Split Demand and Deliveries in an Integrated Three-Level Lot Sizing and Replenishment Problem

Matthieu Gruson[a,*], Jean-François Cordeau[b], Raf Jans[b]

[a] *École des Sciences de la gestion, Université du Québec à Montréal, C.P. 8888, succursale Centre-ville, Montréal, H3C 3P8 Canada*
[b] *HEC Montréal and GERAD, 3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7 Canada*

Abstract

We address a three-level lot sizing and replenishment problem (3LSRP), an extension of the production routing problem. We consider one production plant, with limited capacity, that produces items over a discrete and finite planning horizon. These items are used to replenish warehouses and then retailers. The items are sent from the plant to the warehouses using direct capacitated shipments, and routes are designed to deliver the goods from the warehouses to the retailers. The routes are performed by a homogeneous fleet of capacitated vehicles. The objective is to minimize the sum of the fixed production and replenishment costs, of the variable inventory holding costs, and of the routing costs. We introduce the concept of demand splitting which allows that demand from a specific period can be satisfied by deliveries over multiple periods. We develop a branch-and-cut algorithm to solve this problem and compare it to two heuristics we propose: a top-down approach and a bottom-up approach. The production decisions and the replenishment decisions are the leading decisions in the top-down and bottom-up approach, respectively. We run computational experiments to assess the performance of each heuristic. We analyze the impact of allowing splitting possibilities. The results show that the top-down approach obtains better performance in terms of cost, except when we allow demand splitting only. The bottom-up approach favours the use of delivery splitting. Results also show that we obtain large gains thanks to the splitting possibilities.

Keywords: Production; distribution; integration; heuristics; routing; replenishment; production routing problem

*Corresponding author
Email address: gruson.matthieu@uqam.ca (Matthieu Gruson)

1. Introduction

One of the major challenges in supply chain planning is the coordination and integration of operational decisions. A sequential approach in the decision making process, compared to an integrated approach, will result in sub-optimal or inconsistent plans (see, e.g., Vogel et al., 2017; Absi et al., 2018). The potential benefits of integration, both in terms of cost and customer service, explain why this area has drawn a lot of attention over the last decades. Early studies include those of Chandra & Fisher (1994), Brown et al. (2001) and Çetinkaya et al. (2009). These studies refer to the production routing problem (PRP) and report considerable cost savings for the companies in a context where a central plant replenishes several customers. There is therefore an integration of production, inventory and distribution decisions.

With the growing size and complexity of supply chains, there is now a need to incorporate even more levels of the supply chain in order to benefit from the gains achieved by the integration of operational decisions. In that spirit, Perboli et al. (2011) have introduced the two-echelon vehicle routing problem (2E-VRP). In this problem, goods are sent from a plant to customers through distribution centres (DCs). There are routing decisions between the plant and the DCs, and between the DCs and the customers. However, in a global market, the transportation between the plants and the DCs is not always done using routes visiting multiples DCs. Indeed, the plant and DCs cannot always be reached by ground transportation, especially if the plant and DCs are separated by water. Besides, the quantities transported between the plant and the warehouses are typically much larger than the quantities transported between the warehouses and the retailers. Therefore, the shipments sent between the plant and DCs are more likely to be direct shipments. Furthermore, the 2E-VRP only considers a single time period and does not incorporate any production decisions.

We propose here to follow this line of research by studying a three-level lot sizing and replenishment problem (3LSRP). We consider one production plant (level zero) with limited capacity that produces several items over a discrete and finite planning horizon. The items produced are used to replenish warehouses (level one) and then retailers (level two), in order to satisfy the demand at the retailer level. The items are sent from the plant to the warehouses using direct shipments, and routes are designed to deliver the goods from the warehouses to the retailers. The shipments between the plant and the warehouses are subject to capacity restrictions. In a similar spirit, the shipments between each warehouse and the retailers are performed using homogeneous capacitated vehicles. We also consider production capacity restrictions at the plant. The objective is to min-

imize the sum of the fixed production and replenishment costs, of the variable inventory holding costs at all three levels, and of the routing costs.

A related problem, the three-level lot sizing and replenishment problem with a distribution structure (3LSPD), was analysed by Gruson et al. (2019a). However, there are several differences between that prior work and the work presented here. First, we add transportation capacity requirements. Second, the fixed assignment of retailers to warehouses is relaxed: the retailers can be served by any warehouse in each time period. Third, there are no direct shipments between the warehouses and the retailers. Instead, there are routes constructed to perform the deliveries between the warehouses and the retailers. Finally, in our prior work, the focus was on the modelling aspect while in this paper the focus is on solving the problem efficiently through heuristics.

The motivation for this problem setting is to more closely match the situations faced by companies in practice. Indeed, due to urban constraints, deliveries between warehouses and retailers cannot always be performed by large vehicles. This is different for deliveries between larger facilities, such as the production plant and the warehouses in our case. We also integrate realistic constraints such as the transportation capacity constraints. Note that we assume that we have an unlimited fleet of vehicles to make the deliveries between the plant and the warehouses.

This paper makes three main contributions. First, we extend the work presented in Gruson et al. (2019a) by studying a more realistic version of the 3LSPD, which we call the Integrated Three-Level Lot Sizing and Replenishment Problem (3LSRP). In particular, we add capacity requirements and include routing decisions. The problem is also an extension of the traditional PRP to include a third level. Second, we develop two heuristic algorithms to efficiently solve the problem. The first heuristic is a top-down procedure that represents a case where the production decisions are made first. The second heuristic is a bottom-up procedure that represents a case where the replenishment decisions at the retailer level are made first. This distinction has already been proposed by Darvish & Coelho (2018) in the context of an integrated production, location and distribution problem. The performance of the heuristics is compared to a branch-and-cut algorithm that we develop to solve the problem exactly. Third, we evaluate the achieved gains by allowing split demands and split deliveries. The literature on the PRP and related problems usually assumes that each customer can be visited by one truck only in each time period, but that the delivery of the demand of a specific time period t can be spread over several periods before period t. The first assumption restricts the delivery possibilities, while the second one gives more flexibility. Building on these two

assumptions, we define the notion of split demand and split delivery. Split demand means that for any retailer r and any period t, the demand of retailer r in period t can be shipped over several periods before period t. Split delivery means that in each time period, a retailer can be visited by several trucks coming from the same warehouse. We therefore compare the costs of the solutions obtained when we allow delivery splitting or not, and when we allow demand splitting or not. This leads to four different settings that we investigate.

The remainder of this paper is organized as follows. We first review the literature relevant to our study in Section 2. We then present a mathematical formulation along with valid inequalities for the problem in Section 3. In this section, we also describe the additional constraints needed to impose the various cases with respect to the demand and delivery splitting. We also give details about an exact branch-and-cut algorithm that we designed to solve the problem exactly. The top-down and bottom-up heuristics are presented in Sections 4 and 5, respectively. Section 6 reports the results of the computational experiments performed to assess the performance of our heuristics. This is followed by the conclusion in Section 7.

2. Literature review

We review the literature on several multi-echelon routing problems that relate to the 3LSRP we study. First, we review the literature on the 2E-VRP. The interested reader is referred to Cuda et al. (2015) for a more extensive review of the literature on this topic. Second, we review the literature on the PRP. Recall that the 3LSRP studied here is an extension of the 3LSPD. As the 3LSPD was introduced very recently, we refer the interested reader to the references in Gruson et al. (2019a) and Gruson et al. (2019b) for a review of similar works. Note that recently, Gu et al. (2021) have studied a problem similar to the 3LSRP in the context of a fresh food supply chain. Compared to the 3LSRP, they consider several production plants. They also propose top-down and bottom-up heuristics to solve their problem.

2.1. Two-echelon vehicle routing problem

Most of the papers that consider two-echelon routing problems incorporate routing decisions between levels zero and one, and between levels one and two. The vehicles used for the transportation between levels zero and one are usually larger than the ones used between levels one and two. The reason for this difference is that the vehicles used to do the second part of the routing must cope

with limitations coming from urban areas. Most of these works further consider multiple depots for level zero. In that sense, there is deconsolidation and consolidation of goods at the depots at level one: items received from different depots are deconsolidated and then mixed to make the deliveries to the end customers. Recall that compared to the 2E-VRP we consider direct shipments between levels zero and one and that we include production decisions at level zero.

The two-echelon VRP was initially introduced by Perboli et al. (2011). They propose a mathematical model along with valid inequalities and two matheuristics to solve the problem. They define variants of this problem, which include capacity requirements, satellite synchronization and time windows, among others. These variants have been well studied later in the literature. For instance, Grangier et al. (2016) consider time windows at the customer level, and Dellaert et al. (2019) propose four MIP formulations for the case without synchronization.

Several works incorporate environmental considerations. Soysal et al. (2015) consider fuel consumption in the objective function. They develop a MIP formulation and propose valid inequalities for their problem. Wang et al. (2017) study the same problem and solve it by means of a matheuristic based on variable neighbourhood search. Breunig et al. (2019) consider the electric case where there are some recharging stations available for the vehicles. They develop both a large neighbourhood search metaheuristic and an exact algorithm to solve the problem. Jie et al. (2019) further consider the possibility of swapping batteries for the electric vehicles. They solve this problem by a combined column generation and adaptive large neighbourhood search algorithm. Anderluh et al. (2019) introduce the notion of 'grey zone' for customers that are at the frontier of urban areas. The deliveries to the end customers (echelon two) can originate from depots located in echelon zero or one. They develop a metaheuristic that combines large neighbourhood search and multi-objective methods to find a Pareto front for their problem.

2.2. Production routing problem

In the standard PRP there are two levels, namely the plant and the customers. The objective of the PRP is to jointly optimize the production, distribution and inventory decisions. The distribution between the plant and the customers is done by a fleet of capacitated vehicles that follow some routes that must be constructed during the optimization process. The 3LSRP studied here can be seen as an extension of the PRP to three levels, but with direct shipments between levels zero and one. The interested reader is referred to Adulyasak et al. (2015a) for a review of the PRP.

The PRP has attracted a lot of research since the early work of Chandra & Fisher (1994) and the success stories reported in Brown et al. (2001) and Çetinkaya et al. (2009), at Kellogg and Frito-Lay, respectively. From a modelling perspective, several studies on the PRP rely on combining formulations arising from the lot sizing literature and formulations arising from the VRP literature. Indeed, the PRP can be seen as an integration between a lot sizing problem (LSP) and a VRP. On the LSP side, the classical (Wagner & Whitin, 1958) or transportation (Krarup & Bilde, 1977) formulations have been mostly used. This last formulation is in particular useful in the case of perishable products (see, Dayarian & Desaulniers, 2019). On the VRP side, there have been formulations both with and without vehicle indices. In the VRP, the vehicle index is used when the fleet of vehicles is heterogenous (see, e.g., Baldacci et al., 2008). Note that in the context of the PRP, formulations both with and without vehicle indices have been used (see, e.g., Adulyasak et al., 2015a).

From a methodological perspective, because of its complexity, the PRP has been mainly solved by heuristics or metaheuristics. Adulyasak et al. (2015b) develop an adaptive large neighbourhood search algorithm (ALNS). Absi et al. (2015) propose a two-phase iterative heuristic approach. The first phase is a capacitated lot sizing problem (CLSP) that handles the production and inventory decisions. In this first phase, the routing costs are approximated. The second phase consists in solving a travelling salesman problem (TSP) to actually construct the routes. The procedure iterates by updating the approximated routing costs. Solyalı & Süral (2017) design a heuristic that comprises five phases and makes use of an a priori tour. Recently, Chitsaz et al. (2019) designed a three-phase matheuristic to solve the assembly routing problem (ARP), which also proved successful when applied to the PRP. In the ARP, the objective is to jointly optimize visits at the suppliers to pick up materials that are assembled to produce an end-item whose demand is known.

Recently, numerous extensions of the PRP have been studied. We can mention here the inclusion of delivery time windows (Neves-Moreira et al., 2019), of carbon emissions (Qiu et al., 2017), of visit-spacing decisions (Avci & Yildiz, 2019), of routes extending to several periods (Miranda et al., 2018), or of two echelons (Qiu et al., 2021; Schenekemberg et al., 2021), among others.

3. A transportation based formulation

In this section we present a mathematical formulation for the problem. Table 1 lists all the sets, parameters and decision variables used in the mathematical model. This formulation can be seen

Table 1: Sets, parameters and decision variables used in the combined classical and transportation model

Sets			
F	set of all facilities, $F = P \cup W \cup R$	P	set containing the unique production plant, $P = \{p\} \subset F$
W	set containing the warehouses, $W \subset F$	R	set containing the retailers, $R \subset F$
$V(w)$	set of trucks available at warehouse w	I	set of items
T	set of time periods	E	set of edges, $E = \{(i,j) : i,j \in R \cup W, i < j, i \vee j \in R\}$
$E(S)$	set of edges $(i,j) \in E$ such that $i,j \in S$ and $S \subseteq R \cup W$	$E(i)$	set of edges $(j,j') \in E$ such that $j = i$ or $j' = i$
Parameters			
sc^p_{it}	setup cost at the plant for item i in period t	d^r_{it}	demand of retailer r for item i in period t
sc^w_t	setup cost at warehouse w in period t	d^r_t	total demand of retailer r in period t
sc^r_t	setup cost at retailer r in period t	st_i	setup time for item i
hc^j_{it}	holding cost at facility j in period t for item i	vt_i	variable production time for item i
c_{ij}	cost to go from facility i to facility j	v_i	volume of item i
C^P	production capacity available in any period t	d^p_{itk}	demand at the plant for item i between periods t and k
C^{PW}	capacity of the trucks used to make the deliveries between the plant and the warehouses		
C^{WR}	capacity of the trucks used to make the deliveries between the warehouses and the retailers		
Decision variables			
p_{it}	quantity of item i produced at the plant in period t		
q^w_{it}	quantity of item i ordered by warehouse w in period t		
q^r_{iktw}	quantity of item i ordered by retailer r in period k from warehouse w to satisfy d^r_{it}		
I^j_{it}	inventory of item i in facility j at the end of period t		
y^p_{it}	binary setup variable that takes value 1 if there is a setup for item i in period t		
y^w_t	number of trucks used to do deliveries between the plant and warehouse w in period t		
y^r_t	binary setup variable that takes value 1 if there is an order placed by retailer r in period t		
z^w_t	number of vehicles used by warehouse w in period t to make deliveries to retailers		
z^r_{tw}	total number of times retailer r is visited by the vehicles of warehouse w in period t		
Z^r_{tw}	binary variable that takes value 1 if retailer r is visited by a vehicle of warehouse w in period t		
x_{ijtw}	number of times the edge from i to j is used in period t by vehicles of warehouse w		

as a combination between the classical and the transportation formulations proposed by Gruson et al. (2019a) for the 3LSPD. Note that, regarding the routing part, the formulation used here is similar to the transportation formulation proposed by Alvarez et al. (2020) in the context of the inventory routing problem (IRP) with perishable products. The formulation is as follows:

$$\text{Min} \sum_{t\in T}\left(\sum_{i\in I} sc_{it}^{p} y_{it}^{p} + \sum_{w\in W} sc_{t}^{w} y_{t}^{w} + \sum_{r\in R} sc_{t}^{r} y_{t}^{r}\right)$$
$$+ \sum_{t\in T}\sum_{i\in I}\left(hc_{it}^{p} I_{it}^{p} + hc_{it}^{w} I_{it}^{w} + \sum_{r\in R}\sum_{w\in W}\sum_{k\geq t}\sum_{j=t}^{k-1} hc_{ij}^{r} q_{itkw}^{r}\right) + \sum_{t\in T}\sum_{w\in W}\sum_{(i,j)\in E(W\cup R)} c_{ij} x_{ijtw} \quad (1)$$

$$\text{s. t. } I_{i,t-1}^{p} + p_{it} = \sum_{w\in W} q_{it}^{w} + I_{it}^{p} \qquad \forall\, i\in I, t\in T \quad (2)$$

$$I_{i,t-1}^{w} + q_{it}^{w} = \sum_{k\geq t}\sum_{r\in R} q_{itkw}^{r} + I_{it}^{w} \qquad \forall\, w\in W, i\in I, t\in T \quad (3)$$

$$\sum_{w\in W}\sum_{k\leq t} q_{iktw}^{r} = d_{it}^{r} \qquad \forall\, r\in R, i\in I, t\in T \quad (4)$$

$$p_{it} \leq \min\left\{\frac{C^{p} - st_{i}}{vt_{i}}; d_{it|T|}^{p}\right\} y_{it}^{p} \qquad \forall\, i\in I, t\in T \quad (5)$$

$$q_{it}^{w} \leq \frac{C^{PW}}{v_{i}} y_{t}^{w} \qquad \forall\, w\in W, i\in I, t\in T \quad (6)$$

$$q_{iktw}^{r} \leq d_{it}^{r} y_{k}^{r} \qquad \forall\, r\in R, w\in W, i\in I, 1\leq k\leq t\leq |T| \quad (7)$$

$$\sum_{i\in I}(vt_{i} p_{it} + st_{i} y_{it}^{p}) \leq C^{P} \qquad \forall\, t\in T \quad (8)$$

$$\sum_{i\in I} v_{i} q_{it}^{w} \leq C^{PW} y_{t}^{w} \qquad \forall\, w\in W, t\in T \quad (9)$$

$$\sum_{r\in R}\sum_{i\in I}\sum_{k\geq t} v_{i} q_{itkw}^{r} \leq C^{WR} z_{t}^{w} \qquad \forall\, w\in W, t\in T \quad (10)$$

$$\sum_{k\geq t}\sum_{i\in I} v_{i} q_{itkw}^{r} \leq C^{WR} z_{tw}^{r} \qquad \forall\, r\in R, w\in W, t\in T \quad (11)$$

$$z_{tw}^{r} \leq |V(w)| Z_{tw}^{r} \qquad \forall\, r\in R, w\in W, t\in T \quad (12)$$

$$\sum_{w\in W} Z_{tw}^{r} \leq 1 \qquad \forall\, r\in R, t\in T \quad (13)$$

$$z_{t}^{w} \leq |V(w)| \qquad \forall\, w\in W, t\in T \quad (14)$$

$$\sum_{(j,j')\in E(r)} x_{jj'tw} = 2 z_{tw}^{r} \qquad \forall\, r\in R, w\in W, t\in T \quad (15)$$

$$C^{WR}\sum_{(i,j)\in E(S)} x_{ijtw} \leq \sum_{r\in S}\left(C^{WR} z_{tw}^{r} - \sum_{k\geq t}\sum_{i\in I} v_{i} q_{itkw}^{r}\right) \qquad \forall\, w\in W, S\subseteq R, |S|\geq 2, t\in T \quad (16)$$

$$I^p_{it}, p_{it} \geq 0 \qquad \forall\ i \in I, t \in T \tag{17}$$

$$I^w_{it}, q^w_{it} \geq 0 \qquad \forall\ w \in W, i \in I, t \in T \tag{18}$$

$$q^r_{iktw} \geq 0 \qquad \forall\ r \in R, w \in W, i \in I, 1 \leq k \leq t \leq |T| \tag{19}$$

$$y^p_{it} \in \{0; 1\} \qquad \forall\ i \in I, t \in T \tag{20}$$

$$y^w_t \in \mathbb{N} \qquad \forall\ t \in T \tag{21}$$

$$y^r_t \in \{0; 1\} \qquad \forall\ t \in T \tag{22}$$

$$z^w_t \in \mathbb{N} \qquad \forall\ w \in W, t \in T \tag{23}$$

$$z^r_{tw} \in \mathbb{N} \qquad \forall\ r \in R, w \in W, t \in T \tag{24}$$

$$Z^r_{tw} \in \{0; 1\} \qquad \forall\ r \in R, w \in W, t \in T \tag{25}$$

$$x_{ijtw} \in \mathbb{N} \qquad \forall\ (i, j) \in E(W \cup R), t \in T, w \in W. \tag{26}$$

The objective function (1) minimizes the sum of the setup costs, of the inventory holding costs at each facility, and of the routing costs. We consider a fixed setup cost whenever there is a truck used for delivery between the plant and any warehouse. The routing costs are computed based on the distance travelled by the trucks available at the warehouses. Note that we do not consider unit production costs nor unit transportation costs. Indeed, with static cost it would lead to a constant in the objective function since the full demand must be satisfied. Constraints (2)-(3) are the inventory balance constraints for the plant and the warehouses, respectively. Constraints (4) are the demand satisfaction constraints at the retailer level. Constraints (5)-(7) are the setup constraints for the plant, the warehouses and the retailers, respectively. Constraints (6) compute the number of trucks used for the deliveries between the plant and each warehouse. Constraints (8)-(10) are the capacity restrictions for production at the plant, for transportation between the plant and the warehouses, and for transportation between the warehouses and the retailers, respectively. Constraints (11) link the order and visit variables for the retailers. Constraints (12) link the different visit variables. Constraints (13) state that each retailer can be visited by one warehouse only in each time period, possibly by several trucks. Constraints (14) limit the number of trucks available at each warehouse. Constraints (15) are the degree constraints. Constraints (16) are the subtour elimination constraints. These are sometimes referred to in the literature as general fractional

subtour elimination constraints (GFSEC). Finally, constraints (17)-(26) define the domains of the decision variables. Note that in the numerical experiments we further add valid inequalities adapted from the lot sizing and routing literature.

3.1. Split demands and deliveries

The concept of split deliveries has been discussed in detail in the vehicle routing literature ((Archetti & Speranza, 2008)). If split deliveries are prohibited, then only one truck can visit the retailer in a given period. Since most VRP problems are single period problems, this means that the complete demand is delivered in that single period. However, the PRP is a multi-period problem, and in that case the distinction between splitting deliveries and splitting demand becomes relevant. Indeed, prohibiting split deliveries does not prohibit that the total demand of a specific period is delivered in multiple time periods. Therefore, we explicitly introduce the concept of demand splitting. Recall that delivery splitting means that a retailer can be visited by several trucks (from the same warehouse) in the same time period while demand splitting means that, for a specific retailer r, its total demand over all items in a specific time period, i.e., d_t^r, can be shipped over several periods. These two concepts can be combined in several ways. You can have no demand splitting, but still split deliveries when (for a specific demand period) all the items are delivered in the same period but using multiple vehicles. Inversely, you can have split demand, but not split deliveries. In such a case, it is allowed that the complete demand in a specific period is delivered in multiple periods, but in each of these delivery periods only one vehicle can visit the truck. Finally, if both demand and delivery splitting is prohibited, then the total demand of a specific period must be delivered as a whole in one period and by one vehicle.

The proposed model (1)-(26) allows both demand and delivery splitting. However, if we prevent delivery splitting, we add the following constraints, which impose that at most one vehicle can visit each retailer in any period:

$$\sum_{w \in W} z_{tw}^r \leq 1 \qquad \forall\, r \in R, t \in T. \tag{27}$$

If we prevent demand splitting, we need an additional set of variables. Let Y_{ktw}^r be a binary variable taking the value 1 if some of the demand of retailer r in period t is delivered in period k from warehouse w. We further add the following constraints:

$$q_{iktw}^r \leq d_{it}^r Y_{ktw}^r \qquad \forall\, r \in R, w \in W, i \in I, k \leq t \in T \tag{28}$$

Table 2: Constraints to be added to (1)-(26) based on the splitting possibilities

	No demand splitting	Demand splitting
No delivery splitting	(27)-(30)	(27)
Delivery splitting	(28)-(30)	-

$$\sum_{w \in W} \sum_{k \leq t} Y^r_{ktw} \leq 1 \qquad \forall\ r \in R, t \in T \tag{29}$$

$$Y^r_{ktw} \in \{0;1\} \qquad \forall\ r \in R, w \in W, k \leq t \in T. \tag{30}$$

Constraints (28) link the ordering variables and the newly defined Y^r_{ktw} variables. Constraints (29) impose that the deliveries for all items must be done in one period and come from one warehouse. Table 2 summarizes the constraints that must be added to the original model (1)-(26), depending on the splitting possibilities.

3.2. Branch-and-cut algorithm

The number of GFSECs (16) in the model is exponential, making it intractable. Therefore, it is appropriate to use a branch-and-cut approach to try and solve the problem exactly. Several branch-and-cut approaches have been proposed in the context of the PRP and IRP, see in particular Adulyasak et al. (2014) and Alvarez et al. (2020), respectively. To separate the GFSEC, we use the four heuristic separation algorithms proposed by Lysgaard et al. (2004) in the context of the VRP. We call this procedure for each period and each warehouse. Let $\overline{q}^r_{iktw}$, $\overline{x}_{ijtw}$ and $\overline{z}^r_{tw}$ be the current values of variables q^r_{iktw}, x_{ijtw} and z^r_{tw}, respectively. For each warehouse w and each time period t, we start by building a graph from the set of nodes such that $\overline{z}^r_{tw} > 0$. In these graphs, the edges have a weight equal to $\overline{x}_{ijtw}$ and the delivery quantities to each retailer are set to $\sum_{i \in I} \sum_{k \geq t} \overline{q}^r_{itkw}$.

Note that we can also separate the traditional subtour elimination constraints (SECs). The SECs are defined as follows:

$$\sum_{i \in S} \sum_{j \in S, j \neq i} x_{ijtw} \leq \sum_{i \in S} z^i_{tw} - z^e_{tw} \qquad \forall\ S \subseteq R, |S| \geq 2, w \in W, t \in T, e \in S. \tag{31}$$

To separate the SEC, we use an exact separation algorithm that solves a series of minimum $s-t$ cut problems. The interested reader is referred to Solyalı & Süral (2011) for details on the separation procedure. Note that both the GFSECs and the SECs are separated at the root node. In the

branch-and-bound tree, these constraints are separated only when an integer solution is found. This is done in order not to add too many cuts in the tree.

4. A top-down heuristic

Because of the intractability of the model, we develop a top-down heuristic algorithm to solve the problem efficiently. This heuristic is an iterative procedure that decomposes the problem into two smaller problems. We start with an initial assignment of the retailers to the warehouses, for each time period (Step 1). Then, the first problem determines the production quantities at the plant along with the ordering decisions at the warehouse level (Step 2). The solution of this first problem is used as an input to the second problem, which determines the ordering quantities at the retailer level (Step 3). These two problems are specific versions of the one-warehouse multi-retailer problem (OWMR). In the OWMR, a central warehouse replenishes a set of retailers over a discrete and finite horizon. The formulation used for the different OWMRs is the multi-commodity formulation proposed by Cunha & Melo (2016). We do so in light of both the theoretical and practical observations made in Cunha & Melo (2016) and Gruson et al. (2019a). Between two resolutions of the first OWMR (Step 2), we add diversification constraints that enforce a change in the plant or warehouse setup plans. Between two iterations of the resolution of the second OWMR (Step 3), we update an approximation of the routing costs. After a certain number of consecutive sequences of Steps 2 and 3, we diversify the search by changing the retailer assignment to the warehouses, for each time period (Step 4). Note that we perform several iterations of Step 2, and that between two iterations of Step 2, we further perform several iterations of Step 3. The following sections present in more detail the different steps of the top-down heuristic.

4.1. Step 1: initial assignment of the retailers

We start the heuristic by assigning the retailers to the different warehouses, for each time period. Initially, we assign each retailer to its closest warehouse, for each time period. We then check if this initial assignment is consistent with the transportation capacity requirements. Otherwise, we heuristically reassign retailers so as to satisfy these requirements, by moving some retailers to their second closest warehouse. We chose the retailers to be moved as the furthest in the ordered set of retailers. In case the transportation requirements are still not satisfied, we continue the reassignment to the warehouses further, until the transportation capacity requirements are satisfied. This initial

assignment allows us to solve one travelling salesman problem (TSP) for each warehouse and its assigned retailers in each time period. The tours obtained are used later in the heuristic to define the routes, based on the values of the retailer ordering variables. This idea of solving a TSP beforehand has been proposed in the context of the IRP by Solyalı & Süral (2011). In the sequel we denote by $R(w,t)$ the set of retailers r whose demand in period t will be satisfied by warehouse w.

4.2. Step 2: production quantities and ordering quantities at the warehouses

In the second step of the heuristic, we solve a OWMR considering only the plant and the warehouses. The objective is to obtain the production quantities at the plant level, and the ordering quantities at the warehouse level, together with the respective setup decisions. Note that the assignment of retailers to warehouses allows us to derive the notion of a demand for the warehouses. We set the demand for the warehouses as $d^w_{it} = \sum_{r \in R(w,t)} d^r_{it}$. Let X^p_{wikt} be the amount produced at the production plant in period k to satisfy d^w_{it} and let X^w_{ikt} be the amount transported from the production plant to the warehouse w in period k to satisfy d^w_{it}. Let s^p_{wikt} and s^w_{ikt} be the amounts stocked at the production plant and at warehouse w at the end of period k to satisfy d^w_{it}, respectively. We use the same setup variables for the plant and the warehouses as the ones used in Section 3. Finally, we denote by δ_{kt} the Kroenecker delta that takes the value 1 iff $k = t$. The OWMR we solve in this second step, that we call $OWMR_1$, is as follows:

$$\text{Min} \sum_{t \in T} \left(\sum_{i \in I} sc^p_{it} y^p_{it} + \sum_{w \in W} sc^w_t y^w_t + \sum_{i \in I} \sum_{w \in W} \sum_{k \leq t} (hc^p_{ik} s^p_{wikt} + hc^w_{ik} s^w_{ikt}) \right) \tag{32}$$

$$\text{s. t. } s^p_{w,i,k-1,t} + X^p_{wikt} = X^w_{ikt} + s^p_{wikt} \qquad \forall\, w \in W, i \in I, 1 \leq k \leq t \leq |T| \tag{33}$$

$$s^w_{i,k-1,t} + X^w_{ikt} = \delta_{kt} d^w_{it} + (1 - \delta_{kt}) s^w_{ikt} \qquad \forall\, w \in W, i \in I, 1 \leq k \leq t \leq |T| \tag{34}$$

$$X^p_{wikt} \leq d^w_{it} y^p_{ik} \qquad \forall\, w \in W, i \in I, 1 \leq k \leq t \leq |T| \tag{35}$$

$$X^w_{ikt} \leq d^w_{it} y^w_k \qquad \forall\, w \in W, i \in I, 1 \leq k \leq t \leq |T| \tag{36}$$

$$\sum_{w \in W} \sum_{i \in I} \sum_{k \geq t} vt_i X^p_{witk} + \sum_{i \in I} st_i y^p_{it} \leq C^P \qquad \forall\, t \in T \tag{37}$$

$$\sum_{i \in I} \sum_{k \geq t} v_i X^w_{itk} \leq C^{PW} y^w_t \qquad \forall\, w \in W, t \in T \tag{38}$$

$$X^p_{wikt}, X^w_{ikt}, s^p_{wikt}, s^w_{ikt} \geq 0 \qquad \forall\, w \in W, i \in I, 1 \leq k \leq t \leq |T| \tag{39}$$

$$y^p_{it} \in \{0; 1\} \qquad \forall\, i \in I, t \in T \tag{40}$$

Table 3: Demands for a small instance

Period	Retailer					Total demand
	1	2	3	4	5	
1	5	25	10	10	20	70
2	30	20	15	20	25	110
3	5	5	5	5	5	25
4	20	30	15	20	40	125

$$y_t^w \in \mathbb{N} \qquad \forall\ w \in W, t \in T. \tag{41}$$

If we solve this problem without additional constraints, we may obtain infeasible solutions for the third step of the heuristic where we decide on the ordering decisions at the retailer level. Indeed, we may have ordering quantities at the warehouse level that are higher than the transportation capacities to deliver the goods to the different retailers. Let $V_{kt}^w = \sum_{i \in I} v_i d_{ikt}^w$ and let $C_{kt}^{WR,w} = (t-k+1)|V(w)|C^{WR}$. We add the following constraints:

$$\sum_{i \in I} \sum_{l<k} \sum_{u=k}^{t} v_i X_{ilu}^w \geq V_{kt}^w - C_{kt}^{WR,w} \qquad \forall\ 1 \leq k \leq t \leq |T|, w \in W,\ |\ V_{kt}^w > C_{kt}^{WR,w}. \tag{42}$$

Considering a time interval from periods k to t, the left hand side of constraints (42) calculates the amount of space occupied by items sent strictly before period k to satisfy demand in periods k to t. This amount must be at least as big as the additional transportation capacity needed in excess of the available transportation capacity in periods k to t, i.e., the total volume occupied by d_{kt}^w minus the available transportation volume available between periods k and t. If we do not allow demand splitting, constraints (42) are still valid but do not specifically take into account that there must be no demand splitting. We devise stronger lower bounds for this case. Example 4.1 illustrates this on a small instance.

Example 4.1. *We use a small example in order to illustrate how to devise stronger lower bounds for the case where we do not allow demand splitting. This lower bound imposes a minimum quantity that must be ordered by a warehouse if its cumulative demand exceeds the transportation capacity available to deliver the goods to its retailers. Table 3 is used to illustrate the problem that arises when we do not allow demand splitting. In Table 3, we display the demands for a four-period example with one item and one warehouse being in charge of 5 retailers. The volume of each item is equal to 1. In Table 3 each line represents one period and each column one retailer.*

In this example, we set the transportation capacity C^{WR} equal to 100 and we consider that we have one truck available. In period 2, we do not have enough transportation capacity. One could argue that we must deliver in

period 1 a volume corresponding to the missing capacity, being $110 - 100 = 10$ *units. However, this would imply demand splitting since the minimum demand of a retailer is 15 units for retailer 3 in period* 2*. Therefore, before period 2, we must at least deliver 15 units. This minimum value will be calculated as* $min_{k \leq u \leq t, r \in R(w,u)}\{V_u^r\}$ *in the right-hand side of our new constraint, with* V_t^r *the demand of retailer* r *in period* t*, converted into volume, i.e.,* $V_t^r = \sum_{i \in I} v_i d_{it}^r$.

Let us consider now the case of period 4. In period 4, we have 125 units to deliver but 100 units of transportation capacity. As in the case of period 2, one could argue that we need to deliver in the previous periods the minimum demand of the retailers, being 15 units for retailer 3. This is, however, not enough since we would still have $125 - 15 = 110$ *units to deliver. Actually we would need to deliver at least 30 units in the previous periods. This represents the demand of retailer 2 in period* 4 *and will allow us to satisfy both the non demand splitting and the transportation capacity requirements. To find this minimum quantity, i.e.,* $C_{kt}^{min,w}$*, we actually solve a knapsack problem. Let* k *and* $t, k \leq t$*, be two periods such that* $V_{kt}^w > C_{kt}^{WR,w}$*. We define* y_{ru} *as a binary variable that takes the value 1 iff the total demand for retailer* r *in period* $u \in [k;t]$ *will be delivered strictly before period* k*. We solve the following knapsack problem:*

$$Min \sum_{u=k}^{t} \sum_{r \in R(w,u)} \left(\sum_{i \in I} v_i d_{iu}^r \right) y_{ru} \tag{43}$$

$$s.\ t. \sum_{u=k}^{t} \sum_{r \in R(w,u)} \left(\sum_{i \in I} v_i d_{iu}^r \right) y_{ru} \geq V_{kt}^w - C_{kt}^{WR,w} \tag{44}$$

$$y_{ru} \in \{0;1\} \qquad \forall\ k \leq u \leq t \leq |T|, r \in R(w,u). \tag{45}$$

The objective (43) *minimizes the total volume that must be delivered strictly before period* k *and constraint* (44) *defines a lower bound on this minimum volume, which is the missing capacity. In our new constraint, the parameter* $C_{kt}^{min,w}$ *is equal to the optimal value of this objective function.*

When can now use the following constraints for the case when we do not allow demand splitting:

$$\sum_{i \in I} \sum_{l<k} \sum_{u=k}^{t} v_i X_{ilu}^w \geq \max\{\min_{k \leq u \leq t, r \in R(w,u)}\{V_u^r\}; C_{kt}^{min,w}\} \quad \forall\ 1 \leq k \leq t \leq |T|, w \in W, |V_{kt}^w > C_{kt}^{WR,w}. \tag{46}$$

Constraints (46) indicate that, if the cumulative demand of warehouse w between periods k and t, converted into volume, i.e., V_{kt}^w, exceeds the cumulative transportation capacity available at this warehouse between the same periods, i.e., $C_{kt}^{WR,w}$, then the quantities ordered strictly before period k to satisfy demand between periods k and t, i.e., the left hand side of (46), must be greater than the lowest demand of retailers linked to warehouse w, converted into volume, and $C_{kt}^{min,w}$, where $C_{kt}^{min,w}$ represents the minimum quantity that must be delivered to satisfy the non splitting constraint. The addition of constraints (46) makes a link between the second and third steps of the heuristic, and helps the heuristic find good integrated plans.

Between two iterations of Step 2, we add diversification constraints. The idea of the diversification constraint comes from the local branching strategy of Fischetti & Lodi (2003) and has already been proposed by Fischetti et al. (2004). Let $\overline{y}^p_{it}$ and $\overline{y}^w_t$ be the optimal values of the plant and warehouse setup variables after a specific iteration of Step 2, respectively. We add the following constraints:

$$\sum_{i\in I}\sum_{t|\overline{y}^p_{it}=1}(1-y^p_{it})+\sum_{i\in I}\sum_{t|\overline{y}^p_{it}=0}y^p_{it}\geq 1 \tag{47}$$

$$\sum_{w\in W}\sum_{t|\overline{y}^w_{t}>0}(1-y^w_{t})+\sum_{w\in W}\sum_{t|\overline{y}^w_{t}=0}y^w_{t}\geq 1. \tag{48}$$

In our heuristic, we start by adding diversification constraints linked to the warehouses, i.e., constraints (48). Once we have performed a certain number of iterations of Step 2, we add the diversification constraints (47) linked to the plant, and continue with new iterations of Step 2. If we add a new diversification constraint (47) related to the plant, we remove all the current diversification constraints (48) present in the model. If we add a new diversification constraint (48), we do not remove any other diversification constraints present in the model. Note that we remove all diversification constraints after Step 4.

4.3. Step 3: ordering quantities at the retailers

In the third step of our heuristic, we solve a second OWMR. This OWMR does not integrate the routing decisions. The routes will be constructed once this second OWMR is solved, based on the ordering decisions at the retailer level. Recall that when we execute Step 3, we already have an assignment of the retailers to the different warehouses. We denote by $W(r,t)$ the warehouse assigned to retailer r for its demand in period t. Let X^w_{rikt} be the amount ordered by warehouse w to the plant in period k to satisfy d^r_{it}, and let X^r_{iktv} be the amount delivered to retailer r by vehicle v belonging to $W(r,t)$ in period k to satisfy d^r_{it}. Let s^w_{rikt} and s^r_{ikt} be the amount stocked at warehouse w and at retailer r at the end of period k to satisfy d^w_{it}, respectively. Let $tc^r_{tW(r,t)}$ be a temporary cost linked to the retailer visit variable $z^r_{tW(r,t)}$, which approximates the routing costs. Let $\overline{X}^w_{itk}$ be the optimal value of the X^w_{itk} variables obtained after an execution of Step 2. Let O^w_{it} be the quantity of item i available at warehouse w in period t, computed as $O^w_{it}=\sum_{k\geq t}\overline{X}^w_{itk}$. Let finally $V_{rt}=V(W(r,t))$. The second OWMR we solve, which we call $OWMR_2$, is as follows:

$$\text{Min}\sum_{t\in T}\left(\sum_{r\in R}sc^r_t y^r_t+\sum_{r\in R}tc^r_{tW(r,t)}z^r_{tW(r,t)}+\sum_{r\in R}\sum_{k\leq t}\sum_{i\in I}\left(hc^{W(r,t)}_{ik}s^{W(r,t)}_{rikt}+hc^r_{ik}s^r_{ikt}\right)\right) \tag{49}$$

$$\text{s. t. } s^{W(r,t)}_{r,i,k-1,t} + X^{W(r,t)}_{rikt} = \sum_{v \in V_{rt}} X^{r}_{iktv} + s^{W(r,t)}_{rikt} \quad \forall\, r \in R, i \in I, 1 \leq k \leq t \leq |T| \tag{50}$$

$$s^{r}_{i,k-1,t} + \sum_{v \in V_{rt}} X^{r}_{iktv} = \delta_{kt} d^{r}_{it} + (1 - \delta_{kt}) s^{r}_{ikt} \quad \forall\, r \in R, i \in I, t \in T, k \leq t \in T \tag{51}$$

$$\sum_{r \in R(w,t)} \sum_{t \geq k} X^{w}_{rikt} \leq O^{w}_{ik} \quad \forall\, w \in W, i \in I, k \in T \tag{52}$$

$$\sum_{r \in R(w,t)} \sum_{t \geq k} X^{r}_{iktv} \leq C^{WR} \quad \forall\, w \in W, k \in T, v \in V(w) \tag{53}$$

$$X^{r}_{iktv} \leq d^{r}_{it} y^{r}_{k} \quad \forall\, r \in R, i \in I, 1 \leq k \leq t \leq |T|, v \in V_{rt} \tag{54}$$

$$X^{r}_{iktv} \leq d^{r}_{it} z^{r}_{kw} \quad \forall\, w \in W, k \leq t \in T, r \in R(w,t), i \in I \tag{55}$$

$$\sum_{w \in W} z^{r}_{tw} \leq 1 \quad \forall\, r \in R, t \in T \tag{56}$$

$$X^{w}_{rikt}, s^{w}_{rikt} \geq 0 \quad \forall\, r \in R, w \in W, 1 \leq k \leq t \leq |T| \tag{57}$$

$$s^{r}_{ikt} \geq 0 \quad \forall\, r \in R, 1 \leq k \leq t \leq |T| \tag{58}$$

$$X^{r}_{iktv} \geq 0 \quad \forall\, r \in R, 1 \leq k \leq t \leq |T|, v \in V_{rt} \tag{59}$$

$$y^{r}_{t} \in \{0;1\} \quad \forall\, r \in R, t \in T \tag{60}$$

$$z^{r}_{tw} \in \{0;1\} \quad \forall\, r \in R, w \in W, t \in T. \tag{61}$$

Constraints (50)-(51) are the inventory balance constraints at the warehouse and retailer level, respectively. Constraints (52) make the link between the second and third step of the heuristic. Constraints (53) are the transportation capacity constraints. Constraints (54) are the setup constraints at the retailer level. Constraints (55) link the ordering and visit variables. Constraints (56) state that each retailer can be visited only by one warehouse in each time period. Finally, constraints (57)-(61) define the bounds and domains of the decision variables.

If we prevent demand splitting only, we add a new set of binary variables z'^{r}_{kt} taking the value 1 iff retailer r is visited in period k for its demand in period t. We add the following constraints:

$$X^{r}_{iktv} \leq d^{r}_{it} z'^{r}_{kt} \quad \forall\, r \in R, i \in I, 1 \leq k \leq t \leq |T|, v \in V_{rt} \tag{62}$$

$$\sum_{k \leq t} z'^{r}_{kt} \leq 1 \quad \forall\, r \in R, t \in T \tag{63}$$

$$z'^{r}_{kt} \in \{0;1\} \quad \forall\, r \in R, 1 \leq k \leq t \leq |T|. \tag{64}$$

Constraints (62) link the order variables to the new binary variables. Constraints (63) state

that the demand of a specific retailer for a specific time period cannot be split over several periods. If we prevent delivery splitting only, we define a new set of binary variables z''^r_{kv} taking the value 1 iff vehicle v visits retailer r in period k. We add the following constraints:

$$X^r_{iktv} \leq d^r_{it} z''^r_{kv} \qquad \forall\, r \in R, i \in I, 1 \leq k \leq t \leq |T|, v \in V_{rt} \tag{65}$$

$$\sum_{v \in V_{rt}} z''^r_{tv} \leq 1 \qquad \forall\, r \in R, t \in T \tag{66}$$

$$z''^r_{tv} \in \{0;1\} \qquad \forall\, r \in R, t \in T, v \in V_{rt}. \tag{67}$$

Constraints (65) link the order variables to the new binary variables. Constraints (66) state that deliveries cannot be split per truck. Finally, if we prevent demand and delivery splitting, we define a new set of binary variables Y^r_{ktv} taking the value 1 iff vehicle v visits retailer r in period k for its demand of period t. We add the following constraints:

$$X^r_{iktv} \leq d^r_{it} Y^r_{ktv} \qquad \forall\, r \in R, i \in I, k \leq t \in T, v \in V_{rt} \tag{68}$$

$$\sum_{k \leq t} \sum_{v \in V_{rt}} Y^r_{ktv} \leq 1 \qquad \forall\, r \in R, t \in T \tag{69}$$

$$Y^r_{ktv} \in \{0;1\} \qquad \forall\, r \in R, k \leq t \in T, v \in V_{rt}. \tag{70}$$

Constraints (68) link the order variables to the new binary variables. Constraints (69) state that the deliveries per period and vehicle cannot be split.

The solution of the OWMR problem in Step 3 does not provide a feasible solution for the original problem since we need to construct routes. To construct the routes, we initially solve TSPs with the current assignment of retailers, i.e., the assignment in Step 1 or the new assignment in Step 4. We call these tours a priori tours, in the same vein as the idea used by Solyalı & Süral (2011) for the PRP. This makes the cost update mechanism between two iterations of Step 3 go faster since we already have a tour to follow. If we identify a retailer which is visited but not present in the a priori tour, we actually solve a TSP. In our experiments, this rarely happens. Based on these a priori tours and on the solution obtained in Step 3, we construct the routes. These routes follow the sequence of retailers in the a priori tours, but we skip the retailers that are not visited by the given vehicle, i.e., the ones for which all variables X^r_{iktv} are equal to zero for all items i and all periods $k \leq t$. This allows to construct a feasible solution for the whole problem and to compute the exact routing costs. These exact routing costs are used between two iterations of Step 3 to update the temporary costs $tc^r_{tW(r,t)}$.

We tested several cost update mechanisms but only present the one that gave good results in initial experiments. For any retailer r visited in a route, we set the cost $tc^r_{tW(r,t)}$ to $c_{r_p r} + c_{r r_s} - c_{r_p r_s}$, where r_p and r_s are, in the a priori tour, the facilities that are visited in the current solution just before and after retailer r, respectively. Note that if a retailer is visited in several routes, we sum all the costs related to the different visits. For any retailer r not visited in period t, we identify the best insertion in the current routes. We then set the cost tc^r_{tw} to $c_{r_p r} + c_{r r_s} - c_{r_p r_s}$, where r_p and r_s are the facilities visited in the best insertion identified, just before and after retailer r, respectively. A similar update mechanism was used by Absi et al. (2015) and by Chitsaz et al. (2019).

4.4. Step 4: diversification of the search

Once we have performed Steps 2 and 3, we diversify the search. Indeed, the solution obtained after executing Steps 2 and 3 is closely related to the assignment of the retailers to the different warehouses, done in Step 1. For each time period t, we construct a set $W^+(t)$ containing warehouses that can accept more retailers, and a set $W^-(t)$ containing warehouses that should have fewer retailers. For each warehouse $w \in W^-(t)$, we take, in the set $R(w,t)$, the furthest retailer r from warehouse w. We then find the warehouse w' that will lead to the smallest increase of cost in the a priori tour when we add r to $R(w',t)$. If $w' \in W^+(t)$, we actually move retailer r from $R(w,t)$ to $R(w',t)$. Otherwise, we do not reassign retailer r and move to the next warehouse in $W^-(t)$. We compute the total cost C^w_t for each warehouse, in each time period. This total cost comprises the inventory holding costs and the setup costs at the warehouse and its retailers, and the routing costs between the warehouse and its retailers. We then divide C^w_t by $\sum_{i \in I} d^w_{it}$. We finally compute the average cost per period $\overline{C}_t$, over all warehouses. For each period t, the set $W^-(t)$ contains the warehouses that have a cost C^w_t greater than $\overline{C}_t$. In the sequel we refer to this strategy as the CD strategy. Note that we have tested other diversification strategies that did not prove successful.

4.5. Step 5: improving the solution

Once we have obtained a solution to our problem, we perform an improvement step. We take the routes of each vehicle and each time period and solve a TSP problem on each of these routes. Indeed, the best tours are not necessarily the subtours we can extract from the a priori tours. This last step is performed using the Concorde solver (Applegate et al., 2011).

4.6. Pseudo-code for the top-down heuristic

A sketch of the full top-down heuristic is presented hereafter. In Algorithm 1, the parameters it_P and it_W, it_R and it_D represent the maximum number of iterations allowed for Steps 2, 3 and 4, respectively.

Algorithm 1 Sketch of the top-down heuristic

$it = 0$
Step 1: initial assignment of the retailers and solving of initial TSPs
while $it < it_D$ **do**
  $it_1 = 0$
  while $it_1 < it_P$ **do**
    $it_2 = 0$
    while $it_2 < it_W$ **do**
      Step 2: solve $OWMR_1$
      Update the available quantities for each warehouse and set $it_3 = 0$
      while $it_3 < it_R$ **do**
        Step 3: solve $OWMR_2$
        Build routes and apply the cost update mechanism and set $it_3 = it_3 + 1$
      end while
      Add diversification constraint (48) and set $it_2 = it_2 + 1$
    end while
    Add diversification constraint (47), remove all diversification constraints (48) and set $it_1 = it_1 + 1$
  end while
  Step 4: reassign the retailers to warehouses, build new priori tours and remove all diversification constraints (47) and (48)
end while
Step 5: improve the best routes by solving TSPs

5. A bottom-up heuristic

We present a bottom-up heuristic which represents a situation where the distribution decisions at the retailer level lead the operational decisions of the company. In this heuristic we start by solving a series of single-item uncapacitated lot sizing problems (SI-ULSP), one for each retailer. This gives the best replenishment plan for each retailer. We then solve a facility location problem that will decide which warehouse will be responsible to replenish which retailers in each time period. This second step also allows us to build delivery routes. In this second step, we approximate the routing

costs in a similar way as in the top-down heuristic. We perform some iterations in this second step to improve the routes obtained. At the end of this second step, we have a replenishment plan for the warehouses. In a third step, we solve a specific OWMR for the plant and the warehouses. In this OWMR, the demand is computed based on the replenishment plan for each warehouse. Finally, we diversify the search by changing the initial replenishment plans of the retailers and repeat the whole procedure. We give details on each step in the following sections.

5.1. Step 1: replenishment plans for the retailers

We start the bottom-up heuristic by solving a SI-ULSP for each retailer. We set the demand in each time period as a total demand over all items. The setup costs considered are the sc_t^r values. These SI-ULSPs are solved by a backward dynamic programming algorithm as described in Pochet & Wolsey (2006). Solving these SI-ULSPs gives us the best replenishment plan for each retailer individually. We then make some adjustments to these plans, to take into account both transportation and production capacity requirements. The adjustment of the replenishment plans prevents the bottom-up heuristic from being myopic about the rest of the decisions to be made. The adjustments related to the transportation and production requirements are presented in Algorithm 2. Let $I(t)$ be the set of items that are ordered in period t, and denote by $\overline{o}_{it}^r$ the quantity of item i ordered by retailer r in period t in the solution obtained from solving the SI-ULSPs. In Algorithm 2, the parameter 'capacity' is replaced by $\sum_{w \in W} |V(w)| \times C^{WR}$ or by $C^P - \sum_{i \in I(t)} st_i$ if we are adjusting the replenishment plans to meet the transportation or production requirements, respectively. Besides, the parameter α_i is replaced by v_i or by vt_i if we are adjusting the replenishment plans to meet the transportation or production requirements, respectively. The function 'findBestRetailer(r_1, t)' returns the retailer r_1 whose cost of moving part of the orders from period t to $t+1$ is minimized. The shift includes a new setup in period $t+1$ and the quantity ordered in period $t+1$ for item i is $d_{i,t+1,t'-1}^{r_1}$, where t' represents, after period t, the next period with an order placed in the initial replenishment plan of retailer r_1. Note that there is still a setup in period t and that the quantity ordered in period t is d_t^r. In the sequel, we denote by $Y(r)$ the set of periods with an order placed by retailer r. We first adjust the replenishment plans to satisfy the transportation capacity requirements, and then to satisfy the production capacity requirements, if needed.

Algorithm 2 Adjustment of the replenishment plans to satisfy the capacity requirements

for $t \in T$ **do**
  capacityUsed = $\sum_{r\in R}\sum_{i\in I}\alpha_i \overline{o}^r_{it}$
  while capacityUsed > capacity **do**
    findBestRetailer(r_1, t)
    capacityGained = $\sum_{i\in I}\alpha_i d^{r_1}_{i,t+1,t'-1}$
    $\overline{o}^{r_1}_{it} = \overline{o}^{r_1}_{it} - d^{r_1}_{i,t+1,t'-1}$, $\overline{o}^{r_1}_{i,t+1} = \overline{o}^{r_1}_{i,t+1} + d^{r_1}_{i,t+1,t'-1}$
    capacityUsed = capacityUsed - capacityGained
  end while
end for

5.2. Step 2: assignment of retailers to warehouses

Once we have a replenishment plan for each retailer, we turn to the assignment of retailers to warehouses. Let X^r_{tw} be a binary variable equal to 1 iff warehouse w delivers to retailer r in period t and let a^r_{iktwv} be the proportion of $\overline{o}^r_{it}$ delivered by vehicle v of warehouse w in period k. Let vc^r_{wt} be a temporary cost linked to variable X^r_{wt} which approximates the routing costs. Finally, let χ^r_{it} be equal to 1 iff $\overline{o}^r_{it} > 0$ and 0 otherwise. The model we use is as follows:

$$\text{Min} \sum_{r\in R}\sum_{w\in W}\sum_{t\in T}\left(vc^r_{tw}X^r_{tw} + \sum_{i\in I}\sum_{v\in V(w)}\sum_{k\le t}\sum_{u=k}^{t-1} h^r_u \overline{o}^r_{it} a^r_{iktwv}\right) \tag{71}$$

$$\text{s. t.} \sum_{k\le t|k\in Y(r)}\sum_{w\in W}\sum_{v\in V(w)} a^r_{iktwv} = \chi^r_{it} \qquad \forall\, r\in R, i\in I, t\in T \tag{72}$$

$$a^r_{iktwv} \le X^r_{kw} \qquad \forall\, r\in R, i\in I, w\in W, 1\le k\le t\le |T|, v\in V(w) \tag{73}$$

$$\sum_{r\in R}\sum_{t\ge k}\sum_{i\in I} v_i \overline{o}^r_{it} a^r_{iktwv} \le C^{WR} \qquad \forall\, w\in W, k\in T, v\in V(w) \tag{74}$$

$$\sum_{w\in W} X^r_{tw} \le 1 \qquad \forall\, r\in R, t\in T \tag{75}$$

$$a^r_{iktwv} \ge 0 \qquad \forall\, r\in R, w\in W, i\in I, 1\le k\le t\le |T|, v\in V(w) \tag{76}$$

$$X^r_{tw} \in \{0;1\} \qquad \forall\, r\in R, w\in W, t\in T. \tag{77}$$

The objective function (71) minimizes the sum of the visiting costs and of the inventory holding costs at the retailer level. Constraints (72) are the demand satisfaction constraints. Constraints (73) link the assignment variables and the delivery variables. Constraints (74) are the transportation capacity constraints. Constraints (75) state that a retailer can be visited by one warehouse only in

each period. Finally constraints (76)-(77) define the bounds and domains of the decision variables. Formulation (71)-(77) allows demand and delivery splitting. If we prevent demand and delivery splitting, we impose that the a^r_{iktwv} variables must be binary.

If we prevent delivery splitting only, we define binary variables X'^r_{wkv} equal to 1 iff vehicle v belonging to warehouse w visits retailer r in period k. We then add the following constraints:

$$a^r_{iktwv} \leq X'^r_{kwv} \qquad \forall\, r \in R, w \in W, i \in I, 1 \leq k \leq t \leq |T|, v \in V(w) \tag{78}$$

$$X'^r_{kwv} \leq X^r_{kw} \qquad \forall\, r \in R, w \in W, k \in T, v \in V(w) \tag{79}$$

$$\sum_{v \in V(w)} X'^r_{kwv} \leq 1 \qquad \forall\, r \in R, w \in W, k \in T \tag{80}$$

$$X'^r_{twv} \in \{0;1\} \qquad \forall\, r \in R, w \in W, t \in T, v \in V(w). \tag{81}$$

Constraints (78) link the delivery variables and the new binary variables. Constraints (79) link the new binary variables and the assignment variables. Finally constraints (80) state that at most one vehicle can visit each retailer in each time period.

If we prevent demand splitting only, we define binary variables X''^r_{kt} equal to 1 iff the demand of retailer r in period t will be delivered in period k. We then add the following constraints:

$$a^r_{iktwv} \leq X''^r_{kt} \qquad \forall\, r \in R, w \in W, i \in I, 1 \leq k \leq t \leq |T|, v \in V(w) \tag{82}$$

$$\sum_{k \leq t} X''^r_{kt} \leq 1 \qquad \forall\, r \in R, t \in T \tag{83}$$

$$X''^r_{kt} \in \{0;1\} \qquad \forall\, r \in R, k \leq t \in T. \tag{84}$$

Constraints (82) link the delivery variables to the new binary variables. Constraints (83) impose that the demand of a specific time period for a specific retailer is served in at most one period.

At each iteration of Step 2, we construct routes by solving a TSP for each vehicle of each warehouse, based on the obtained solution of the assignment problem. This route construction allows us to update the costs vc^r_{tw}. This is done in order to intensify the search and obtain better routes, in terms of costs. The cost update of the parameter vc^r_{tw} is the same as for the parameter tc^r_{tw} in Section 4.3.

5.3. Step 3: production quantities and ordering quantities at the warehouses

Once we have performed several iterations of Step 2, we proceed to Step 3. In this step, the objective is to find an optimal production plan, given the orders of the warehouses obtained in Step

2. We solve an OWMR problem similar to the one solved in Step 2 of the top-down heuristic. Let $\overline{X}^r_{tw}$ and $\overline{a}^r_{iktwv}$ be, in the solution obtained after Step 2, the values of variables X^r_{tw} and a^r_{iktwv}, respectively. The demand d^w_{it} at the warehouses is computed as $\sum_{r\in R|\overline{X}^r_{tw}=1}\sum_{k\leq t}\sum_{v\in V(w)}\overline{o}^r_{ik}\overline{a}^r_{itkwv}$. The decision variables used in this third step are the same as in the OWMR solved in Step 2 of the top-down heuristic. The problem to be solved is given by (32)-(41).

5.4. Step 4: diversification of the search

Once we have performed Step 3, we have a feasible solution for our problem. However, this solution is highly dependent on the initial replenishment plans of the retailers. We diversify the search by changing the initial replenishment plans at the retailers. We change the setup costs of the retailers based on the adjustment of the plans done in Step 1. For each retailer whose replenishment plan has been changed because of capacity requirements in period t, we set its setup cost sc^r_t to a large value. If we do not have to adjust the replenishment plans, we compute the total production quantities at the plant for each time period. We denote by t_{min} the period whose production quantity at the plant is the lowest, but strictly positive. We finally set the setup cost of the retailers to a large value in period t_{min}.

5.5. Pseudo-code for the bottom-up heuristic

A sketch of the full top-down heuristic is presented hereafter. In Algorithm 3, it_R and it_W are used to represent the maximum number of iterations done to diversify and intensify the search, respectively.

6. Computational experiments

In order to assess the performance of our heuristics, we conducted numerical experiments on instances adapted from Gruson et al. (2019a). We generated smaller instances than in Gruson et al. (2019a) since we tackle a more difficult problem. We set the production capacity as a given factor C of the average total demand. The production capacity imposed is thus $C^P = C\sum_{r\in R}\sum_{t\in T}\sum_{i\in I} vt_i d^r_{it} \ / \ |T|$. We set the capacity factor C equal to 2 or 1.75. We set the capacity of the trucks available at the plant as $C^{PW} = \sum_{r\in R}\sum_{i\in I}\sum_{t\in T} v_i d^r_{it} \ / \ |T|$. We set the capacity of the trucks available at the warehouses as a given factor C'^{WR} of the capacity of the trucks used to make the deliveries between the plant and the warehouses. This transportation

Algorithm 3 Sketch of the bottom-up heuristic

it = 0
while $it < it_R$ **do**
  Step 1: solve a SI-ULSP for each retailer
  Adjust the replenishment plans to satisfy the transportation then production capacity requirements using Algorithm 2 and set $it_2 = 0$
  while $it_2 < it_W$ **do**
    Step 2: solve the assignment problem (71)-(77)
    Build the routes from the solution to the assignment problem
    Update the costs vc^r_{tw} and set $it_2 = it_2 + 1$
  end while
  Compute the demand at each warehouse
  Step 3: solve the OWMR (32)-(41)
  Change the setup costs of the retailers and set $it = it + 1$
end while

capacity is therefore $C^{WR} = C'^{WR} \times C^{PW}$. We set the factor C'^{WR} equal to 0.5 or 0.4. The number of trucks $|V(w)|$ available at each warehouse w is set equal to 2 or 5. The number of items $|I|$ is set equal to 1, 3 or 5. The volume v_i of each item is set equal to 1. Finally, the production time vt_i to produce one unit of item i is set equal to 1 for all items.

To put the facilities on a map, we consider a square whose side length is 100 units. Both the retailers and warehouses are randomly placed on each square using a uniform distribution. The distances between the warehouses and their retailers are computed as the Euclidean distance. The number of retailers is set equal to 10 or 20, and the number of warehouses is set equal to 2 or 4. The number of time periods is set equal to 6.

As in Gruson et al. (2019a), the demand at the retailers is generated both in a static and in a dynamic way from U[5, 100]. In the case of a static demand, we have $d^r_{it} = d^r_i \ \forall \ t \in T, r \in R, i \in I$. The fixed costs at all levels are also generated in a static and in a dynamic way. For the production plant, the fixed costs are generated from U[30000, 45000]. For the warehouses, the fixed costs are generated from U[1500, 4500]. For the retailers, the fixed costs are generated from U[5, 100]. All the demands and fixed costs are generated as integer values. The unit inventory holding costs are static and are set to 0.25 for the production plant and 0.5 for the warehouses. For the retailers, the unit inventory holding costs are generated from U[0.5, 1]. The holding costs take continuous values and are the same for all items.

For the top-down heuristic, the number of iterations is set equal to 2, 5, 10 and 5 for Steps 2, 3, 4 and 5, respectively. The OWMR problems of Steps 2 and 3 are solved using CPLEX, with a gap limit of 1%. However, we stop the solution process if the time spent between obtaining two consecutive integer solutions is large compared to the improvement obtained. Let z_i and z_{i+1} be the objective value of the two consecutive integer solutions i and $i+1$ obtained during the search process, respectively. Let also cpu_i and cpu_{i+1} be the CPU times to obtain solutions i and $i+1$, respectively. We define a threshold S to stop the solution process. This threshold is computed as $3600/(it_D \times it_P \times it_W \times it_R)$, where it_D, it_P, it_W and it_R are the number of iterations performed for Steps 4, 2 and 3, respectively. We stop the solution process if $0.01 z_i \frac{cpu_{i+1} - cpu_i}{z_i - z_{i+1}} > S$. Note that we made that choice after initial experiments have shown that the CPU time spent on Step 2 was long, even if we fine-tuned the CPLEX parameters or used multiple threads. For the bottom-up heuristic, the number of iterations is set equal to 10 and 5 for Steps 2 and 4, respectively. For both heuristics, the choice of values for the number of iterations done in each step is based on the results of initial experiments.

We performed the experiments on a 6.67 GHz Intel Xeon X5650 Westmere processor with one thread. We used the CPLEX 12.9.0.0 C++ library and turned off CPLEX's parallel mode. For the branch-and-cut algorithm, we set the CPLEX MIP emphasis parameter to 2. The emphasis is therefore on optimality over feasibility. Initial experiments have highlighted better results with this setting. All the other CPLEX parameters are set to their default value. The time limit imposed is one hour for the heuristics, and three hours for the branch-and-cut algorithm.

In the following sections, the performance of the heuristics is measured as the total CPU time taken and with the gap between the cost of the solution given by the heuristic and the cost of a solution obtained with a sequential approach. In a sequential approach, there is no diversification phase. These two measures are denoted by Total CPU and Gap seq in the tables, respectively. We also report the cost of the solution found by the heuristic, denoted as BUB. For the top-down heuristic, we report the CPU time spent on Steps 2 and 3, and the CPU time taken to improve the solution. These values are denoted by CPU_2, CPU_3 and CPU_{TSP}, respectively. For the bottom-up heuristic, we report the CPU time taken for Steps 2 and 3. These values are denoted by CPU_2, CPU_3, respectively. All the CPU times reported are expressed in seconds.

Table 4: Results obtained by CPLEX

Delivery splitting	Demand splitting	BUB	BLB	CPU time	Nodes	Optimality gap
x	x	293474	290687	4174	24626	1.5
x	✓	295410	293983	3224	15357	0.41
✓	x	296843	293034	4040	27450	1.4
✓	✓	299018	292616	3349	16179	1.17

Table 5: Results of the heuristic

Delivery splitting	Demand splitting	Top-down heuristic						Bottom-up heuristic				
		BUB	CPU_2	CPU_3	CPU_{TSP}	Total CPU	Gap (%)	BUB	CPU_2	CPU_3	Total CPU	Gap (%)
x	x	311937	5.6	211.3	0.2	217.1	5.84	315796	858	1.44	861	4.1
x	✓	309593	5.5	860.3	0.4	866.3	4.19	310389	665	1.4	667	5.3
✓	x	317429	5.8	245.3	0.2	251.3	5.37	301350	1200	1.0	1202	1.03
✓	✓	312498	5.6	235.9	0.4	241.9	5.16	299907	1409	1.0	1411	0.7

6.1. Quality of the branch-and-cut and of the heuristics

The results obtained by our branch-and-cut algorithm are reported in Table 4, where the best upper and lower bounds are given by BUB and BLB, respectively. The columns CPU time, Nodes and Optimality gap report the total CPU time taken by the branch-and-cut algorithm, the number of nodes explored in the search tree, and the optimality gap at the end of the time limit, respectively.

Table 5 gives the results obtained for the two heuristics. In Table 5, we also report the gap between the cost of the solution of a heuristic approach and the cost of the solution given by the branch-and-cut algorithm. This is denoted by Gap. One can see that the solutions obtained by the top-down heuristic are of acceptable quality, with a gap of around 5% compared to the solution found by the branch-and-cut algorithm. Note, however, that these solutions are obtained in less time compared to the branch-and-cut algorithm.

Regarding the use of the CPU time, the majority of the time is spent on solving the second OWMR. This is expected because of the numerous calls to the third step of the heuristic. The CPU time taken to solve the first OWMR is relatively small compared to the total CPU time. This is explained by our procedure that tries to identify a tailing-off effect, and also because we solve the first OWMR less often than the second one. We finally note that for the case where we allow split

demand only, the third step takes more CPU time. This may be explained by the specific structure of the problem with demand splitting only.

In Table 5, one can see that for the cases where delivery splitting is not allowed, the quality of the solutions obtained by the bottom-up heuristic is worse than the quality of the solutions obtained by the top-down heuristic. We still have a gap of around 4-5% compared to the best solution obtained by our branch-and-cut algorithm. However, if we allow delivery splitting, the bottom-up approach is able to reduce this gap to 0.7 and to 1.03 % for the case with and without demand splitting, respectively. This increase in solution quality comes, however, with a large increase in CPU time. When we allow delivery splitting, the CPU time now reaches around 1300 seconds for the bottom-up heuristic, compared to around 250 seconds with the top-down approach. This change in CPU time can be explained by the complexity of the assignment of retailers to warehouses (Step 2 of the bottom-up heuristic). Indeed, the assignment problem is bigger in size, due to a higher number of decision variables, in particular the assignment variables a^r_{iktwv}. Besides, allowing delivery splitting offers more flexibility, giving a wider search space to explore before reaching optimality. Finally, the CPU time taken to solve the OWMR is also quite low. This can be explained by the size of the problem to be solved, and the use of a strong formulation.

6.2. Influence of the splitting possibilities

We now analyze the results obtained depending on the splitting possibilities. We first note that if we do not allow any demand nor delivery splitting, the CPU time taken by our branch-and-cut algorithm is higher than in the other settings, as reported in Table 4. This is expected because of the strong restrictions.

As far as our heuristics are concerned, they have a different performance depending on the splitting possibilities. As illustrated in Table 5, the top-down heuristic finds solutions of roughly the same cost regardless of the splitting possibilities. This illustrates a drawback of this approach, which seems to handle worse the splitting possibilities. Indeed, the cost when we allow for both demand and delivery splitting is higher than when we allow for demand splitting only. As the former case is less restrictive than the later one, such a situation should not appear. It may be beneficial to have more diversification iterations in such a setting. Furthermore, the case with only delivery splitting also gives higher costs compared to the case of no splitting at all. This is also contrary to what we theoretically would expect.

Table 6: Proportion of splitting possibilities

		Top-down heuristic				Bottom-up heuristic				CPLEX	
Delivery splitting	Demand splitting	Del split (%)	Dem split (%)	Gap seq (%)	CPU_{seq}	Del split (%)	Dem split (%)	Gap seq (%)	CPU_{seq}	Del split (%)	Dem split (%)
x	x	0	0	0.2	2.6	0	0	1	92.5	0	0
x	✓	0	0.9	0.1	30.3	0	2.2	14.3	99.1	0	4.12
✓	x	31.7	0	0.2	2.5	73.5	0	5	185.1	2.45	0
✓	✓	32.9	0.8	0.4	2.1	64.6	1.9	12.7	221.7	2.43	3.89

On the contrary, the bottom-up approach seems to better handle the splitting possibilities. Indeed, one can see in Table 5 that the cost of the solutions obtained is lower when we have some flexibility compared to the case where we do not allow any demand or delivery splitting. When we have delivery splitting, the bottom-up approach obtains the best performance, in terms of quality of the solution.

We further analyzed the proportion of time with demand and delivery splitting that actually occurs in the integrated plans. Table 6 illustrates the proportion of demands where delivery and/or demand splitting occurs, on the same instances as the ones used previously. In the columns Gap seq we further report the gain, in terms of total cost, between the solution obtained by our approaches compared to the solution obtained by a sequential approach. In a sequential approach, there is no diversification phase. Let z^{int} and z^{seq} be the costs of the solution obtained for an integrated and sequential approach, respectively. The gap compared to a sequential approach is computed as $\frac{z^{seq}-z^{int}}{z^{int}}$. We finally report the CPU time taken by the sequential approach, denoted as CPU_{seq}.

In Table 6, one can see that for both heuristic approaches, when delivery splitting is allowed, it is more used compared to when demand splitting is allowed. This may be explained by the fact that with delivery splitting, the fixed setup cost at the retailer level is shared by a higher number of vehicles, thus bringing more economies. On the contrary, demand splitting must be combined with a route that already exists in order not to have an additional setup cost at the retailer level. The bottom-up heuristic uses more the splitting possibilities compared to the top-down approach. This is expected since the bottom-up approach focuses on the retailer, therefore optimizing the splitting possibility directly. On the contrary, the top-down approach first deals with the flow of goods between the plant and the warehouses, which limits the splitting possibilities at the time

of solving the second OWMR of this approach. Regarding the gains compared to a sequential approach, they are higher with the bottom-up approach, and when demand splitting is allowed. Again, the bottom-up approach first takes advantage of the splitting possibilities, leading to higher values compared to the top-down approach. The second reason is that most of the costs are spent on the setup costs at the plant and warehouse level. As such, there is little flexibility in terms of the setup plans at the plant and warehouse level. Therefore, the best setup pattern identified in the top-down approach is highly likely to be the same in the sequential approach too. Note that the sequential bottom-up approach is worse (both with respect to the quality of the solution and the CPU time) compared to the sequential top down approach. The sequential top-down heuristic also provides solution which are almost as good as the full top-down heuristic but in a much lower CPU time.

6.3. Sensitivity analysis

To better understand the settings that impact the performance of the two heuristics, we further performed experiments with inventory limits at the retailer level, with a supply network having a fixed structure or not, with initial inventory available at the retailer level, and with different values for the setup, holding and routing costs. In case we add inventory limits, let I' be a factor to limit the inventory kept on hand. We add the following constraints:

$$\sum_{i\in I}\sum_{w\in W}\sum_{u=1}^{t}\sum_{l=u}^{|T|} v_i \left(q^r_{iulw} - d^r_{i1t}\right) \leq I' \frac{\sum_{i\in I}\sum_{t\in T} v_i d^r_{it}}{|T|} \qquad \forall\, r \in R, t \in T \tag{85}$$

$$\sum_{i\in I}\sum_{k\leq t} v_i X^{W(r,t)}_{rikt} \leq I' \frac{\sum_{i\in I}\sum_{t\in T} v_i d^r_{it}}{|T|} \qquad \forall\, r \in R, t \in T \tag{86}$$

$$\sum_{i\in I}\sum_{k\leq t|k\in Y(r)}\sum_{w\in W}\sum_{v\in V(w)} v_i a^r_{iktwv} \leq I' \frac{\sum_{i\in I}\sum_{t\in T} v_i d^r_{it}}{|T|} \qquad \forall\, r \in R, t \in T. \tag{87}$$

Constraints (85), (86) and (87) are added to the full MIP model, to the second OWMR solved in the top-down approach, and to the facility location problem in the bottom-up approach, respectively. In the experiments, we set the inventory factor I' to 1 and 2.

In case we have a fixed network, the retailers are assigned to their closest warehouse. In case we have initial inventory available, this initial inventory, for each retailer, is set as the average demand of the retailer, i.e., $d^r_{1|T|}/|T|$. Finally, we define three levels for the setup, holding and routing costs: high, medium and low. The values of the costs parameters used in the experiments reported in

Table 7: Adjustment of the value of the cost parameters

Cost setting	Setup cost	Holding costs	Routing costs
Low	/100	×1	×1
Medium	/5	×100	×100
High	×1	×200	×200

Table 8: Global results of the top-down heuristic

Delivery splitting	Demand splitting	BUB	CPU_2	CPU_3	CPU_{TSP}	Total CPU	Del split (%)	Dem split (%)	Gap seq (%)
x	x	776352	0.89	31.2	1	65.1	0	0	2.9
x	✓	681878	0.88	77.8	0.9	157.8	0	0.6	3.4
✓	x	638331	0.88	29.4	1	61.1	16.9	0	4.4
✓	✓	632335	0.87	25.9	1	54.1	16.5	0.2	4.8

Sections 6.1 and 6.2 are our base values, defined as high for the setup costs, and low for the holding and routing costs. Table 7 displays how we adjust the values of the cost parameters compared to the base costs, depending on the cost level. The rest of the parameters used in the experiments run in this section are the same as previously defined. For this section, the total number of instances solved for each heuristic and each splitting possibility is 622,080. We did not do any comparison with our branch-and-cut algorithm directly as it would have taken too much time.

We first report the results aggregated per splitting possibilities. Tables 8 and 9 display the results obtained with the top-down and the bottom-up approach, respectively. In Tables 8 and 9, one can see that the top-down heuristic outperforms the bottom-up heuristic on average, in terms of cost of the solution, except when we allow demand splitting only. Detailed results indicate that the top-down approach outperforms the bottom-up one, in terms of cost, for 72% of the instances. The fact that the bottom-up heuristic performs better when we allow demand splitting only can be explained by the fact that it works on the retailer level first, so on the distribution side first, which benefits from demand splitting possibilities. We also note that demand splitting occurs in very few cases. This can be explained by the fact that demand splitting implies paying twice a fixed cost at the retailer level compared to delivery splitting. In the same spirit, we note that having demand splitting only leads to solutions that are more expensive than when we allow delivery splitting only.

Table 9: Global results of the bottom-up heuristic

Delivery splitting	Demand splitting	BUB	CPU_1	CPU_2	CPU_3	Total CPU	Del split (%)	Dem split (%)	Gap seq (%)
x	x	836209	0	33.8	0.3	34.3	0	0	3.2
x	✓	667688	0.3	26.4	0.3	26.9	0	0.1	4.5
✓	x	708396	0	42.8	0.2	43.3	56.6	0	15.9
✓	✓	735492	0	65.7	0.2	66.1	45.3	0	18.7

In Table 8, one can see that the gains compared to a sequential approach are relatively small, between 2.9 and 4.8%. The highest gains are achieved when we allow both delivery and demand splitting, which is expected since we have more freedom. The CPU time spent in Step 3 is high because of the numerous calls to solving a OWMR in Step 3. Note that the CPU time spent on Step 2 is relatively low since we stop the solution process early as explained in Section 6. Compared to the findings reported in Section 6.2, we see that the cost of the solution is less homogeneous depending on the splitting possibilities. This shows the influence of the parameters values on the performance of the top-down approach.

In Table 9, one can see that the gains compared to a sequential approach are higher than for the top-down heuristic, between 3.2 and 18.7%. Again, the high gains are achieved when we allow both delivery and demand splitting. The highest gains compared to a sequential approach indicate that the use of a bottom-up sequential approach is unlikely to lead to good solutions, especially if we allow for delivery splitting.

To better highlight the influences of the parameters, we report in Table 10 the cost and total CPU time taken by the top-down (TD) and bottom-up (BU) heuristics, depending on the splitting possibilities. In Table 10, one can see which heuristic obtains the best results depending on the splitting possibilities and values of the different parameters. In case we have both demand and delivery splitting, the top-down heuristic obtains solutions of a better cost and in a lower CPU time compared to the bottom-up heuristic, regardless of the parameter setting. In case we have demand splitting only, there is no clear impact of the heuristic in terms of costs, but the bottom-up heuristic obtains its solutions in a shorter CPU time, making it a better choice in this setting. In case we have delivery splitting only, the top-down heuristic is also better, both in terms of CPU time and cost of the solution. Finally, in case there is no splitting allowed, the parameter values have more impact on the performance of each heuristic. In such a case, the choice of the heuristic

Table 10: Performance of the top-down and bottom-up heuristics

Parameter	Value	Cost								CPU time (s)							
		Del split ✓	Dem split ✓	Del split x	Dem split ✓	Del split ✓	Dem split x	Del split x	Dem split x	Del split ✓	Dem split ✓	Del split x	Dem split ✓	Del split ✓	Dem split x	Del split x	Dem split x
		TD	BU	TD	BU	TD	BU	TD	BU	TD	BU	TD	BU	TD	BU	TD	BU
$\|V\|$	2	596819	656125	794388	640509	599516	689715	863941	718206	19	37	30	25	23	27	25	28
	5	667852	816547	569368	695228	677145	727080	688763	954232	35	95	128	29	39	60	41	41
$\|I\|$	1	473445	511401	802442	537204	476442	508319	814240	508283	15	55	87	14	19	30	21	19
	3	645754	756708	562510	674518	654614	719045	689006	888313	26	67	61	27	30	44	33	39
	5	777807	938365	680681	791482	783936	897850	825810	1111946	40	76	90	40	44	56	45	45
$\|R\|$	10	485074	628806	684201	564773	489883	616358	751204	703637	16	51	36	15	19	32	19	24
	20	779597	842177	679554	770655	786778	800427	801500	968758	38	81	122	39	43	55	47	45
$\|W\|$	2	721610	737251	855973	651756	725861	705394	1008200	766349	22	64	100	24	26	39	29	35
	4	543061	733732	507783	683481	550800	711399	544504	906057	32	68	58	30	35	48	37	33
C^{WR}	0,4	649334	781668	810716	686838	654488	754671	928221	865866	27	66	85	26	31	42	34	35
	0,5	615337	688333	553039	648916	622173	662125	624483	806557	27	66	74	28	31	45	32	34
C^{P}	1,75	633878	743224	684314	672679	639967	709581	778428	841765	27	65	75	24	31	42	33	34
	2	630793	727595	679442	662545	636694	707212	774276	830653	27	67	83	30	31	44	33	35
Setup costs	DF	632449	732712	556737	666921	638213	707302	651060	839078	27	66	76	26	31	43	33	34
	SF	632222	738271	807019	668426	638448	709491	901644	833339	27	66	83	28	31	43	33	34
Demand	DD	633593	779441	809317	663652	640588	749608	904942	832345	27	64	73	28	31	41	33	33
	SD	631078	691542	554439	671693	636073	667181	647762	840072	27	68	85	25	31	45	33	36
Inventory	1	632710	734043	689378	668063	640634	708379	783267	835882	25	70	62	27	31	49	33	37
	2	632213	734043	679403	668116	637340	708445	774682	835882	29	74	81	27	31	47	34	36
	inf	632083	738483	676852	666810	637017	708365	771107	836863	28	53	95	27	30	33	32	30
Network	free	598110	736971	527765	667286	602824	708423	617795	836536	47	65	132	26	52	39	57	35
	fix	666561	734043	835990	668050	673837	708370	934909	835881	8	67	26	27	9	47	9	34
Init inv	0	632344	734043	681876	668100	638342	708379	776347	835949	26	66	79	28	31	45	32	34
	1	632327	736971	681880	667234	638319	708414	776357	836469	28	66	80	26	31	42	33	35
Setup cost level	L	439717	556397	510274	487240	443150	528651	580789	663032	23	75	32	29	24	47	28	40
	M	728622	825072	767692	758102	735957	798289	874130	922708	30	70	103	27	34	45	36	35
	H	728667	825005	767667	758082	735885	798249	874137	922888	29	53	102	25	34	37	35	28
Holding cost level	L	551188	580231	532435	527987	563846	553169	681413	678002	34	67	168	28	41	42	43	35
	M	661572	789015	746177	727414	664597	761664	814429	906069	24	68	35	27	26	45	28	35
	H	684246	837229	767021	747668	686549	810356	833214	924555	24	63	34	26	25	43	27	33
Routing cost level	L	199936	194100	324282	197363	200032	194407	325179	198275	26	67	94	28	30	44	32	35
	M	613916	748570	674552	668545	621649	721891	766353	836058	28	66	72	26	31	43	34	34
	H	1083154	1263805	1046799	1137157	1093311	1208891	1237525	1474293	28	65	72	26	31	42	33	34
Average		632335	735492	681878	667688	638331	708396	776352	836209	27	66	79	27	31	43	33	34

to use should be done more carefully.

7. Conclusion

In this paper we have addressed the 3LSRP, an extension of the 3LSPD introduced in Gruson et al. (2019a), and of the PRP. We have added production and transportation capacity constraints to this prior work, along with routing decisions and flexibility in the assignment of retailers to warehouses. We have designed two heuristics to solve this problem: a top-down one and a bottom-up one. In the top-down heuristic the production decisions are the leading decisions while in the bottom-up heuristic the replenishment decisions at the retailer level are the leading decisions. In both heuristics we decompose the whole problem into several subproblems that exchange information and are solved iteratively. Both heuristics comprise intensification and diversification phases. The intensification phase works on improving the routes we construct while the diversification phase propose new setup plans or new assignment of retailers to warehouses.

To assess the performance of our heuristics we have developed an exact branch-and-cut algorithm. This algorithm allows us to obtain optimal solutions on a restricted set of instances that are used to measure the performance of the solutions given by our heuristics. On these instances the heuristics are able to find solutions that are on average 4.03 and 2.78% from the solution given by CPLEX for the top-down and bottom-up heuristics, respectively. Those solutions are, however, found in a CPU time lower than the one taken by CPLEX to obtain the optimal solution.

We also performed a deep sensitivity analysis to highlight the influence of the parameters and see the impact of the splitting possibilities. We observe that the top-down approach obtains better performance in terms of cost, except when we allow demand splitting only. The bottom-up approach is better suited for the use of delivery splitting. We also see that the gains obtained when allowing delivery or demand splitting are large. We finally see that some parameters are more critical than others in terms of obtaining more gains when there is no splitting allowed.

We have finally considered the possibility of having demand or delivery splitting possibilities. We have observed that when we give more flexibility, the CPU time taken to solve those instances tends to get larger. Interestingly, the bottom-up approach seems more suitable for the settings with demand splitting.

In future research we would like to investigate algorithms that can compute good lower bounds. In particular, we could develop mathematical programming-based heuristics that provide a lower

bound during the search process. We would also like to further explore the possibility of solving the problem exactly by the use of decomposition methods.

Acknowledgements

This research was enabled in part by support provided by Calcul Québec and Compute Canada. The first author gratefully acknowledges the support of the Government of Canada (grant CGV-151506).

References

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

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

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

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

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

Alvarez, A., Cordeau, J.-F., Jans, R., Munari, P., & Morabito, R. (2020). Formulations, branch-and-cut and a hybrid heuristic algorithm for an inventory routing problem with perishable products. *European Journal of Operational Research*, *283*.

Anderluh, A., Nolz, P. C., Hemmelmayr, V. C., & Crainic, T. G. (2019). Multi-objective optimization of a two-echelon vehicle routing problem with vehicle synchronization and 'grey zone' customers arising in urban logistics. *CIRRELT Technical Report*, .

Applegate, D., Bixby, R., Chvátal, V., & Cook, W. (2011). Concorde TSP solver. http://www.math.uwaterloo.ca/tsp/concorde.html. Accessed 2019-10-07.

Archetti, C., & Speranza, M. G. (2008). The split delivery vehicle routing problem: A survey. In B. Golden, S. Raghavan, & E. Wasil (Eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges* (pp. 103–122). Boston, MA: Springer US. URL: https://doi.org/10.1007/978-0-387-77778-8_5. doi:10.1007/978-0-387-77778-8_5.

Avci, M., & Yildiz, S. T. (2019). A matheuristic solution approach for the production routing problem with visit spacing policy. *European Journal of Operational Research*, *279*, 572 – 588.

Baldacci, R., Battarra, M., & Vigo, D. (2008). Routing a heterogeneous fleet of vehicles. In B. Golden, S. Raghavan, & E. Wasil (Eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges,* chapter 1. (pp. 3–27). Boston, MA: Springer.

Breunig, U., Baldacci, R., Hartl, R. F., & Vidal, T. (2019). The electric two-echelon vehicle routing problem. *Computers & Operations Research*, *103*, 198–210.

Brown, G., Keega, J., Vigus, B., & Wood, K. (2001). The Kellog company optimizes production, inventory, and distribution. *Interfaces*, *31*, 1–15.

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

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

Cuda, R., Guastaroba, G., & Speranza, M. G. (2015). A survey on two-echelon routing problems. *Computers & Operations Research*, *55*, 185–199.

Cunha, J. O., & Melo, R. A. (2016). On reformulations for the one-warehouse multi-retailer problem. *Annals of Operations Research*, *238*, 99–122. doi:10.1007/s10479-015-2073-4.

Darvish, M., & Coelho, L. (2018). Sequential versus integrated optimization: Production, location, inventory control, and distribution. *European Journal of Operational Research*, *268*, 203–214.

Dayarian, I., & Desaulniers, G. (2019). A branch-price-and-cut algorithm for a production-routing problem with short-life-span products. *Transportation Science*, *53*, 829–849.

Dellaert, N., Van Woensel, T., Crainic, T. G., & Saridarq, F. D. (2019). A multi-commodity two-echelon capacitated vehicle routing problem with time windows: model formulations and solution approach. *CIRRELT Technical Report*, .

Çetinkaya, S., Uster, H., Easwaran, G., & Keskin, B. B. (2009). An integrated outbound logistics model for Frito-Lay: coordinating aggregate-level production and distribution decisions. *Interfaces*, *39*, 460–475.

Fischetti, M., & Lodi, A. (2003). Local branching. *Mathematical Programming*, *98*, 23–47.

Fischetti, M., Polo, C., & Scantamburlo, M. (2004). A local branching heuris- tic for mixed-integer programs with 2-level variables, with an application to a telecommunication network design problem. *Networks*, *44*, 61–72.

Grangier, P., Gendreau, M., Lehuédé, F., & Rousseau, L.-M. (2016). An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. *European Journal of Operational Research*, *254*, 80–91.

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

Gruson, M., Cordeau, J.-F., & Jans, R. (2019b). Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure. *European Journal of Operational Research*, *291*, 206–217.

Gu, W., Archetti, C., Cattaruzza, D., Ogier, M., Semet, F., & Speranza, M. G. (2021). A sequential approach for a multi-commodity two-echelon distribution problem. *hal-03167379*, .

Jie, W., Yang, J., Zhang, M., & Huang, Y. (2019). The two-echelon capacitated electric vehicle routing problem with battery swapping stations: Formulation and efficient methodology. *European Journal of Operational Research*, *272*, 879–904.

Krarup, K., & Bilde, O. (1977). *Plant location, set covering and economic lot-sizes: An $O(mn)$ algorithm for structured problems*. Birkhauser Verlag, Bassel: L.Collatz (Editor).

Lysgaard, J., Letchford, N., & Eglese, R. W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, *100*, 423–445.

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

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

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

Pochet, Y., & Wolsey, L. A. (2006). *Production Planning by Mixed Integer Programming*. New York, NY, USA: Springer.

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

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

Schenekemberg, C., Scarpin, C., Pécora, J., Guimaraes, T., & Coelho, L. (2021). The two-echelon production-routing problem. *European Journal of Operational Research*, *288*, 436–449.

Solyalı, O., & Süral, H. (2011). A Branch-and-Cut Algorithm Using a Strong Formulation and an A Priori Tour-Based Heuristic for an Inventory Routing Problem. *Transportation Science*, *45*, 335–345.

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

Soysal, M., Bloemhof-Ruwaard, J. M., & Bektaş, T. (2015). The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations. *International Journal of Production Economics*, *164*, 366–378.

Vogel, T., Almada-Lobo, B., & Almeder, C. (2017). Integrated versus hierarchical approach to aggregate production planning and master production scheduling. *OR Spectrum*, *39*, 193–227.

Wagner, H. M., & Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management Science*, *5*, 89–96.

Wang, K., Shao, Y., & Zhou, W. (2017). Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service. *Transportation Research Part D*, *57*, 262–276.