Submitted to *Transportation Science*
manuscript (Please, provide the manuscript number!)

Stochastic Production-Distribution Planning with Transportation Mode-Dependent Lead Times

Dorian Dumez
Department of Logistics and Operations Management, HEC Montréal, H3T 2A7, Canada, dorian.dumez@hec.ca

Jean-François Cordeau
Department of Logistics and Operations Management, HEC Montréal, H3T 2A7, Canada, jean-francois.cordeau@hec.ca

Raf Jans
Department of Logistics and Operations Management, HEC Montréal, H3T 2A7, Canada, raf.jans@hec.ca

Plants and distribution centers often deliver their products to numerous customers spread over a vast territory and can thus rely on a combination of road, air, rail, and maritime transportation to make their deliveries. These means of transportation have different costs but also different lead times, and there is thus a fundamental trade-off to be considered: a shorter lead time typically comes at a higher cost but offers more flexibility to react quickly to changes in demand. Hence, the plant faces the complex problem of making simultaneous production and transportation decisions, which include the selection of the transportation modes to use for shipping products to different customers. The objective is often to minimize the expected cost of production, transportation, inventory, and lost sales. We consider this problem in a setting with a discrete and finite time horizon during which customers face a stochastic demand. In each time period, the plant has to make production and transportation decisions before the demand is revealed. We solve the resulting multi-stage problem approximately in a rolling horizon framework that relies on a static-dynamic representation of the problem. To efficiently solve this static-dynamic problem, we present a tree-search heuristic based on Anytime Column Search and Limited Discrepancy Search. For the node selection strategy, we develop a guide heuristic that aggregates all the considered scenarios to quickly improve the current solution with respect to the set-up decisions of the current tree-search node. The tree-search framework and the guide heuristic are evaluated on medium-size instances and compared to CPLEX. The guide heuristic proves to be able to select solutions with an average gap below 0.1% compared to the best one while being 316 times faster than CPLEX on average. Overall, the efficiency of the proposed tree-search heuristic method is validated by the fact that CPLEX never finds a better solution than the one found by the proposed method, even with ten times more computing time.

Key words: Production-distribution planning, Multi-stage stochastic programming, Tree-search heuristic

1. Introduction

Production plants and distribution centers often forward their products to numerous customers spread over a large territory, and can thus use multiple carriers to perform their distribution. For example, an industrial company could rely on road, air, rail, and maritime shipping modes to deliver its products. These transportation modes typically have different costs, but also different delivery lead times. Consequently, many services with various prices and delivery times can be combined to achieve the best trade-off between delivery performance and cost. In many markets, demand is uncertain and the short lead times offered by faster transportation modes give the distributor more flexibility to adjust its production level and react quickly to changes in the demand. However, cheaper and slower modes are better tailored to transport large amounts of products. Hence, a judicious combination of these different modes of transportation can reduce transportation, inventory holding and stockout costs for the company.

In this paper, we consider a multi-stage stochastic decision problem for a production plant and several customers that face an uncertain demand and are connected to the plant by multiple transportation modes. The objective is to minimize the total expected cost of the supply chain including production, transportation, and inventory holding costs as well as the cost of unsatisfied demand. It extends the well-known *One-Warehouse Multi-Retailer Problem (OWMRP)* by taking into account stochastic demand and lead times. We call this problem the *Stochastic Production-Distribution Problem with Lead times (SPDPL)*. To the best of our knowledge, this is the first time that a stochastic production-distribution planning problem with multiple transportation modes and lead times is studied. Note that the considered problem can be indistinctly applied to a production plant supplying national warehouses, or to a distribution center delivering products to retailers or customers. In the remainder of the paper, for the sake of consistency, we will refer to a *plant* that *produces* goods to supply *customers*.

To model the demand uncertainty over multiple periods, we use the concept of scenario trees. In such a tree, each node represents a possible realization of the demand in a given time period, each arc represents a probabilistic transition, and a path from the root to a leaf node is a scenario. When the demand at the customers is continuous, it is not possible to enumerate all possible scenarios to solve the multi-stage problem exactly. Consequently, in our implementation, scenarios are sampled with a Quasi Monte-Carlo method (Niederreiter 1978).

To mitigate the impact of the length of the planning horizon, a rolling horizon framework is used. For each time window in the rolling horizon, we solve the static-dynamic version of the SPDPL (i.e., binary variables are common to all the considered demand scenarios). However, because we consider possibly long lead times, the width of the window considered in the rolling horizon needs to be large enough to accommodate all the lead times. In order to solve sub-problems with a long enough window, we present a tree-search heuristic based on Anytime Column Search (ACS, Vadlamudi et al. (2012)) and Limited Discrepancy Search (LDS, Harvey and Ginsberg (1995)). For the node selection strategy, we develop a flow-based guide heuristic that aggregates all the considered scenarios to quickly improve the current solution with respect to the set-up decisions of the current tree-search node. Thus, the proposed tree-search heuristic re-optimizes the solution obtained for the previous time period by navigating the space of related solutions w.r.t. the set-up decisions, and by guiding the exploration using a flow-based cost-to-go estimation heuristic.

To summarize, the methodological contributions of this article are threefold: (1) we propose a heuristic that scales well with the number of customers and periods, something that mixed integer programming (MIP) formulations and dynamic programming (DP) methods have trouble with; (2) this heuristic can handle any probability distribution of the demand, which is a challenge for most policies; and (3) the method allows for any number of carriers with any lead time, while MIP solvers, DP methods, and reordering-point policies have trouble with long lead times (the first two because of the explosion in the number of nodes and states, and the latter because of the complexity of the involved analytical formulations). Thus, this study shows the relevance of considering tree-search methods guided by heuristics to solve stochastic optimization problems. In addition, we make three modelling contributions: (4) we propose a MIP model for the multi-stage SPDPL; (5) we compute the value of stochastic solutions for this problem; and (6) we derive managerial insights on synthetic instances.

The rest of the paper is structured as follows. Section 2 discusses the related literature. Section 3 formalizes the problem and provides a MIP formulation for it. Section 4 presents the solution methods, one directly based on the MIP formulation and one based on the tree-search heuristic. Finally, Section 5 reports the results of computational experiments.

2. Review of related problems

The SPDPL considers production and transportation planning in a network composed of one plant and multiple customers over a discrete and finite time horizon. Thus, we will first review studies considering similar contexts. Because one major feature of the present work is the management of inventories with lead times for their resupply, the second section of the literature review will focus on this aspect. Finally, some other closely related studies will be mentioned, and the contributions of this article with respect to the existing literature will be explained in more detail.

2.1. Production-distribution in a discrete time horizon

The *One-Warehouse Multi-Retailer Problem* (OWMRP, Roundy (1985)) consists in planning the production of a product at a plant, as well as the shipping of the product to multiple retailers to satisfy the customer demand. In the OWMRP, the shipping is restricted to back and forth routes between the plant and each retailer, generally representing the use of third-party carriers managing the routing externally. For deterministic production-distribution problems, Solyali and Sural (2012), Cunha and Melo (2016), Gruson et al. (2019), and Cunha and Melo (2021) compare many MIP formulations that are solved by general-purpose solvers.

For a summary of the early literature on the *Stochastic OWMRP (SOWMRP)*, we refer to Aliev et al. (2007). They develop a fuzzy approach to a problem with multiple production plants, multiple distribution centers, and multiple customers. A genetic algorithm is developed to solve instances with up to two plants, two distribution centers, and two customers over two time periods. Gruson, Cordeau, and Jans (2021) describe a variant of the two-stage SOWMRP in a three-level fixed distribution structure: plant, warehouses, and retailers. Goods have to be produced and sent to retailers, facing stochastic demand, through their pre-assigned warehouse. A Benders decomposition method is developed and it can solve instances with up to 200 retailers and 30 time periods within six hours. Chavarro et al. (2021) design a method to compute an ordering policy for a discrete-period infinite-horizon SOWMRP. It was tested on instances with up to 100 retailers facing a stationary, normally distributed demand.

To summarize, these articles propose various approaches to manage production and multiple inventories, but, compared to the objectives of this study, they are limited by the size of the instances they can solve in reasonable time, or by their flexibility to consider other settings (e.g., demand probability distribution, transportation network, lead times).

The *Production Routing Problem (PRP)* is an extension of the OWMRP which involves vehicle routing decisions instead of only direct deliveries from the plant to the retailers. It is also an extension of the *Inventory Routing Problem (IRP)*, which is a simpler problem that does not consider production. We refer the readers interested in the deterministic PRP to Adulyasak, Cordeau, and Jans (2015b) and to Hrabec, Hvattum, and Hoff (2022), and those interested in the IRP to Coelho, Cordeau, and Laporte (2014).

An extension of the PRP considering stochastic demand (the SPRP) was first introduced in Adulyasak, Cordeau, and Jans (2015a). A Benders-based branch-and-cut method is presented to solve the two-stage problem, and is embedded in a rolling horizon framework for the multi-stage problem. Tests were conducted on instances with up to 30 customers and six time periods. Agra, Requejo, and Rodrigues (2018) consider the two-stage SPRP with a known distribution of the demand. A hybrid sample average approximation with a relax-and-fix matheuristic method is developed, and is tested on instances with up to 80 customers and ten periods. Bertazzi, Bosco, and Laganà (2015) consider a stochastic IRP with transportation procurement and direct shipments. For the case with an order-up-to level (OU) policy, an exact and a heuristic stochastic dynamic programming methods are developed. The methods are tested on instances with up to 20 retailers.

Finally, the *Carriers' Selection and Shipments' Assignment Problem (CSSAP)* also addresses the resupply of distribution centers from warehouses by relying on multiple carriers. We refer the interested readers to Boujemaa, Rekik, and Hajji (2022), who solve a CSSAP with unknown demand using a stochastic dual dynamic programming method. The method was tested on instances with up to 12 periods, 10 warehouses, 15 distribution centers, four products and four carriers.

Despite the fact that these articles present advanced techniques for the optimization of production and inventory management over multiple periods, most of their contributions lie in the computation of the routing part of the considered problems. Consequently, they do not tackle the challenges we are facing with the transportation of products in a multi-stage problem. In particular, none of these articles considers transportation lead times.

2.2. Inventory management with lead times

The dual-sourcing inventory management problem consists in maintaining enough inventory at a warehouse to meet the demand by choosing when to order a given product from multiple vendors with possibly different lead times and costs. The transportation networks considered in dual sourcing problems are the reverse of the ones considered in the OWMRP. Indeed, the inventory at the plant has to be managed with multiple modes for incoming supply arcs and one mode for outgoing delivery arcs, while in the SPDPL, there is only one incoming production arc and multiple modes for outgoing transportation arcs. Both the deterministic and the stochastic versions of this problem have been studied since the early days of operations research. We refer the interested readers to Engebrethsen and Dauzère-Pérès (2019) and to Svoboda, Minner, and Yao (2021).

Some papers have proposed ordering policies for discrete time periods and for various probability distributions of the demand and various constraints on the lead time, e.g., Kiesmüller, de Kok, and Fransoo (2005), Sun and Van Mieghem (2019), Boulaksil et al. (2021), Boute et al. (2021). In addition, Zhou et al. (2011) consider the dual-sourcing problem with no lead time and a single mode, but, to the best of our knowledge, it presents the first dynamic programming method developed to handle stochastic demand and multiple suppliers.

Lot-sizing problems consist in choosing in which periods to start production, and how much to make of the considered product, in order to meet the demand. We refer readers interested in stochastic lot sizing to Brahimi et al. (2017). In particular, Alp, Erkip, and Güllü (2003) present a lot-sizing problem with deterministic demand but stochastic lead times. A dynamic-programming solution method is developed to solve instances with up to 80 time periods.

In these two streams of literature, the impacts of lead times and of multiple modes have been considered, but their impact on the management of multiple distinct inventories facing different demands remains to be studied.

2.3. Closely related problems

Bertazzi, Moezi, and Maggioni (2021) consider a system with one plant and one customer facing deterministic demand, and whose inventory is monitored by the former. The products can be transferred between the two locations via multiple carriers, either with full container loads or less-than-container loads, with different prices and lead times. The problem then consists in finding an optimal production rate and replenishment frequency for each carrier. A MIP model is described and solved with up to four transport options (two carriers, each with full and less-than-container load). This article is insightful with respect to the combination of multiple transportation modes for the management of multiple inventories. Unfortunately, the proposed model cannot be easily generalized to a more flexible shipment management setting.

Farhangi (2021) considers a network with multiple plants, multiple warehouses, and multiple retailers, with transportation lead times between them. Considering multiple warehouses between the plants and the retailers creates several possible paths for the products, similar to the presence of multiple transportation modes. The demand is unknown, with the only information being the upper and lower bounds for each retailer. Two models are devised: optimizing with the average demand or maximizing the minimum satisfied demand. A MIP-based fix-and-optimize method is developed. Instances with up to four plants, five warehouses, 10 retailers and 15 time periods were solved. However, in the case considered by this article, there are no trade-offs between cost and lead time as they are proportional across all plant-to-retailer paths. In addition, the former model implicitly assumes that the probability distribution of the demand is symmetric, while the latter is overly conservative for most distributions. Finally, we refer readers interested in advanced methods for the three-echelon production-distribution problem to Wu et al. (2022), who also identify the uncertainty of customer demand as an important and challenging extension.

2.4. Contributions with respect to the literature

Table 1 provides a summary of the main articles discussed above. For each article, columns 2 to 4 indicate whether production, transportation, and inventory decisions are taken into account, respectively. Column 5 specifies the number of transportation modes taken into account, and Column 6 gives their lead times. Here, "Const" indicates that the lead time can take any value, but should be constant over time. Otherwise, if the proposed method depends on specific values of the lead time, it is given. Column 7 specifies the probability distribution of the demand that can be managed by the proposed method. The word "Stat" means that the distribution is stationary over time and specific probability distributions are indicated ($\mathcal{B}$ for binomial, $Pois$ for Poisson, $\mathcal{N}$ for normal, and $\mathcal{U}$ for uniform). Finally, Column 8 summarizes the solution method.

This table highlights the fact that, to the best of our knowledge, no previously developed method is able to optimize production and multiple inventories, linked by multiple carriers with lead times, with various probability distributions of the demand using only a reasonable amount of computing time.

	Source	Production	Transport	Inventory	#Modes	Lead times	Distribution	Method
OWMRP	Aliev et al. (2007)	✓	✓	✓	1	0	Any	Fuzzy GA
	Gruson, Cordeau, and Jans (2021)	✓	✓	✓	1	0	Any	2-stage MIP
	Chavarro et al. (2021)	✓	✓	✓	1	0	Stat $\mathcal{N}$	Ratio policy
	Farhangi (2021)	✓	✓	✓	1	Const	Any	MIP on avg. demand
PRP	Adulyasak, Cordeau, and Jans (2015a)	✓	✓	✓	1	0	Any	Benders B&C
	Bertazzi, Bosco, and Laganà (2015)		✓	✓	Any	0	Stat $\mathcal{B}$	OU policy
	Agra, Requejo, and Rodrigues (2018)	✓	✓	✓	1	0	Any	SAA
	Boujemaa, Rekik, and Hajji (2022)		✓	✓	Any	0	Any	SDDP
Dual sourcing	Kiesmüller, de Kok, and Fransoo (2005)		✓	✓	2	Const	Stat	OU policy
	Zhou et al. (2011)		✓	✓	1	0	Any discrete	DP
	Sun and Van Mieghem (2019)		✓	✓	2	Const	Any	Dual index policy
	Boulaksil et al. (2021)		✓	✓	2	0 and 1	Any i.i.d.	Ordering policies
	Boute et al. (2021)		✓	✓	2	1 and $\mathbb{N}^*$	i.i.d. $\mathcal{N}$	Linear ordering policy

Table 1 Summary of related problems studied in the literature

3. The Stochastic Production-Distribution Problem with Lead times

This section is divided into two parts: Section 3.1 first presents the SPDPL and Section 3.2 then describes a multi-stage MIP formulation of the problem.

3.1. Problem description

The *Stochastic Production-Distribution Problem with Lead times (SPDPL)* considers a single production plant, a single product, multiple customers, and multiple carriers, over multiple time periods. It consists of managing the inventory and the production of the product at the plant and at the customers, which are facing stochastic demands. To transport the products, the company relies on multiple carriers, typically third-party companies, often called logistics services providers. Figure 1 depicts an example of an SPDPL instance with (from left to right) the plant, three carriers, and two customers.

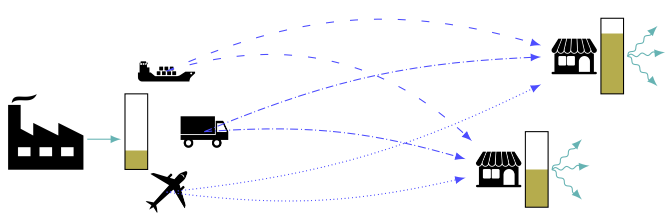

Figure 1 Network representation of a problem instance with one plant and three customers

In this article, we consider that the production plant and the customers belong to the same company, or that they are bound by a vendor management inventory (VMI) or an equivalent contract. Thus, the objective is to minimize the expected total cost at the two levels of the supply chain over the considered planning horizon. This includes the production costs, the inventory holding costs, the transportation costs, and the lost sales penalties. We consider a discrete time horizon, composed of $|T|$ periods $t \in T = \{1, ..., |T|\}$. During each of these periods, some decisions must be made first before the demand becomes known for the considered period. These decisions are made at the plant, and they include: production setups, production quantities, and quantities sent via each carrier to each customer. The recourse decisions are computed after the demand is revealed, and they correspond to the resulting lost sales and inventory levels.

Let us represent the plant by index 0. At period $t \in T$, it has a production setup cost of s_t, a unit production cost p_t, and a production capacity Q_t. During all time periods, the unit holding cost is h_0. We do not consider a storage capacity at the plant. The plant has an initial inventory that we denote by i_0^0.

The products are destined to the customers. During each time period $t \in T$, each customer $r \in R$ has an expected demand d_r^t. The unit holding cost per period at customer $r \in R$ is h_r. We do not consider storage capacities at the customers. Customer $r \in R$ has an initial inventory level i_r^0. If, during a time period, the demand is not met, a per-unit lost sales penalty a is incurred.

To transport the products from the plant to the customers, the services of the carriers in set F can be used. Only direct deliveries are considered. The unit cost to transport products from the plant to customer $r \in R$, in period $t \in T$, with carrier $f \in F$ is u_{rf}^t, and the fixed cost is w_{rf}^t. Without loss of generality, we assume that the holding cost during transportation is included in the unit transportation cost so that it can encompass the opportunity cost. The lead time is v_{rf}^t, i.e., the products shipped in period t will be available at the customer to satisfy its demand starting from period $t + v_{rf}^t$.

We model the stochasticity of the demand using a scenario tree composed of the nodes in set $\mathcal{N}$. Each node $n \in \mathcal{N}$ represents a realization of the demand at each customer during one time period. The demand of customer $r \in R$ in node $n \in \mathcal{N}$ is d_r^n. The time period at which the node $n \in \mathcal{N}$ takes place is denoted by $\mathcal{T}(n)$, and the set of nodes taking place at period $t \in T$ is denoted by $\mathcal{N}(t)$. If $\mathcal{T}(n) > 1$, the father of the node $n \in \mathcal{N}$ is denoted by $\mathcal{A}(n)$. More generally, its unique ancestor at period $t < \mathcal{T}(n)$ is $\mathcal{A}(n,t)$. In the following equations, to simplify the notation, we let $\mathcal{A}(n, \mathcal{T}(n)) = n$. If $\mathcal{T}(n) < |T|$, the set of descendants of the node $n \in \mathcal{N}$ is denoted by $\mathcal{D}(n)$. More generally, its set of descendants in period $t > \mathcal{T}(n)$ is $\mathcal{D}(n,t)$. Finally, let us denote the unconditional probability of realization of node $n \in \mathcal{N}$ by π^n.

In the presence of uncertainty, let us clarify the order in which decisions are made and at what moment demand is revealed during each time period. This sequence is represented in Figure 2 (with a transportation lead time of one) and is the following:

1. Items are produced and are available are the plant.
2. Items are shipped from the plant.

3. Shipped quantities arrive at the customers according to their lead times. These incoming products are available to satisfy the demand.
4. Demand is revealed for all the customers.
5. Inventory and lost sales are computed.

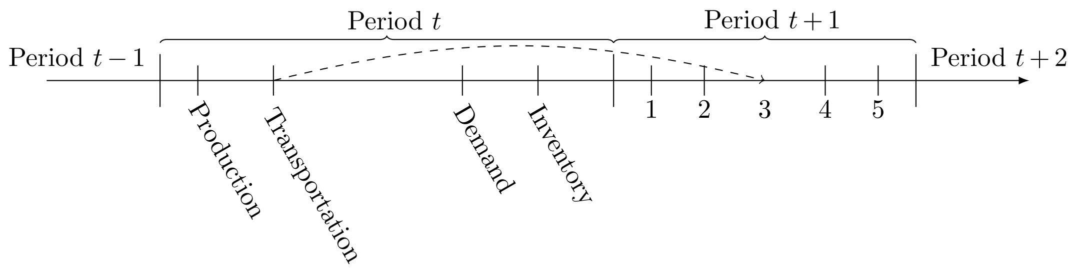

Figure 2 Decomposition of the time periods

3.2. Multi-stage dynamic model

The SPDPL is a multi-stage dynamic problem. The modifications to get its static-dynamic counterpart, which will be used in our solution method, are described in Section 4.

In the presence of uncertainty, to model the fact that production and transportation decisions must be made before the demand of the corresponding period becomes known, the corresponding variables must be shifted one period backward in time. Indeed, since production and transportation decisions are made at the beginning of a period, before demand is revealed, this is equivalent to making these decisions at the end of the previous period (after the calculation of inventory and lost sales). That is to say, the production and transportation decisions made in node n refer to decisions made at the beginning of time period $\mathcal{T}(n)+1$. Consequently, we add a virtual root node 0 to represent the decisions for the first period. It has no demand and takes place at the virtual period 0. Let us define $\mathcal{N}^* = \bigcup_{t\in[1;|T|-1]} \mathcal{N}(t) \cup \{0\}$.

The variables used in our formulation are the following. Let y^n be a binary variable taking value 1 if a production setup takes place in node $n \in \mathcal{N}^*$, and 0 otherwise; $x^n \geqslant 0$ be the quantity produced in node $n \in \mathcal{N}^*$; z^n_{rf} be a binary variable taking value 1 if a transportation setup takes place in node $n \in \mathcal{N}^*$ to customer $r \in R$ via carrier $f \in F$, and 0 otherwise; $\theta^n_{rf} \geqslant 0$ be the quantity sent to customer $r \in R$ via carrier $f \in F$ in node $n \in \mathcal{N}^*$; $i^n_0 \geqslant 0$ be the quantity stored at the plant in node $n \in \mathcal{N}$; $i^n_r \geqslant 0$ be the quantity stored at customer $r \in R$ in node $n \in \mathcal{N}$; and $l^n_r \geqslant 0$ be the lost sales quantity at retailer $r \in R$ in node $n \in \mathcal{N}$. Remember that the production and transportation decisions (y^n, x^n, z^n_{rf}, θ^n_{rf}) made in node n refer to decisions applied at the beginning of time period $\mathcal{T}(n)+1$. Using this notation, the SPDPL can be formulated as follows:

Model 1: SPDPL dynamic multi-stage model

$$\begin{aligned}
\texttt{Min} \ & \sum_{t\in T}\sum_{n\in\mathcal{N}(t-1)} \pi^n \left(s_t y^n + p_t x^n + \sum_{r\in R}\sum_{f\in F} \left(u^t_{rf}\theta^n_{rf} + w^t_{rf} z^n_{rf}\right)\right) && (1)\\
& + \sum_{t\in T}\sum_{n\in\mathcal{N}(t)} \pi^n \left(h_0 i^n_0 + \sum_{r\in R} (h_r i^n_r + a.l^n_r)\right) && \\
\texttt{s.t} \ & x^n \leqslant Q_{\mathcal{T}(n)+1} y^n && \forall n\in\mathcal{N}^* && (2)\\
& \theta^n_{rf} \leqslant M z^n_{rf} && \forall n\in\mathcal{N}^*, r\in R, f\in F && (3)\\
& i^n_0 = i^{\mathcal{A}(n)}_0 + x^{\mathcal{A}(n)} - \sum_{r\in R}\sum_{f\in F}\theta^{\mathcal{A}(n)}_{rf} && \forall n\in\mathcal{N} && (4)\\
& i^{\mathcal{A}(n)}_r + \sum_{f\in F}\sum_{t=1}^{\mathcal{T}(n)} \delta_{\{t+v^t_{rf}=\mathcal{T}(n)\}}\theta^{\mathcal{A}(n,t-1)}_{rf} = d^n_r + i^n_r - l^n_r && \forall n\in\mathcal{N}, r\in R && (5)\\
& y^n \in \mathbb{B} && \forall n\in\mathcal{N}^* && (6)\\
& x^n \geqslant 0 && \forall n\in\mathcal{N}^* && (7)\\
& z^n_{rf} \in \mathbb{B} && \forall n\in\mathcal{N}^*, r\in R, f\in F && (8)\\
& \theta^n_{rf} \geqslant 0 && \forall n\in\mathcal{N}^*, r\in R, f\in F && (9)\\
& i^n_0 \geqslant 0 && \forall n\in\mathcal{N} && (10)\\
& i^n_r \geqslant 0 && \forall n\in\mathcal{N}, r\in R && (11)\\
& l^n_r \geqslant 0 && \forall n\in\mathcal{N}, r\in R. && (12)
\end{aligned}$$

The objective (1) is the probability-weighted sum of the production costs, transportation costs, holding costs, and lost-sales penalties. Constraints (2) limit the production in each node, either to 0 if there was no production setup, otherwise to the production capacity $Q_{\mathcal{T}(n)+1}$. Constraints (3) limit the amount transported to each customer, in each node, with each carrier: either to 0 or to a large number M depending on the set-up. Constraints (4) compute the inventory at the production plant. Finally, constraints (5) compute the inventory at the customers. The parameter $\delta_{\{t+v^t_{rf}=\mathcal{T}(n)\}}$ is equal to 1 if the subscripted condition is verified, and to 0 otherwise. The sum of the transported quantities is written as such because it is possible that products sent in different periods arrive at the same time since we allow lead times to vary depending on the shipping period.

Constraints (2) can be tightened, especially for the last periods, by taking the maximum total remaining demand into account. To do so, $Q_{\mathcal{T}(n)+1}$ should be replaced by:

$$\min\left\{ Q_{\mathcal{T}(n)+1}, \max_{n'\in\mathcal{D}(n,|T|)} \left\{ \sum_{t'=\mathcal{T}(n)+1}^{|T|} \sum_{r\in R} d^{\mathcal{A}(n',t')}_r \right\}\right\}. \quad (13)$$

Similarly, constraints (3) can be tightened by replacing M with:

$$\max_{n' \in \mathcal{D}(n,|T|)} \left\{ \sum_{t'=\mathcal{T}(n)+1+v_{rf}^{\mathcal{T}(n)+1}}^{|T|} d_r^{\mathcal{A}(n',t')} \right\}. \tag{14}$$

4. Solution method

The proposed SPDPL is a multi-stage dynamic problem. However, since it is intractable even for instances of modest size, we develop a rolling horizon method that relies on a static-dynamic approximation. For each time window of the rolling horizon, a scenario tree is constructed to sample possible outcomes in the periods to come for building a SPDPL sub-problem. The decisions made for the first considered period are then applied for the current period, before moving to the next one after the demand is revealed, and repeating the process. The developed rolling horizon has a step size of one period. The SPDPL model which is solved in the rolling horizon is a restricted version, imposing that the production and the transportation setups must be the same for all scenarios at a given time period. This is the static-dynamic approximation.

To get the static-dynamic counterpart of Model 1 (see Model 3, in Appendix A), for all $n \in \mathcal{N}^*$, one must replace all occurrences of the y^n production setup variables by $y^{\mathcal{T}(n)}$, and all occurrences of the z_{rf}^n transport setup variables by $z_{rf}^{\mathcal{T}(n)}$. In addition, let us denote the subset of time periods taken into account for the sub-problem during iteration $\tilde{t} \in T$ of the rolling horizon by $T_{\tilde{t}} \subseteq T$.

Despite the abundant literature on stochastic programming, tackling realistic-size production-distribution problems remains challenging (Fahimnia et al. 2013). To bridge this gap, simulation-based heuristics have been developed (Juan et al. 2015), and Monte-Carlo sampling has been extensively used (Homem-de Mello and Bayraksan 2014). The advantage of representing uncertainty with scenario trees sampled with Monte-Carlo is that is can approximate any complex probability distribution as well as various correlations (temporal auto-correlation, inter-customers correlation, etc). In this spirit, the updated review of stochastic optimization presented by Kaut (2021) suggest working with scenario trees as they can represent historical data by using machine learning, albeit doing so accurately is not trivial. To generate sampling scenario trees, we use a *Randomized Quasi Monte-Carlo (RQMC)* procedure. RQMC is described and motivated for stochastic optimization in Homem-de Mello and Bayraksan (2014): compared to crude Monte-Carlo sampling, it guarantees a good representation of the considered distribution with a limited sampling. As underlined by Thevenin, Adulyasak, and Cordeau (2022) it is essential to have a large sampling for the first periods because the corresponding decisions will be fixed afterward.

In addition, in the SPDPL, it is possible that the products will be transported with low-cost but slower modes. Because these carriers have long lead times, replenishment cycles may also be long, and the rolling horizon time windows must be longer than for most similar problems. Consequently, despite the simplification, the restricted problem grows with the size of the scenario tree as well as the number of customers, carriers, and simulated periods. Therefore, general-purpose solvers are impractical for this task. Alternatively, we chose to develop a tree-search-based heuristic to manage the binary variables, i.e., production and transportation setups. To guide the exploration, we propose a cost-to-go estimation, flow-based sub-problem to quickly adapt the quantity of goods produced and delivered w.r.t. the considered decisions.

The proposed solution methods will be presented next. First, Section 4.1 presents the tree-search method developed to solve the static-dynamic sub-problem. Second, Section 4.2 details the flow subproblem used to guide the search. Finally, Section 4.3 provides a summary of the steps of the proposed method.

4.1. Tree-search method

In this section, we describe the arborescent search scheme used to navigate the space $\{0,1\}^{|T_{\tilde{t}}|\times(1+|R|\times|F|)}$ of the binary variables representing the production setups and the transportation setups during the time window considered in the current iteration of the rolling horizon. To identify promising nodes during the tree search, a guide heuristic is used to adjust the production quantities, transported quantities, and stored quantities. This cost-to-go estimation guide heuristic will be presented in Section 4.2.

A leaf of the proposed tree associates a value to each binary variable of Model 3. An internal node encodes a partial solution, giving a value to only a subset of the binary variables. A branch in the tree represents the two possible choices of values for a binary variable not fixed in the father's node partial solution. In both cases, the value of the continuous variables can be either determined by the proposed guide heuristic or set optimally by using the simplex algorithm. Consequently, the goal of the tree search is to select a leaf such that the resulting solution, including the continuous variables, will have the lowest possible cost. The choices made in this solution for its first considered period will then be used by the rolling horizon.

Both the tree search and the guide heuristic need an initial solution, i.e., values for both binary and continuous variables. For the binary variables, we use the one from the previous step $\tilde{t}-1$ of the rolling horizon, or the one obtained by solving the deterministic model on the expected demands if $\tilde{t}=1$. For the continuous variables, they can be adjusted to the newly generated sampling scenario tree by solving Model 3 with the binary variables fixed to the considered decisions. In addition, at the end of the tree search, the aforementioned LP is solved to optimally adjust the continuous variables of the newly selected solution.

Limited Discrepancy Search (LDS, Harvey and Ginsberg (1995)) is a way to limit the number of nodes created during a tree search. LDS takes as input a reference solution. Then, during the search, only a limited number of differences D^{lds} to this reference solution are allowed. Generally, the root node has a label D^{lds}, its first descendant is the choice corresponding to the one made in the reference solution, so this descendant has the same label. The next descendant has a difference of 1 w.r.t. the reference solution, so it has a label with a value one unit lower, and nodes with a negative label are pruned. We chose to use the vocabulary from the tree search literature, but a similar idea exists in the MIP community under the name of *local branching* (Fischetti and Lodi 2003).

Figure 3 presents an example of the execution of a binary tree-search LDS with the parameter $D^{lds} = 2$. Each level of the tree is associated with a binary variable. The reference solution is depicted in the squares on the left. A node has up to two sons: one with the same value as in the reference solution and the one with the other value (i.e., the inverse of the value of the considered variable in the reference solution). Nodes pruned because of a negative value of their label are not drawn.

For the static-dynamic SPDPL, the tree search works on the binary variables. In our case, however, the tree search has to concurrently make two types of decisions. First, let us introduce D^{lds}_{prod} as the maximum number of allowed modifications to the production setup variables. Second, D^{lds}_{trans} is the maximum number of allowed modifications to the transportation setup variables. As mentioned above, we typically use the solution obtained in the previous step of the rolling horizon as the reference solution. This solution is extended with no production and no transportation for the newly considered last period. According to preliminary experiments, production and transportation setup decisions are not prone to change significantly between iterations. Hence, D^{lds}_{prod} and D^{lds}_{trans} can be set to small values. For the sake of simplicity, we consider the variables in ascending order of the time periods. For a given time period, we first branch on the production setup variable and next on the transportation setup variables. These transportation setup variables are considered in ascending order of the customer index number and the carrier index number.

Beam search (Ow and Morton 1988) is a method between the well-known *Breadth First Search (BFS)* and *Depth First Search (DFS)*, but has received much less attention in the operation research community (Libralesso 2020). Its main idea is that, at each level of the search tree, all the descendants of every open node are created, much like in BFS, but beam search keeps only the D^{bs} best ones (according to a chosen guide function). Consequently, DFS is a beam search with a beam width of $D^{bs} = 1$, and BFS is the case with $D^{bs} = \infty$.

To be able to fully use the computing time budget allocated for each step of the rolling horizon, we decided to use a variant of beam search called *Anytime Column Search* (ACS, Vadlamudi et al. (2012)). ACS acts like an iterative beam search by first considering nodes that are the most promising, until the time budget is exhausted. Its key concept is that when the nodes are created

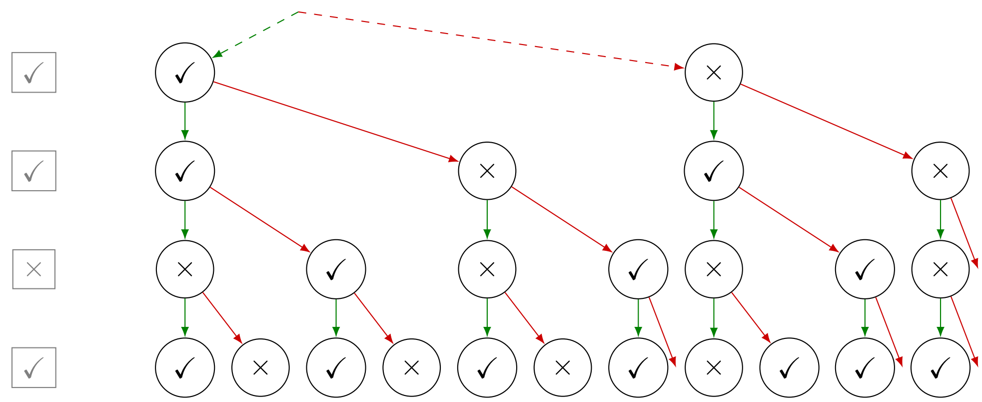

Figure 3 **Example of an LDS with** $D^{\text{lds}} = 2$

for a level, they are sorted by their estimated cost, much like in beam search. Likewise, only the best D^{acs} ones are expanded, the others are kept for later in a priority queue data structure instead of being discarded. Then, once the leaves are reached in the first column of width D^{acs}, the next D^{acs} nodes of the first periods are expanded.

Figure 4 presents an example of the execution of an ACS with the parameter $D^{acs} = 1$. Of course, in practice, nothing prevents us from using a larger initial value, or using a different value for the increment and the initial value of D^{acs}. In this example, the tree presented in Figure 3 is fully explored using ACS.

ACS is a mechanism to prioritize the exploitation of promising branches of the tree to guarantee the identification of a good feasible solution early in the search. Such a strategy is essential because, even with LDS, the number of possible combinations of production and transportation is too large to be fully explored.

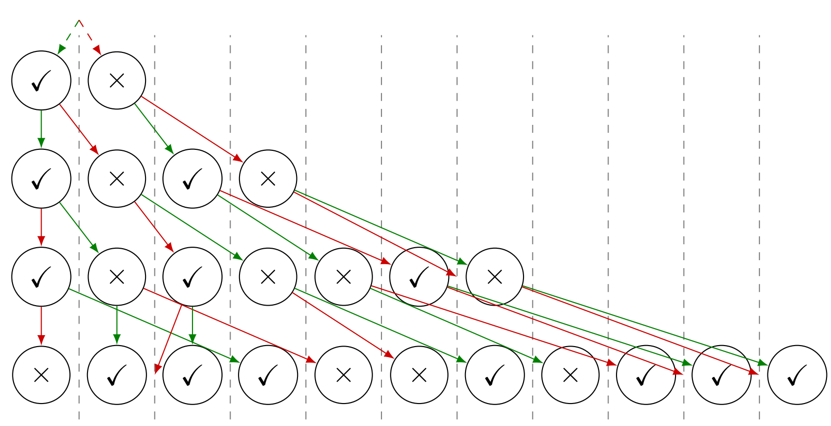

Figure 4 **LDS from Figure 3 combined with an ACS with** $D^{\text{acs}} = 1$

In the literature, a combination of LDS and beam search has already been used by Furcy and Koenig (2005) under the name *Beam Search Using Limited Discrepancy Backtracking* for some combinatorial problems (N-puzzle, tower of Hanoi, Rubik's cube). Our heuristic algorithm is presented in Algorithm 1. Let us designate the initial beam search width by D^{acs}_{init}, and its increment D^{acs}_{incr}. This algorithm also requires the parameters D^{lds}_{prod} and D^{lds}_{trans} to be specified, as well as a reference solution s.

Algorithm 1 relies on an array of priority queues (Line 2), one for each level of the tree search. Each of the priority queues stores the nodes to be expanded and, by definition, gives first the most promising one. The tree is initialized with a dummy node (Line 3), from which two branches are created to decide on the production setup variable of the first considered period (Line 4). Recall that each node n of the tree has two labels $(d^{prod}_n; d^{trans}_n)$ to keep track of its validity w.r.t. D^{lds}_{prod} and D^{lds}_{trans}. For example, at line 4, the first node, i.e., with the same production setup, is labelled with $(d^{prod}_n; d^{trans}_n)$ and the other is labelled with $(d^{prod}_n - 1; d^{trans}_n)$. The main loop, Line 6, continues until the tree has been completely explored or, more likely, until the time limit is reached. The inner loop, Line 8, browses the levels of the tree, each corresponding to a binary variable. For each time period, the production setup variable is considered first, then come all the transportation setup variables, in increasing order of their indices. Finally, the loop at Line 9 is used to open the w most promising un-expanded nodes of the considered level. The node to expand is first retrieved from the priority queue of the current level (Line 10), i.e. the node of the level with the lowest estimated cost. Its children are then created depending on the decision variables which will be modified, and on its LDS counters (Line 11). We consider that the first child node is the one using the decision made in the reference solution s, and that the second child takes the other possible value. When a node is added to a priority queue, it is evaluated with the cost-to-go estimation guide function (discussed in Section 4.2) and sorted w.r.t. the other nodes not yet expanded at this level of the tree. Observe that the value of w is progressively increased at line 25. Finally, the best-found solution, i.e., leaf node, is returned at the end.

4.2. Flow sub-problem

As mentioned in the previous section, Algorithm 1 requires a guide function to evaluate and choose promising nodes (see Line 10). Because this guide function will be called thousands of times during a tree search, and it is a bottleneck, it needs to be as efficient as possible. To this end, we propose a flow sub-problem to adjust the quantity of goods produced and transported from the solution of the father node to the newly considered value of the setup variables.

In order to efficiently search through the solution space, some recent studies focus on estimations. In practice, these methods iteratively improve the quality of their estimation of the recourse costs

Algorithm 1: LDS-ACS for the SPDPL

s : initial solution adapted to the scenario tree
$level$: [empty priority queue$|i \in [0; |T_{\tilde{t}}|(1+|R| \times |F|)]$]
n : create the root node labelled with $(D^{lds}_{prod}; D^{lds}_{trans})$
$level[0]$: add the two descendants of n labelled with $(D^{lds}_{prod}; D^{lds}_{trans})$ and $(D^{lds}_{prod} - 1; D^{lds}_{trans})$
$w = D^{acs}_{init}$
while some nodes are not explored **and** time is remaining **do**
  $i = 1$
  for $t \in T$, $r \in R$, $f \in F$ **do**
    for j from 1 to w **do**
      n : pop of $level[i]$
      if n is a leaf **then**
        Compare n to the current best known solution
      else if n branches on production **and** $d^{prod}_n > 0$ **then**
        Add the first descendant of n to $level[i+1]$ labelled with $(d^{prod}_n; d^{trans}_n)$
        Add the second descendant of n to $level[i+1]$ labelled with $(d^{prod}_n - 1; d^{trans}_n)$
      else if n branches on transport **and** $d^{trans}_n > 0$ **then**
        Add the first descendant of n to $level[i+1]$ labelled with $(d^{prod}_n; d^{trans}_n)$
        Add the second descendant of n to $level[i+1]$ labelled with $(d^{prod}_n; d^{trans}_n - 1)$
      else
        Add the first descendant of n to $level[i+1]$ labelled with $(d^{prod}_n; d^{trans}_n)$
      end if
    end for
    $i = i + 1$
  end for
  $w = w + D^{acs}_{incr}$
end while
return the best solution found

depending on the first-stage variables. Zou, Ahmed, and Sun (2019) and Thevenin, Adulyasak, and Cordeau (2022) review these methods, including nested Benders decomposition, L-shaped decomposition, and Stochastic Dual Dynamic Programming (SDDP). However, to the best of our knowledge, even the recently updated review of Juan et al. (2021) does not mention tree-search based methods using such a strategy for stochastic production-distribution problems (except for the usual branch-and-cut embedded methods previously mentioned). Nonetheless, we can find some examples for other problems. We can cite the tabu search presented in Gendreau, Laporte, and Séguin (1996) for a VRP with stochastic demand, in which the evaluation of the moves is based on an analytical expression using calculations updated regularly, or Bianchi et al. (2004) who also consider a VRP with stochastic demand and perform evaluations either with an analytical expression or a fast linear-time procedure.

An internal node of the search tree encodes a partial solution, as the binary decisions for later periods are not made yet. To evaluate these partial solutions, we complete them with the choices made in the reference solution used for the considered tree search. By definition of LDS, the remainder of the solution cannot be significantly different (because of the D^{lds} parameter) and, regardless, a guide function does not need to be exact. Having said that, the quantities of goods produced and sent need to be adapted to the considered binary decisions. Let us define this flow sub-problem as searching for the best improvement of the produced, stored, transported, and missed quantity of goods such that the modifications are the same for all the scenarios. That is to say, we improve the solution for each scenario uniformly by solving a sub-problem whose size does not depend on the size of the scenario tree.

In order to define the flow sub-problem as a minimum cost flow problem once the binary setup variables have been fixed, let us define an auxiliary graph. The nodes are a source s, a sink S, and for each time period $t \in T_{\tilde{t}}$, a copy of the facility 0_t and a copy of each customer r_t. The arcs are, for each time period $t \in T_{\tilde{t}}$, the production $(s, 0_t)$ if $y_t = 1$, the transportation for each customer r and each carrier f $(0_t, r_{t+v^t_{rf}})$ if $z^t_{rf} = 1$, the demand for each customer r (r_t, S), the storage at the facility $(0_t, 0_{t+1})$, the storage for each customer r (r_t, r_{t+1}), and, finally, the lost sales arc (s, S). The cost of the production arcs is p_t, the cost of the storage arcs is h_0 or h_r, respectively, the cost of the transportation arcs is u^t_{rf}, the cost of the demand arcs is 0, and the cost of the lost sales arc is a. Figure 5 represents an example of such an auxiliary graph. It corresponds to the instance presented in the example of Figure 1, with four periods. For the sake of clarity, the transportation arcs are represented only for the first period, and all production setup variables are set to 1.

As of now, the auxiliary graph does not take into account the stochastic nature of the SPDPL. To do so in a straightforward way would require duplicating the vertices w.r.t. the nodes of the tree search instead of the time. In addition, it would also require duplicating the arcs while imposing that the flow along them, representing quantities, respect the non-anticipativity constraints imposed by Model 3. Even if this was possible, it would require solving a generalized minimum cost flow problem, which has been proven to be much more time-consuming than the standard min cost flow problem (Wayne 2002). Therefore, we only consider the deterministic graph.

To improve the solution with a procedure whose complexity does not depend on the scenario tree size, we aggregate all the scenarios. To do so, let us define the upper (resp. lower) capacity on each arc of the auxiliary graph as the maximum possible increase (resp. decrease) of the flow of products on this arc among all the scenarios of the considered scenario tree for the considered solution. Thus, we use the simplex network algorithm to transport 0 units of flow from the source to the sink. A negative flow value $-x$ on an arc represents a decrease in the quantity of product that is either produced, transported, stored, missed, or fulfilled by x for all the scenarios. On the contrary, a positive flow value x on an arc represents an increase in the corresponding quantity of products by x for all the scenarios. By construction of the auxiliary graph, the resulting quantities are non-negative and respect the capacities for all the scenarios. Accordingly, the cost of the solution is modified by the same value as the objective value of the flow sub-problem (which is negative or null since it is a minimization problem).

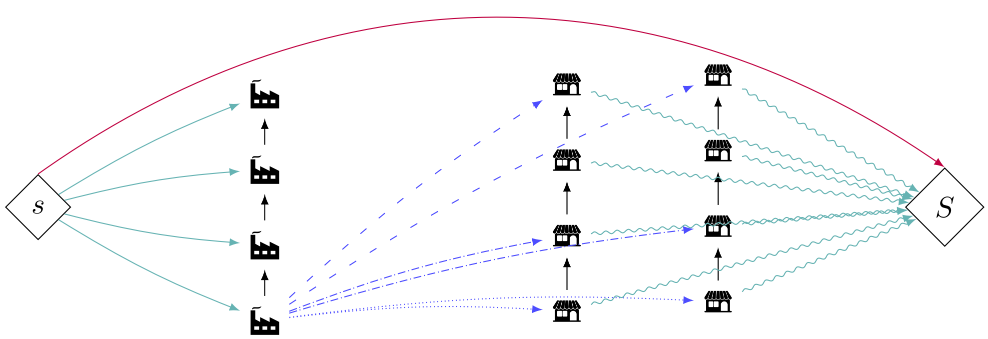

Figure 5 Example of an auxiliary graph for the guiding heuristic sub-problem

For the sake of completeness, let us detail the LP model describing the flow sub-problem. Since this model is defined to improve a given solution of Model 3, this solution is described using the variables of the aforementioned model superscripted by a bar. In addition, let us define the modified production cost at period $t \in T$ $\overline{p_t}$, and the modified transportation cost at period $t \in T$ to customer $r \in R$ with carrier $f \in F$ $\overline{u^t_{rf}}$. $\overline{p_t}$ is equal to p_t if $\overline{y^t} = 1$ and to M otherwise, likewise, $\overline{u^t_{rf}}$ is equal to u^t_{rf} if $\overline{z^t_{rf}} = 1$ and to M otherwise. Indeed, considering the arcs whose setup variables are 0 as highly penalized with a cost M simplifies the problem when the associated variables were set to 0 in the solution to evaluate.

The variables of Model 2 are the following. Let $\widehat{x^t} \in \mathbb{R}$ be the additional quantity produced at period $t \in T_{\tilde{t}}$; $\widehat{\theta^t_{rf}} \in \mathbb{R}$ be the additional quantity sent to customer $r \in R$ via carrier $f \in F$ at period $t \in T_{\tilde{t}}$; $\widehat{i^t_0} \in \mathbb{R}$ be the additional quantity stored at the facility at period $t \in T_{\tilde{t}}$; $\widehat{i^t_r} \in \mathbb{R}$ be the additional quantity stored at customer $r \in R$ at period $t \in T_{\tilde{t}}$; $\widehat{d_r^t} \in \mathbb{R}$ be the additional demand fulfilled by customer $r \in R$ at time $t \in T_{\tilde{t}}$; and $\widehat{l} \in \mathbb{R}$ be the additional lost sales. Using this notation, the SPDPL flow improvement sub-problem can be formulated as follows:

Model 2: Flow improvement sub-problem

$$\texttt{Min}\ a.\widehat{l} + \sum_{t \in T_{\tilde{t}}} \left(\overline{p_t}\widehat{x^t} + h_0 \widehat{i_0^t} + \sum_{r \in R} \left(h_r \widehat{i_r^t} - a.\widehat{d_r^t} + \sum_{f \in F} \overline{u_{rf}^t}\widehat{\theta_{rf}^t} \right) \right) \tag{15}$$

$$\texttt{s.t}\ \sum_{t \in T_{\tilde{t}}} \widehat{x^t} + \widehat{l} = 0 \tag{16}$$

$$\sum_{t \in T_{\tilde{t}}} \sum_{r \in R} \widehat{d_r^t} - \widehat{l} = 0 \tag{17}$$

$$\widehat{i_0^{t-1}} + \widehat{x^t} = \widehat{i_0^t} + \sum_{r \in R} \sum_{f \in F} \widehat{\theta_{rf}^t} \qquad \forall t \in T_{\tilde{t}} \tag{18}$$

$$\widehat{i_r^{t-1}} + \sum_{f \in F} \sum_{t'=\min(T_{\tilde{t}})}^{t} \delta_{\{t'+v_{rf}^{t'}=t\}} \widehat{\theta_{rf}^{t'}} = \widehat{d_r^t} + i_r^t \qquad \forall t \in T_{\tilde{t}}, r \in R \tag{19}$$

$$-\min_{n \in \mathcal{N}(t-1)} \overline{x^n} \leqslant \widehat{x^t} \leqslant \delta_{\overline{y^{t-1}}=1} \left(Q_t - \max_{n \in \mathcal{N}(t-1)} \overline{x^n} \right) \qquad \forall t \in T_{\tilde{t}} \tag{20}$$

$$-\min_{n \in \mathcal{N}(t-1)} \overline{\theta_{rf}^n} \leqslant \widehat{\theta_{rf}^t} \leqslant \delta_{\overline{z_{rf}^{t-1}}=1} M \qquad \forall t \in T_{\tilde{t}}, r \in R, f \in F \tag{21}$$

$$-\min_{n \in \mathcal{N}(t)} \overline{i_0^n} \leqslant \widehat{i_0^t} \leqslant M \qquad \forall t \in T_{\tilde{t}} \tag{22}$$

$$-\min_{n \in \mathcal{N}(t)} \overline{i_r^n} \leqslant \widehat{i_r^t} \leqslant M \qquad \forall t \in T_{\tilde{t}} \tag{23}$$

$$-\min_{n \in \mathcal{N}(t)} (d_r^n - \overline{l_r^n}) \leqslant \widehat{d_r^t} \leqslant \min_{n \in \mathcal{N}(t)} \overline{l_r^n} \qquad \forall t \in T_{\tilde{t}}, r \in R \tag{24}$$

$$-\sum_{t \in T_{\tilde{t}}} \sum_{r \in R} \min_{n \in \mathcal{N}(t)} \overline{l_r^n} \leqslant \widehat{l} \leqslant \sum_{t \in T_{\tilde{t}}} \sum_{r \in R} \min_{n \in \mathcal{N}(t)} (d_r^n - \overline{l_r^n}). \tag{25}$$

The objective (15) is the sum of the cost of the applied modifications to production, storage, transport, and lost sales. Let us call this value the estimated cost variation induced by the considered modified binary variable. Constraints (16)–(19) are the flow conservation constraints, and (20)–(25) are the bound constraints. Constraint (16) is the sum of the additional goods produced and lost sales. Constraint (17) is the sum of the newly met demand minus the additional lost sales. Constraints (18) consider the variation of the inventory at the plant, of the production, and of the transport. Constraints (19) consider the variation of the inventory at the customers, of the transport, and of the met demand. The modification of production at period t, $\widehat{x^t}$, is bound by (20) between the negative of the minimum production on nodes taking place at period t, and the maximum remaining production capacity on nodes of this period (or 0 if there is no production setup). The modification of the quantity transported at period t to customer r with carrier f, $\widehat{\theta_{rf}^t}$, is bounded in (21) by the negative of the minimum amount transported on nodes of this period to r via f, and by M if this transport is set-up (otherwise by 0). Constraints (22) and (23) limit the modification of the stored quantity by the negative of the minimum quantity stored at the considered location among nodes of the considered period, and by a large number M. The modification of the demand fulfilled at period t at customer r, $\widehat{d}_r^t$, is bounded in (24) by the negative of the minimal amount of demand fulfilled at r on nodes of period t, and by the minimum of unmet demand among the same nodes. Finally, the modification of the amount of lost sales among all periods and customers, $\widehat{l}$, is bounded in (25) by the negative of the sum over the periods and customers of the minimum amount of lost sales at the considered customer among nodes at the considered time period, and by the same sum of the minimum of met demand.

To conclude, Model 2 represents a minimum cost flow problem with bounded flow. First, its size does not depend on the number of scenarios. Second, in practice, the underlying graph has much fewer arcs than the complete one presented in the example of Figure 5 since only a fraction of the transport possibilities are available. Third, since we modify only one binary variable at a time in Algorithm 1, the flow in the improvement sub-problem will be non-null on only a few arcs of the considered network. Thus, we can evaluate a solution very efficiently with the network simplex algorithm (Frangioni and Manca 2006), instead of solving a complex generalized minimum cost flow problem on the original solution (Wayne 2002).

4.3. Rolling horizon

To summarize this section, Algorithm 2 indicates the different steps of the proposed rolling horizon heuristic on a given realization of an instance. The algorithm is initialized with the solution obtained by solving the deterministic model of the PDPL with the average demand for each retailer at each period. Then, for each period of the time horizon, the rolling horizon samples scenarios for the time window starting from the considered period. The current solution is adapted to the generated scenario tree by solving the multi-stage static-dynamic Model 3 using the binary variables of the considered solution. If needed, this solution is extended to the newly considered last period with no production and no transportation. This solution is improved by the tree search heuristic described in Section 4.1 using the cost-to-go estimation guide heuristic presented in Section 4.2. The newly obtained solution is also finely tailored to the considered scenario tree by solving Model 3 with its new values for the binary variables. Finally, the demand for the considered period is revealed, the inventory and lost sales are computed at each customer for the period t, before moving to the next period.

5. Experiments

The algorithm was implemented in C++. It was complied with g++ 9.3.0 and run on CentOS Linux 7 with an Intel Xeon E5-2683 processor running at 2.10GHz. Only one thread was used by our code and by CPLEX. We used the IBM Ilog CPLEX 20.1.0 MIP solver (IBM 2020). The

Algorithm 2: Rolling horizon using the tree-search heuristic

```
s : Initial solution from the deterministic model with the expected values of the demand
for t = 1 to |T| do
    Build the sampling scenario tree S_t with the t − 1 first periods fixed to the known demand
    s ← Solve Model 3 with the binary variables fixed to the value of s on S_t
    s ← Call Algorithm 1, guided by Model 2, on s and S_t with the first t − 1 periods fixed
    s ← Solve Model 3 with the binary variables fixed to the value of s on S_t
    Reveal the demand for period t
    s ← Compute the inventories and lost sales at period t in solution s w.r.t. the demand
end for
return s
```

generator vector used in our implementation of RQMC is created by Lattice Builder (L'Écuyer and Munger 2016). Probability distributions are managed by boost 1.78.0 (Boost 2021). The LDS and the ACS are implemented using the CATS-Framework (Bouhassoun and Libralesso 2020, Libralesso 2020). The simplex network solver for the flow sub-problem is implemented using the Lemon library (Dezső, Jüttner, and Kovács 2011, 2014).

For a given instance, a test set of 16 realizations was constructed. It is unknown to the solution method. They are generated using a crude Monte-Carlo method. The averages are computed over the various evaluated realizations.

For a given instance, realization and rolling horizon time window starting in period $t \in T$, a sampling scenario tree is built. It is not assumed to encompass the considered realization, and most certainly will not since they are generated independently. The sampling scenario trees are generated using Randomized Quasi Monte Carlo (RQMC). A sampling scenario tree represents 512 scenarios and is constructed as follows: eight nodes at period t, four child nodes for the periods $t+1$ and $t+2$, two child nodes for the next two periods, and one for the next six periods. For the aforementioned six additional periods, the expected demand is used instead of RQMC. These parameters were chosen based on the recommendations of Thevenin, Adulyasak, and Cordeau (2022) and on preliminary experiments.

As a preliminary experiment for the rolling-horizon framework, we also adapted OWMRP models of Solyali and Sural (2012), Cunha and Melo (2016) and Gruson et al. (2019): echelon stock model, multi-commodity model, multi-commodity echelon-stock model, plant-location model, and pre-processed shortest path model. On average, for the multi-stage static-dynamic problem solved in the rolling horizon, they were not more efficient than the intuitive model. Hence, Model 3 was kept for the sake of simplicity.

In the tree-search method, we use $D^{lds}_{prod} = 2$, $D^{lds}_{trans} = 2|R|$, $D^{acs}_{init} = 20$, and $D^{acs}_{incr} = 5$. The tuning of the LDS parameters is based on the analysis of optimal solutions computed by CPLEX, and the ACS parameters do not have a significant impact as long as they remain reasonably small. Finally, the time budget of the tree search for time window in the rolling horizon is of 360 seconds. The tables of Appendix B provide more detail about the results presented in the following sections.

5.1. Instances

Our instances are based on those of Solyali and Sural (2012) for the deterministic One-Warehouse Multi-Retailer Problem. Compared to the original instances, we applied the following modifications. The unit production cost is stationary, whereas the fixed-cost can be stationary or dynamic, but both are multiplied by 2 w.r.t. the original instances so that the production costs are not negligible with respect to the modified transportation costs. Since the original instances do not have a production capacity Q, for all periods we set it to 1.4 times the total average demand of all customers in each period. For the customers, we set the unit lost sales penalty a to 50 (the unit production cost is of 1), and we set the initial inventory to 100 (the average demand is in the range $[1; 100]$). Three carriers are available, their fixed costs are the same and they are equal to ten times the unit transportation cost used in the original instances (they do not consider fixed transportation cost). The first carrier has a lead time of 1 and a unit cost equal to the one of the original instances. We refer to it as "air transport". The second has a lead time of 2 and the original unit cost is divided by 3, and we refer to it a "road transport". The third one has a lead time of 4 and the original unit cost is divided by 5, and we refer to it as "maritime transport". Since we introduce positive lead times, we set the demand at the customers to 0 for the first four periods to accurately simulate the system setups.

For the uncertainty, we consider that the demand follows normal distributions centered at the expected demand. Let ρ be the coefficient of variation of the customers' demand, we set it to $\rho = 30\%$. Thus, the demand at customer $r \in R$ at period $t \in T$ follows the normal distribution $\mathcal{N}(d^t_r, \rho d^t_r)$.

Instances have 50 or 100 customers over 15 or 30 time periods. The production setup cost and the demand can either be stationary or dynamic. "SD" stands for stationary demand, as opposed to "DD", which stands for dynamic demand. Likewise, "SF" stands for stationary fixed costs, and "DF" for dynamic fixed costs. With 10 instances for each setting, we obtain a total of 160 instances. The presented results are averages over the instances considered and over 16 realizations of the demand per instance.

5.2. Evaluation of the developed method

This section focuses on the evaluation of the proposed method. To solve the considered multi-stage dynamic SPDPL, four approximations are used: (1) the rolling horizon framework relaxes the dynamic aspect by repetitively solving smaller sub-problems; (2) solving the static-dynamic model on a sample scenario tree reduces the sub-problem to a manageable size; (3) the tree-search heuristically solves the multi-stage static-dynamic problem; (4) the flow heuristic aggregates the multi-stage component. Unfortunately, we cannot evaluate the effectiveness of the rolling horizon approximation because of the difficulty of solving the complete multi-stage dynamic problem. To the best of our knowledge, no method has been proposed for such problems, and it is out of the reach of CPLEX to solve instances of significant size.

To evaluate the impact of the sample scenario tree, Table 2 shows the results on the instances with 15 periods and 50 customers using three different structures for the scenario tree. Each line corresponds to a configuration: Line 2 to a branching factor in the sampling scenario tree of two for the first five periods, Line 3 to the chosen configuration, and, Line 4 to twice the previous number of scenarios. Column 2 gives the average cost of the solutions produced with the considered configuration. Columns 3 to 6 detail the average production costs, transportation costs, holding costs and lost sales penalties, respectively. Finally, Column 7 indicates the average total computing time using each structure of the sampling scenario tree. We can see that reducing the number of scenarios severely degrades the quality of the solutions, and that increasing it complicates the problem to the point where it also degrades the quality of the solutions.

Configuration	Total cost	Prod. cost	Transport. cost	Holding cost	Lost sales	Time (sec)
Scenario tree 22222	269 744.9	64 364.2	136 410.9	52 708.5	16 261.3	5 395.9
Scenario tree 84422	266 976.8	64 397.4	136 436.2	53 069.2	13 074.1	5 854.6
Scenario tree 88442	269 082.7	64 401.2	137 231.6	53 768.0	13 681.8	11 617.2

Table 2 Comparison of the sampling strategies on the instances with 50 customers and 15 periods

To test the impact of the third approximation, we used CPLEX to solve the static-dynamic multi-stage problem for each time window of the rolling horizon instead of the proposed tree search. CPLEX was allowed to use one thread and 25Gb of memory with a computing time budget of one hour per time window The default settings were used. The instances with 50 retailers and 15 periods were solved. The cost of the solutions found by CPLEX is, on average, 17.57% higher than the cost of the solutions found by the proposed method. At the end of the time budget, the reported relative MIP gap is consistently over 20%, and often above 30%.

In terms of the total run time, the proposed method scales especially well with the number of customers. The total run time only increases by 11% and 61% when the number of customers doubles on instances with 15 and 30 time periods, respectively. The reader should keep in mind that the average run time per time window is not impacted by the length of the planning horizon as it does not change the number of periods considered at each iteration of the rolling horizon. Besides, we can notice that stationary demand and costs do not have a major impact on the run time. Table 3 details these results on instances with 50 or 100 customers and with 15- and 30-period time horizons. In addition, when fixed transportation costs are set to 0, all the transportation setup binary variables can be trivially set to 1. Thus, the LDS tree search can be completely performed at each period, dividing the total run time by a factor of 5.3.

Instances	Total cost	Prod. cost	Transport. cost	Holding cost	Lost sales	Time (sc)
$\|R\| = 50; \|T\| = 15$	266 976.8	64 397.4	136 436.2	53 069.2	13 074.1	5 854.6
$\|R\| = 50; \|T\| = 30$	669 414.3	185 027.9	292 596.6	135 348.4	56 441.4	16 476.7
$\|R\| = 100; \|T\| = 15$	507 405.5	92 264.7	285 635.9	106 624.1	22 880.7	6 599.3
$\|R\| = 100; \|T\| = 30$	1 212 777.4	258 860.0	620 487.3	258 298.3	75 444.9	26 647.3

Table 3 Summary of the results on various instance sizes

With respect to the fourth approximation, we present results to assess the approximation heuristic used to guide the tree search. To do so, an additional evaluation of the leaves of the tree was performed by CPLEX. Each leaf of the tree defines a partial solution, giving a value to the binary variables but not to the continuous ones. Thus, for each explored combination of the binary variables, Model 3 was solved to get the true cost of the considered solution. Then all the leaves can be sorted in increasing order of their cost computed by CPLEX. Indeed, in the heuristic approach the leaves are evaluated only with the guide heuristic, giving only an approximate cost to the solution since it does not generally find the optimal values for the continuous variables. This evaluation was performed on the sub-problems solved during the first seven iterations of the rolling horizon, the last period were not concerned because the time window is truncated with the end of the planning horizon. The instances with 50 customers and 15 periods were solved.

Performance-wise, the gap between the true cost of the solutions selected by the heuristic and by CPLEX is negligible: it is consistently below 0.1%, with an average of 0.08%. However, there is a gap of 9% between the estimated cost variation of the chosen leaf (i.e., the best estimated cost variation computed by the guide heuristic) and the estimated cost variation of the leaf chosen by CPLEX (i.e., the estimated cost variation of the best leaf). Nonetheless, these results validate the use of approximation guide heuristics as they can identify near-optimal solutions quickly even if the estimated costs are not very accurate. In addition, the leaf selected by the heuristic is often identified early in the search. Thus, exploration mechanisms may be a promising path for future research.

During these experiments, the average time spent to solve a sub-problem with CPLEX and with the heuristic was recorded. On average, CPLEX took 0.465 seconds to solve Model 3 with fixed binary variables, and Model 2 was solved by the network simplex in 0.00147 seconds on average. That is to say, a node is evaluated by our heuristic 316 times faster than by CPLEX.

5.3. Impact of uncertainty

To quantify the effects of uncertainty with multiple carriers, we re-solved the instances with 50 customers and 15 periods with different values of the coefficient of variation of the customers' demand. Its value for the initial experiments was $\rho = 30\%$. This section presents results with no uncertainty, i.e., $\rho = 0\%$, and with triple uncertainty, i.e., $\rho = 90\%$.

To evaluate the use of sampling scenario trees for the SPDPL, we computed the *dynamic value of the stochastic solution* (VSS^{D}, Escudero et al. (2007)). This metric evaluates the quality of a method embedded in a rolling horizon by comparing its results with a reference rolling horizon. The reference rolling horizon solves the deterministic model of the considered problem with the average value of the uncertain parameters to take the decisions for the considered period. The gap can then be computed between the solutions obtained at the end of the planning horizon by the reference rolling horizon and the developed rolling horizon, in our case the one solving Model 2 with a sampling scenario tree.

Table 4 presents the results of the experiments considering different levels of uncertainty. Columns 2, 3 and 4 correspond to a value of the coefficient of variation of the customers' demand, from 0% to 90%. Columns 5 and 6 give the average VSS^{D} in the case of $\rho = 30\%$ and 90%, respectively. The second line reports the average cost of the solutions obtained by the developed method and by using all three carriers. The third line presents the average cost of the solutions obtained by the reference rolling horizon solving the deterministic version of Model 1 with the expected demand and with all three carriers. The fourth and fifth, sixth and seventh, and eighth and ninth lines show the results of the two methods but using only air, road, or maritime transportation, respectively. Of course, it is not relevant to perform a sampling of possible outcomes when there is no uncertainty.

The average cost of the solutions found by the reference rolling horizon is 3.69% more expensive on the instances with three carriers and with $\rho = 30\%$. On the instances with the uncertainty increased to $\rho = 90\%$, the dynamic value of the stochastic solution is of 4.03%. When only one transportation mode is available, the VSS^{D} is much lower, to the point where it is barely significant when only the fast transportation mode is used. The fact that the VSS^{D} is much lower when only one transportation mode is available underlines the importance of using scenario trees to make decisions instead of the expected scenario when multiple transportation modes are available.

	$\rho=0\%$	$\rho=30\%$	$\rho=90\%$	$\text{VSS}^{\text{D}}_{\rho=30\%}$	$\text{VSS}^{\text{D}}_{\rho=90\%}$
All carriers with sampling	-	266 976.8	278 918.4		
All carriers without sampling	258 231.3	276 836.4	290 177.8	3.69%	4.03%
Air only with sampling	-	577 937.0	587 405.3		
Air only without sampling	565 528.4	579 575.9	589 719.7	0.28%	0.39%
Road only with sampling	-	327 607.8	338 021.6		
Road only without sampling	315 857.8	333 228.2	345 720.1	1.71%	2.28%
Maritime only with sampling	-	298 446.4	311 775.4		
Maritime only without sampling	285 462.9	303 660.7	316 816.2	1.74%	1.62%

Table 4 **Average cost of the solutions with different levels of uncertainty**

5.4. Application example

To understand the consequences of the combination of multiple transportation modes on the supply chain, this section presents examples based on synthetic data. Several variations of the randomly generated instances presented in Section 5.1 were solved. If not stated otherwise, the experiments presented in this section were performed on the instances with 50 customers and 15 periods. In the figures, the three transportation modes of our instances (air, road, maritime) will be depicted by a plane, a truck, and a ship, respectively. Note that the aim of this section is to provide an example. The presented results are hence only valid for the specific setting and if cost or lead times change, the results will of course be different. Nonetheless, these data show that finding efficient solutions to the proposed instances requires non-trivial combinations of the transportation modes.

Figure 6 presents the average cost of the solutions as a function of the available transportation modes. For each combination, the bar depicts the average cost of the solutions obtained at the end of the rolling horizon on each instance and for each evaluated realization. The bar is divided into four parts to represent the share of each type of cost in the total cost: production costs (fixed and unit), holding costs (at the plant and at the customers), transportation costs (fixed and unit), and lost sales penalties.

Figure 7 presents the average quantity of products transported by each carrier along the planning horizon depending on the available transportation modes. In addition, Figure 8 presents the average number of times each carrier has been used along the planning horizon depending on the available transportation modes.

When maritime transport is available, the majority of the products are transported by this mode, thus confirming its importance. In the first three combinations, an average of 84% of the products are transported using this mode. Nonetheless, using only the cheapest transportation mode does not lead to the lowest overall cost. Indeed, when only using the maritime transportation, the overall cost increases on average by 11.4% and the quantity of lost sales increases by 38.9% compared to using all modes.

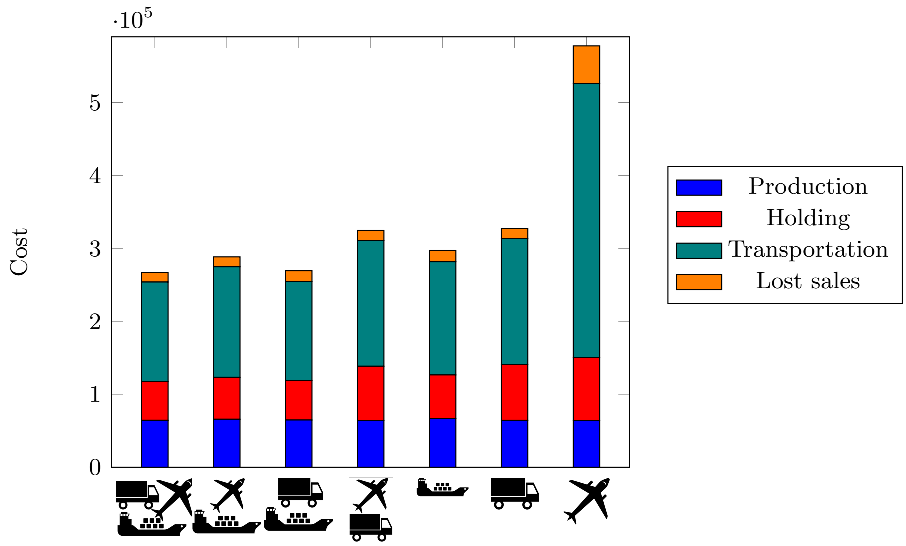

Figure 6 **Average cost of the solutions depending on the available transportation modes**

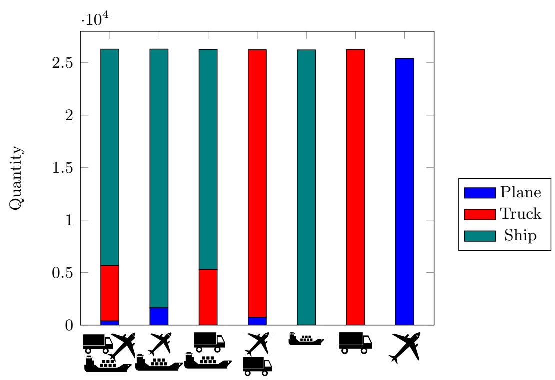

Figure 7 **Average quantity transported depending on the available transportation modes**

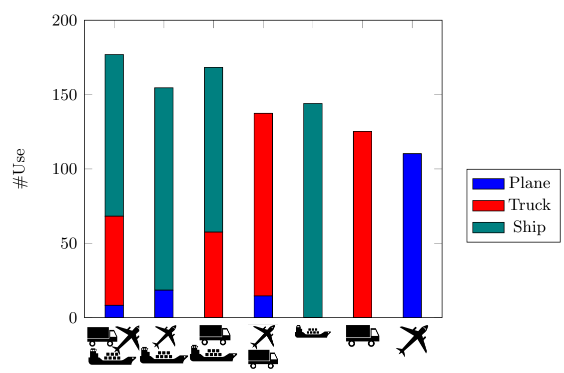

Figure 8 **Average number of calls to carriers depending on the available transportation modes**

Second, adding road transport in the mix has a significant impact. Indeed, the cost is on average 10.48% lower when both the road and maritime mode are used compared to the case when only the maritime mode is used. It also improves the service quality by decreasing the lost sales by 22.1%. In this configuration, 20.2% of products are carried by road, but it represents 34.1% of the number of deliveries to customers. In addition, the time between orders of the maritime mode increases by 28.4%, from 3.34 to 4.29 periods. Road transportation is thus a good addition to maritime transportation to mitigate demand fluctuations.

Third, when road and maritime transport are already integrated, adding air freight in the mix only reduces the average cost by 0.78%. It can be seen by the fact that the fast transportation mode is only used 8.4 times on average, and rarely twice for the same customer. Nonetheless, it can be of great use to counterbalance unforeseen fluctuations. Indeed, over all the instances and realizations, its inclusion reduces the cost by up to 9.1% and the lost sales by up to 56.3% compared to the solution obtained with only the road and the maritime modes.

When the number of customers goes from 50 to 100, the cost increases by only 90%, i.e., the cost per customer is reduced by 5%. Instead of doubling, the lost sales increase by only 75%, and the increase of the production cost is only 43.3%. When only the maritime, the road or the air transportation is available, the cost increases by 91.6%, 93.8% and 98% when the number of customers is doubled, respectively. Consequently, the combination of multiple carriers even reinforces the economies of scale that come from considering more customers at once.

Allowing backlogs reduces the cost by 2.18%, while it increases the computing time by 1.32%. The backlogging cost per unit per period was equal to a third of the lost sales penalty cost. The amount of lost sales is reduced by 44.53% on average. Otherwise, the total production and holding costs are not significantly changed. When looking at the transportation modes, we notice that the quantity of products transported by air is reduced by 3.06%, while the share of the road is slightly reduced, and that of maritime transport slightly increased.

When fixed transportation costs are set to 0, the holding costs and the lost sales are reduced by 18% and 55.66% respectively. The amounts transported by air and by road increase by 15.38% and 20.35%, respectively, leading to a reduction of the usage of the maritime mode by 5.15%. The storage at the plant increases by 3.04% to reduce the storage at the customers by 19.48%.

Figure 9 depicts the average cost of the solutions as a function of the uncertainty level and of the lead time of the carriers. The first axis represents the coefficient of variation of the demand ranging from $\rho = 0$ (i.e., deterministic demand) to $\rho = 75\%$ (i.e., highly volatile demand). The second axis represents four sets of lead times. The case 'base+0' represents the base case in which the lead times are not increased. The case '+1' represents the case in which the lead times of all modes are increased by 1, giving a lead time of 2, 3 and 4 for air, road and maritime transportation, respectively. Similarly, for the cases '+2' and '+3' the base lead times for all transportation modes are increased by 2 and 3, respectively. The z-axis indicates the average cost of the solutions obtained when solving the selected instances with the considered settings. For this experiment, to limit the computing time, only the 10 instances with 50 customers, 15 periods, dynamic demand and dynamic fixed costs were considered.

In addition, Figures 10 and 11 provide some details when the coefficient of variation is 30% and with the considered sets of lead times. Figure 10 gives the share of each type of cost in the total cost of the solutions. Figure 11 presents the average number of times each carrier has been used along the planning horizon depending on the lead times.

For the base case, i.e., no increase in lead times, the cost of the solution with an uncertainty of $\rho = 75\%$ is 8% higher than with no uncertainty, whereas, with the highest considered lead times, the gap is 12.5%. Even with deterministic demand, the cost of the solutions with the highest transportation times (+3) is 28% higher than the cost the solutions obtained in the base case, and considering uncertain demand with a coefficient of variation of 75%, this gap becomes 33%. Contrary to what is expected based on the literature with one transportation mode, in our case, when the lead times increase, the holding costs decrease. In fact, there is a modal shift to achieve the best balance between holding and transportation costs: road transportation gets used more, while maritime transportation usage declines, which decreases the average time between orders since the ratio between the fixed cost and the per unit cost is lower with road transportation. To conclude, the solutions obtained with the high lead times and the high uncertainty are 44% more expensive than the ones with low lead times and known demand, especially because of an increase of 69% of the transportation costs and of 1335% of the lost sales penalties.

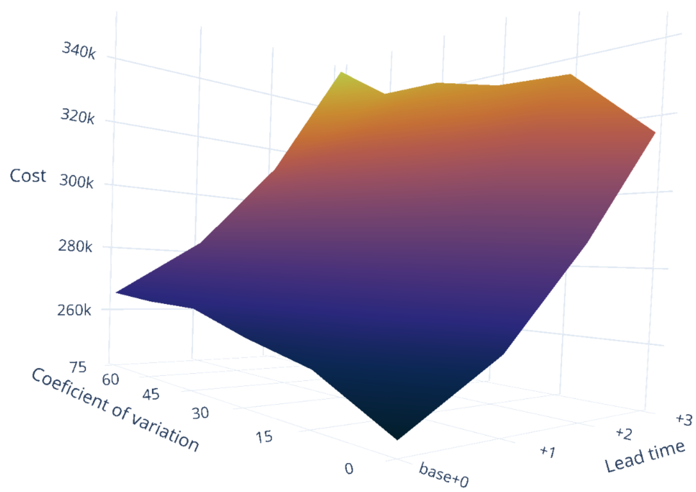

Figure 9 Average cost of the solutions depending on the lead time and on the uncertainty

6. Conclusion

In this paper, we have introduced the *Stochastic Production-Distribution Problem with Lead times (SPDPL)*. It extends the SOWMRP by considering multiple carriers to transport products from the plant to the customers. In order to solve real-size instances of the multi-stage version of the SPDPL, a rolling horizon based method was developed. At each period of the planning horizon, a static-dynamic multi-stage problem is solved with a scenario tree that samples possible outcomes of the demand. However, since we consider possibly long lead times, the width of the window considered in the rolling horizon needs to be large enough to accommodate all the lead times. Consequently, we presented a tree-search heuristic based on Anytime Column Search and Limited

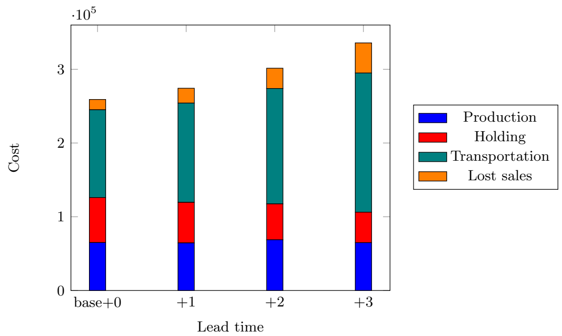

Figure 10 **Average cost of the solutions depending on the lead time**

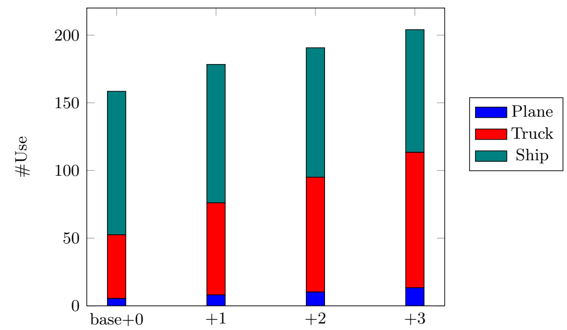

Figure 11 **Average number of calls to carriers depending on the lead times**

Discrepancy Search. To select promising nodes, we developed a flow-based guide heuristic that aggregates all the considered scenarios to quickly improve the current solution with respect to the set-up decisions of the current tree-search node.

First, it was observed that the developed guide heuristic allows to select solutions that are only 0.08% more expensive than those selected by CPLEX, while being 316 times faster on average since it does not explicitly consider the whole scenario tree. Second, the Limited Discrepancy Search combined with Anytime Column search guided by the aforementioned heuristic outperformed CPLEX on medium-size instances, since the latter was never able to improve the solutions produced in 360 seconds, even when given one hour per time window. Third, the proposed method scales well with the number of customers. Indeed, its total run time only increases by 11% and 61% when the number of customers doubles on instances with 15 and 30 time periods, respectively.

To conclude, the results presented in this study motivate the use of approximation guide heuristics as they can identify near-optimal solutions quickly even if the estimated costs are not very accurate. In future research, the SPDPL could be extended in several promising ways. First, multiple production sites could be considered, raising the issue of plants to customer assignment. Second, different types of carrier contracts could be considered, such as one-time contracts mixed with multiple-periods contracts, or less-than-truckload mixed with full-truckload companies. Third, and foremost, it would be challenging to consider uncertainty in the lead times.

Acknowledgments

This research was supported by the Group for Research in Decision Analysis (GERAD) under a post-doctoral fellowship grant 2022-2023. It was also suported by the Chair in Logistics and Transportation and the Chair in Supply Chain Operations Planning at HEC Montréal. This support is gratefully acknowledged.

References

Adulyasak Y, Cordeau JF, Jans R, 2015a *Benders decomposition for production routing under demand uncertainty. Operations Research* 63(4):851–867, URL http://dx.doi.org/10.1287/opre.2015.1401.

Adulyasak Y, Cordeau JF, Jans R, 2015b *The production routing problem: A review of formulations and solution algorithms. Computers & Operations Research* 55:141–152, URL http://dx.doi.org/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(1):5–24, URL http://dx.doi.org/10.1002/net.21796.

Aliev R, Fazlollahi B, Guirimov B, Aliev R, 2007 *Fuzzy-genetic approach to aggregate production–distribution planning in supply chain management. Information Sciences* 177(20):4241–4255, URL http://dx.doi.org/10.1016/j.ins.2007.04.012.

Alp O, Erkip NK, Güllü R, 2003 *Optimal lot-sizing/vehicle-dispatching policies under stochastic lead times and stepwise fixed costs. Operations Research* 51(1):160–166, URL http://dx.doi.org/10.1287/opre.51.1.160.12799.

Bertazzi L, Bosco A, Laganà D, 2015 *Managing stochastic demand in an inventory routing problem with transportation procurement. Omega* 56:112–121, URL http://dx.doi.org/10.1016/j.omega.2014.09.010.

Bertazzi L, Moezi SD, Maggioni F, 2021 *The value of integration of full container load, less than container load and air freight shipments in vendor–managed inventory systems. International Journal of Production Economics* 241:108260, URL http://dx.doi.org/10.1016/j.ijpe.2021.108260.

Bianchi L, Birattari M, Chiarandini M, Manfrin M, Mastrolilli M, Paquete L, Rossi-Doria O, Schiavinotto T, 2004 *Metaheuristics for the vehicle routing problem with stochastic demands*. Yao X, Burke EK, Lozano JA, Smith J, Merelo-Guervós JJ, Bullinaria JA, Rowe JE, Tiňo P, Kabán A, Schwefel HP, eds., *Parallel Problem Solving from Nature - PPSN VIII*, 450–460 (Berlin, Heidelberg: Springer Berlin Heidelberg), ISBN 978-3-540-30217-9.

Boost, 2021 *Boost 1.78.0*. https://www.boost.org/.

Bouhassoun AM, Libralesso L, 2020 *CATS-framework*. https://gitlab.com/librallu/cats-framework.

Boujemaa R, Rekik M, Hajji A, 2022 *An SDDP-based solution approach for carriers' selection and shipments assignment under dynamic stochastic demand*. Technical report, CIRRELT, URL https://www.cirrelt.ca/documentstravail/cirrelt-2022-17.pdf.

Boulaksil Y, Hamdouch Y, Ghoudi K, Fransoo JC, 2021 *Comparing policies for the stochastic multi-period dual sourcing problem from a supply chain perspective. International Journal of Production Economics* 232:107923, URL http://dx.doi.org/10.1016/j.ijpe.2020.107923.

Boute RN, Disney SM, Gijsbrechts J, Van Mieghem JA, 2021 *Dual sourcing and smoothing under nonstationary demand time series: Reshoring with speedfactories. Management Science* mnsc.2020.3951, URL `http://dx.doi.org/10.1287/mnsc.2020.3951`.

Brahimi N, Absi N, Dauzère-Pérès S, Nordli A, 2017 *Single-item dynamic lot-sizing problems: An updated survey. European Journal of Operational Research* 263(3):838–863, URL `http://dx.doi.org/10.1016/j.ejor.2017.05.008`.

Chavarro G, Fresen M, González ER, Ferro DB, López-Ospina H, 2021 *Solving the one-warehouse N-retailers problem with stochastic demand: An inter-ratio policies approach. International Journal of Industrial Engineering Computations* 131–142, URL `http://dx.doi.org/10.5267/j.ijiec.2020.7.001`.

Coelho LC, Cordeau JF, Laporte G, 2014 *Thirty years of inventory routing. Transportation Science* 48(1):1–19, URL `http://dx.doi.org/10.1287/trsc.2013.0472`.

Cunha JO, Melo RA, 2016 *On reformulations for the one-warehouse multi-retailer problem. Annals of Operations Research* 238(1-2):99–122, URL `http://dx.doi.org/10.1007/s10479-015-2073-4`.

Cunha JO, Melo RA, 2021 *Valid inequalities, preprocessing, and an effective heuristic for the uncapacitated three-level lot-sizing and replenishment problem with a distribution structure. European Journal of Operational Research* 295(3):874–892, URL `http://dx.doi.org/10.1016/j.ejor.2021.03.029`.

Dezső B, Jüttner A, Kovács P, 2011 *LEMON – an open source C++ graph template library. Electronic Notes in Theoretical Computer Science* 264(5):23–45, URL `http://dx.doi.org/10.1016/j.entcs.2011.06.003`.

Dezső B, Jüttner A, Kovács P, 2014 *LEMON graph library*. `https://lemon.cs.elte.hu/trac/lemon`.

Engebrethsen E, Dauzère-Pérès S, 2019 *Transportation mode selection in inventory models: A literature review. European Journal of Operational Research* 279(1):1–25, URL `http://dx.doi.org/10.1016/j.ejor.2018.11.067`.

Escudero LF, Garín A, Merino M, Pérez G, 2007 *The value of the stochastic solution in multistage problems. TOP* 15(1):48–64, URL `http://dx.doi.org/10.1007/s11750-007-0005-4`.

Fahimnia B, Farahani RZ, Marian R, Luong L, 2013 *A review and critique on integrated production–distribution planning models and techniques. Journal of Manufacturing Systems* 32(1):1–19, URL `http://dx.doi.org/10.1016/j.jmsy.2012.07.005`.

Farhangi H, 2021 *Multi-echelon supply chains with lead times and uncertain demands. Operations Research Forum* 2(3):46, URL `http://dx.doi.org/10.1007/s43069-021-00080-1`.

Fischetti M, Lodi A, 2003 *Local branching. Mathematical Programming* 98(1):23–47, URL `http://dx.doi.org/10.1007/s10107-003-0395-5`.

Frangioni A, Manca A, 2006 *A computational study of cost reoptimization for min-cost flow problems. INFORMS Journal on Computing* 18(1):61–70, URL `http://dx.doi.org/10.1287/ijoc.1040.0081`.

Furcy D, Koenig S, 2005 *Limited discrepancy beam search. IJCAI* (Edinburgh), URL https://asset-pdf.scinapse.io/prod/20974380/20974380.pdf.

Gendreau M, Laporte G, Séguin R, 1996 *A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. Operations Research* 44(3):469–477, URL http://dx.doi.org/10.1287/opre.44.3.469.

Gruson M, Bazrafshan M, Cordeau JF, Jans R, 2019 *A comparison of formulations for a three-level lot sizing and replenishment problem with a distribution structure. Computers & Operations Research* 111:297–310, URL http://dx.doi.org/10.1016/j.cor.2019.07.005.

Gruson M, Cordeau JF, Jans R, 2021 *Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure. European Journal of Operational Research* 291(1):206–217, URL http://dx.doi.org/10.1016/j.ejor.2020.09.019.

Harvey WD, Ginsberg ML, 1995 *Limited discrepancy search. IJCAI*, 607–615 (Montreal), URL https://ai.dmi.unibas.ch/research/reading_group/harvey-ginsberg-ijcai1995.pdf.

Homem-de Mello T, Bayraksan G, 2014 *Monte Carlo sampling-based methods for stochastic optimization. Surveys in Operations Research and Management Science* 19(1):56–85, URL http://dx.doi.org/10.1016/j.sorms.2014.05.001.

Hrabec D, Hvattum LM, Hoff A, 2022 *The value of integrated planning for production, inventory, and routing decisions: A systematic review and meta-analysis. International Journal of Production Economics* 108468, URL http://dx.doi.org/10.1016/j.ijpe.2022.108468.

IBM, 2020 *CPLEX*. https://www.ibm.com/docs/en/icos/20.1.0.

Juan AA, Faulin J, Grasman SE, Rabe M, Figueira G, 2015 *A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. Operations Research Perspectives* 2:62–72, URL http://dx.doi.org/10.1016/j.orp.2015.03.001.

Juan AA, Keenan P, Martí R, McGarraghy S, Panadero J, Carroll P, Oliva D, 2021 *A review of the role of heuristics in stochastic optimisation: From metaheuristics to learnheuristics. Annals of Operations Research* URL http://dx.doi.org/10.1007/s10479-021-04142-9.

Kaut M, 2021 *Scenario generation by selection from historical data. Computational Management Science* 18(3):411–429, URL http://dx.doi.org/10.1007/s10287-021-00399-4.

Kiesmüller G, de Kok A, Fransoo J, 2005 *Transportation mode selection with positive manufacturing lead time. Transportation Research Part E: Logistics and Transportation Review* 41(6):511–530, URL http://dx.doi.org/10.1016/j.tre.2005.07.003.

Libralesso L, 2020 *Anytime tree search for combinatorial optimization*. Ph.D. thesis, Université Grenoble Alpes, URL https://tel.archives-ouvertes.fr/tel-03014697/.

L'Écuyer P, Munger D, 2016 *Algorithm 958: Lattice builder: A general software tool for constructing rank-1 lattice rules. ACM Transactions on Mathematical Software* 42(2):1–30, URL http://dx.doi.org/10.1145/2754929.

Niederreiter H, 1978 *Quasi-Monte Carlo methods and pseudo-random numbers. Bulletin of the American mathematical society* 84(6):957–1041.

Ow PS, Morton TE, 1988 *Filtered beam search in scheduling. The International Journal Of Production Research* 26(1):35–62, URL http://dx.doi.org/10.1080/00207548808947840.

Roundy R, 1985 *98%-Effective integer-ratio lot-sizing for one-warehouse multi-retailer systems. Management Science* 31(11):1416–1430, URL http://dx.doi.org/10.1287/mnsc.31.11.1416.

Solyali O, Sural H, 2012 *The one-warehouse multi-retailer problem: Reformulation, classification, and computational results. Annals of Operations Research* 196(1):517–541, URL http://dx.doi.org/10.1007/s10479-011-1022-0.

Sun J, Van Mieghem JA, 2019 *Robust dual sourcing inventory management: Optimality of capped dual index policies and smoothing. Manufacturing & Service Operations Management* 21(4):912–931, URL http://dx.doi.org/10.1287/msom.2018.0731.

Svoboda J, Minner S, Yao M, 2021 *Typology and literature review on multiple supplier inventory control models. European Journal of Operational Research* 293(1):1–23, URL http://dx.doi.org/10.1016/j.ejor.2020.11.023.

Thevenin S, Adulyasak Y, Cordeau JF, 2022 *Stochastic dual dynamic programming for multiechelon lot sizing with component substitution. INFORMS Journal on Computing* ijoc.2022.1215, URL http://dx.doi.org/10.1287/ijoc.2022.1215.

Vadlamudi SG, Gaurav P, Aine S, Chakrabarti PP, 2012 *Anytime column search. Australasian Joint Conference on Artificial Intelligence*, 254–265 (Springer), URL https://link.springer.com/chapter/10.1007/978-3-642-35101-3_22.

Wayne KD, 2002 *A polynomial combinatorial algorithm for generalized minimum cost flow. Mathematics of Operations Research* 27(3):445–459, URL http://dx.doi.org/10.1287/moor.27.3.445.313.

Wu T, Zhang C, Chen W, Liang Z, Zhang X, 2022 *Unsupervised learning-driven matheuristic for production-distribution problems. Transportation Science* 56(6):1677–1702, URL http://dx.doi.org/10.1287/trsc.2022.1149.

Zhou Y, Zhao L, Zhao X, Jiang J, 2011 *A supplier selection and order allocation problem with stochastic demands. International Journal of Systems Science* 42(8):1323–1338, URL http://dx.doi.org/10.1080/00207721.2010.517864.

Zou J, Ahmed S, Sun XA, 2019 *Stochastic dual dynamic integer programming. Mathematical Programming* 175(1-2):461–502, URL http://dx.doi.org/10.1007/s10107-018-1249-5.

Appendix A: Additional models

Model 3: SPDPL static dynamic model (counterpart of Model 1)

$$Min \sum_{t\in T}\sum_{n\in\mathcal{N}(t-1)} \pi^n \left(s_t y^{\mathcal{T}(n)} + p_t x^n + \sum_{r\in R}\sum_{f\in F}\left(u^t_{rf}\theta^n_{rf} + w^t_{rf} z^{\mathcal{T}(n)}_{rf}\right)\right) \tag{26}$$

$$+\sum_{t\in T}\sum_{n\in\mathcal{N}(t)} \pi^n \left(h_0 i^n_0 + \sum_{r\in R}(h_r i^n_r + a.l^n_r)\right)$$

$$s.t\ x^n \leqslant Q_{\mathcal{T}(n)+1} y^{\mathcal{T}(n)} \qquad \forall n\in\mathcal{N}^* \tag{27}$$

$$\theta^n_{rf} \leqslant M z^{\mathcal{T}(n)}_{rf} \qquad \forall n\in\mathcal{N}^*, r\in R, f\in F \tag{28}$$

$$i^{\mathcal{A}(n)}_0 + x^{\mathcal{A}(n)} = \sum_{r\in R}\sum_{f\in F}\theta^{\mathcal{A}(n)}_{rf} + i^n_0 \qquad \forall n\in\mathcal{N} \tag{29}$$

$$i^{\mathcal{A}(n)}_r + \sum_{f\in F}\sum_{t=1}^{\mathcal{T}(n)} \delta_{\{t+v^t_{rf}=\mathcal{T}(n)\}}\theta^{\mathcal{A}(n,t-1)}_{rf} = d^n_r + i^n_r - l^n_r \qquad \forall n\in\mathcal{N}, r\in R \tag{30}$$

$$y^t \in \mathbb{B} \qquad \forall t\in T^* \tag{31}$$

$$x^n \geqslant 0 \qquad \forall n\in\mathcal{N}^* \tag{32}$$

$$z^t_{rf} \in \mathbb{B} \qquad \forall t\in T^*, r\in R, f\in F \tag{33}$$

$$\theta^n_{rf} \geqslant 0 \qquad \forall n\in\mathcal{N}^*, r\in R, f\in F \tag{34}$$

$$i^n_0 \geqslant 0 \qquad \forall n\in\mathcal{N} \tag{35}$$

$$i^n_r \geqslant 0 \qquad \forall n\in\mathcal{N}, r\in R \tag{36}$$

$$l^n_r \geqslant 0 \qquad \forall n\in\mathcal{N}, r\in R \tag{37}$$

For backlogging demand, let us introduce the backlog per-unit penalty α, and the quantity of backlogged demand at customer $r\in R$ at node $n\in\mathcal{N}$: $\lambda^n_r \in \mathbb{R}$.

Model 4: SPDPL static-dynamic model with backlog

$$Min \sum_{t\in T}\sum_{n\in\mathcal{N}(t-1)} \pi^n \left(s_t y^{\mathcal{T}(n)} + p_t x^n + \sum_{r\in R}\sum_{f\in F}\left(u^t_{rf}\theta^n_{rf} + w^t_{rf} z^{\mathcal{T}(n)}_{rf}\right)\right) \tag{38}$$

$$+\sum_{t\in T}\sum_{n\in\mathcal{N}(t)} \pi^n \left(h_0 i^n_0 + \sum_{r\in R}(h_r i^n_r + a.l^n_r + \alpha.\lambda^n_r)\right)$$

$$s.t\ x^n \leqslant Q_{\mathcal{T}(n)+1} y^{\mathcal{T}(n)} \qquad \forall n\in\mathcal{N}^* \tag{39}$$

$$\theta^n_{rf} \leqslant M z^{\mathcal{T}(n)}_{rf} \qquad \forall n\in\mathcal{N}^*, r\in R, f\in F \tag{40}$$

$$i^{\mathcal{A}(n)}_0 + x^{\mathcal{A}(n)} = \sum_{r\in R}\sum_{f\in F}\theta^{\mathcal{A}(n)}_{rf} + i^n_0 \qquad \forall n\in\mathcal{N} \tag{41}$$

$$i^{\mathcal{A}(n)}_r + \sum_{f\in F}\sum_{t=1}^{\mathcal{T}(n)} \delta_{\{t+v^t_{rf}=\mathcal{T}(n)\}}\theta^{\mathcal{A}(n,t-1)}_{rf} = d^n_r + i^n_r - l^n_r - \lambda^n_r + \lambda^{\mathcal{A}(n)}_r \qquad \forall n\in\mathcal{N}, r\in R \tag{42}$$

$$y^t \in \mathbb{B} \qquad \forall t\in T^* \tag{43}$$

$$x^n \geqslant 0 \qquad \forall n\in\mathcal{N}^* \tag{44}$$

$$z^t_{rf} \in \mathbb{B} \qquad \forall t \in T^*, r \in R, f \in F \tag{45}$$

$$\theta^n_{rf} \geqslant 0 \qquad \forall n \in \mathcal{N}^*, r \in R, f \in F \tag{46}$$

$$i^n_0 \geqslant 0 \qquad \forall n \in \mathcal{N} \tag{47}$$

$$i^n_r \geqslant 0 \qquad \forall n \in \mathcal{N}, r \in R \tag{48}$$

$$l^n_r \geqslant 0 \qquad \forall n \in \mathcal{N}, r \in R \tag{49}$$

$$\lambda^n_r \geqslant 0 \qquad \forall n \in \mathcal{N}, r \in R \tag{50}$$

In this model, with the way that the backlogging arcs are constructed, the action of backlogging can be interpreted as sending inventory back in time. The modification of the amount of products sent backward in time from period t at customer r is denoted $\widehat{\lambda^t_r}$.

Model 5: Flow improvement sub-problem with backlogs

$$Min\ a.\widehat{l} + \sum_{t\in T_{\tilde{t}}} \left(\overline{p_t}\widehat{x^{t-1}} + h_0 \widehat{i^t_0} + \sum_{r\in R} \left(h_r \widehat{i^t_r} - a.\widehat{d^t_r} + \alpha.\widehat{\lambda^t_r} + \sum_{f\in F} \overline{u^t_{rf}} \widehat{\theta^{t-1}_{rf}} \right) \right) \tag{51}$$

$$s.t \sum_{t\in T_{\tilde{t}}} \widehat{x^{t-1}} + \widehat{l} = 0 \tag{52}$$

$$\sum_{t\in T_{\tilde{t}}} \sum_{r\in R} \widehat{d^t_r} - \widehat{l} = 0 \tag{53}$$

$$\widehat{i^{t-1}_0} + \widehat{x^{t-1}} = \widehat{i^t_0} + \sum_{r\in R}\sum_{f\in F} \widehat{\theta^{t-1}_{rf}} \qquad \forall t \in T_{\tilde{t}} \tag{54}$$

$$\widehat{i^{t-1}_r} + \widehat{\lambda^{t+1}_r} + \sum_{f\in F} \sum_{t'=\min(T_{\tilde{t}})}^{t} \delta_{\{t'+v^{t'}_{rf}=t\}} \widehat{\theta^{t'-1}_{rf}} = \widehat{d^t_r} + i^t_r + \widehat{\lambda^t_r} \qquad \forall t \in T_{\tilde{t}}, r \in R \tag{55}$$

$$-1 * \min_{n\in\mathcal{N}(t)} \overline{x^n} \leqslant \widehat{x^{t-1}} \leqslant \delta_{\overline{y^t}=1} \left(Q_t - \max_{n\in\mathcal{N}(t)} \overline{x^n} \right) \qquad \forall t \in T_{\tilde{t}} \tag{56}$$

$$-1 * \min_{n\in\mathcal{N}(t)} \overline{\theta^n_{rf}} \leqslant \widehat{\theta^{t-1}_{rf}} \leqslant \delta_{\overline{z^t_{rf}}=1} M \qquad \forall t \in T_{\tilde{t}}, r \in R, f \in F \tag{57}$$

$$-1 * \min_{n\in\mathcal{N}(t)} \overline{i^n_0} \leqslant \widehat{i^t_0} \leqslant M \qquad \forall t \in T_{\tilde{t}} \tag{58}$$

$$-1 * \min_{n\in\mathcal{N}(t)} \overline{i^n_r} \leqslant \widehat{i^t_r} \leqslant M \qquad \forall t \in T_{\tilde{t}} \tag{59}$$

$$-1 * \min_{n\in\mathcal{N}(t)} (d^n_r - \overline{l^n_r}) \leqslant \widehat{d^t_r} \leqslant \min_{n\in\mathcal{N}(t)} \overline{l^n_r} \qquad \forall t \in T_{\tilde{t}}, r \in R \tag{60}$$

$$-1 * \sum_{t\in T_{\tilde{t}}} \sum_{r\in R} \min_{n\in\mathcal{N}(t)} \overline{l^n_r} \leqslant \widehat{l} \leqslant \sum_{t\in T_{\tilde{t}}} \sum_{r\in R} \min_{n\in\mathcal{N}(t)} (d^n_r - \overline{l^n_r}) \tag{61}$$

$$-1 * \min_{n\in\mathcal{N}(t-1)} \overline{\lambda^n_r} \leqslant \widehat{\lambda^t_r} \leqslant \sum_{t'=\min(T_{\tilde{t}})}^{t-1} \min_{n\in\mathcal{N}(t')} d^n_r \qquad \forall t \in T_{\tilde{t}}, r \in R \tag{62}$$

Appendix B: Tables of results

Tables 5, 7, 8 and 9 provide the average results on the four classes of instances. Tables 5 and 7 show the results on instances with 50 customers on a 15- or a 30-period planning horizon, respectively. Tables 8 and 9 present the results on instances with 100 customers on a 15- or a 30-period planning horizon, respectively.

Table 6 reports the average results over all four classes on instances with 50 customers on a 15-period planning horizon under various values of the coefficient of variation of the customers' demand. Columns 2 to 6 present the cost of the solutions: their total cost, their production costs (fixed and unit), their transportation costs (fixed and unit), their holding costs (at the plant and at the customers) and their lost sales penalties. Finally, Column 7 gives the total time used to solve the instances (all included from the first to the last period).

Type	Total cost	Production cost	Transportation cost	Holding cost	Lost sales	Time (sec)
SD_SF	268 414.3	62 803.3	140 853.7	53 643.4	11 114.0	5 796.8
SD_DF	263 814.9	62 718.0	135 307.4	53 938.1	11 851.4	5 793.4
DD_SF	276 266.7	69 302.3	139 582.6	51 707.3	15 674.5	5 950.5
DD_DF	259 411.5	62 765.9	130 001.0	52 988.1	13 656.5	5 877.7
Average	266 976.8	64 397.4	136 436.2	53 069.2	13 074.1	5 854.6

Table 5 **Average cost of the solutions with all transportation modes on instances with 50 customers and 15 periods**

Type	Total cost	Production cost	Transportation cost	Holding cost	Lost sales
$\rho = 0\%$	258 276.3	66 838.4	135 515.7	50 129.6	5 792.6
$\rho = 30\%$	266 976.8	64 397.4	136 436.2	53 069.2	13 074.1
$\rho = 90\%$	278 918.4	62 991.7	135 604.5	54 330.3	25 991.9

Table 6 **Average cost of the solutions with all transportation modes on instances with 50 customers and 15 periods with different coefficients of variation of the customers' demand**

Type	Total cost	Production cost	Transportation cost	Holding cost	Lost sales	Time (sec)
SD_SF	671 539.6	188 146.7	287 628.7	137 148.8	58 615.3	16 476.7
SD_DF	666 882.8	188 787.9	293 154.7	135 087.6	49 852.6	16 086.3
DD_SF	681 457.4	191 384.0	290 201.8	135 023.9	64 847.8	15 263.1
DD_DF	657 777.4	171 793.2	299 401.0	134 133.3	52 449.8	15 298.8
Average	669 414.3	185 027.9	292 596.6	135 348.4	56 441.4	16 476.7

Table 7 **Average cost of the solutions with all transportation modes on instances with 50 customers and 30 periods**

Type	Total cost	Production cost	Transportation cost	Holding cost	Lost sales	Time (sec)
SD_SF	501 426.6	88 741.8	283 849.3	104 661.1	24 174.4	6 728.4
SD_DF	512 120.7	92 236.4	291 453.8	106 287.4	22 143.0	6 561.0
DD_SF	505 017.9	91 724.5	283 019.4	108 185.8	22 088.2	6 606.3
DD_DF	511 056.6	96 356.2	284 221.0	107 362.1	23 117.4	6 501.5
Average	507 405.5	92 264.7	285 635.9	106 624.1	22 880.7	6 599.3

Table 8 Average cost of the solutions with all transportation modes on instances with 100 customers and 15 periods

Type	Total cost	Production cost	Transportation cost	Holding cost	Lost sales	Time (sec)
SD_SF	1 214 416.7	257 741.5	625 682.2	254 129.9	77 545.2	30 632.8
SD_DF	1 196 833.3	252 190.3	617 073.3	256 725.5	70 422.4	25 734.4
DD_SF	1 199 150.0	257 041.4	610 208.8	257 311.4	74 679.2	25 603.1
DD_DF	1 240 709.5	268 466.9	628 985.0	265 026.5	79 133.0	24 619.0
Average	1 212 777.4	258 860.0	620 487.3	258 298.3	75 444.9	26 647.3

Table 9 Average cost of the solutions with all transportation modes on instances with 100 customers and 30 periods

Table 10 presents the average cost of the solutions obtained with backlog on each type of instance. The backlog penalty was set to a third of the lost sales penalty (per unit and per time period). Columns 2 to 6 present the cost of the solutions: their total cost, their production costs (fixed and unit), their transportation costs (fixed and unit), their holding costs (at the plant and at the customers) and their lost sales penalties. Finally, Column 7 gives the total time used to solve the instance (all included from the first to the last period). The last line recalls the average cost of the solutions obtained without backlog on these instances.

Type	Total cost	Production cost	Transportation cost	Holding cost	Lost sales	Backlog	Time (sec)
SD_SF	262 482.8	63 579.8	135 673.3	53 757.5	6 020.4	3 451.7	5 862.2
SD_DF	249 555.5	59 661.6	127 965.4	54 813.7	3 875.8	3 239.0	6 044.5
DD_SF	276 699.8	71 904.1	138 958.8	51 745.3	11 462.7	2 628.8	5 852.0
DD_DF	255 885.9	63 218.5	128 828.3	53 083.6	7 652.2	3 103.2	5 970.7
Average	261 156.0	64 591.0	132 856.4	53 350.0	7 252.8	3 105.7	5 932.4
Without	266 976.8	64 397.4	136 436.2	53 069.2	13 074.1	-	5 854.6

Table 10 Average cost of the solutions with all transportation modes on instances with 50 customers and 15 periods with backlogs

Table 11 details some indicators on the solution obtained with and without backlog. Columns 2 and 3 give the number of periods during which production takes place and the total number of units produced. Columns 4 and 5 give the total number of units stored per period at the plant and at all the customers.

Finally, columns 6 to 8 give the total number of units transported by each carrier (air, road, and maritime).

Backlog	Nb prod	Qt prod	Store depot	Store customers	Qt trans A	Qt trans R	Qt trans M
With	7.33	23 719.62	6 306.07	67 982.56	378.50	5 266.30	20 648.99
Without	7.32	23 709.13	6 710.98	66 941.15	390.47	5 287.95	20 620.99
Relative diff.	0.13%	0.04%	−6.02%	1.55%	−3.06%	−0.39%	0.13%

Table 11 Comparison of indicators on the instances with 50 customers and 15 periods without and with backlogs

Table 12 presents the average cost of the solutions obtained without fixed transportation costs on each type of instance. Columns 2 to 6 present the cost of the solutions: their total cost, their production costs (fixed and unit), their transportation costs (fixed and unit), their holding costs (at the plant and at the customers) and their lost sales penalties. Finally, Column 7 gives the total time used to solve the instance (all included from the first to the last period). The last line reminds the average cost of the solutions obtained with fixed transportation costs on these instances.

Type	Total cost	Production cost	Transportation cost	Holding cost	Lost sales	Time (sec)
SD_SF	226 724.1	61 279.4	113 954.6	44 562.6	6 927.5	884.1
SD_DF	220 125.2	59 209.1	110 660.8	43 797.7	6 457.7	1 101.9
DD_SF	228 728.4	68 655.5	112 655.6	42 715.8	4 701.4	1 058.0
DD_DF	213 744.0	57 965.7	107 697.6	42 979.3	5 101.4	1 373.9
Average	222 330.4	61 777.4	111 242.1	43 513.9	5 797.0	1 104.5
With	266 976.8	64 397.4	136 436.2	53 069.2	13 074.1	5 854.6

Table 12 Average cost of the solutions with all transportation modes on instances with 50 customers and 15 periods without transportation fixed costs

Table 13 compares the solution indicators to the case without transportation fixed costs. Columns 2 and 3 give the number of periods during which production takes place and the total number of units produced. Columns 4 and 5 give the total number of units stored per period at the plant and at all the customers. Finally, columns 6 to 8 give the total number of units transported by each carrier (air, road, and maritime).

Fixed Costs	Nb prod	Qt prod	Store depot	Store customers	Qt trans A	Qt trans R	Qt trans M
Without	7.08	23 781.79	6 914.86	53 900.10	450.93	6 363.29	19 557.52
With	7.32	23 709.13	6 710.98	66 941.15	390.47	5 287.95	20 620.99
Relative Diff.	−3.28%	0.30%	3.04%	−19.48%	15.38%	20.35%	−5.15%

Table 13 Comparison of indicators on the instances with 50 customers and 15 periods with and without transportation fixed costs