the branch-and-bound tree.

For solving the problem, we set the CPLEX MIP emphasis parameter to prioritize optimality, and allocate 32 GB of memory and 10 threads.

In addition to SECs, we add several classes of valid inequalities to improve the efficiency of the BC algorithm. The first set consists of logical inequalities that ensure a vehicle $k \in \mathcal{K}_w$ can only visit customer $i \in \mathcal{N}_c$ if it is dispatched from warehouse w:

$$z^s_{ikt} \leq z^s_{wkt} \qquad \forall i \in \mathcal{N}_c, w \in \mathcal{N}_w, k \in \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S}. \quad \text{(D56)}$$

The second set of logical inequalities imposes that an edge $(i,j) \in \mathcal{E}_{wc}$ can only be used if both nodes i and j are visited:

$$x^s_{ijkt} \leq z^s_{ikt} \qquad \forall i,j \in \mathcal{N}_c \cup \mathcal{N}_w, k \in \bigcup_{w \in \mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S}. \quad \text{(D57)}$$

$$x^s_{ijkt} \leq z^s_{jkt} \qquad \forall i,j \in \mathcal{N}_c \cup \mathcal{N}_w, k \in \bigcup_{w \in \mathcal{N}_w} \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S}. \quad \text{(D58)}$$

The third set includes symmetry-breaking constraints that prevent a vehicle with index k from being dispatched unless the vehicle with index $k-1$ is already dispatched from the same warehouse:

$$z^s_{wkt} \leq z^s_{w,k-1,t} \qquad \forall w \in \mathcal{N}_w, k \in \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S}. \quad \text{(D59)}$$

The last set of valid inequalities are the lexicographic constraints, which further help break symmetry by imposing an ordering on customer visits across vehicles. These constraints ensure that the sequence of customers visited by vehicle k is lexicographically smaller than or equal to that of vehicle $k-1$:

$$\sum_{i=1}^{j} 2^{j-i} z^s_{ikt} \leq \sum_{i=1}^{j} 2^{j-i} z^s_{i,k-1,t} \qquad \forall j \in \mathcal{N}_c, w \in \mathcal{N}_w, k \in \mathcal{K}_w, t \in \mathcal{T}, s \in \mathcal{S}. \quad \text{(D60)}$$

Lexicographic constraints are especially helpful when multiple vehicles are assigned to the same warehouse, as they eliminate redundant solutions that differ only in the indexing of vehicles. All of these valid inequalities are also adjusted and added to the deterministic problem.