A Progressive Hedging-based Matheuristic for the Stochastic Production Routing Problem with Adaptive Routing

Ali Kermani, Jean-François Cordeau, Raf Jans

[a] *HEC Montréal, CIRRELT, and GERAD, 3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7, QC, Canada*

Abstract

The production routing problem (PRP) arises in the context where a manufacturing facility manages its production schedule, the delivery of goods to customers by a fleet of vehicles, and the inventory levels both at the plant and at the customers. The presence of uncertainty often complicates the planning process. In particular, production and distribution costs may significantly increase if demand uncertainty is ignored in the planning phase. Nevertheless, only a few studies have considered demand uncertainty in the PRP. In this article, we propose a two-stage stochastic programming approach for a one-to-many PRP with a single product and demand uncertainty. Unlike previous studies in the literature, we consider the case where routing decisions are made in the second stage after customer demands become known. This offers more flexibility, which can decrease transportation costs by preventing unnecessary customer visits. In addition to the static-dynamic case, in which setup decisions are made first and production quantities are decided in the second stage, we also consider the static-static setting where both sets of decisions must be made prior to the demand realization. A progressive hedging algorithm combined with a matheuristic is developed to solve the problem. The role of the progressive hedging algorithm is to decompose the stochastic problem into more tractable scenario-specific subproblems and lead the first-stage variables toward convergence by modifying their Lagrangean multipliers. However, solving the subproblems remains challenging since they include routing decisions, and we thus propose a matheuristic to exploit the structural characteristics of the subproblems. First, a Traveling Salesman Problem (TSP) is solved to find the optimal tour for all customers regardless of demand and capacity. Utilizing the sequence obtained from the TSP, we then solve a restricted PRP while taking into account the other constraints of the original problem. Finally, for each period and scenario, a vehicle routing problem is solved to improve the quality of the solutions. Computational experiments are conducted to analyze the algorithm's efficiency and to assess the benefits that can be achieved by handling routing in the second stage.

Keywords: Production Routing, Stochastic Programming, Progressive Hedging, Matheuristic, Mixed-integer Programming

1. Introduction.

The production routing problem (PRP) aims to make production, inventory, and routing decisions simultaneously by integrating the lot-sizing and vehicle routing problems. Adulyasak et al. (2015b) provide an overview of the PRP literature, including both formulations and solution methods. The majority of the formulations discussed in this survey focus on the deterministic PRP, but in recent years, the stochastic version of the problem has attracted more attention. In particular, demand uncertainty is known to have a great impact on both production planning and vehicle routing. However, one should keep in mind that the PRP is an NP-hard problem and that considering stochasticity further increases the difficulty of the problem. Adulyasak et al. (2015a) introduced a two-stage and a multi-stage model for the stochastic production routing problem (SPRP) with demand uncertainty. In both problems, setup and routing decisions are made in the first stage, while the remaining decisions, including production quantities, delivery quantities and inventory levels, belong to the second stage. When solving the two-stage formulation, the demand for the entire planning horizon is assumed to become known at the beginning of the second stage, whereas in the multi-stage problem, the demand of each period is realized at the beginning of that period. Although the multi-stage setting offers more flexibility, the two-stage model can be solved more efficiently. It can also be used in a rollout algorithm to provide a heuristic solution to the multi-stage problem.

Considering the routing decisions among the first-stage decisions may lead to unnecessary customer visits and higher costs. In this paper, we introduce a two-stage stochastic programming formulation for the PRP with adaptive routing, i.e., the routing becomes a second-stage decision, which provides more flexibility in response to the observed demand. We assume a plant that produces and distributes a single product to multiple customers in a finite horizon using a fleet of homogeneous vehicles. If the demand of a customer cannot be fully satisfied from the available inventory, a cost per unit of unmet demand is imposed to represent the outsourcing costs.

We provide below a simple example with two periods, three customers, and two scenarios to illustrate the possible benefits of considering routing decisions in the second stage. In this example, we assume that there are no holding or production costs, that the production and inventory capacities are large, and that there are two vehicles with a capacity of 50. The two scenarios have the same probability $p_{s_1} = p_{s_2} = 0.5$. The rest of the parameters is provided in Table 1, where the first two rows are the x and y coordinates of each node and the next four rows are the demands of the customers. The transportation costs are calculated using Euclidean distances. Note that the scenario and period pairs are denoted by (s, t). The production plant is denoted by 0 and customers are indexed from 1 to 4. Figure 1 displays the optimal solution to this example under both routing strategies. The numbers written next to the nodes are the delivered quantities. If we solve this problem with first-stage routing, the optimal expected cost is 1854.2. However, by considering adaptive routing, the optimal expected cost is 1594.7, demonstrating how this flexibility can lower the cost.

To the best of our knowledge, this is the first study to consider adaptive routing in the SPRP, which

Table 1: Parameters of the example

Node No.		0	1	2	3	4
x coordinate		143	89	76	285	401
y coordinate		99	159	314	63	325
demand	(1,1)	-	4	14	21	0
	(1,2)	-	9	17	31	6
	(2,1)	-	24	12	3	13
	(2,2)	-	29	15	10	18

is the first contribution of our study. Our second contribution is to develop a progressive hedging (PH) approach combined with a three-phase matheuristic algorithm. Through the PH algorithm, the problem is decomposed into subproblems, and the first-stage variables are guided toward convergence by changing their corresponding costs in the objective function. The first phase of the process involves solving a TSP to find an a priori tour. Production setup, quantities, and visit decisions are determined by restricting subproblems to the route defined in the first phase of the algorithm and by solving them until a consensus is reached in the first-stage variables or a stopping criterion is met. An aggregation procedure is applied at this point to construct the solution, and in the final stage of the solution process, a capacitated vehicle routing problem (CVRP) is solved for each period and every scenario to further improve the routing decisions. The third contribution of this study is the consideration of the static-static case that has been adopted from the lot sizing literature, where the production quantities are also considered in the first stage of the problem. Finally, computational experiments are conducted in order to demonstrate the effectiveness of the proposed algorithm and the improvements that can be obtained by determining the routing decisions in the second stage.

The remainder of the paper is organized as follows. A literature review of the PRP and SPRP is provided in Section 2 to show how the current work relates to previous studies. The mathematical formulation and notations are provided in Section 3 while Section 4 describes the PH-based three-phase matheuristic algorithm. Computational experiments are presented in Section 5. Finally, we discuss our findings and draw conclusions in Section 6.

2. Literature Review.

First introduced by Chandra (1993), the PRP aims to improve supply chain efficiency and reduce costs by increasing collaboration between partners and incorporating production and distribution decisions into a single problem. The classic PRP considers a production plant, multiple customers, and a fleet of vehicles to transport the goods to customers. However, several variants of the problem have also been investigated to address specific real-world situations. In order to find high-quality solutions for the PRP in a reasonable amount of time, many heuristic and metaheuristic algorithms have been developed over the years. One

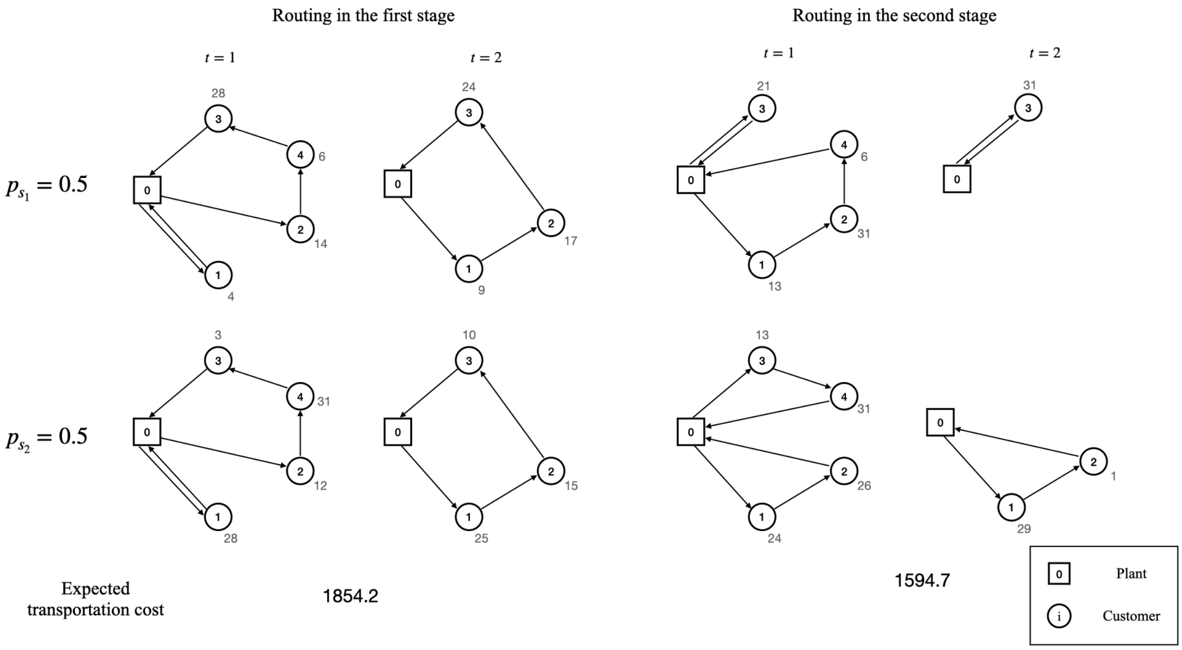


Figure 1: An example of considering routing in the first stage vs. the second stage

heuristic approach that is commonly employed is to take advantage of the problem's structure and decompose it into less complex subproblems.

Boudia et al. (2008) introduce the "uncoupled" and the "coupled" heuristics to solve the problem in two phases. Bard and Nananukul (2009) provide a two-phase reactive tabu search (TS) algorithm with path-relinking. Archetti et al. (2011) compare the order-up-to-level (OU) and maximum level (ML) replenishment policies and propose a hybrid heuristic to solve the production and routing subproblems sequentially. Armentano et al. (2011) deal with a multi-product, multi-vehicle PRP and present two variants of TS as the solution algorithm. Adulyasak et al. (2014b) decompose the main problem into less complex subproblems and employ an ALNS-based heuristic. Absi et al. (2015) propose a two-phase heuristic that solves a two-level lot sizing problem with approximate routing costs and uses a back-and-forth iteration between the two phases to improve the solutions. Solyalı and Süral (2017) propose a multi-phase matheuristic algorithm that first solves a restricted PRP using a determined set of routes and improves the production and routing decisions in the following phases. Russell (2017) develops two matheuristic algorithms. The first one utilizes the set partitioning reformulation of the problem followed by a TS algorithm to improve the solutions. In the second algorithm, by assigning artificial demands to customers, they are divided into relatively similar-sized clusters, and approximate routes are constructed using the idea of seed routes.

Miranda et al. (2018) extend the decomposition heuristic presented by Absi et al. (2015) to solve a rich PRP derived from a real-world problem at a Brazilian furniture manufacturer. A heterogeneous fleet, multiple products, and routes that can extend over multiple days are all taken into account in their formulation. Considering the multi-product PRP with the possibility of outsourcing, Li et al. (2019) solve the problem using a three-level heuristic algorithm. Avci and Topaloglu Yildiz (2020) investigate the PRP with transshipment between retailers or between a retailer and the production plant. They propose a multi-phase matheuristic algorithm to solve the problem for small and medium size instances. A two-echelon PRP for the petrochemical industry is introduced by Schenekemberg et al. (2021) such that both pickups and deliveries may occur at suppliers, plants, and customers. A local search is combined with a branch-and-cut algorithm to solve the problem. Gruson et al. (2023) also discuss a two-echelon extension of the PRP, in which there are direct shipments from the plant to several warehouses, and routes have to be established for the distribution from the warehouses to the customers. They consider the case of split deliveries and split demand. Manousakis et al. (2022) propose a two-commodity flow formulation for the PRP under the ML policy. A matheuristic algorithm is proposed that first solves the relaxed PRP considering approximate routing costs, and the routing problem is then enhanced using a Greedy Randomized Adaptive Search Procedure (GRASP). Finally, a local search algorithm that explores both feasible and infeasible search spaces is utilized to improve the quality of the solutions. In their study, Neves-Moreira et al. (2019) present a three-phase matheuristic algorithm to solve a multi-product PRP taking into account the perishability of the products and time windows. A study by Alvarez et al. (2022) addresses PRPs with perishable products and transshipment in which the profit decreases gradually over time as the products reach their shelf life. In order to solve the problem, they propose a hybrid heuristic algorithm based on iterative local search (ILS). Chitsaz et al. (2019) study the assembly routing problem (ARP), where there is an assembly unit that requires components from different suppliers to produce a final product. A unified decomposition matheuristic is developed that produces solutions to the ARP and related problems in three phases.

A few studies have introduced exact algorithms to solve small to medium size PRPs to optimality. Bard and Nananukul (2010) propose a branch-and-price algorithm to solve the PRP with a homogeneous fleet. Adulyasak et al. (2014a) propose four formulations for the PRP which differ with respect to the replenishment policy considered and whether or not they include a vehicle index. They develop a branch-and-cut algorithm to solve the problems utilizing several valid inequalities to strengthen the problem formulation. Qiu et al. (2019) study the PRP with perishable products and propose a mixed integer programming (MIP) formulation based on the Miller–Tucker–Zemlin (MTZ) subtour elimination constraints. A branch-and-cut algorithm using valid inequalities is applied to solve this problem. Chitsaz et al. (2020) extend the ARP to consider products that can be made by multiple suppliers, and they solve the problem using a branch-and-cut algorithm. In a study conducted by Schenekemberg et al. (2023), they introduce a novel three-front parallel algorithm designed to exploit the two-index and the three-index formulations using the Branch-and-Cut (BC) algorithm and a local search matheuristic approach, independently. The core concept of this algorithm involves the simultaneous solution of these three distinct fronts, all within an integrated framework that facilitates information sharing among these fronts. The experimental results demonstrate the remarkable performance of their algorithm. Specifically, it has exhibited the ability to identify previously undiscovered optimal solutions, as well as achieving the best-known solutions for a significant subset of benchmark instances.

Kumar et al. (2016) present a bi-objective formulation for the PRP where the second objective function evaluates the carbon emissions by considering the fuel consumption based on the vehicles' total travel time. Qiu et al. (2017) propose a PRP formulation for considering Greenhouse Gases (GHG) and specifically carbon emission costs under the carbon cap-and-trade policy and solve the problem using a branch-and-price algorithm. Qiu et al. (2018) propose a model with simultaneous pickup and delivery to incorporate remanufacturing into the PRP and solve it using a branch-and-price algorithm. This problem involves products being returned to remanufacturing facilities different from the production plant.

As mentioned earlier, despite the significance of the SPRP, only a few studies have addressed this problem due to its complexity. Adulyasak et al. (2015a) introduce the stochastic PRP that considers the routing and production setup decision as the first-stage and the production and delivery quantities as the second-stage decisions. In their study, a Benders decomposition is combined with a branch-and-cut algorithm to solve the problem. In the two-stage case, the Benders algorithm is strengthened by lower bound lifting inequalities, scenario group cuts, and Pareto-optimal cuts. Mousavi et al. (2022) propose a two-stage stochastic programming formulation for the PRP with a single perishable product and demand uncertainty. In this study, a five-stage matheuristic solution approach based on the algorithm proposed by Solyalı and Süral (2017) is developed to solve the problem. Agra et al. (2018) propose a two-stage stochastic production-inventory routing problem in which production periods, quantities and routing decisions are the first-stage variables. The second-stage variables are the delivery quantities, inventory levels, and backlogging quantities. Several sets of valid inequalities are introduced and customized to the problem, and two solution algorithms based on the sample average approximation (SAA) approach are presented.

Zhang et al. (2018) study the PRP problem with recycling and remanufacturing in which both pickup and delivery can occur at a retailer. This research proposes a deterministic and a two-stage stochastic optimization problem considering demand uncertainty. In the stochastic problem, production quantities in both manufacturing and remanufacturing firms, as well as the routing decisions, are considered in the first stage, while the inventory and the number of worn-out items that are carried back to the remanufacturing firms are the second-stage variables. Shuang et al. (2019) study the PRP problem with recycling and remanufacturing with carbon taxes as well as the carbon cap-and-trade policy in a similar context.

In all prior SPRP studies, routing decisions are made in the first stage (SPRP-FSR) and unnecessary visits may occur in the second stage. In order to address this issue, we propose an adaptive routing strategy to reduce routing costs. Adaptive routing is especially relevant when delivery costs are high and more flexible decisions are required based on the actual demand. In each realized scenario, routing decisions are part of the solution that is influenced by both inventory and demand. Routing decisions and customer visits that are scenario-specific increase the complexity of the problem. Thus, we develop an algorithm to obtain high-quality solutions to the problem. Moreover, the static-static strategy has been neglected in the SPRP studies while it constitutes a significant part of the lot sizing problem (LSP) literature. By modifying the formulation as well as adapting our heuristic algorithm, this paper intends to fill this gap in the existing literature.

3. Problem Formulation.

This section presents a formulation for the two-stage stochastic PRP with demand uncertainty and adaptive routing for a single product that must be produced and delivered to N customers, where $\mathcal{N} = \{0, ..., N\}$ is the set of all nodes including $\{0\}$, which denotes the production plant, and $\mathcal{N}_c$, which represents the set of customers. Let $G = (\mathcal{N}, \mathcal{E})$ be a complete and undirected graph where $\mathcal{E} = \{(i, j) : i, j \in \mathcal{N}, i < j\}$ represents the set of edges that connect each pair of nodes in $\mathcal{N}$. Homogeneous vehicles with capacity Q deliver the product to the customers, and they must start and end their routes at the production plant in each period. The set of all vehicles is defined by $\mathcal{K} = \{1, ..., K\}$. We consider a discrete and finite planning horizon of T periods, and the set of periods is denoted by $\mathcal{T} = \{1, ..., T\}$. In order to model demand uncertainty, we define a finite set of scenarios $\phi = \{1, ..., S\}$, each of which may occur with probability $\pi_\varphi > 0,\ \forall \varphi \in \phi$ and $\sum_{\varphi \in \phi} \pi_\varphi = 1$. The demand of each customer is assumed to be an independent random variable with a known distribution, and the realized demand of customer $i \in \mathcal{N}_c$ in period $\tau \in \mathcal{T}$ under scenario $\varphi \in \phi$ is denoted by $d^\varphi_{i\tau}$.

In this study, we adopt two of the strategies proposed by Bookbinder and Tan (1988) for the stochastic LSP: The static-static strategy in which both setup and production quantity decisions are made prior to the demand realization and the more flexible static-dynamic strategy where production quantities are determined in the second stage while setup decisions are made in the first stage. The visiting, delivery quantity, outsourcing, and routing decisions are made in the second stage in both variants of the PRP.

The binary variable y_τ takes value 1 if production takes place in period $\tau \in \mathcal{T}$, with a fixed setup cost of F, and 0 otherwise. In addition, a per unit production cost u applies to the production quantity, which is represented by variable p^φ_τ (p_τ for the static-static case). This production quantity is bounded by the production capacity C. The integer variable $x^\varphi_{ij\kappa\tau}$ is a recourse variable that denotes how many times edge $(i, j) \in \mathcal{E}$ is visited by vehicle $\kappa \in \mathcal{K}$ in period $\tau \in \mathcal{T}$ under scenario $\varphi \in \phi$. Traversing edge (i, j) by a vehicle induces a transportation cost of c_{ij}. A unit holding cost of h_i applies to the inventory at the end of each period for all nodes $i \in \mathcal{N}$. Accordingly, we define variable $I^\varphi_{i\tau}$ as the inventory of node $i \in \mathcal{N}$ at the end of each period τ and for each scenario φ. An initial inventory of I_{i0} exists at each node $i \in \mathcal{N}$ at the beginning of the planning horizon. If customer $i \in \mathcal{N}_c$ faces a stockout, outsourcing to a third-party supplier can take place with cost β_i and $o^\varphi_{i\tau}$ is the recourse variable that represents the amount of demand at this customer which is satisfied using this outsourcing option under scenario φ. We define binary variable $z^\varphi_{i\kappa\tau}$ to take value 1 if node $i \in \mathcal{N}$ is visited by vehicle $\kappa \in \mathcal{K}$ in period $\tau \in \mathcal{T}$ under scenario $\varphi \in \phi$, and 0 otherwise. In addition, variable $q^{\varphi}_{i\kappa\tau}$ denotes the amount delivered to customer $i \in \mathcal{N}_c$ by vehicle $\kappa \in \mathcal{K}$ in period τ under scenario φ. Table 2 provides a summary of all sets, parameters, and decision variables used in the mathematical formulation. The SPRP with adaptive routing (SPRP-AR) and assuming the static-dynamic strategy ($SPRP-AR_{SD}$) can now be formulated as the following MIP:

$$(SPRP-AR_{SD}) \quad \min \quad \sum_{\tau\in\mathcal{T}} \Big(Fy_\tau + \sum_{\varphi\in\phi} \pi_\varphi \Big(up^{\varphi}_{\tau} + \sum_{(i,j)\in\mathcal{E}} \sum_{\kappa\in\mathcal{K}} c_{ij} x^{\varphi}_{ij\kappa\tau} + \sum_{i\in\mathcal{N}} h_i I^{\varphi}_{i\tau} + \sum_{i\in\mathcal{N}_c} \beta_i o^{\varphi}_{i\tau} \Big) \Big) \tag{1}$$

s.t.

$$p^{\varphi}_{\tau} \leq \mathcal{M}^{\varphi}_{\tau} y_\tau \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \tag{2}$$

$$I^{\varphi}_{0\tau} = I^{\varphi}_{0,\tau-1} + p^{\varphi}_{\tau} - \sum_{i\in\mathcal{N}_c} \sum_{\kappa\in\mathcal{K}} q^{\varphi}_{i\kappa\tau} \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \tag{3}$$

$$I^{\varphi}_{i\tau} = I^{\varphi}_{i,\tau-1} + \sum_{\kappa\in\mathcal{K}} q^{\varphi}_{i\kappa\tau} + o^{\varphi}_{i\tau} - d^{\varphi}_{i\tau} \qquad \forall i \in \mathcal{N}_c, \tau \in \mathcal{T}, \varphi \in \phi \tag{4}$$

$$I^{\varphi}_{0\tau} \leq L_0 \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \tag{5}$$

$$I^{\varphi}_{i\tau} + d^{\varphi}_{i\tau} \leq L_i \qquad \forall i \in \mathcal{N}_c, \tau \in \mathcal{T}, \varphi \in \phi \tag{6}$$

$$\sum_{i\in\mathcal{N}_c} q^{\varphi}_{i\kappa\tau} \leq Q z^{\varphi}_{0\kappa\tau} \qquad \forall \kappa \in \mathcal{K}, \tau \in \mathcal{T}, \varphi \in \phi \tag{7}$$

$$\sum_{\kappa\in\mathcal{K}} z^{\varphi}_{i\kappa\tau} \leq 1 \qquad \forall i \in \mathcal{N}_c, \tau \in \mathcal{T}, \varphi \in \phi \tag{8}$$

$$q^{\varphi}_{i\kappa\tau} \leq \mathcal{W}^{\varphi}_{i\tau} z^{\varphi}_{i\kappa\tau} \qquad \forall i \in \mathcal{N}_c, \kappa \in \mathcal{K}, \tau \in \mathcal{T}, \varphi \in \phi \tag{9}$$

$$\sum_{(j,j')\in\epsilon(\{i\})} x^{\varphi}_{jj'\kappa\tau} = 2 z^{\varphi}_{i\kappa\tau} \qquad \forall i \in \mathcal{N}, \kappa \in \mathcal{K}, \tau \in \mathcal{T}, \varphi \in \phi \tag{10}$$

$$\sum_{(i,j)\in\mathcal{E}(\eta)} x^{\varphi}_{ij\kappa\tau} \leq \sum_{i\in\eta} z^{\varphi}_{i\kappa\tau} - z^{\varphi}_{e\kappa\tau} \qquad \forall \eta \subseteq \mathcal{N}_c, |\eta| \geq 2, e \in \eta, \kappa \in \mathcal{K}, \tau \in \mathcal{T}, \varphi \in \phi \tag{11}$$

$$y_\tau \in \{0,1\} \qquad \forall \tau \in \mathcal{T} \tag{12}$$

$$p^{\varphi}_{\tau} \geq 0 \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \tag{13}$$

$$q^{\varphi}_{i\kappa\tau} \geq 0 \qquad \forall i \in \mathcal{N}, \kappa \in \mathcal{K}, \tau \in \mathcal{T}, \varphi \in \phi \tag{14}$$

$$I^{\varphi}_{i\tau} \geq 0 \qquad \forall i \in \mathcal{N}, \tau \in \mathcal{T}, \varphi \in \phi \tag{15}$$

$$o^{\varphi}_{i\tau} \geq 0 \qquad \forall i \in \mathcal{N}, \tau \in \mathcal{T}, \varphi \in \phi \tag{16}$$

$$z^{\varphi}_{i\kappa\tau} \in \{0,1\} \qquad \forall i \in \mathcal{N}, \kappa \in \mathcal{K}, \tau \in \mathcal{T}, \varphi \in \phi \tag{17}$$

$$x^{\varphi}_{ij\kappa\tau} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{E}, i \neq 0, \kappa \in \mathcal{K}, \tau \in \mathcal{T}, \varphi \in \phi \tag{18}$$

$$x^{\varphi}_{0j\kappa\tau} \in \{0,1,2\} \qquad \forall j \in \mathcal{N}_c, \kappa \in \mathcal{K}, \tau \in \mathcal{T}, \varphi \in \phi. \tag{19}$$

The objective function (1) minimizes the setup costs and the expected cost of the second-stage variables including the production, routing, holding, and outsourcing costs. Constraints (2) impose that the setup must take place if production is required in a period. The production quantity in each period is bounded by the minimum of either the production capacity or the sum of all customer demands for the rest of the periods, denoted by $\mathcal{M}_\tau^\varphi = \min\left\{C,\ \sum_{l=\tau}^{T} \sum_{i\in\mathcal{N}_c} d_{il}^\varphi\right\}$. Constraints (3) and (4) are the inventory balance constraints for the plant and customers, respectively. Each node has a maximum storage capacity of L_i. Constraints (5) control the inventory capacity at the plant and constraints (6) impose that the inventory of each customer at the end of the period plus the demand of that period cannot exceed the capacity of that customer. Constraints (7) force the vehicle to leave the plant if it is assigned to deliver products to customers and impose the vehicle capacity. Constraints (8) and (9) indicate that each customer can only be visited by one vehicle in a period (split deliveries are not allowed) and the maximum delivery to a customer must not exceed the smallest value of the vehicle capacity, customer's storage, or the customer's demand for the remaining periods of the planning horizon, which is indicated by $\mathcal{W}_{i\tau}^\varphi = \min\left\{Q,\ L_i,\ \sum_{l=\tau}^{T} d_{il}^\varphi\right\}$.

We define $\mathcal{E}(\eta)$ as a set of edges $(i, j) \in \mathcal{E}$ such that $i, j \in \eta$ and $\eta \subseteq \mathcal{N}$ is a given set of nodes. Moreover, $\epsilon(\{i\})$ is the set of edges incident to node $i \in \mathcal{N}$. Constraints (10) indicate that two edges must be incident to a visited node and constraints (11) are the subtour elimination constraints (SEC). As mentioned earlier, the routing problem depends on the realized scenario, which means that the routing is decided after the demand realization. Therefore, these constraints must be considered for every possible demand realization and all periods. Finally, constraints (12)–(19) define the domain of the decision variables.

We introduce a second formulation, called $SPRP-AR_{SS}$, for the static-static strategy, assuming that production quantities are also determined in the first stage:

$$(SPRP-AR_{SS}) \qquad \min \quad \sum_{\tau\in\mathcal{T}}\left(Fy_\tau + up_\tau + \sum_{\varphi\in\phi}\pi_\varphi\Big(\sum_{(i,j)\in\mathcal{E}}\sum_{\kappa\in\mathcal{K}} c_{ij}x_{ij\kappa\tau}^\varphi + \sum_{i\in\mathcal{N}} h_i I_{i\tau}^\varphi + \sum_{i\in\mathcal{N}_c}\beta_i o_{i\tau}^\varphi\Big)\right) \tag{20}$$

s.t. (4) - (12) and (14) - (19),

$$p_\tau \leq \mathcal{M}_\tau^{(max)} y_\tau \qquad \forall \tau \in \mathcal{T} \tag{21}$$

$$I_{0\tau}^\varphi = I_{0,\tau-1}^\varphi + p_\tau - \sum_{i\in\mathcal{N}_c}\sum_{\kappa\in\mathcal{K}} q_{i\kappa\tau}^\varphi \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \tag{22}$$

$$p_\tau \geq 0 \qquad \forall \tau \in \mathcal{T}. \tag{23}$$

In the above formulation, the objective function (20) minimizes the production costs instead of their expected value since production is no longer a scenario-dependent variable.

In addition, in constraints (21) the production quantity must be decided regardless of the scenario and is bounded by $\mathcal{M}_\tau^{max} = \min\left\{C,\ \max_{\varphi\in\phi}\{\sum_{l=\tau}^{T} \sum_{i\in\mathcal{N}_c} d_{il}^\varphi\}\right\}$. Constraints (22) are the adapted inventory balance

Table 2: Sets, parameters, and decision variables.

Sets:	
$\mathcal{N}$	Set of nodes, $i \in \{0, ..., N\}$, $\{0\}$ is the plant.
$\mathcal{N}_c$	Set of customers, $i \in \{1, ..., N\}$.
$\mathcal{E}$	Set of edges, $\mathcal{E} = \{(i, j) : i, j \in \mathcal{N}, i < j\}$.
$\mathcal{T}$	Set of time periods, $\tau \in \{0, ..., T\}$.
$\mathcal{K}$	Set of vehicles, $\kappa \in \{1, ..., K\}$.
ϕ	Set of scenarios, $\varphi \in \{1, ..., S\}$.
$\mathcal{E}(\eta)$	Subset of edges $(i, j) \in \mathcal{E}$ such that $i, j \in \eta$ and $\eta \subseteq \mathcal{N}$ is a given set of nodes.
$\epsilon(\{i\})$	Subset of edges incident to node $i \in \mathcal{N}$.
Parameters:	
F	Setup cost.
u	Unit production cost.
C	Production capacity.
c_{ij}	Cost of visiting edge $(i, j) \in \mathcal{E}$.
h_i	Per unit holding cost at node $i \in \mathcal{N}$.
Q	Vehicle capacity.
β_i	Unit outsourcing cost at node $i \in \mathcal{N}_c$.
π_φ	Probability of scenario $\varphi \in \phi$.
$d_{i\tau}^{\varphi}$	Demand of node $i \in \mathcal{N}_c$, in period $\tau \in \mathcal{T}$, under scenario $\varphi \in \phi$.
I_{i0}	Initial inventory at node $i \in \mathcal{N}$.
L_i	Storage capacity at node $i \in \mathcal{N}$.
Decision variables:	
y_τ	Setup decision, equal to 1 if setup takes place at period $\tau \in \mathcal{T}$, and 0 otherwise.
p_τ	Production quantity at period $\tau \in \mathcal{T}$ (for $SPRP - AR_{SS}$).
p_τ^{φ}	Production quantity at period $\tau \in \mathcal{T}$ under scenario $\varphi \in \phi$ (for $SPRP - AR_{SD}$).
$x_{ij\kappa\tau}^{\varphi}$	Number of times edge $(i, j) \in \mathcal{E}$ is traversed by vehicle $\kappa \in \mathcal{K}$ in period $\tau \in \mathcal{T}$ under scenario $\varphi \in \phi$.
$I_{i\tau}^{\varphi}$	Inventory of node $i \in \mathcal{N}$ at the end of each period $\tau \in \mathcal{T}$ under scenario $\varphi \in \phi$.
$z_{i\kappa\tau}^{\varphi}$	Visiting decision, equal to 1 if node $i \in \mathcal{N}$ is visited by vehicle $\kappa \in \mathcal{K}$ in period $\tau \in \mathcal{T}$ under scenario $\varphi \in \phi$, and 0 otherwise.
$o_{i\tau}^{\varphi}$	Quantity of outsourced products at node $i \in \mathcal{N}_c$ in period $\tau \in \mathcal{T}$ under scenario $\varphi \in \phi$.
$q_{i\kappa\tau}^{\varphi}$	Amount of product delivered to customer $i \in \mathcal{N}_c$ by vehicle $\kappa \in \mathcal{K}$ in period $\tau \in \mathcal{T}$ under scenario $\varphi \in \phi$.

constraints at the plant to take into account the scenario-independent production quantity.

4. Progressive Hedging-Based Matheuristic Algorithm.

The goal of this section is to introduce a PH algorithm that is combined with a three-phase matheuristic in order to solve the SPRP-AR. The PH algorithm was first proposed by Rockafellar and Wets (1991) as a scenario decomposition approach. This method works by replacing the first-stage variables with scenario-specific variables, leading to a set of independent subproblems representing each scenario, thereby reducing the complexity of the problem. It should be noted that the solution to each subproblem is only optimal in the context of that specific scenario. Accordingly, in order to reach a consensus over all the variables involved in the first stage of the stochastic process and achieve a feasible solution, an augmented Lagrangean strategy is adopted.

A number of studies have investigated the use of the PH approach in combination with other heuristics in order to obtain high-quality solutions to integer stochastic problems. Løkketangen and Woodruff (1996) present a PH-based TS for stochastic binary MIP problems. A study by Haugen et al. (2001) implements a PH algorithm for solving the stochastic LSP. Crainic et al. (2011) propose a heuristic adjustment strategy and compare the heuristic and the Lagrangean strategies combined with the TS algorithm for a time-dependent stochastic network design problem. Alvarez et al. (2021) solve the two-stage stochastic IRP with stochastic demand and supply using a heuristic PH algorithm and an ILS heuristic. They consider visit and routing decisions as the first-stage variables and propose an aggregation strategy for each of the three possible cases, i.e., consensus on both sets of variables, consensus on visit variables but not on routing variables or consensus on neither.

In order to implement the PH algorithm, we must duplicate the first-stage variables and divide the problem into scenario-specific subproblems, which we call SSPRP$^{(\varphi)}$ for the static-static strategy and SDPRP$^{(\varphi)}$ for the static-dynamic strategy. The decomposition procedure and the formulations are described in Section 4.1 for both strategies. In order to solve the scenario-specific subproblems (which are basically deterministic PRPs), an effective solution algorithm is needed. The heuristic solution algorithm to solve the scenario-specific subproblem is a three-phase algorithm: the first phase is to find an a priori tour over all the nodes, the second phase is to solve restricted PRPs (RPRPs) and the third phase is to solve CVRPs for each period. When integrating this heuristic with the PH algorithm, we only need to solve the TSP once at the beginning. Furthermore, the third phase (solving CVRPs) is only done for the best solution found by PH. The three-phase algorithm and its integration with the PH approach are explained in Section 4.3.

In this study, the heuristic adjustment approach proposed by Crainic et al. (2011) is used to accomplish the goal of getting consensus on the first-stage variables. Throughout the algorithm, costs associated with each of these variables can be adjusted at every iteration to achieve consensus. After the subproblems have been solved sequentially in each iteration, the stopping criteria are checked to determine whether to stop the algorithm or proceed to the next iteration with the updated parameters. At the end of each iteration, the following four criteria must be evaluated:

- If a consensus over all variables has been achieved;
- If the maximum number of iterations has been reached;
- If the maximum run-time has been exceeded;
- If the number of non-improving iterations has reached a specified number.

If the first criterion is met (consensus on all first-stage variables), the solution is feasible for the stochastic problem. If, after several iterations, the inconsistency persists, an aggregation strategy is applied to obtain a feasible solution. The adjustment and the aggregation strategy are discussed in detail in Section 4.2. The pseudo-code of the overall algorithm is given in Algorithm 1.

Algorithm 1 PH-based matheuristic algorithm

Solve the TSP:
return α_i and σ_i for all $i \in \mathcal{N}$
Initialize the PH:
    $v \leftarrow 0$
    for $\varphi \in \phi$ **do**
        $\bar{F}_{\tau}^{\varphi(v)} \leftarrow F$
        Solve $RPRP^{(\varphi)}$
        Set $ReferenceSolution$ (Eq. (38))
        Set $BestSolution$ (Eq. (42))
    end for
End Initialization
repeat
    $v \leftarrow v + 1$
    Perform global adjustment (Eq. (39))
    for $\varphi \in \phi$ **do**
        Perform local adjustment (Eq. (40))
        Fix eligible variables (Eq. (41))
        Solve $RPRP^{(\varphi)}$
        Update $ReferenceSolution$ (Eq. (38))
        Update $BestSolution$ (Eq. (42))
    end for
until Stopping criterion is met
return Setup, production, and inventory decisions
for $\varphi \in \phi$ **do**
    for $\tau \in \mathcal{T}$ **do**
        Solve CVRP
    end for
end for
return Routing and delivery decisions

4.1. *Scenario decomposition*

One can observe that the SPRP-AR formulations presented in Section 3 have a block-diagonal structure, where each block represents a deterministic PRP for a given scenario $\varphi \in \phi$. Constraints (2) and constraints (21) and (22) are the linking constraints that connect the first-stage and the second-stage variables. Thus, by duplicating the setup variables (y_τ), we introduce a specific setup decision variable y_τ^φ for each scenario $\varphi \in \phi$ and the static-dynamic problem can be reformulated as follows:

$$(PRP_{SD}) \qquad \min \quad \sum_{\varphi\in\phi} \pi_\varphi \Bigg(\sum_{\tau\in\mathcal{T}} \Big(F y_\tau^\varphi + u p_\tau^\varphi + \sum_{(i,j)\in\mathcal{E}} \sum_{\kappa\in\mathcal{K}} c_{ij} x_{ij\kappa\tau}^\varphi + \sum_{i\in\mathcal{N}} h_i I_{i\tau}^\varphi + \sum_{i\in\mathcal{N}_c} \beta_i o_{i\tau}^\varphi \Big) \Bigg) \tag{24}$$

s.t. (3) - (11) and (13) - (19),

$$p_\tau^\varphi \le \mathcal{M}_\tau^\varphi y_\tau^\varphi \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \tag{25}$$

$$y_\tau^\varphi = \bar{y}_\tau \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \tag{26}$$

$$y_\tau^\varphi \in \{0,1\} \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \tag{27}$$

$$\bar{y}_\tau \in \{0,1\} \qquad \forall \tau \in \mathcal{T}. \tag{28}$$

Scenario-decomposable constraints (25) replace the original linking constraints in the above formulation, while the non-anticipativity constraints (26) ensure that the setup decisions do not depend on the scenario and that the final solution of the first-stage variables is consistent across all scenarios. In this regard $\bar{y}_\tau \in \{0,1\}$, $\forall \tau \in \mathcal{T}$ is defined and is known as the "overall design vector".

In a similar manner, by duplicating the setup and production variables (y_τ and p_τ), and defining y_τ^φ and p_τ^φ for each scenario $\varphi \in \phi$ for the static-static case, we can reformulate the second problem as follows:

$$(PRP_{SS}) \qquad \min \quad \sum_{\varphi\in\phi} \pi_\varphi \Bigg(\sum_{\tau\in\mathcal{T}} \Big(F y_\tau^\varphi + u p_\tau^\varphi + \sum_{(i,j)\in\mathcal{E}} \sum_{\kappa\in\mathcal{K}} c_{ij} x_{ij\kappa\tau}^\varphi + \sum_{i\in\mathcal{N}} h_i I_{i\tau}^\varphi + \sum_{i\in\mathcal{N}_c} \beta_i o_{i\tau}^\varphi \Big) \Bigg) \tag{29}$$

s.t. (4) - (11) and (14) - (19),

$$p_\tau^\varphi \le \mathcal{M}_\tau^{max} y_\tau^\varphi \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \tag{30}$$

$$I_{0\tau}^\varphi = I_{0,\tau-1}^\varphi + p_\tau^\varphi - \sum_{i\in\mathcal{N}_c} \sum_{\kappa\in\mathcal{K}} q_{i\kappa\tau}^\varphi \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \tag{31}$$

$$y_\tau^\varphi = \bar{y}_\tau \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \tag{32}$$

$$p_\tau^\varphi = \bar{p}_\tau \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \tag{33}$$

$$y_\tau^\varphi \in \{0,1\} \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \tag{34}$$

$$\bar{y}_\tau \in \{0,1\} \qquad \forall \tau \in \mathcal{T} \quad (35)$$

$$p_\tau^\varphi \geq 0 \qquad \forall \tau \in \mathcal{T}, \varphi \in \phi \quad (36)$$

$$\bar{p}_\tau \geq 0 \qquad \forall \tau \in \mathcal{T}. \quad (37)$$

Again, the original linking constraints are replaced by constraints (30) and (31), the non-anticipativity constraints (32) and (33) are added, and $\bar{y}_\tau \in \{0,1\}, \ \forall \tau \in \mathcal{T}$ and $\bar{p}_\tau \geq 0, \ \forall \tau \in \mathcal{T}$ are defined as the overall design vector.

Rockafellar and Wets (1991) propose an augmented Lagrangean approach to relax non-anticipativity constraints (26), (32), and (33) and make the problem scenario-decomposable. Using Lagrangean multipliers, the relaxed constraints are incorporated into the objective function. These multipliers are updated at each iteration in order to maintain convergence. It is shown that this strategy leads to a global optimum in continuous problems, whereas this is not necessarily the case for integer problems (Rockafellar and Wets, 1991). In this study, we employ the heuristic adjustment strategy proposed by Crainic et al. (2011) to lead the first-stage variables to a consensus as it is shown to be more efficient, especially for binary first-stage variables.

4.2. Adjustment Strategy.

The heuristic adjustment strategy relies on adjusting the cost of the first-stage variables based on their frequency in the solutions of the scenario subproblems instead of updating the Lagrangean multipliers. We use a similar approach to that introduced in Crainic et al. (2011) to modify the setup costs during each iteration. As noted earlier, the solution to the SDPRP$^{(\varphi)}$(SSPRP$^{(\varphi)}$) is only optimal for a given scenario, while through an aggregation operator, we can benefit from these local solutions to lead our search toward the consensus of the first-stage variables. Thus, we define the reference solution $\bar{y}_\tau^{(\nu)}$ for each iteration ν, which is the weighted sum of scenario-determined setup decisions, i.e., $\hat{y}_\tau^{\varphi(\nu)}$, where the probability of the occurrence of a scenario is considered as the weight of that scenario. The reference solution for the setup variables that can be used in both strategies is calculated for a given iteration ν as follows:

$$\bar{y}_\tau^{(\nu)} = \sum_{\varphi \in \phi} \pi_\varphi \hat{y}_\tau^{\varphi(\nu)} \qquad \forall \tau \in \mathcal{T}. \quad (38)$$

For a given period $\tau \in \mathcal{T}$, $\bar{y}_\tau^{(\nu)} \in \{0,1\}$ implies that the consensus is achieved for this period over all scenarios in this iteration. If this situation applies to all first-stage variables, it means that the algorithm reached a consensus and the solution is feasible. However, the value of setup decisions in the reference solution is usually such that $0 < \bar{y}_\tau^{(\nu)} < 1$, which results in an infeasible solution due to the integrality conditions on the setup variables. Nevertheless, this solution contains valuable information about the likelihood of production occurring in a given period. Global adjustment strategies can use this information to guide the algorithm toward consensus. Lower values of $\bar{y}_\tau^{(v)}$ indicate that it is less probable that production takes place at that period in the final solution, while higher values state the opposite. In this regard, we define two thresholds θ_L and θ_H on $\bar{y}_\tau^{(v)}$ values of a given period. If $\bar{y}_\tau^{(v)}$ is greater than θ_H the corresponding cost is decreased while for values less than θ_L the setup cost is increased. Thus, the global adjustment strategy (39) aims to modify the setup costs according to the average value of variables through all scenarios:

$$\bar{F}_\tau^{(v)} = \begin{cases} \lambda \bar{F}_\tau^{(v-1)} & \text{if } \bar{y}_\tau^{(v-1)} < \theta_L \\ \frac{1}{\lambda} \bar{F}_\tau^{(v-1)} & \text{if } \bar{y}_\tau^{(v-1)} > \theta_H \\ \bar{F}_\tau^{(v-1)} & \text{otherwise,} \end{cases} \tag{39}$$

where $\lambda > 1$ is the modification rate and $\bar{F}_\tau^{(v)}$ is the modified global setup cost for period τ in iteration v in all scenarios to replace F in (43). Note that $0 < \theta_L < 0.5$ and $0.5 < \theta_H < 1$ in equation (39). Moreover, another adjustment can be applied based on the solution of a specific scenario. In this regard, when the absolute difference of a scenario's solution from its reference point is larger than γ_F, the corresponding cost must be adjusted as a factor of λ. Thus, we define the local adjustment strategy (40) to be only applied locally for a specific scenario φ to the corresponding costs of variables that are far from their reference point:

$$\bar{F}_\tau^{\varphi(v)} = \begin{cases} \lambda \bar{F}_\tau^{(v)} & \text{if } |y_\tau^{\varphi(v-1)} - \bar{y}_\tau^{(v-1)}| \geq \gamma_F \text{ and } y_\tau^{\varphi(v-1)} = 1 \\ \frac{1}{\lambda} \bar{F}_\tau^{(v)} & \text{if } |y_\tau^{\varphi(v-1)} - \bar{y}_\tau^{(v-1)}| \geq \gamma_F \text{ and } y_\tau^{\varphi(v-1)} = 0 \\ \bar{F}_\tau^{(v)} & \text{otherwise,} \end{cases} \tag{40}$$

where the distance threshold is $0.5 < \gamma_F < 1$ and $\bar{F}_\tau^{\varphi(v)}$ is the modified local setup cost for scenario $\varphi \in \phi$. Finally, we define another threshold $0 < \gamma_N < 0.5$ to fix the solutions that are close to the reference point for the next iteration, using the following equation:

$$y_\tau^{\varphi(v)} = y_\tau^{\varphi(v-1)} \qquad \text{if } |y_\tau^{\varphi(v-1)} - \bar{y}_\tau^{(v-1)}| \leq \gamma_N. \tag{41}$$

The above equation can improve the speed of the algorithm by reducing the size of the problem at each iteration. At the end of each iteration, we need to define the best solution that aggregates all variables to provide a feasible solution to the master problem. This function fixes the setup decisions to one, even if production is required for a single scenario in that period. Thus, the best solution is calculated using the following equation:

$$y_\tau^{best(v)} = \max_{\varphi \in \phi} (y_\tau^{\varphi(v)}) \qquad \forall \tau \in \mathcal{T}. \tag{42}$$

If the PH algorithm stops due to the convergence on the first-stage variables, the solution is feasible for the original problem, allowing the algorithm to proceed to the next phase. However, if the algorithm stops because of a stopping criterion other than convergence, it implies that a consensus among all scenarios concerning the first-stage variables has not been reached. In such cases, we designate the values of $y_{\tau}^{best(v)}$ and the corresponding second-stage solutions as the best feasible solution for the problem.

4.3. Solving Subproblems

The purpose of this section is to present a three-phase matheuristic algorithm for solving the scenario-specific subproblems sequentially. We must solve S deterministic PRPs per iteration in order to solve the SPRP-AR. The performance of the algorithm is improved by applying some modifications, since this is a demanding task even for a heuristic algorithm. The first step in the process consists of solving a TSP disregarding the capacity and demands of each node to determine an a priori tour over all nodes. This problem only needs to be solved once at the beginning, as the optimal TSP tour is the same for all subproblems. A restricted problem can then be solved using the sequence obtained in the first phase. The restricted PRPs (RPRP) must be solved for each scenario at each iteration until one of the stopping criteria of the PH algorithm is met. During the third phase of the algorithm, once the production and delivery variables have been fixed, a CVRP is solved for each pair of period and scenario, in order to enhance the quality of the routing decisions. Note that this third phase is not executed for every subproblem in the PH, but only at the end of the PH for the best solution. The proposed algorithm is inspired by the five-phase heuristic developed by Solyalı and Süral (2017) since this algorithm has shown promising results for both the PRP and IRP (Solyalı and Süral, 2022). Even so, there are significant differences between these two approaches in the second phase of the algorithm, where we propose a different MIP formulation which provides high-quality feasible solutions.

4.3.1. The First Phase: A priori tour.

In this phase, the Concorde solver is used to solve a TSP by considering all nodes while ignoring their demands (Applegate et al., 2020). This yields an a priori tour that starts from the plant, visits all customers and eventually returns to the starting point. Using the solution of this problem we define two sets α_i and σ_i for each node $i \in \mathcal{N}$ which will be used to impose the sequence of the a priori tour in the second-phase problem. The set α_i denotes the nodes that can be visited after node i, which includes the plant for all $i \in \mathcal{N}_c$ since the vehicle can return to the plant after making a delivery to a customer. The set σ_i defines the nodes that can be reached prior to node i, which also always includes $\{0\}$ since the vehicle can go to any customer after it departs from the plant. Finally, α_0 and σ_0 contain all the customers, meaning that they can be visited both before and after the plant since routes start and end at the plant.

4.3.2. The Second Phase: The restricted PRP.

In the second stage of the algorithm, we present an MIP formulation that uses the visiting sequence obtained from phase one to provide feasible solutions to the main problem. The sequence of the a priori tour is imposed, but it is possible to skip nodes. Solving this restricted stochastic problem is easier since subtour elimination constraints are no longer required. In order to solve this problem, we temporarily change the structure of the problem into a directed graph, and we assume the same transportation costs for both directions between any two vertices. In the current formulation, imposing an integrality condition on $x^{\varphi}_{ij\kappa\tau}$ is no longer necessary since integer $z^{\varphi}_{i\kappa\tau}$ variables in combination with sequence sets (α_i and σ_i) instead of subtour elimination constraints result in the integrality property on continuous $x^{\varphi}_{ij\kappa\tau}$. Thus, we replace the routing variables with the continuous variables $0 \leq \delta^{\varphi}_{ij\kappa\tau} \leq 1$. The formulation for the $RPRP^{(\varphi)}$ of a given scenario $\varphi \in \phi$ is presented below. We omit the φ index for the sake of notational simplicity.

$$(RPRP^{(\varphi)}) \quad \min \quad \sum_{\tau \in \mathcal{T}} \Big(F y_\tau + u p_\tau + \sum_{i \in \mathcal{N}} \sum_{j \in \alpha_i} \sum_{\kappa \in \mathcal{K}} c_{ij} \delta_{ij\kappa\tau} + \sum_{i \in \mathcal{N}} h_i I_{i\tau} + \sum_{i \in \mathcal{N}_c} \beta_i o_{i\tau} \Big) \tag{43}$$

s.t.

$$p_\tau \leq \mathcal{M}_\tau y_\tau \qquad \forall \tau \in \mathcal{T} \tag{44}$$

$$I_{0\tau} = I_{0,\tau-1} + p_\tau - \sum_{i \in \mathcal{N}_c} \sum_{\kappa \in \mathcal{K}} q_{i\kappa\tau} \qquad \forall \tau \in \mathcal{T} \tag{45}$$

$$I_{i\tau} = I_{i,\tau-1} + \sum_{\kappa \in \mathcal{K}} q_{i\kappa\tau} + o_{i\tau} - d_{i\tau} \qquad \forall i \in \mathcal{N}_c, \tau \in \mathcal{T} \tag{46}$$

$$I_{0\tau} \leq L_0 \qquad \forall \tau \in \mathcal{T} \tag{47}$$

$$I_{i\tau} + d_{i\tau} \leq L_i \qquad \forall i \in \mathcal{N}_c, \tau \in \mathcal{T} \tag{48}$$

$$\sum_{i \in \mathcal{N}_c} q_{i\kappa\tau} \leq Q z_{0\kappa\tau} \qquad \forall \kappa \in \mathcal{K}, \tau \in \mathcal{T} \tag{49}$$

$$\sum_{\kappa \in \mathcal{K}} z_{i\kappa\tau} \leq 1 \qquad \forall i \in \mathcal{N}_c, \tau \in \mathcal{T} \tag{50}$$

$$q_{i\kappa\tau} \leq \mathcal{W}_{i\tau}\, z_{i\kappa\tau} \qquad \forall i \in \mathcal{N}_c, \kappa \in \mathcal{K}, \tau \in \mathcal{T} \tag{51}$$

$$\sum_{j \in \alpha_i} \delta_{ij\kappa\tau} = z_{i\kappa\tau} \qquad \forall i \in \mathcal{N}, \kappa \in \mathcal{K}, \tau \in \mathcal{T} \tag{52}$$

$$\sum_{j \in \sigma_i} \delta_{ji\kappa\tau} = z_{i\kappa\tau} \qquad \forall i \in \mathcal{N}, \kappa \in \mathcal{K}, \tau \in \mathcal{T} \tag{53}$$

$$y_\tau \in \{0, 1\} \qquad \forall \tau \in \mathcal{T} \tag{54}$$

$$p_\tau \geq 0 \qquad \forall \tau \in \mathcal{T} \tag{55}$$

$$q_{i\kappa\tau} \geq 0 \qquad \forall i \in \mathcal{N}, \kappa \in \mathcal{K}, \tau \in \mathcal{T} \tag{56}$$

$$I_{i\tau} \geq 0 \qquad \forall i \in \mathcal{N}, \tau \in \mathcal{T} \quad (57)$$

$$o_{i\tau} \geq 0 \qquad \forall i \in \mathcal{N}, \tau \in \mathcal{T} \quad (58)$$

$$z_{i\kappa\tau} \in \{0, 1\} \qquad \forall i \in \mathcal{N}, \kappa \in \mathcal{K}, \tau \in \mathcal{T} \quad (59)$$

$$0 \leq \delta_{ij\kappa\tau} \leq 1 \qquad \forall i \in \mathcal{N}, j \in \alpha_i, \kappa \in \mathcal{K}, \tau \in \mathcal{T}. \quad (60)$$

The objective function (43) is similar to the deterministic PRP except that if a vehicle is in node i the next visited node must be chosen from the set α_i. In addition, we remove the subtour elimination constraints (11) since they are no longer required. To impose the sequence sets on routing decisions, we replace constraints (10) with sets of constraints (52) and (53).

The same RPRP$^{(\varphi)}$ can be used for the static-static case where the only difference is that constraints (44) are replaced with the following constraints:

$$p_\tau \leq \mathcal{M}_\tau^{(max)} y_\tau \qquad \forall \tau \in \mathcal{T}. \quad (61)$$

Within the PH-based matheuristic, after solving the subproblems for all scenarios, the algorithm moves to the next iteration and the corresponding costs of first-stage variables are updated in order to lead the PH algorithm to a consensus over all the first-stage variables. If a stopping criterion is met, we go to the third phase of the algorithm. The goal of this phase is to improve routing decisions for a specific period within a scenario.

In the static-static strategy, the production quantities are also scenario-independent while the result of the RPRPs provides scenario-specific values for these variables. Thus, another step is required to obtain a consensus for the production quantities as well. Since the p_τ variables are aligned with the setup decisions, the setup decisions obtained from the PH algorithm are used to find the production quantities. In addition, we fix the routing and visit decisions as they will be improved further in the next phase. Thus, the resulting model is a linear programming (LP) problem that can be solved to find the production quantities.

4.3.3. The Third Phase: Route improvement.

The second-phase problem provides feasible solutions to the subproblems, and as a result, the final iteration of the PH algorithm provides a feasible solution to the SPRP-AR. However, using the predefined sets in the RPRPs may result in non-optimal routes in some periods. To improve routing decisions, we solve the CVRP for all periods of every scenario to reduce routing costs whenever possible. In the CVRP, we consider the $q_{i\kappa\tau}^{\varphi}$ from the previous phase as the demand of node $i \in \mathcal{N}$, in period $\tau \in \mathcal{T}$, and under the scenario $\varphi \in \phi$. Note that the CVRP is solved for a (τ, φ) pair only if there exists at least one $z_{i\kappa\tau}^{\varphi}$ that is equal to one.

4.4. *Branch-and-Cut algorithm*

To solve the SPRP-AR, we also propose a BC algorithm to compare its results with the ones from the PH-based matheuristic. Several sets of valid inequalities are employed to strengthen the problem's formulation and enhance the LP bounds.

4.4.1. *Symmetry breaking inequalities*

When there is a homogeneous fleet of vehicles, considering the vehicle index leads to redundant enumerations. Assuming a fleet of K vehicles, only a subset of n^{φ}_{τ} ($n^{\varphi}_{\tau} < K$) may be dispatched in period $\tau \in \mathcal{T}$ under scenario $\varphi \in \phi$ out of the $\binom{K}{n^{\varphi}_{\tau}}$ options with the same cost for selecting the required vehicles (Adulyasak et al., 2014b). Thus, by adding constraints (62), this symmetry issue can be prevented:

$$z^{\varphi}_{0\kappa\tau} \leq z^{\varphi}_{0,\kappa-1,\tau} \qquad 2 \leq \kappa \leq K, \forall \tau \in \mathcal{T}, \varphi \in \phi. \tag{62}$$

Other symmetry-breaking inequalities are the lexicographic constraints (63) that impose an order to assign customers to vehicles in each period (Jans, 2009; Adulyasak et al., 2014b):

$$\sum_{i=1}^{j} 2^{(j-i)} z^{\varphi}_{i\kappa\tau} \leq \sum_{i=1}^{j} 2^{(j-i)} z^{\varphi}_{i,\kappa-1,\tau} \qquad \forall j \in \mathcal{N}_c, 2 \leq \kappa \leq K, \tau \in \mathcal{T}, \varphi \in \phi. \tag{63}$$

4.4.2. *Logical inequalities*

Logical inequalities can also be applied to reduce the solution space of the problem (Archetti et al., 2007; Coelho and Laporte, 2014). We add constraints (64) to the formulation to ensure that if edge $(i, j) \in \mathcal{E}$ is travelled by vehicle $\kappa \in \mathcal{K}$, then node $j \in \mathcal{N}_c$ is visited by the same vehicle:

$$x^{\varphi}_{ij\kappa\tau} \leq z^{\varphi}_{j\nu\tau} \qquad \forall i \in \mathcal{N}_c, j \in \mathcal{N}_c, \kappa \in \mathcal{K}, \tau \in \mathcal{T}, \varphi \in \phi. \tag{64}$$

Note that variable $x^{\varphi}_{ij\kappa\tau}$ can take 0, 1, or 2 when one of the incident nodes is the depot while it can only take values 0 and 1 otherwise. The last set of valid inequalities imposes that the vehicle $\kappa \in \mathcal{K}$ is able to visit customer $i \in \mathcal{N}_c$ only if it is dispatched from the depot:

$$z^{\varphi}_{i\kappa\tau} \leq z^{\varphi}_{0\kappa\tau} \qquad \forall i \in \mathcal{N}_c, \kappa \in \mathcal{K}, \tau \in \mathcal{T}, \varphi \in \phi. \tag{65}$$

The proposed valid inequalities are also used in solving the SPRP-FSR with the BC algorithm by dropping the scenario index from the variables. In addition, we utilize all valid inequalities except for constraints (64) in the RPRP problems to increase the efficiency of the matheuristic algorithm.

Once the valid inequalities are added to the formulation, we relax the subtour elimination constraints (SECs) (11) and solve a minimum s-t cut problem as the exact separation algorithm to identify and add the violated constraints. In this regard, the Concorde library is employed to solve the separation algorithm (Applegate et al., 2020). The same separation algorithm is used in phases 1 and 3 of the algorithm to deal with the SECs.

5. Computational Experiments.

We performed experiments on two different datasets derived from the deterministic instances presented in Archetti et al. (2011). Monte Carlo simulation is employed as a sampling approach in order to generate scenarios. The demand of a customer in each period is an independent random variable that follows a discrete uniform distribution with a range of $[\bar{d}_{it}(1-\epsilon), \bar{d}_{it}(1+\epsilon)]$, where $\bar{d}_{it}$ represents the nominal demand from the deterministic instances and ϵ is the uncertainty level, where $\epsilon \in [0, 1]$. Each scenario may occur with a probability of $\pi_\varphi = 1/S$, $\forall \varphi \in \phi$, so that all scenarios have the same probability. The outsourcing cost is calculated as $\beta_i = \lceil \hat{\alpha}(u + f/C + 2c_{0i}/Q) \rceil$ as in Adulyasak et al. (2015a), where $\hat{\alpha}$ is a predefined penalty coefficient.

On the first dataset, the experiments were only carried out with the static-dynamic strategy in order to compare the results of the SPRP-AR proposed in this paper and the SPRP-FSR presented in Adulyasak et al. (2015a). Specifically, this dataset contains two test sets: a small set called $\mathscr{P}2$ and a large set called $\mathscr{G}2$. Set $\mathscr{P}2$ consists of 5, 10, 15, and 20 customers with $T = 3$ and $K = 1$ and four instance types with distinct characteristics namely the standard setting, high unit production costs, high transportation costs, and no customer inventory costs. The larger set $\mathscr{G}2$ comprises only the standard setting and instances with up to 3 vehicles, $T = 3$ and 5 to 30 customers or $T = 6$ and up to 20 customers. Our analyses only contained cases with 100 scenarios since increasing the number of scenarios to 500 or 1000 did not provide significant improvements in the results, which is consistent with the findings of Adulyasak et al. (2015a), while they can drastically increase the computation time. All experiments in the first set have a penalty coefficient of $\hat{\alpha} = 5$ and an uncertainty level of $\epsilon = 0.2$.

To evaluate the stochastic solutions, we report the expected value of perfect information (EVPI) and the value of the stochastic solution (VSS) relative to the objective function value of the stochastic problem. The EVPI (66) is the difference between the weighted sum of the optimal objective values of the wait-and-see (WS^*) problems and the optimal objective value of the recourse problem (RP^*). The VSS, on the other hand, is calculated by first solving the expected value (EV) problem based on the average demand. Once the first-stage variables are found, the corresponding second-stage problems must be solved for all scenarios considering the fixed first stage decisions obtained from the EV problem. The expected value of the EV solution (EEV^*) can be calculated as the weighted sum of the optimal objective function value of the EEV^* subproblems. The VSS is the difference between the EEV^* and the RP^*. One of the challenges of this study is that the second stage of the SPRP-AR is not a linear problem that can be solved quickly. Specifically, it is an MIP formulation that includes SECs in each scenario, thus significantly increasing the complexity of the problem. Therefore, we require new measurements to assess the quality of solutions.

Since we cannot solve the SPRP-AR to optimality, we obtain an upper bound (UB) to the stochastic problem (RP^{UB}). As a consequence, when using RP^{UB} instead of RP*, we obtain UBs to the actual EVPIs and lower bounds (LBs) to the actual VSSs. We applied a BC algorithm to solve the WS and EV subproblems with a computation time limit of four hours. Nonetheless, there are some cases where these problems are not optimally solved. When this occurs, we use the LB of the WS subproblems (WS^{LB}) to provide a UB to the actual EVPI which we call EVPI^{UB}.

$$EVPI = RP^* - WS^* \leq RP^{UB} - WS^* \leq RP^{UB} - WS^{LB} = EVPI^{UB}. \tag{66}$$

In addition, we use the LB and the UB on the EEV* (EEV^{LB} and EEV^{UB}) to provide both an LB and a UB to VSS, which are denoted by VSS^{LB} and VSS^{UB}, respectively (67, 68):

$$VSS^{LB} = EEV^{LB} - RP^{UB} \leq EEV^* - RP^{UB} \leq EEV^* - RP^* = VSS \tag{67}$$

$$VSS \leq EEV^{UB} - RP^* \leq EEV^{UB} - RP^{LB} = VSS^{UB}. \tag{68}$$

We propose a new dataset called $\mathscr{M}$ in order to conduct experiments on both static-dynamic and static-static strategies. The reason why we introduce this second dataset is that the setup decisions in the $\mathscr{P}2$ and $\mathscr{G}2$ sets are mostly trivial, meaning that one can often obtain optimal setup decisions without considering the stochasticity. Solving the EV problem provides the optimal value for the first-stage variables, indicating that there is no benefit to solving the more complicated stochastic problem, as shown in Table 3. The new dataset is created based on the same Archetti et al. (2011) benchmark set while it differs in some parameters to yield non-trivial setup decisions. Consequently, the penalty coefficient $\hat{\alpha}$ is set to 50 in $\mathscr{M}$ as the penalty costs appear to be insufficient in the first dataset, resulting in a high level of unmet demand. The number of periods dramatically increases the complexity of the problem, but we do not generate instances with $T = 3$ because in practice setup decisions are often required to be made over a longer period of time. The demands in the deterministic dataset are constant over periods, which allows us to generate sets with longer planning horizons. In this regard, we constructed new sets considering the standard setting and with $T = 6$, $K \leq 2$, $N \leq 30$; $T = 6$, $K = 3$, $N \leq 20$; $T = 9$; $K = 1$, $N \leq 30$; and $T = 9$, $K = 2$, $N \leq 20$. The production capacity is set to $C = 3 \sum_{i \in \mathcal{N}_c} \bar{d}_{it}$. In order to calculate the transportation costs, Euclidean distances are used for all instances, while in set $\mathscr{M}$ we multiply them by a transportation coefficient (tc), where $tc = 4$ in all experiments.

Computational experiments were performed on an Intel Xeon Gold 6148 2.4 GHz processor and 32 GB of RAM. The algorithms were coded in C and Python programming languages while CPLEX 22.1.0 was used to solve the mathematical formulations. In CPLEX, eight threads were assigned for the parallel

Table 3: The results of SPRP-AR on $\mathscr{P}2$ and $\mathscr{G}2$ sets under the Static-Dynamic Strategy

T	K	S	#Ins	PH-M					BC	
				#Best	CPU (secs)	VSS^{LB} (%)	VSS^{UB} (%)	EVPI^{UB} (%)	CPU (secs)	Gap (%)
3	1	100	16	16	7.7	1.3	1.5	0.5	4,084.0	0.3
3	1	100	6	6	11.0	1.3	1.5	0.4	4,639.7	0.2
3	2	100	6	6	67.0	1.4	1.8	0.4	7,962.1	0.4
3	3	100	6	6	492.4	1.7	2.0	0.5	10,096.2	0.5
6	1	100	4	4	42.5	0.0	0.0	0.0	9,264.6	0.0
6	2	100	4	4	291.9	0.0	0.2	0.2	13,675.8	0.2
6	3	100	4	4	1,362.2	0.0	1.0	0.7	15,490.3	0.7
Total			46	46	224.6	1.0	1.3	0.4	9,316.1	0.3

processing and the optimality gap was set to 10^{-6} in all experiments.

5.1. Static-Dynamic Strategy.

In this section, we report the experiments on the Static-Dynamic strategy. In Table 3 the average performance of the heuristic and the BC algorithms on the $\mathscr{P}2$ and $\mathscr{G}2$ sets is provided. The solution of the heuristic algorithm was given to the BC algorithm as a warm start, and the computation time was set to 4 hours. The reported CPU time for BC represents the combined computation time of the BC algorithm and the heuristic. Besides the limit on the computation time, in several cases the limit on the memory caused the BC to stop while the algorithm still had not reached its time limit. Hence, there are cases where the optimality gap is positive while the reported CPU time is less than four hours.

Each row of the table reports the average results of a specific setting. The first three columns display the number of periods, the number of vehicles, and the number of scenarios, respectively. Column **#Ins** is the number of instances in each set. Under the PH-based matheuristic (PH-M), column **#Best** denotes the number of instances for which the BC algorithm could not find a better solution than the one obtained by the heuristic algorithm. The time limit for the PH-M was set to 2 hours and the actual running time is stated in seconds in column **CPU** under the PH-M. As stated in the previous section, we provide both an LB and a UB to VSS and a UB to EVPI which are in the VSS^{LB}, VSS^{UB} and EVPI^{UB} columns, respectively. Column **Gap(%)** reports the relative optimality gap. It is worth mentioning that besides the LB obtained from the BC algorithm, the weighted sum of the LB of the WS subproblems also provides a valid LB to the main recourse problem. We used the maximum of these two values to calculate the LB of the problem. The maximum number of iterations for the PH algorithm was set to 50 and the maximum number of non-improving iterations was set to 10. Moreover, θ_L, θ_H, γ_N, and γ_F were set to 0.4, 0.6, 0.2, and 0.8, respectively.

To demonstrate the advantage of using adaptive routing over fixed routes, as well as the efficacy of the proposed heuristic algorithm, we also solved the SPRP-FSR using the BC algorithm. The optimality gap and solution time for this algorithm are provided in Table 4. The time limit was set to 4 hours and the presented inequalities in Section 4.4 were also incorporated in order to improve the LP bounds. We applied a post-optimization approach to the SPRP-FSR solutions in which customers who were visited in a scenario with no delivery quantity were skipped in order to reduce transportation costs. Thus, after solving the SPRP-FSR using BC, if there was a customer i with $z_{i\kappa\tau} = 1$ and $q^{\varphi}_{i\kappa\tau} = 0$, we removed this customer from the visits in that scenario. In this regard, we calculated the new objective function by subtracting the transportation costs of the incident nodes j' and j'' to node i and adding the direct transportation cost from j' to j'' $(-c_{j'i} - c_{ij''} + c_{j'j''})$. In column **PostOpt**, we show the improvement obtained from this procedure as a percentage compared to the value of the objective function of the FSR (first-stage routing) problem.

Column **ImpLB(%)** demonstrates the LB of the cost improvement in percentage by using adaptive routing instead of first-stage routing. This is obtained by comparing the LB of the FSR problem and the result of the PH-M. We also report the UB of the possible cost improvement in column **ImpUB(%)** using the LB of the SPRP-AR and the UB of the FSR problem. In other words, **ImpLB(%)** and **ImpUB(%)** indicate the LB and the UB of the possible cost improvement by employing adaptive routing rather than fixed routes. It should be noted that value of zero was assigned to VSS^{LB} and ImpLB in all cases where these values were negative due to the fact that zero is always a valid LB for both. We should highlight that the average values for the VSS that are shown in Table 3 usually result from a few high VSS values while most of the instances have a VSS equal to 0. Thus, if the optimal setup decisions can be achieved by solving the EV problem, the result of the EEV problems can lead us to the optimal solution for the stochastic problem. It should be noted that, despite the fact that having VSS = 0 when solving a stochastic problem is not desirable, we can still achieve valuable improvements with adaptive routing over FSR even after applying the post-optimization technique (Table 4). We observe that for these data sets, the LB on the improvement is 0.7%. In addition, in this dataset, we still have cases where the VSS is large enough so that it is reasonable to consider the stochastic problem rather than the mean value problem.

As mentioned earlier, we created the set $\mathcal{M}$ to compare the results where solving the stochastic problem has a more tangible impact on the outcome. Table 5 shows the result of SPRP-AR under the static-dynamic strategy. As expected the CPU time of both the PH-M and the BC algorithm significantly increases when more scenarios are considered. In addition, for 200 scenarios, the BC algorithm was able to find a better solution for one of the instances. This results from the second stage of the RPRPs being unable to reach their optimality within the algorithm's time limit. In these experiments, VSS^{LB}, VSS^{UB}, and EVPI^{UB} remained almost the same regardless of the number of scenarios. When $S = 50$, the BC algorithm was able to find better LBs than the WS subproblems, which resulted in a lower optimality gap compared to cases with more scenarios. Table 6 provides the comparison of the results with the SPRP-FSR. ImpLB decreases with an increase in scenarios, as uncertainty has a greater impact on the FSR problem since routing decisions

Table 4: Comparing the results of the SPRP-AR and the SPRP-FSR on $\mathscr{P}2$ and $\mathscr{G}2$ sets under the Static-Dynamic Strategy

T	K	S	#Ins	SPRP-AR (PH-M) Gap (%)	SPRP-AR (PH-M) CPU (secs)	SPRP-FSR (BC) Gap (%)	SPRP-FSR (BC) CPU (secs)	SPRP-FSR (BC) PostOpt (%)	ImpLB (%)	ImpUB (%)
3	1	100	16	0.3	7.7	0.0	7.1	0.1	0.6	0.8
3	1	100	6	0.2	11.0	0.0	49.6	0.1	0.7	0.9
3	2	100	6	0.4	67.0	0.0	623.1	0.1	0.8	1.2
3	3	100	6	0.5	492.2	0.0	2,095.8	0.3	1.4	1.9
6	1	100	4	0.0	42.5	0.0	322.8	0.0	0.7	0.7
6	2	100	4	0.2	291.9	0.0	2,353.2	0.1	0.8	1.0
6	3	100	4	0.7	1,362.2	3.9	11,246.5	0.4	0.5	4.3
Total			46	0.3	224.6	0.3	1,574.3	0.1	0.7	1.3

must be made in the first stage and reducing the number of scenarios may have a significant impact on the FSR. However, the AR problem does not seem to be affected as much by the number of scenarios. The uncertainty level ϵ was set to 0.2 and the problem was solved with 50, 100, and 200 scenarios. It is worth mentioning that in the heuristic algorithm, the 2-hour time limit was split into one hour for solving the phase two subproblems and one hour for the phase three subproblems, as phase 1 only takes a few seconds to solve.

Figure 2a presents the average value of the objective function of the SPRP-AR model under different numbers of scenarios. One can observe that the objective function values for the AR problem remain relatively consistent across all scenarios. However, the objective function values for the FSR problem and FSR(PO) (FSR after post-optimization) are consistently higher than those of the AR. Notably, for larger problem instances, the BC algorithm often failed to find high-quality feasible solutions for SPRP-FSR, resulting in instances where the only viable solution was to outsource all demands. Consequently, such cases exhibited significantly larger objective function values and optimality gaps. To ensure a more realistic comparison, instances where the optimality gap for the FSR exceeded 20% in all cases were disregarded in cost comparisons. Otherwise, the cost of FSR would be considerably higher than that of AR. In Figure 2b, the routing costs of AR, FSR, and FSR(PO) are depicted.

The results of the SPRP-AR with different uncertainty levels including $\epsilon = 0.4, 0.6, 0.8$ with 100 scenarios are shown in Table 7. The BC algorithm was not able to find a better solution than the one obtained by PH-M in 151 of the 156 cases, keeping in mind that the BC algorithm was started with the PH-M solution as a starting point. This denotes the deficiency of the exact algorithms in solving such a complicated problem. For the instances in which the BC was able to obtain a better solution, the average gap improved by 0.3%, which is indicative of the efficiency of the PH-M algorithm. The average optimality gap of PH-M for all instances of set $\mathscr{M}$ under the static-dynamic strategy with different levels of uncertainty was 3%.

Table 5: The results of the SPRP-AR for different number of scenarios on set $\mathscr{M}$ under the Static-Dynamic Strategy

T	K	S	#Ins	PH-M					BC	
				#Best	CPU (secs)	VSS^{LB} (%)	VSS^{UB} (%)	EVPI^{UB} (%)	CPU (secs)	Gap (%)
6	1	50	6	6	59.2	3.4	4.2	0.8	7,581.2	0.8
6	2	50	6	6	891.5	3.9	6.0	1.6	13,088.6	1.6
6	3	50	4	4	904.8	4.0	9.3	3.2	13,530.1	3.2
9	1	50	6	6	283.6	2.2	3.1	0.9	10,618.7	0.9
9	2	50	4	4	1,269.5	2.0	7.7	3.6	13,606.3	3.6
Total			26	26	619.4	3.1	5.7	1.8	11,395.3	1.8
6	1	100	6	6	114.7	3.3	4.1	0.8	12,405.5	0.8
6	2	100	6	6	973.5	3.2	5.9	2.5	13,234.9	2.5
6	3	100	4	4	1,212.7	3.5	9.0	3.5	12,997.5	3.5
9	1	100	6	6	644.0	2.0	3.1	1.1	11,665.6	1.1
9	2	100	4	4	1,694.2	2.1	7.8	4.0	15,211.7	4.0
Total			26	26	847.0	2.8	5.6	2.2	12,948.9	2.2
6	1	200	6	6	289.1	3.3	4.1	0.8	12,701.7	0.8
6	2	200	6	6	2,505.8	3.2	5.8	2.2	16,145.1	2.2
6	3	200	4	4	1,576.9	3.7	9.1	4.2	13,416.7	4.2
9	1	200	6	5[1]	2,036.6	1.8	2.9	1.3	16,439.3	1.3
9	2	200	4	4	1,903.9	1.5	7.5	5.3	16,307.6	5.3
Total			26	25	1,650.5	2.7	5.5	2.4	15,023.6	2.4

[1] The Branch-and-Cut UB has improved the optimality gap by 0.1%

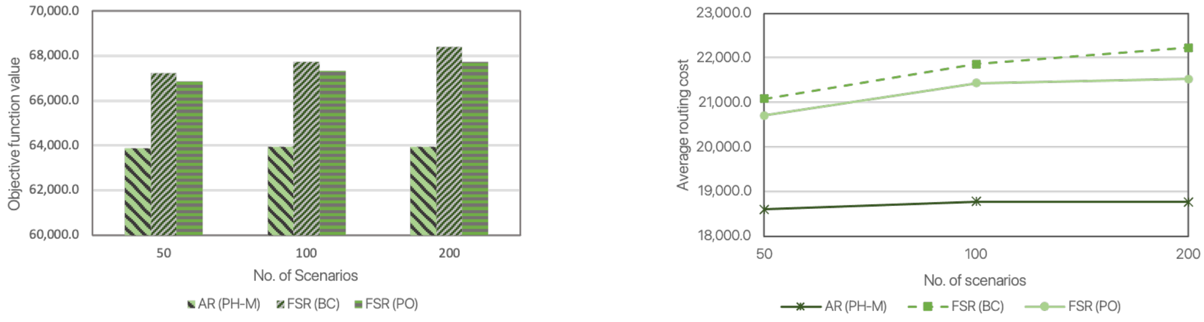
Table 6: Comparing the results of the SPRP-AR and the SPRP-FSR for different number of scenarios on set $\mathscr{M}$ under the Static-Dynamic Strategy

T	K	S	#Ins	SPRP-AR (PH-M)		SPRP-FSR (BC)				
				Gap (%)	CPU (secs)	Gap (%)	CPU (secs)	PostOpt (%)	ImpLB (%)	ImpUB (%)
6	1	50	6	0.8	59.2	0.0	303.5	0.5	3.9	4.7
6	2	50	6	1.6	891.5	4.3	10,049.5	1.0	2.5	8.3
6	3	50	4	3.2	904.8	8.0	12,132.2	1.1	2.0	11.6
9	1	50	6	0.9	283.6	1.4	9,556.4	0.5	2.5	4.8
9	2	50	4	3.6	1,269.5	13.7	14,403.5	0.7	0.7	15.1
Total			26	1.8	619.4	4.7	8,676.9	0.7	2.7	8.2
6	1	100	6	0.8	114.7	0.0	1,413.7	0.7	4.0	4.8
6	2	100	6	2.5	973.5	9.5	11,699.5	1.5	2.2	12.7
6	3	100	4	3.5	1,212.7	12.0	14,416.2	1.5	1.8	14.4
9	1	100	6	1.1	644.0	3.9	10,349.7	0.5	2.0	7.0
9	2	100	4	4.0	1,694.2	20.2	14,402.3	1.0	0.5	20.4
Total			26	2.2	847.0	8.1	9,848.1	1.0	2.4	11.0
6	1	200	6	0.8	289.1	0.2	6,153.1	0.7	4.0	5.0
6	2	200	6	2.2	2,505.8	39.3	12,893.1	2.4	1.7	40.5
6	3	200	4	4.2	1,576.9	51.7	14,401.8	1.4	1.8	53.4
9	1	200	6	1.3	2,036.6	18.5	12,389.3	0.6	1.8	20.8
9	2	200	4	5.3	1,903.9	55.3	14,401.0	0.5	0.2	55.0
Total			26	2.4	1,650.5	29.8	11,685.6	0.8	2.1	32.0

Table 7: Summary of the SPRP-AR results for different uncertainty levels on set $\mathscr{M}$ under the Static-Dynamic Strategy

T	K	ϵ	#Ins	PH-M					BC	
				#Best	CPU (secs)	VSS^{LB} (%)	VSS^{UB} (%)	EVPI^{UB} (%)	CPU (secs)	Gap (%)
6	1	0.4	6	6	119.2	10.3	11.0	0.8	12,000.3	0.8
6	2	0.4	6	6	1,139.7	11.3	13.2	1.7	12,426.7	1.7
6	3	0.4	4	4	1,241.8	12.0	16.7	3.0	13,015.7	3.0
9	1	0.4	6	6	796.5	6.7	7.7	1.0	14,703.7	1.0
9	2	0.4	4	4	1,971.6	5.5	12.5	4.0	16,041.0	4.0
Total			26	26	968.7	9.2	11.9	1.9	13,500.4	1.9
6	1	0.6	6	5[1]	121.3	16.8	18.4	2.0	13,084.8	1.9
6	2	0.6	6	6	1,013.7	18.4	21.3	3.1	12,476.5	3.1
6	3	0.6	4	4	1,257.5	19.3	25.1	5.2	10,725.5	5.2
9	1	0.6	6	5[2]	753.9	11.6	13.2	2.2	14,600.0	1.6
9	2	0.6	4	4	1,985.5	10.1	17.6	5.4	16,388.7	5.4
Total			26	24	934.8	15.3	18.8	3.3	13,439.4	3.3
6	1	0.8	6	5[3]	162.9	21.6	24.0	3.5	11,813.1	3.4
6	2	0.8	6	6	1,102.1	23.5	27.5	4.6	12,084.9	4.6
6	3	0.8	4	4	1,403.4	24.8	31.5	7.1	13,345.0	7.1
9	1	0.8	6	5[4]	841.2	15.3	17.7	2.9	12,105.3	2.9
9	2	0.8	4	4	2,359.0	14.0	21.8	6.4	16,762.1	6.4
Total			26	24	1,064.9	19.9	24.2	4.6	12,940.3	4.6

[1] The Branch-and-Cut UB has improved the optimality gap by 0.05%; [2] The Branch-and-Cut UB has improved the optimality gap by 0.6%; [3] The Branch-and-Cut UB has improved the optimality gap by 0.5%; [4] The Branch-and-Cut UB has improved the optimality gap by 0.3%.



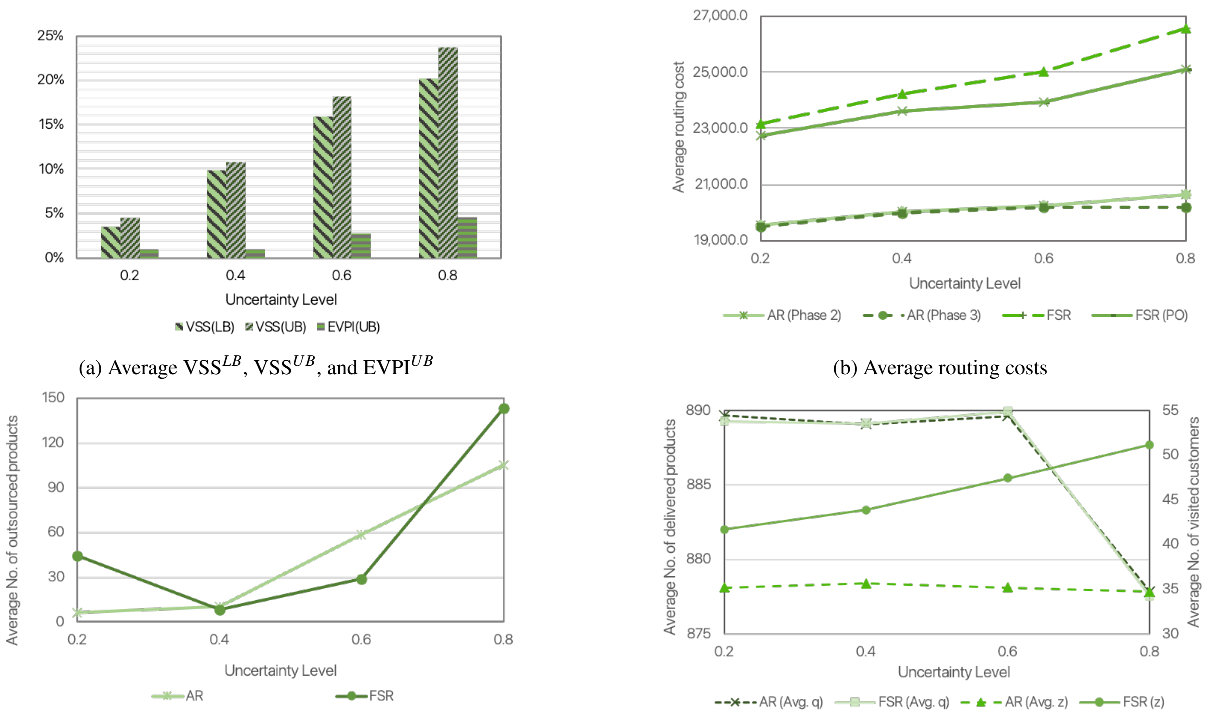
(a) Objective function value

(b) Average routing costs

Figure 2: Comparison of costs for different number of scenarios on set $\mathcal{M}$

We summarize the results of the AR and the FSR problems with different uncertainty levels, solved with the PH-M and BC algorithms, in Table 8. As can be seen from the table, the optimality gap for the BC algorithm remained relatively constant for various levels of uncertainty. In contrast, the PH-M gap increased for higher uncertainty levels. There are two reasons for this. First, since LBs are derived from WS subproblems, increasing uncertainty naturally increases the EVPI, resulting in worse LBs; second, using the PH algorithm we decompose the stochastic problem into subproblems each representing a specific scenario. The presence of greater stochasticity makes it more challenging to find consensus and aggregate the first-stage variables. However, it is important to keep in mind that PH-M still has a lower optimality gap while taking less computational resources. When observing the improvement of the objective function value from the **ImpLB** column, it is important to remember that this is the minimum possible saving of using the AR, whereas the actual savings could be greater, as in some cases the optimality gap for the BC algorithm was so high that for those cases we had to consider a 0% improvement on the LB. Even so, with 100 scenarios the LB of the average improvement is as low as 2.4% with $\epsilon = 0.2$ and goes up to 6.4% when the $\epsilon = 0.8$. There is a substantial improvement in the objective function which is primarily attributable to routing costs. Based on a comparison between the objective function of the AR and FSR(PO), we find that considering flexible routes results in better solutions, while at the same time providing a more reasonable solution time for the PH-M. The AR becomes even more relevant when routing costs are high or when the environmental impact of transportation is taken into consideration.

One may observe that in larger instances, specifically for those with $T = 9$, $K = 2$, the improvement obtained by applying post-optimization on the FSR problem is more than ImpLB. We should note that these two values are not directly comparable unless the optimality gap of the FSR is zero. It is due to the fact that the former is obtained by improving the UB of the FSR while the latter represents the relative difference between the AR UB and the FSR LB. Thus, a weak LB for the FSR can result in a low value of ImpLB which is the LB of the potential improvements. However, by looking at ImpUB we can see in all such cases the UB of the potential improvement is also so high which means that we can expect to have higher



(a) Average VSS^{LB}, VSS^{UB}, and EVPI^{UB}

(b) Average routing costs

(c) Average outsourced products

(d) Average visits and delivered products

Figure 3: An analysis of the behavior of solutions under the static-dynamic strategy with different uncertainty levels

improvements by employing adaptive routing in these cases, as well.

We provide Figure 3 to better illustrate the behavior of the solutions while having different values of uncertainty level. Figure 3a indicates the average values of VSS^{LB}, VSS^{UB}, and EVPI^{UB} for different uncertainty levels. We can observe the rapid increase of VSS for higher levels of uncertainty. The VSS^{LB} goes from 3.5% when $\epsilon = 0.2$ to 20.2% when $\epsilon = 0.8$. It is evident, therefore, that when the level of uncertainty is high, it is crucial to consider the stochastic problem instead of the mean value problem. As shown by the low values of EVPI^{UB}, more accurate information would not be of significant value, demonstrating the robustness of the solutions.

Figure 3b presents the routing cost of both the AR and the FSR problems. Since the heuristic PH-M algorithm includes two stages of routing decisions, we present the average routing cost after phases two and three. To assess the cost improvement that is achieved by utilizing flexible routes, we report the routing costs of the FSR before and after the post-optimization process. The third phase of the PH-M usually leads to a slight improvement in all cases, which is beneficial as this phase takes only a small portion of the solution time and often enhances the result. Here, one of the most significant observations is the difference between the AR and the FSR routing costs, which emphasizes the importance of considering adaptive routing. As can be seen from Table 7, this improvement is associated with a lower CPU time, as the average CPU time for SPRP-AR is approximately 15 minutes, whereas the average CPU time for SPRP-FSR is slightly less

Table 8: Comparing the results of the SPRP-AR and the SPRP-FSR for different uncertainty levels on set $\mathscr{M}$ under the Static-Dynamic Strategy

T	K	ϵ	#Ins	SPRP-AR (PH-M)		SPRP-FSR (BC)				
				Gap (%)	CPU (secs)	Gap (%)	CPU (secs)	PostOpt (%)	ImpLB (%)	ImpUB (%)
6	1	0.4	6	0.8	119.2	0.2	3,826.7	0.8	4.3	5.2
6	2	0.4	6	1.7	1,139.7	6.0	10,555.7	1.1	2.5	9.4
6	3	0.4	4	3.0	1,241.8	12.5	14,404.4	1.9	2.2	15.9
9	1	0.4	6	1.0	796.5	2.8	9,957.5	0.7	2.7	6.4
9	2	0.4	4	4.0	1,971.6	13.3	14,402.3	1.1	1.0	14.4
Total			26	1.9	968.7	6.1	10,048.7	1.1	2.9	9.5
6	1	0.6	6	1.9	121.3	0.0	2,107.2	1.8	4.5	5.9
6	2	0.6	6	3.1	1,013.7	6.6	10,801.6	1.8	3.0	12.3
6	3	0.6	4	5.2	1,257.5	11.0	14,403.6	2.5	2.5	16.9
9	1	0.6	6	2.2	753.9	1.9	9,794.6	1.0	3.6	7.1
9	2	0.6	4	5.4	1,985.5	13.5	14,402.1	1.7	1.6	16.7
Total			26	3.2	934.8	5.7	9,670.9	1.7	3.5	11.0
6	1	0.8	6	3.4	162.9	0.1	3,205.9	2.1	7.6	9.9
6	2	0.8	6	4.6	1,102.1	7.5	11,446.0	3.0	7.1	17.8
6	3	0.8	4	7.1	1,403.4	11.6	14,402.1	3.3	4.1	22.3
9	1	0.8	6	2.9	841.2	1.9	9,766.1	1.7	6.6	10.9
9	2	0.8	4	6.4	2,359.0	13.0	14,403.9	2.3	2.2	20.2
Total			26	4.4	1,064.9	6.0	10,066.6	2.4	6.4	15.5

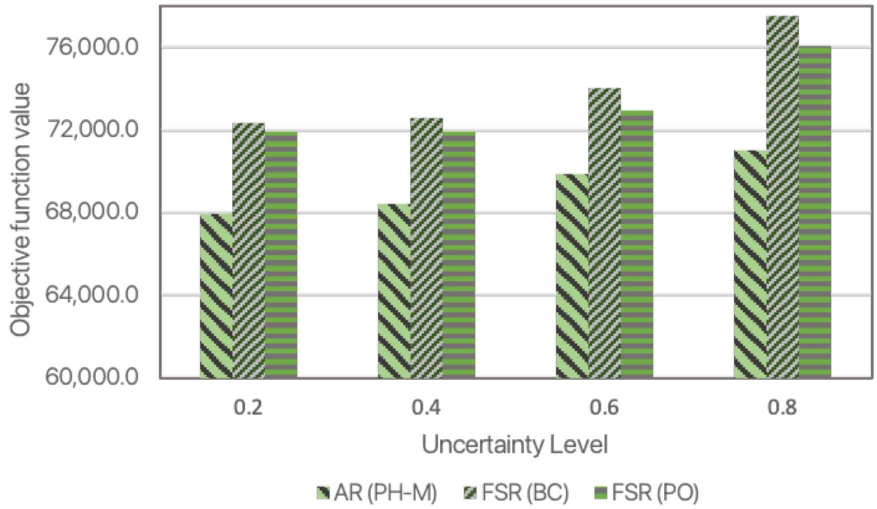


Figure 4: Objective function values for the AR (PH-M), FSR (BC), and FSR after post-optimization

than three hours, with the heuristic algorithm yielding a reasonable gap.

In both AR and the FSR, the trend line for outsourced products indicate that the average number of outsourced products increases as the uncertainty level rises with one exception for FSR when $\epsilon = 0.2$, as depicted in Figure 3c. This discrepancy is likely attributed to the higher optimality gap observed in FSR under this specific setting. Additionally, the analysis presented in Figure 3d indicates that the number of delivered products remains approximately equal for both problems, while the number of visits required by the AR problem is significantly lower. Consequently, an equivalent quantity of products can be delivered to customers with a reduced number of visits, showcasing the advantages of employing the AR problem. These observations were consistent across all cases considered, with a focus on instances where $S = 100$.

Figure 4 presents a comparison of the average value of the objective function for these two problems. The objective function values for AR solved with the PH-M, the UB of the FSR solved with BC, and the value of FSR after applying the post-optimization on the UB. This figure provides a comprehensive understanding of why ImpLB is not directly comparable to FSR(PO), as the objective function values of AR are consistently lower than those of FSR(PO) in all cases.

5.2. Static-static Strategy.

The following section presents the findings of the SPRP-AR considering the "static-static" setting. This method limits the flexibility of the problem by requiring the determination of lot sizes at the beginning of the planning horizon. However, it can reduce the nervousness of the system due to the fact that both setups and production quantities are planned in advance (Tunc et al., 2013). It is particularly useful when respecting production capacity is crucial since deterministic variables for production quantities can ensure that these

Table 9: The results of the SPRP-AR for different number of scenarios on set $\mathcal{M}$ under the Static-Static Strategy

T	K	S	#Ins	PH-M					BC	
				Gap (%)	CPU (secs)	VSS^{LB} (%)	VSS^{UB} (%)	$EVPI^{UB}$ (%)	CPU (secs)	Gap (%)
6	1	50	6	3.2	68.4	5.1	8.1	4.7	12,083.6	3.2
6	2	50	6	5.5	980.8	5.4	11.0	5.5	13,004.4	5.5
6	3	50	4	6.8	1,026.2	4.4	15.4	6.8	15,429.2	6.8
9	1	50	6	3.7	310.2	3.1	6.8	3.9	14,714.4	3.7
9	2	50	4	7.0	1,588.9	2.1	12.0	7.0	14,956.5	6.9
Total			26	5.0	716.0	4.6	10.2	5.4	13,859.9	5.0
6	1	100	6	3.9	140.0	4.4	8.2	5.0	14,543.7	3.9
6	2	100	6	6.4	1,234.4	4.4	10.5	6.5	15,638.1	6.4
6	3	100	4	7.4	1,554.3	3.7	14.6	7.4	15,960.6	7.4
9	1	100	6	4.7	873.1	2.5	7.0	4.9	15,276.1	4.6
9	2	100	4	7.9	1,717.0	2.1	12.3	7.9	14,228.5	7.9
Total			26	5.8	1,021.9	4.0	10.1	6.1	15,134.8	5.8
6	1	200	6	4.6	268.5	4.1	8.5	4.9	14,548.3	4.5
6	2	200	6	6.5	1,979.2	3.7	10.3	6.5	16,384.0	6.5
6	3	200	4	8.5	1,919.3	3.6	14.4	8.5	16,323.3	8.5
9	1	200	6	4.8	1,189.3	1.9	6.7	4.8	15,591.8	4.7
9	2	200	4	11.0	2,858.7	1.5	11.9	11.0	17,262.0	11.0
Total			26	6.7	1,528.2	3.4	9.9	6.8	15,903.3	6.6

limitations are addressed throughout the planning horizon (Tempelmeier and Herpers, 2010).

In Table 9, we present the results of the SPRP-AR with the static-static case for different scenarios using similar settings as in the previous section. Note that the instances from the previous section have been used for comparative purposes. As expected, optimality gaps are higher for this strategy. However, this does not necessarily imply that algorithm's performance is inferior since finding a better LB becomes more difficult in this case. The reason is that considering production quantities in the first stage also increases the EVPI. In order to demonstrate the efficacy of the presented algorithm for the static-static strategy, we provide the optimality gaps for both PH-M and BC algorithms. It can be observed that, in some cases, the BC algorithm provided better solutions than the PH-M algorithm. Even so, these improvements are usually so small that the average total improvement is 0.1%. It should be emphasized once again that the BC algorithm was given the PH-M solution as a warm start and was still unable to improve it within the limits of the computation in most instances.

A significant observation in the static-static case is the higher VSS^{LB} values, which is again due to the presence of more first-stage variables. As a result, stochastic programming becomes even more important

Table 10: Comparing the results of the SPRP-AR and the SPRP-FSR for different number of scenarios on set $\mathscr{M}$ under the Static-Static Strategy

T	K	S	#Ins	SPRP-AR (PH-M)		SPRP-FSR (BC)				
				Gap (%)	CPU (secs)	Gap (%)	CPU (secs)	PostOpt (%)	ImpLB (%)	ImpUB (%)
6	1	50	6	3.2	68.4	0.0	331.6	0.5	3.2	6.6
6	2	50	6	5.5	980.8	6.0	9,990.3	1.2	2.0	12.7
6	3	50	4	6.8	1,026.2	8.5	12,456.4	1.3	1.9	15.1
9	1	50	6	3.7	310.2	1.4	9,708.0	0.4	2.1	7.1
9	2	50	4	7.0	1,588.9	12.5	14,403.5	0.9	0.7	16.4
Total			26	5.0	716.0	4.9	8,754.6	0.8	2.3	11.0
6	1	100	6	3.9	140.0	0.0	2,059.7	0.7	3.3	7.3
6	2	100	6	6.4	1,234.4	8.1	10,784.2	1.2	1.9	14.0
6	3	100	4	7.4	1,554.3	10.6	14,402.2	1.1	1.4	15.8
9	1	100	6	4.7	873.1	2.3	10,139.7	0.4	1.1	8.1
9	2	100	4	7.9	1,717.0	35.1	14,403.4	1.3	0.3	37.5
Total			26	5.8	1,021.9	9.4	9,735.5	0.9	1.8	15.0
6	1	200	6	4.6	268.5	0.2	6,916.3	0.6	3.4	8.0
6	2	200	6	6.5	1,979.2	37.5	12,699.3	1.0	1.8	41.4
6	3	200	4	8.5	1,919.3	33.1	14,401.2	1.6	0.9	36.2
9	1	200	6	4.8	1,189.3	3.7	12,170.5	0.6	1.4	9.0
9	2	200	4	11.0	2,858.7	39.3	14,401.4	0.3	0.1	41.5
Total			26	6.7	1,528.2	19.9	11,766.4	0.8	1.8	24.8

when the company is required to employ this strategy. Table 10 presents a comparison between the results of the first-stage and second-stage routing problems under the static-static strategy. Based on the results, the LB of the cost improvement is slightly worse than in the previous case. The BC algorithm for the FSR also has a larger optimality gap, which may be one of the factors contributing to the difference. The ImpUB is also significantly higher, which makes it difficult to draw any conclusive comparison between the two cases in terms of possible improvements. Even so, we can observe that adaptive routing reduces routing costs and the improvements are better than those obtained by post-optimization. A comparison of the results of the static-static strategy with different uncertainty levels can be found in Tables 11 and 12. As can be seen, the VSS values are significantly higher than the static-dynamic strategy.

Figure 5 displays the comparison of the static-static and the static-dynamic strategies. We can observe the value of the objective function for different numbers of scenarios in Figure 5a and for different uncertainty levels in Figure 5b. In both figures, the trend line of the static-static strategy is above the static-dynamic strategy, indicating higher objective function values for the former. A significant portion of the higer values can be attributed to higher holding costs since the model under the static-static strategy tends to produce more products in earlier periods to avoid outsourcing in later periods. This can be seen in Fig-

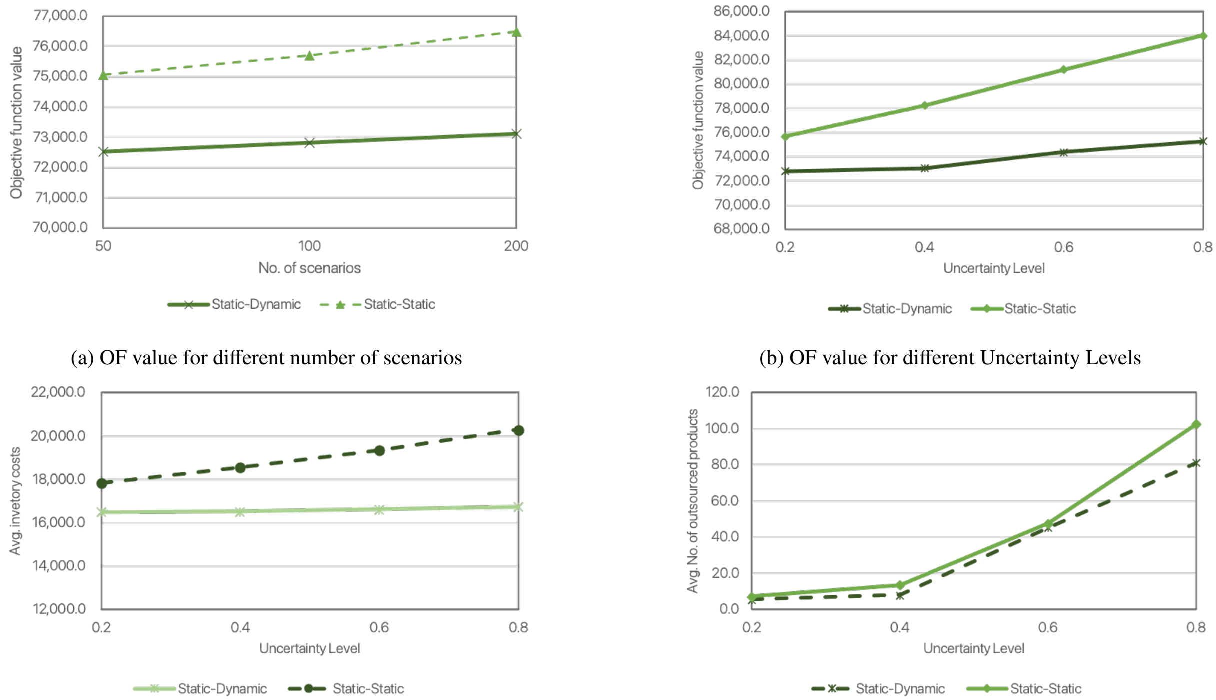
Table 11: Summary of the SPRP-AR results for different uncertainty levels on set $\mathscr{M}$ under the Static-Static Strategy

T	K	ϵ	#Ins	PH-M					BC	
				Gap (%)	CPU (secs)	VSS^{LB} (%)	VSS^{UB} (%)	EVPI^{UB} (%)	CPU (secs)	Gap (%)
6	1	0.4	6	4.0	177.6	15.6	19.1	7.5	12,606.4	4.0
6	2	0.4	6	8.5	1,201.5	15.3	23.0	9.0	14,579.8	8.5
6	3	0.4	4	11.1	1,394.4	11.7	28.9	11.1	12,717.4	11.1
9	1	0.4	6	5.8	773.5	9.0	14.3	6.7	13,988.7	5.6
9	2	0.4	4	10.2	1,940.4	5.9	21.5	10.2	16,342.3	10.1
Total			26	7.5	1009.8	13.3	20.8	8.6	13,972.6	7.4
6	1	0.6	6	5.1	178.1	26.5	30.2	10.9	12,405.0	5.1
6	2	0.6	6	9.1	1,284.6	26.7	33.7	12.0	14,118.2	9.1
6	3	0.6	4	12.2	1,263.9	20.3	40.2	13.6	12,571.6	12.2
9	1	0.6	6	6.4	890.5	16.5	22.1	9.6	15,293.4	6.3
9	2	0.6	4	12.0	2,622.5	11.6	30.2	13.3	17,025.6	12.0
Total			26	8.5	1,140.9	23.5	30.7	11.6	14,203.4	8.4
6	1	0.8	6	6.3	191.5	31.6	36.0	14.6	11,763.9	6.3
6	2	0.8	6	10.9	1,246.5	31.0	38.9	16.6	13,456.8	10.9
6	3	0.8	4	12.6	1,328.5	23.5	44.3	17.4	12,520.0	12.6
9	1	0.8	6	6.6	854.0	20.3	25.8	12.3	15,257.1	6.5
9	2	0.8	4	11.0	2,565.6	14.2	32.3	15.3	16,327.3	10.9
Total			26	9.1	1,128.0	27.8	35.0	15.1	13,779.1	9.1

Table 12: Comparing the results of the SPRP-AR and the SPRP-FSR for different uncertainty levels on set $\mathscr{M}$ under the Static-Static Strategy

T	K	ϵ	#Ins	SPRP-AR (PH-M) Gap (%)	SPRP-AR (PH-M) CPU (secs)	SPRP-FSR (BC) Gap (%)	SPRP-FSR (BC) CPU (secs)	SPRP-FSR (BC) PostOpt (%)	ImpLB (%)	ImpUB (%)
6	1	0.4	6	4.0	177.6	0.4	5,125.9	0.9	3.8	8.0
6	2	0.4	6	8.5	1,201.5	6.2	11,030.9	0.8	2.3	14.5
6	3	0.4	4	11.1	1,394.4	13.3	14,401.9	1.5	1.9	21.6
9	1	0.4	6	5.8	773.5	2.6	9,946.0	0.6	2.1	10.1
9	2	0.4	4	10.2	1,940.4	13.1	14,402.6	1.4	0.8	18.9
Total			26	7.5	1,009.8	6.2	10,455.2	1.0	2.5	13.8
6	1	0.6	6	5.1	178.1	0.0	1,931.3	1.5	3.6	8.5
6	2	0.6	6	9.1	1,284.6	9.1	10,816.9	2.4	2.3	19.0
6	3	0.6	4	12.2	1,263.9	10.8	14,402.5	2.0	2.2	22.4
9	1	0.6	6	6.4	890.5	1.4	9,714.7	0.9	2.5	10.1
9	2	0.6	4	12.0	2,622.5	33.3	14,403.8	1.5	1.3	40.0
Total			26	8.5	1,140.9	9.2	9,615.5	1.7	2.8	18.3
6	1	0.8	6	6.3	191.5	0.0	4,484.9	1.8	6.3	11.8
6	2	0.8	6	10.9	1,246.5	7.2	11,526.7	2.6	4.5	20.8
6	3	0.8	4	12.6	1,328.5	12.0	14,403.2	3.1	3.2	26.4
9	1	0.8	6	6.6	854.0	2.3	9,324.2	1.4	5.7	13.9
9	2	0.8	4	11.0	2,565.6	13.7	14,402.7	3.1	2.0	24.8
Total			26	9.1	1,128.0	6.1	10,278.4	2.3	5.0	18.6

ure 5c. However, the number of outsourced products is also higher in the static-static strategy (Figure 5d), particularly at higher levels of uncertainty.



(a) OF value for different number of scenarios

(b) OF value for different Uncertainty Levels

(c) Average inventory costs

(d) Average outsourced products

Figure 5: The comparison of Static-Static and Static-Dynamic strategies

5.3. Other Probability Distributions

To further demonstrate the applicability of the proposed algorithm, we extended the experiments conducted in the previous section to include stochastic demand generated from the Normal and Gamma distribution functions. The same algorithm was applied, and a similar Monte Carlo Sampling technique was employed to generate scenarios. It is worth noting that for consistency in the results, we utilized the same mean and standard deviation for all probability distribution functions (PDF). Specifically, we used the nominal demand as the mean value, and the standard deviation obtained from the Uniform distribution as the given standard deviation for both the Normal and Gamma distributions.

Furthermore, to handle the potential occurrence of negative values in the Normal distribution, we set the demand to zero in such cases. Moreover, to maintain integer demand values, we truncate the obtained numbers from the probability function. This approach ensures that the demand values remained integer throughout the analysis.

For these experiments, we employed identical instances featuring 100 scenarios and four distinct levels of demand uncertainty. The time limit for the PH-M algorithm remained consistent at two hours, and for all the BC algorithms, we set a four-hour time limit. The parameters of the heuristic algorithm remain unchanged, as previously detailed in Section 5.1. In Table 13, we present a summary of the results for the Normal and Gamma distributions. Notably, the average optimality gap for different uncertainty levels remains below 4% across all studied PDFs. However, this gap is higher for the Normal and Gamma distributions, primarily due to their higher EVPI^{UB}. The increased variability in demand values within the Normal and Gamma distributions explains this higher EVPI^{UB}.

Moreover, the average values of the VSS^{LB} underscore the importance of considering stochastic programming over mean-value problem formulations, particularly when demand uncertainty is high. Additionally, the total average value of ImpLB falls between 3% and 4% across all PDFs. It is worth noting that the proposed heuristic has an average CPU time ranging from 16 to 20 minutes, whereas solving the SPRP-FSR using the BC algorithm requires over 2.5 hours on average.

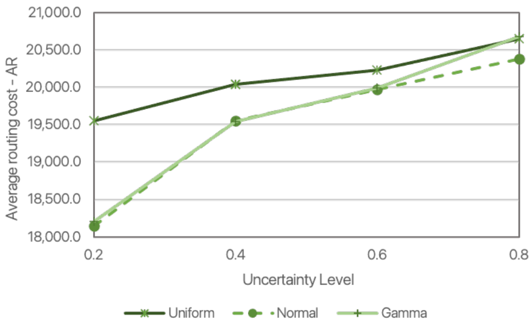
We present a comparative analysis of the average routing costs associated with three different routing algorithms: AR, FSR, and FSR(PO). The results are depicted in Figure 6, where instances exhibiting a difference of more than 20% for FSR across all cases have been excluded. Across all probability functions examined, we observe a clear trend of increased routing costs as the level of uncertainty rises. Notably, the costs are comparatively lower for scenarios following the Normal and Gamma distributions, particularly at lower uncertainty levels. This phenomenon can be attributed to the fact that the probability distribution in the Normal and Gamma cases is more concentrated around the mean value. Consequently, a larger proportion of scenarios fall within a narrow range around the mean, whereas in the Uniform distribution, the probability remains constant across the entire range. However, it is worth noting that the effect of this characteristic diminishes as ϵ increases, as evidenced in Figure 6a. Despite this observation, the routing cost of FSR(PO) consistently exceeds that of AR, providing evidence of the algorithm's effectiveness across all instances.

Figure 7 presents a comparison of three performance metrics: VSS^{LB}, VSS^{UB}, and EVPI^{UB} for different probability functions. As anticipated, when $\epsilon = 0.2$, scenarios generated by the Normal and Gamma distributions tend to cluster closely around the mean value. Consequently, the EV problem yields a high-quality first-stage solution, leading to lower VSS^{LB} values. However, as ϵ increases, the VSS^{LB} also increases. Figure 8 illustrates the objective function values for various probability distribution functions across different uncertainty levels. Regardless of the distribution, we consistently observe that the average objective function value is lower for AR, even after applying post-optimization to FSR. On average, the improvement achieved by utilizing AR instead of FSR amounts to 6.5%, 6.2%, and 7.8% for the Uniform, Normal, and Gamma distributions, respectively. If we consider the post-optimization on FSR, the improvement percentages become 5.3%, 5.2%, and 6.4% for the respective distributions.

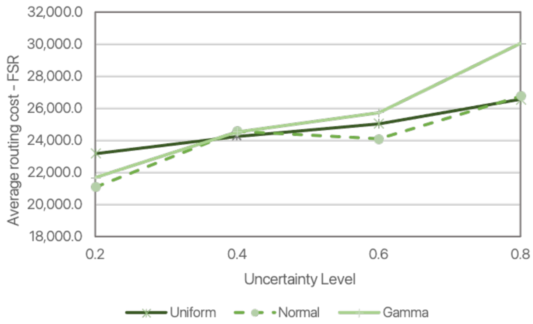
These results highlight the superior performance of AR over FSR in terms of achieving lower objective function values across all instances and also indicate the effectiveness and efficiency of the PH-M algorithm.

Table 13: Summary of the results of the Normal and Gamma distribution for different uncertainty levels on set $\mathscr{M}$ under the Static-Dynamic Strategy

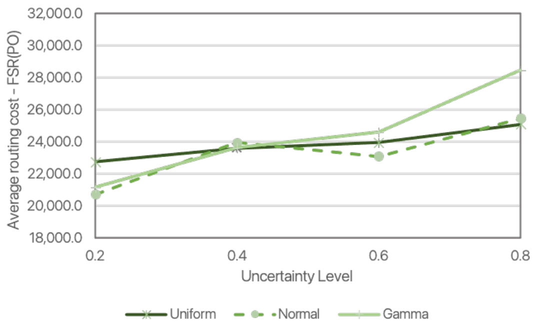
	ϵ	#Ins	SPRP-AR (PH-M)					SPRP-FSR (BC)				
			Gap (%)	CPU (secs)	VSS^{LB} (%)	VSS^{UB} (%)	EVPI^{UB} (%)	Gap (%)	CPU (secs)	PostOpt (%)	ImpLB (%)	ImpUB (%)
Normal	0.2	26	3.2	1,143.2	0.4	3.7	3.2	9.9	9,800.2	0.7	1.3	12.4
	0.4	26	3.3	1,063.4	6.3	9.3	3.3	13.7	10,078.6	1.0	2.0	17.3
	0.6	26	3.0	1,149.4	11.6	15.0	3.0	14.9	9,776.4	1.4	3.7	20.1
	0.8	26	5.3	1,008.3	16.6	21.7	5.3	6.7	8,793.8	2.3	5.7	16.0
Total		104	3.7	1,091.1	8.8	12.4	3.7	11.3	9,612.3	1.4	3.2	16.4
Gamma	0.2	26	3.1	1,146.2	0.5	3.6	3.1	10.5	9,994.5	0.8	1.3	12.8
	0.4	26	4.6	1,174.3	5.9	9.3	4.6	9.9	10,065.0	1.2	1.5	13.7
	0.6	26	3.4	1,233.1	11.8	15.6	3.4	9.3	9,202.8	1.7	4.6	16.1
	0.8	26	4.7	1,073.8	17.9	22.4	4.7	5.9	9,398.8	2.2	8.5	18.3
Total		104	3.9	1,156.9	9.0	12.7	3.9	8.9	9,665.2	1.5	4.0	15.2



(a) Avg. routing cost for AR

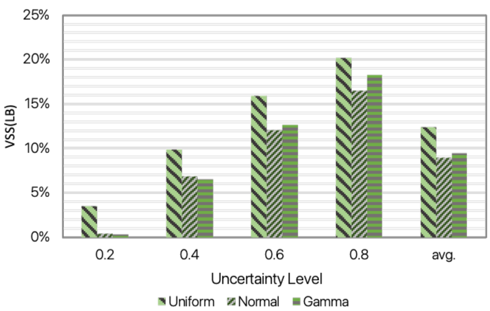


(b) Avg. routing cost for FSR

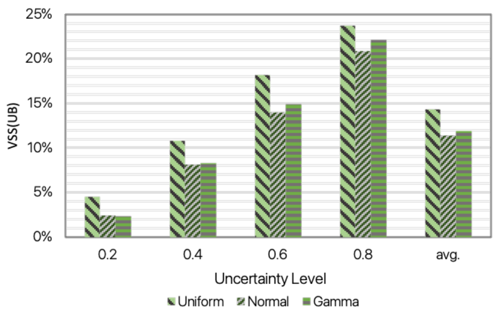


(c) Avg. routing cost for FSR(PO)

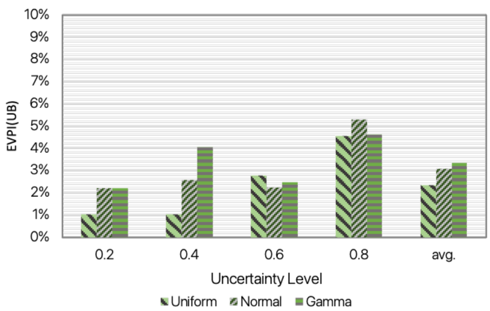
Figure 6: The average value of routing costs under the SD strategy with different ϵ for different PDFs



(a) Avg. VSS^{LB}

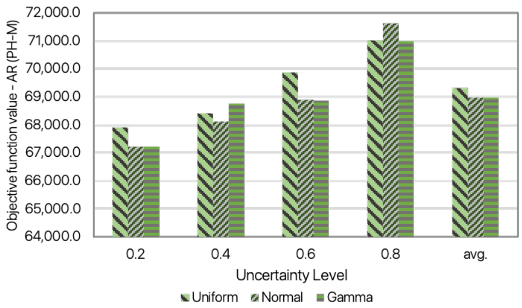


(b) Average VSS^{UB}

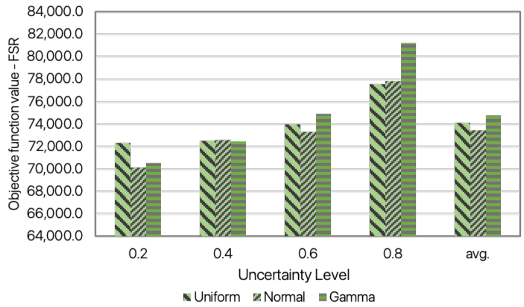


(c) Avg. EVPI^{UB}

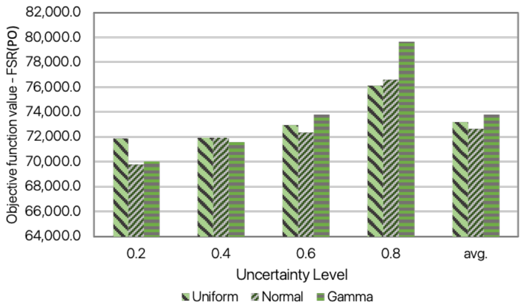
Figure 7: Comparison of relative VSS^{LB}, VSS^{UB}, and EVPI^{UB} under the SD strategy with different ϵ for different PDFs



(a) Avg. OF for AR



(b) Avg. OF for FSR



(c) Avg. OF for FSR(PO)

Figure 8: The average objective function values under the SD strategy with different ϵ for different PDFs

6. Conclusion.

This study addresses a significant research gap in the field of the SPRP by focusing on adaptive second-stage routing decisions. By introducing this novel concept in the stochastic PRP, the study enhances the flexibility of the problem, resulting in reduced routing costs and improved overall cost efficiency for the system. This increased flexibility provides valuable managerial advantages by enabling quick responses to unexpected demands. To tackle this complex problem, the study proposes a progressive hedging-based matheuristic algorithm. This algorithm is designed to generate high-quality solutions within a reasonable timeframe. The effectiveness of this approach is demonstrated through extensive computational experiments on 566 instances. Additionally, the algorithm is extended to incorporate the static-static strategy, which further stabilizes the production schedule but reduces system flexibility.

The study compares the results of the SPRP with adaptive routing to those obtained using fixed routes or fixed routes after applying a post-optimization procedure. The results show that incorporating adaptive routing in the stochastic PRP using the proposed algorithm, considering the static-dynamic strategy, leads to an average minimum improvement of 6.5%. This finding suggests that flexible routing can reduce not only routing costs but also inventory and outsourcing costs within the system. Furthermore, the proposed algorithm achieves these high-quality results with an average CPU time of less than 20 minutes, highlighting its efficiency compared to the long CPU time required by the BC algorithm. The study also analyzes the UB to the average expected value of perfect information ($EVPI^{UB}$) and the LB to the average minimum value of the stochastic solution (VSS^{LB}). The $EVPI^{UB}$ remains relatively low, barely exceeding 5%, indicating that the value of additional information is limited. On the other hand, the VSS^{LB} can reach as high as 12.4%, emphasizing the importance of considering uncertainty in the decision-making process.

Acknowledgment

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 Montreal 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.

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. doi:10.1287/trsc.2014.0523.

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. doi:10.1287/ijoc.2013.0550.

Adulyasak, Y., Cordeau, J.F., Jans, R., 2014b. Optimization-based adaptive large neighborhood search for the production routing problem. Transportation Science 48, 20–45. doi:10.1287/trsc.1120.0443.

Adulyasak, Y., Cordeau, J.F., Jans, R., 2015a. Benders decomposition for production routing under demand uncertainty. Operations Research 63, 851–867. doi:10.1287/opre.2015.1401.

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. doi:10.1016/j.cor.2014.01.011.

Agra, A., Requejo, C., Rodrigues, F., 2018. An adjustable sample average approximation algorithm for the stochastic production-inventory-routing problem. Networks 72, 5–24. doi:10.1002/net.21796.

Alvarez, A., Cordeau, J.F., Jans, R., Munari, P., Morabito, R., 2021. Inventory routing under stochastic supply and demand. Omega 102, 102304. doi:10.1016/j.omega.2020.102304.

Alvarez, A., Miranda, P., Rohmer, S., 2022. Production routing for perishable products. Omega 111, 102667. doi:10.1016/j.omega.2022.102667.

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., Laporte, G., Speranza, M.G., 2007. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. Transportation Science 41, 382–391. doi:10.1287/trsc.1060.0188.

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. doi:10.1016/j.cor.2011.03.002.

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. doi:10.1016/j.cor.2010.10.026.

Avci, M., Topaloglu Yildiz, S., 2020. A mathematical programming-based heuristic for the production routing problem with transshipments. Computers & Operations Research 123, 105042. doi:10.1016/j.cor.2020.105042.

Bard, J.F., Nananukul, N., 2009. The integrated production–inventory–distribution–routing problem. Journal of Scheduling 12, 257–280. doi:10.1007/s10951-008-0081-9.

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.

Bookbinder, J.H., Tan, J.Y., 1988. Strategies for the probabilistic lot-sizing problem with service-level constraints. Management Science 34, 1096–1108. doi:10.1287/mnsc.34.9.1096.

Boudia, M., Louly, M.A.O., Prins, C., 2008. Fast heuristics for a combined production planning and vehicle routing problem. Production Planning & Control 19, 85–96. doi:10.1080/09537280801893356.

Chandra, P., 1993. A dynamic distribution model with warehouse and customer replenishment requirements. Journal of the Operational Research Society 44. doi:10.1057/jors.1993.117.

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. doi:10.1287/ijoc.2018.0817.

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. doi:10.1016/j.ejor.2019.10.007.

Coelho, L.C., Laporte, G., 2014. Improved solutions for inventory-routing problems through valid inequalities and input ordering. International Journal of Production Economics 155, 391–397. doi:10.1016/j.ijpe.2013.11.019.

Crainic, T.G., Fu, X., Gendreau, M., Rei, W., Wallace, S.W., 2011. Progressive hedging-based metaheuristics for stochastic network design. Networks 58, 114–124. doi:10.1002/net.20456.

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 , 106434doi:https://doi.org/10.1016/j.cor.2023.106434.

Haugen, K.K., Løkketangen, A., Woodruff, D.L., 2001. Progressive hedging as a meta-heuristic applied to stochastic lot-sizing. European Journal of Operational Research 132, 116–122. doi:10.1016/S0377-2217(00)00116-8.

Jans, R., 2009. Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints. INFORMS Journal on Computing 21, 123–136. doi:10.1287/ijoc.1080.0283.

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. doi:10.1016/j.cie.2015.07.003.

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. doi:10.1016/j.ejor.2018.07.018.

Løkketangen, A., Woodruff, D.L., 1996. Progressive hedging and tabu search applied to mixed integer (0, 1) multistage stochastic programming. Journal of Heuristics 2, 111–128. doi:10.1007/BF00247208.

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. doi:10.1016/j.ejor.2021.05.037.

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. doi:10.1016/j.cor.2018.05.004.

Mousavi, R., Bashiri, M., Nikzad, E., 2022. Stochastic production routing problem for perishable products: Modeling and a solution algorithm. Computers & Operations Research 142, 105725. doi:10.1016/j.cor.2022.105725.

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. doi:10.1016/j.omega.2018.07.006.

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. doi:10.1016/j.tre.2018.01.009.

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. doi:10.1016/j.omega.2016.06.001.

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. doi:10.1016/j.omega.2018.01.006.

Rockafellar, R.T., Wets, R.J.B., 1991. Scenarios and policy aggregation in optimization under uncertainty. Mathematics of Operations Research 16, 119–147. doi:10.1287/moor.16.1.119.

Russell, R.A., 2017. Mathematical programming heuristics for the production routing problem. International Journal of Production Economics 193, 40–49. doi:10.1016/j.ijpe.2017.06.033.

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 doi:10.1287/trsc.2022.0261.

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. doi:10.1016/j.ejor.2020.05.054.

Shuang, Y., Diabat, A., Liao, Y., 2019. A stochastic reverse logistics production routing model with emissions control policy selection. International Journal of Production Economics 213, 201–216. doi:10.1016/j.ijpe.2019.03.006.

Solyalı, O., Süral, H., 2017. A multi-phase heuristic for the production routing problem. Computers & Operations Research 87, 114–124. doi:10.1016/j.cor.2017.06.007.

Solyalı, O., Süral, H., 2022. An effective matheuristic for the multivehicle inventory routing problem. Transportation Science 56, 1044–1057. doi:10.1287/trsc.2021.1123.

Tempelmeier, H., Herpers, S., 2010. Abc β–a heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint. International Journal of Production Research 48, 5181–5193. doi:10.1080/00207540903179782.

Tunc, H., Kilic, O.A., Tarim, S.A., Eksioglu, B., 2013. A simple approach for assessing the cost of system nervousness. International Journal of Production Economics 141, 619–625. doi:10.1016/j.ijpe.2012.09.022.

Zhang, Y., Alshraideh, H., Diabat, A., 2018. A stochastic reverse logistics production routing model with environmental considerations. Annals of Operations Research 271, 1023–1044. doi:10.1007/s10479-018-3045-2.