Les Cahiers du GERAD ISSN: 0711–2440

Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure

M. Gruson,
J.-F. Cordeau, R. Jans

G–2019–51

July 2019

Citation suggérée : M. Gruson, J.-F. Cordeau, R. Jans (Juillet 2019). Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure, Rapport technique, Les Cahiers du GERAD G–2019–51, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (https://www.gerad.ca/fr/papers/G-2019-51) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

Suggested citation: M. Gruson, J.-F. Cordeau, R. Jans (July 2019). Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure, Technical report, Les Cahiers du GERAD G–2019–51, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (https://www.gerad.ca/en/papers/G-2019-51) to update your reference data, if it has been published in a scientific journal.

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

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

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2019
– Bibliothèque et Archives Canada, 2019

Legal deposit – Bibliothèque et Archives nationales du Québec, 2019
– Library and Archives Canada, 2019

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

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

Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure

Matthieu Gruson
Jean-François Cordeau
Raf Jans

Department of Logistics and Operations Management, HEC Montréal, Montréal (Québec), Canada, H3T 2A7

matthieu.gruson@gerad.ca
jean-francois.cordeau@gerad.ca
raf.jans@gerad.ca

July 2019
Les Cahiers du GERAD
G–2019–51

Abstract: We address a stochastic three-level lot sizing and replenishment problem with a distribution structure in a two-stage decision process. We consider one production plant that produces one type of item over a discrete and finite planning horizon. The items produced are transported to warehouses and then to retailers using direct shipments. Each retailer is linked to a unique warehouse and there are no transfers between warehouses nor between retailers. The stochasticity comes from the uncertainty in the demand at the retailer level and is modelled through scenarios. The setup decisions are made in the first stage and the production, transportation and inventory decisions are made in the second stage, once the demands are revealed. The objective is to minimize the sum of the fixed production and replenishment costs, and of the expected variable inventory holding costs among all scenarios. We use a Benders decomposition approach and develop a Benders-based branch-and-cut algorithm to efficiently solve the problem. We take advantage of the substructures identified in the decomposition and design efficient procedures to solve the subproblems obtained. We also propose computational enhancements to speed up the solution process. Finally, we perform extensive computational experiments to assess the performance of our decomposition approach and analyze the impact of these enhancements. The Benders-based branch-and-cut algorithm we propose clearly outperforms CPLEX.

Keywords: Production, stochastic lot sizing, supply chain management, Benders decomposition, multi-level

Résumé : Nous étudions un problème stochastique de planification de production et réapprovisionnement sur trois échelons via un processus de décision en deux étapes. Nous prenons en compte une usine de production qui produit un seul type d'article au cours d'un horizon temporel fini et discret. Les articles produits sont trasportés vers des entrepôts puis vers des détaillants, via des livraisons directes. Chaque détaillant est relié à un unique entrepôt et les transferts entre entrepôts ou entre détaillants sont prohibés. L'aspect stochastique vient de l'incertitude entourant la demande au niveau des détaillants, et est modélisée via des scénarios de demande. Les décisions de mise en route sont prises lors de la première étape et les décisions de production, transport et stockage sont prises lors de la deuxième étape, une fois que les demandes ont été révélées. L'objectif est de minimiser la somme des coûts fixes de production et réapprovisionnement, et de l'espérance des coûts variables de stockage à travers tous les scénarios. Nous utilisons une décomposition de Benders et développons un algorithme de *branch-and-cut* fondé sur la décomposition de Benders pour résoudre efficacement le problème. Nous tirons profit des sous-structures identifiées dans la décomposition et développons des procédures efficaces pour résoudre les sous-problèmes obtenus. Nous intégrons également des améliorations de calcul pour accélérer le processus de résolution. Enfin, nous réalisons de nombreuses expériences numériques pour évaluer la performance de notre approche par décomposition et pour analyser l'impact des améliorations proposées. L'algorithme de *branch-and-cut* fondé sur la décomposition de Benders que nous proposons surpasse de manière notable CPLEX.

Mots clés : Production, lotissement stochastique, gestion de la chaîne d'approvisionnement, décomposition de Benders, multi-niveaux

---

Acknowledgments: The authors gratefully acknowledge the support of Calcul Québec, of the Natural Sciences and Engineering Research Council of Canada (grants 2014–03849 and 2014–04959), and of the Fonds de Recherche du Québec –Nature et Technologies (grant 2014–PR–174190). The first author gratefully acknowledges the support of the Government of Canada (grant CGV–151506).

1 Introduction

Lot sizing problems (LSP) have numerous applications in production, distribution and inventory management, three cornerstones of supply chain planning. Usually the customers and the production plant of a given company are located in different areas. This implies that the company must decide when to deliver products to its customers so as to minimize the replenishment costs. At the production plant level, the company must also make lot sizing decisions. Solving these two operational problems sequentially, as it is often the case in practice, leads to solutions that can be much more costly compared to the solution of an integrated lot sizing and replenishment problem. The integration of these two operational problems has proven to be very effective in practice, see, e.g., Dhaenens-Flipo and Finke [14], Zhang and Song [39] and Abdullah et al. [1].

Supply chain planning tools often take as a starting point forecasts of the future demand, in the form of point estimates. As a result, most of the lot sizing literature considers deterministic demand. However, as pointed out by Adulyasak et al. [2], if these forecasts are misleading, it can result in wrong and costly decisions. Taking uncertainty into account can be very beneficial but also increases the difficulty of the operational problems to be solved.

We address here an integrated three-level lot sizing and replenishment problem with a distribution structure, in a two-stage decision process (2S-3LSPD). We consider the supply chain of a general manufacturing company. This supply chain comprises one production plant, several warehouses and multiple retailers which compose the levels zero, one and two, respectively. Each warehouse is linked to the production plant and each retailer is linked to a unique warehouse, leading to a distribution structure. There are no links between the different warehouses, nor between the different retailers. Therefore, the flow of goods ordered by the retailers is entirely fixed: the product goes from the production plant (where it is produced), to a warehouse (where it is stored) and finally to a retailer (where it is sold). Figure 1 illustrates this flow of goods in a distribution network composed by one production plant, three warehouses and three retailers linked to each warehouse.

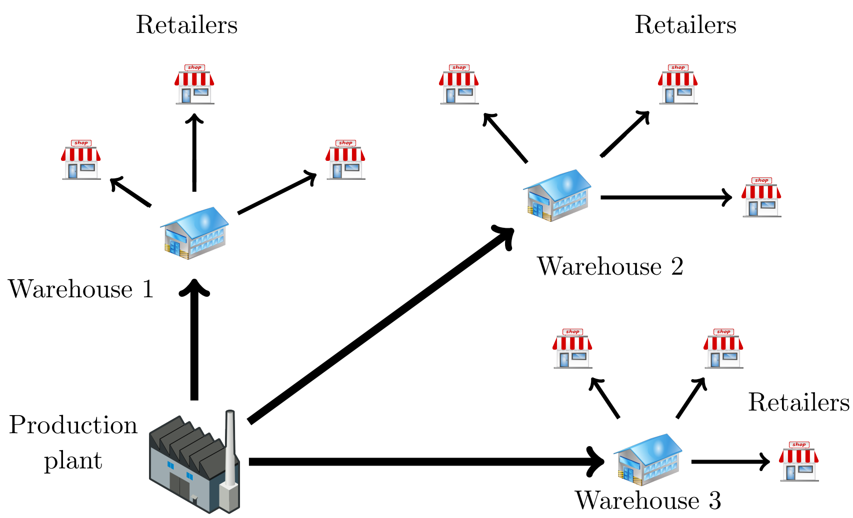

Figure 1: Graphical representation of the problem considered

The objective of the problem is to determine, for each time period, the production quantities at the plant and the flow of goods between the facilities so as to minimize the operational costs of the whole system. These costs are the sum of fixed and expected variable costs. The fixed setup costs are incurred whenever there is production at the production plant or an order is placed by a warehouse or a retailer to its predecessor. Given a finite set T of time periods, indexed by t, we denote by sc_{pt}, sc_{wt} and sc_{rt} the setup costs for production at the production plant level and for placing a replenishment order at the warehouse and retailer level, respectively. The variable costs are the inventory holding costs, incurred whenever there is some inventory on hand at the end of a time period. We denote by hc_{it} the holding cost to keep one unit of item at the end of period t at facility i. Note that we do not include any unit production cost nor any unit replenishment cost. Indeed, if we consider that these costs are constant through time they will lead to a constant term in the objective function since the complete demand of the retailers must be satisfied.

The retailers face a stochastic and dynamic demand for a unique item. The distribution of the demand for each retailer is assumed to be known, and uncertainty is taken into account through the use of demand scenarios. In our two-stage decision process, the demands of each retailer for the entire time horizon are revealed once the first stage decisions are made. These first stage decisions correspond to the production and ordering setup decisions for each facility and each time period. The second stage involves production, replenishment and inventory decisions. This separation between the first and second stage decisions corresponds exactly to the static-dynamic uncertainty strategy first proposed by Bookinder and Tan [11] for the stochastic single item LSP.

We consider that the shipments which are performed between the production plant and the warehouses, and between a warehouse and its retailers are uncapacitated. We only consider direct transportation between facilities and as such we exclude routing decisions. Finally, we do not impose any restrictions on the inventory level at any facility. In a disaggregated context, each facility faces a basic LSP. The basic LSP has been extensively studied since the seminal paper of Wagner and Whitin [37], who proposed a dynamic programming approach to solve the single item uncapacitated lot sizing problem (SI-ULSP). The reader is referred to Brahimi et al. [12] and to Pochet and Wolsey [28] for a review of the work done on the SI-ULSP and its extensions. Note that at the third level, each retailer faces a stochastic LSP. The interested reader is referred to Tempelmeier [34] and Aloulou et al. [4] for reviews on stochastic lot sizing.

Our paper makes four main contributions. First, we extend the work of Gruson et al. [17] by studying a stochastic version of the three-level lot sizing problem with a distribution structure (3LSPD). Second, we apply a Benders decomposition to the 2S-3LSPD. This decomposition exploits the substructures that appear in the MIP formulation we propose. Third, we develop two simple yet efficient procedures to solve the subproblems obtained. These procedures exploit the structure of the holding costs as well as the fact that the optimal solution to one subproblem can be easily obtained from the solution of a previous subproblem. They allow us to solve numerous minimum cost flow problems in a very short amount of CPU time. Finally, we develop a Benders-based branch-and-cut algorithm. This algorithm takes advantage of the substructures yielded by the decomposition approach. We further incorporate computational enhancements in this algorithm to speed up the solution process. Each enhancement tackles one specific issue raised in the literature when using Benders-based branch-and-cut algorithms. To the best of our knowledge, this is one of the few attempts to solve such a problem with an exact method.

The remainder of this paper is organized as follows. First, we survey the work linked to our study in Section 2. Then, we give a formal problem description in Section 3 along with one mathematical formulation for the problem. This formulation is used as a basis for the application of a Benders decomposition in Section 4. In this section, we further present a Benders-based branch-and-cut algorithm along with computational enhancements. Section 5 details the computational experiments performed to assess the performance of the algorithm we developed, and to analyze the impact of the enhancements we proposed. Note that we also analyse the deterministic version of the problem in this section. This is followed by the conclusion in Section 6.

2 Literature review

This section first briefly reviews the stochastic lot sizing literature related to our problem. It then focuses on the multi-level lot sizing literature. Finally, it details the application of Benders decomposition to lot sizing problems.

2.1 Stochastic lot sizing

The stochastic lot sizing literature usually considers the demand as a random parameter. To counter the resulting uncertainty and make both setup and lot sizing decisions, several strategies have been proposed in Bookbinder and Tan [11]: the static strategy, where both lot sizes and setup decisions are made before the uncertainty is revealed; the dynamic strategy, where production decisions are made after the uncertainty is revealed; and the static-dynamic strategy, where setup decisions are fixed before the first period and the production quantity decisions are the recourse decisions. To represent demand uncertainty while maintaining a tractable problem, it is common to use scenarios of demand instead of more general demand distributions. The challenge in such cases is to have a good balance between a large number of scenarios, which would make the problem hard to solve, and too few scenarios, which would give a bad representation of the demand distribution. The use of demand scenarios in stochastic lot sizing can be seen in Haugen et al. [21], who study the stochastic single item lot sizing problem with backlogging and embed the progressive hedging algorithm of Rockafellar and Wets [30] in a metaheuristic. Gutiérrez et al. [20] address the stochastic single item lot sizing problem with concave production costs, where the costs and demand distribution depend on the scenario considered. They solve the problem based on a multiobjective branch-and-bound approach. Taskin and Lodree [32] consider the challenges of procurement and production decisions at the time of the hurricane season. They use scenario reduction methods to be able to use a general purpose solver to solve the problem. Finally, Helber et al. [22] use demand scenarios to approximate a non linear version of the stochastic capacitated lot sizing problem (CLSP). Note that there are alternative approaches to tackle demand uncertainty in stochastic lot sizing. This is the case in Tunc et al. [36] who use static-dynamic uncertainty strategy with random demand. They consider that the production periods are fixed in advance and use the expected inventory holding costs in their non-linear objective function. Instead of using scenarios to approximate this objective function, they propose a novel MIP formulation and develop a dynamic cut generation approach.

Some work has also been done on polyhedral results by Guan et al. [19], still with demand scenarios. They adapt the well known (l, S) valid inequalities to the stochastic case and call them $(\mathcal{Q}, S_{\mathcal{Q}})$ valid inequalities. They further establish necessary and sufficient conditions for these $(\mathcal{Q}, S_{\mathcal{Q}})$ inequalities to be facet defining. These inequalities have been later simplified by Di Summa and Wolsey [15] who also propose several reformulations for the stochastic lot sizing problem with constant capacity.

Another stream of research in stochastic lot sizing incorporates different service level constraints in the models. Service level constraints include, among others, the cycle service level, which limits the stockout probability during a replenishment cycle; and the fill rate, which limits the amount of backorders. The reader is referred to Tempelmeier [33] for a comparison of various service level constraints and their implications in a stochastic setting and to Gruson et al [18] for a discussion of inventory service levels in deterministic lot sizing problems.

2.2 Multi-level lot sizing

In multi-level lot sizing problems, the production of one or more end items requires the production of one or more components, used as inputs to produce the end items. The multi-level lot sizing literature considers four different product structures [28]: assembly, where each component has a unique successor; in series, where each component has a unique successor and a unique predecessor; distribution, where each component has a unique predecessor; and general. Because of the distribution structure, the problem we address in this work is a specific case of the general multi-level lot sizing problem. In this section, we briefly review the literature related to this general problem. This problem has been tackled mainly with heuristics because of its difficulty.

Tempelmeier and Helber [35] propose a general heuristic for the multi-item multi-level capacitated lot sizing problem. Their heuristic solves a series of CLSPs using a modified Dixon-Silver heuristic (see Dixon and Silver [16]). They propose four variants of this general heuristic where the differences come from the ordering at the different levels. Maes et al. [23] address the same problem and propose a LP-based heuristic to solve it. They are able to solve small instances to optimality. More recently, Sahling et al. [31] proposed a fix-and-optimize heuristic to solve this problem while also considering setup carry-over. The idea of their heuristic is to sequentially solve a series of small mixed-integer programs and to use the solution to fix a large proportion of the integer variables to a specific value in the next iterations.

2.3 Benders decomposition in lot sizing

The use of Benders decomposition in the lot sizing literature is very scarce. Bahl and Zionts [5] apply Benders decomposition to a multi item CLSP with setup times. They take advantage of the transportation problem obtained as a subproblem to efficiently solve the problem. Bayley et al. [6] apply a combination of Benders decomposition and evolutionary algorithm to the CLSP with setup times. They consider production families for the different setup times imposed. They improve both the lower and upper bounds of the problem using their procedure. Adulyasak et al. [2] propose a Benders decomposition algorithm to tackle a combined production planning and routing problem with demand uncertainty.

3 Mathematical formulation for the 2S-3LSPD

In this section we present the mathematical formulation for the 2S-3LSPD on which we will apply Benders decomposition. Let $G = (F, A)$ be a graph with F the set of nodes (facilities in our problem) and A the set of arcs. Let $P = \{p\} \subset F$, $W \subset F$ and $R \subset F$ be the sets containing the unique production plant, the warehouses and the retailers, respectively. Given the problem description in Section 1, we have $F = P \cup W \cup R$. Let $\delta(i)$ be the set of all direct successors of facility i and $\delta_w(r)$ be the warehouse linked to the retailer $r \in R$.

To account for stochasticity, let Ω be the finite set of demand scenarios and let p_ω be the probability of scenario $\omega \in \Omega$. We denote by $d_{rt\omega}$ the demand of retailer r in period t under scenario ω. We further define a demand for any warehouse and for the production plant by extending the demand faced by any retailer, in the following fashion:

$$d_{it\omega} = \begin{cases} \sum_{r \in R} d_{rt\omega} & \text{if } i = p \\ \sum_{r \in \delta(i)} d_{rt\omega} & \text{if } i \in W. \end{cases}$$

The formulation we present is the multi-commodity (MC) formulation proposed by Melo and Wolsey [26] for a two-level lot sizing problem, and later extended by Gruson et al. [17] to the deterministic 3LSPD. The idea of this formulation is to work on distinct commodities $d_{rt\omega}$. In the following formulation, we denote by δ_{kt} the Kronecker delta that takes the value 1 if $k = t$ and 0 otherwise. We denote by $x^{lr}_{kt\omega}$ the quantities produced or ordered in level l in period k to satisfy $d_{rt\omega}$ and by $\sigma^{lr}_{kt\omega}$ the stock at level l at the end of period k to satisfy $d_{rt\omega}$. Let y_{it} be a boolean setup variable taking the value 1 if and only if there is production or an order placed by facility i in period t. The MC formulation for the 2S-3LSDP is as follows:

$$\text{Min} \sum_{t \in T} \left(\sum_{i \in F} sc_{it} y_{it} + \sum_{\omega \in \Omega} p_\omega \sum_{r \in R} \sum_{k \leq t} \left(hc_{pk} \sigma^{0r}_{kt\omega} + hc_{\delta_w(r)k} \sigma^{1r}_{kt\omega} + hc_{rk} \sigma^{2r}_{kt\omega} \right) \right) \tag{1}$$

$$\sigma^{0r}_{k-1,t,\omega} + x^{0r}_{kt\omega} = x^{1r}_{kt\omega} + \sigma^{0r}_{kt\omega} \qquad \forall\ t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{2}$$

$$\sigma^{1r}_{k-1,t,\omega} + x^{1r}_{kt\omega} = x^{2r}_{kt\omega} + \sigma^{1r}_{kt\omega} \qquad \forall\ t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{3}$$

$$\sigma^{2r}_{k-1,t,\omega} + x^{2r}_{kt\omega} = \delta_{kt} d_{rt\omega} + (1 - \delta_{kt}) \sigma^{2r}_{kt\omega} \qquad \forall\ t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{4}$$

$$x^{0r}_{kt\omega} \leq d_{rt\omega} y_{pk} \qquad \forall\ t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{5}$$

$$x^{1r}_{kt\omega} \leq d_{rt\omega} y_{\delta_w(r)k} \qquad \forall\ t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{6}$$

$$x^{2r}_{kt\omega} \leq d_{rt\omega} y_{rk} \qquad \forall\ t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{7}$$

$$x^{0r}_{kt\omega}, x^{1r}_{kt\omega}, x^{2r}_{kt\omega}, \sigma^{0r}_{kt\omega}, \sigma^{1r}_{kt\omega}, \sigma^{2r}_{kt\omega} \geq 0 \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{8}$$

$$y_{it} \in \{0,1\} \qquad \forall t \in T, i \in F. \tag{9}$$

The objective function (1) minimizes, for the first stage decisions, the sum of the setup costs and, for the second stage decisions, the expected inventory holding costs at each facility for each time period. Constraints (2)–(4) represent the inventory balance equations for each commodity $d_{rt\omega}$ at the production plant, the warehouse and the retailer level, respectively. Constraints (5)–(7) are the setup forcing constraints for the production plant, the warehouses and the retailers, respectively.

4 Benders reformulation

We apply here a Benders decomposition, starting from the MC formulation. Next, we present a Benders-based branch-and-cut (B&C) algorithm to solve the 2S-3LSPD.

4.1 The reformulation

In the MC formulation, when the binary setup decisions are fixed, we obtain a continuous linear problem which can be solved efficiently. This framework is well suited for the use of Benders decomposition. The original idea of Benders decomposition [7] is to partition the complete problem into two smaller problems, namely the master problem and the subproblem. The master problem is a simplified version of the original problem where only some variables have been kept, along with the constraints in which they are the only ones to appear. The master problem also contains an artificial variable representing a lower bound on the cost of the subproblem. The subproblem is exactly the original problem without the constraints that have been kept in the master problem. In this subproblem, the variables present in the master problem are fixed to given values. In our case, we keep the binary setup variables y_{it} in the master problem. The production and inventory variables x and σ are present in the subproblem, along with constraints (2)–(8). For a recent review on Benders decomposition, the reader is referred to Rahmaniani et al. [29].

We start by presenting the primal subproblem when applying Benders decomposition to the MC formulation. Let $\widehat{y}_{it}$ denote the values of the fixed binary setup variables. The primal subproblem PSP is defined by:

$$\text{Min} \sum_{\omega \in \Omega} p_\omega \sum_{t \in T} \sum_{r \in R} \left(\sum_{k \leq t} hc_{pk} \sigma^{0r}_{kt\omega} + \sum_{k \leq t} hc_{\delta_w(r)k} \sigma^{1r}_{kt\omega} + \sum_{k \leq t} hc_{rk} \sigma^{2r}_{kt\omega} \right) \tag{10}$$

$$\text{s. t. } (2) - (4), (8) \tag{11}$$

$$x^{0r}_{kt\omega} \leq d_{rt\omega} \widehat{y}_{pk} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{12}$$

$$x^{1r}_{kt\omega} \leq d_{rt\omega} \widehat{y}_{\delta_w(r)k} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{13}$$

$$x^{2r}_{kt\omega} \leq d_{rt\omega} \widehat{y}_{rk} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega. \tag{14}$$

Let $\psi^{0r}_{kt\omega}$, $\psi^{1r}_{kt\omega}$, $\psi^{2r}_{kt\omega}$, $\phi^{0r}_{kt\omega}$, $\phi^{1r}_{kt\omega}$ and $\phi^{2r}_{kt\omega}$ be the dual variables associated with constraints (2)–(4) and (12)–(14), respectively. The dual subproblem DSP corresponding to PSP is given as follows:

$$\text{Max} \sum_{\omega \in \Omega} \sum_{t \in T} \sum_{r \in R} \left(d_{rt\omega} \psi^{2r}_{tt\omega} - \sum_{k \leq t} d_{rt\omega} \left(\widehat{y}_{pk} \phi^{0r}_{kt\omega} + \widehat{y}_{\delta_w(r)k} \phi^{1r}_{kt\omega} + \widehat{y}_{rk} \phi^{2r}_{kt\omega} \right) \right) \tag{15}$$

$$\psi^{0r}_{k+1,t,\omega} - \psi^{0r}_{kt\omega} \leq p_\omega hc_{pk} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{16}$$

$$\psi^{1r}_{k+1,t,\omega} - \psi^{1r}_{kt\omega} \leq p_\omega hc_{\delta_w(r)k} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{17}$$

$$\psi^{2r}_{k+1,t,\omega} - (1 - \delta_{kt}) \psi^{2r}_{kt\omega} \leq p_\omega hc_{rk} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{18}$$

$$\psi^{0r}_{kt\omega} - \phi^{0r}_{kt\omega} \leq 0 \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{19}$$

$$\psi^{1r}_{kt\omega} - \psi^{0r}_{kt\omega} - \phi^{1r}_{kt\omega} \leq 0 \qquad \forall\ t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{20}$$

$$\psi^{2r}_{kt\omega} - \psi^{1r}_{kt\omega} - \phi^{2r}_{kt\omega} \leq 0 \qquad \forall\ t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{21}$$

$$\phi^{0r}_{kt\omega}, \phi^{1r}_{kt\omega}, \phi^{2r}_{kt\omega} \geq 0 \qquad \forall\ t \in T, k \leq t \in T, r \in R, \omega \in \Omega. \tag{22}$$

The DSP can be decomposed into $|R||T||\Omega|$ subproblems, denoted by $DSP_{rt\omega}$, one for each commodity $d_{rt\omega}$. Let $z_{rt\omega}$ be an additional variable representing a lower bound on the cost of the subproblem associated with commodity $d_{rt\omega}$. Let also $\Delta_{SP}(r,t,\omega)$ represent the polyhedron defined by constraints (16)–(22) for commodity $d_{rt\omega}$. The Benders reformulation, denoted as BD-MC, is defined by:

$$\text{Min} \sum_{\omega\in\Omega}\sum_{r\in R}\sum_{t\in T} z_{rt\omega} + \sum_{t\in T}\sum_{i\in F} sc_{it}y_{it} \tag{23}$$

$$\text{s.t.} \quad d_{rt\omega}\left(\psi^{2r}_{tt\omega} - \sum_{k\leq t}\left(\phi^{0r}_{kt\omega}y_{rk} + \phi^{1r}_{kt\omega}y_{\delta_w(r)k} + \phi^{2r}_{kt\omega}y_{pk}\right)\right) \leq z_{rt\omega}$$

$$\forall\ t \in T, r \in R, \omega \in \Omega,\ \forall\ \left(\phi^{0r}_{kt\omega}, \phi^{1r}_{kt\omega}, \phi^{2r}_{kt\omega}, \psi^{0r}_{kt\omega}, \psi^{1r}_{kt\omega}, \psi^{2r}_{kt\omega}\right) \in \Delta_{SP}(r,t,\omega) \tag{24}$$

$$y_{it} \in \{0,1\} \qquad \forall\ t \in T, i \in F. \tag{25}$$

The objective function (23) minimizes the sum of the setup costs and of the lower bound on the cost of the subproblems. Constraints (24) are the optimality cuts for each subproblem.

4.2 A specialized algorithm to solve the subproblem

The use of a Benders decomposition naturally leads to an iterative procedure to solve the original problem. Indeed, each polyhedron $\Delta_{SP}(r,t,\omega)$ may contain a huge number of extreme points, leading to an equally large number of cutting planes for the master problem. Each iteration of the procedure consists of the solution of the master problem with only a subset of constraints (24) and of the dual subproblem DSP. The master problem is solved to obtain values for the coupling variables. These values are passed to the dual subproblems $DSP_{rt\omega}$, which are then solved. The solution of the dual subproblem of each commodity leads to the generation of a so-called Benders cut which is either an optimality cut (in the case where the dual subproblem is feasible) or a feasibility cut (in the case where the dual subproblem is infeasible because of the values of the coupling variables). The iterative procedure can be seen as a cutting plane method where information is transferred from the subproblems to the master problem in the form of cuts. At each iteration of the procedure, both a lower and an upper bound can be derived from the solutions obtained for the master problem and the subproblems. Note that in our case, since there is no capacity requirement, the dual subproblems will always be feasible.

Each dual subproblem $DSP_{rt\omega}$ can be solved by means of a general purpose solver to generate an optimality cut. However, a closer look at the primal subproblem indicates the presence of a network substructure in constraints (16)–(18) for each commodity $d_{rt\omega}$. This substructure is illustrated in Figure 2, where we consider an example for which $t = 4$, one retailer r and one scenario ω.

In Figure 2, each node represents a pair (level, time period) and each arc represents the flow between the facilities or the stock on hand at the end of a time period for a specific facility, in order to satisfy $d_{r4\omega}$. Costs are incurred whenever there is some positive inventory on hand at a particular facility, i.e., if $\sigma^{lr}_{kt\omega} > 0$. In Figure 2 we have considered that all setup variables take the value 1 but if there is a binary variable that takes the value 0 in the solution of the master problem, we remove the corresponding vertical arc in the network. In such a case, we are still left with a shortest path problem to solve for each commodity d_{rtw}, which gives us a solution to the primal subproblem. To obtain a dual solution, we use the properties of network flow duality. Indeed, each node in Figure 2 is linked to a flow conservation constraint in the PSP. For each network representing a specific subproblem, i.e., a specific commodity d_{rtw}, the dual value linked to each node can be computed as the shortest path to

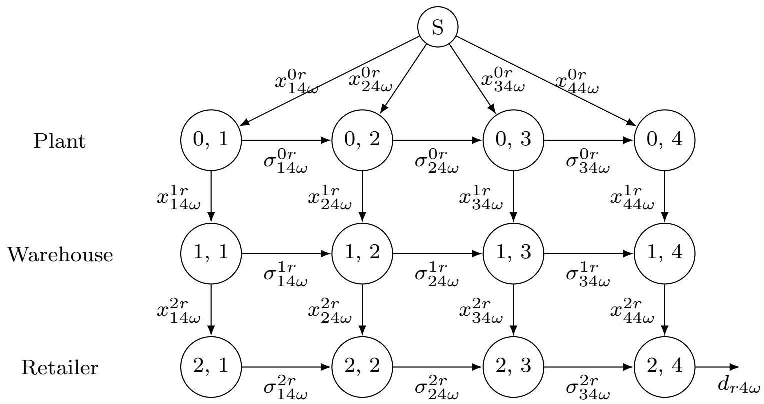

Figure 2: Graphical representation of one subproblem $(t = 4)$

go from the source node to this particular node (see Ahuja et al. [3]). These dual values correspond to the optimal values of the ψ variables in DSP. Using the structure of (19)–(21), the other dual values related to subproblem $DSP_{rt\omega}$ are computed as follows:

$$\phi^{0r}_{kt\omega} = \psi^{0r}_{kt\omega} \qquad \forall\ k \le t \in T \tag{26}$$
$$\phi^{1r}_{kt\omega} = \max\{\psi^{1r}_{kt\omega} - \psi^{0r}_{kt\omega}; 0\} \qquad \forall\ k \le t \in T \tag{27}$$
$$\phi^{2r}_{kt\omega} = \max\{\psi^{2r}_{kt\omega} - \psi^{1r}_{kt\omega}; 0\} \qquad \forall\ k \le t \in T. \tag{28}$$

We can further exploit the decomposition mentioned previously to quickly compute the dual values, i.e., the shortest paths to go to each node in Figure 2. We first compute the shortest paths at the plant level, in $O(|T|)$. We then use these values to compute the shortest paths at the warehouse level, in $O(|W||T|)$. We finally use these values to compute the shortest paths at the retailer level, in $O(|R||T|)$. Note that these shortest paths do not depend on the commodity but only on the setup values. The length of the shortest path, however, depends only on the retailer considered. We can therefore compute all the shortest paths in $O(|F||T|)$. These values are assigned to the ψ variables, which are commodity specific. The values of the ψ variables are further used to compute the values of the ϕ variables. We are therefore able to compute an optimal dual solution in $O(|R||T|^2|\Omega|)$.

4.3 A Benders-based branch-and-cut algorithm

It is well known that the optimality cuts (24) can be generated from any solution and not only from an optimal integer solution to the master problem. Therefore, we can solve the 2S-3LSPD in a standard B&C framework with the use of callbacks. At each node of the B&C tree for the Benders reformulation, the dual subproblem is solved, thus generating an optimality cut. Indeed, each $DSP_{rt\omega}$ acts as a separation problem to generate cuts. Such an implementation is possible through the use of callbacks available in general-purpose solvers, see, e.g., Adulyasak et al. [2] for such an implementation.

4.4 Enhancements

Even when implemented using callbacks, there are several features that slow down the Benders-based B&C. We implemented several ideas to speed up the solution process, which are presented in the following sections. The first three ideas aim to improve the lower bound during the search process. The fourth idea deals with the choice of a good optimality cut and the fifth one explores the different ways of aggregating cuts from the different subproblems.

4.4.1 Lower bound lifting inequalities

One major drawback of the Benders approach is the poor value of the lower bound during the search process, especially in the earlier stages. We can tackle this issue by adding some lower bound lifting inequlities (LBL) to the master problem. The purpose of these inequalities is to give a better approximation of the cost of the primal subproblem, given a set of binary setup values. The use of such inequalities has proven successful in Adulyasak et al. [2] in the context of the production routing problem under demand uncertainty. Here, we compute the minimal holding costs that will be incurred, given a feasible integer solution to the master problem. This solution is seen as a replenishment plan for each facility and we associate a new set of binary variables to these plans.

Let μ_{ivt} be a binary variable that takes the value 1 if there is production or an order placed by facility i in period v and the next production or order period is in period t ($v < t$). For these variables, we further define a dummy period $|T| + 1$ to give the possibility for a plan to have a setup in the last period of the actual time horizon. All the fixed costs, variable costs and demands associated to this dummy period for each facility are 0. We associate a cost c_{ivt} to the μ_{ivt} variables which is defined as follows:

$$c_{ivt} = \begin{cases} \sum_{\omega\in\Omega} p_\omega \sum_{k=v}^{t-1} \sum_{l=v}^{k-1} hc_{pl} d_{pk\omega} & \text{if } i = p \\ \sum_{\omega\in\Omega} p_\omega \sum_{k=v}^{t-1} \sum_{l=v}^{k-1} (hc_{il} - hc_{pl})\, d_{ik\omega} & \text{if } i \in W \\ \sum_{\omega\in\Omega} p_\omega \sum_{k=v}^{t-1} \sum_{l=v}^{k-1} \left(hc_{il} - hc_{\delta_w(i)l}\right) d_{ik\omega} & \text{if } i \in R. \end{cases}$$

These costs c_{ivt} represent the cost one has to pay at the plant for the holding cost and the additional holding cost one has to pay when the goods are transferred to the warehouse or retailer. Note that there is no assumption made about the holding costs to define the costs c_{ivt}. The following LBL inequalities are added to the master problem:

$$\sum_{v=1}^{t-1} \mu_{ivt} = y_{it} \qquad \forall\, i \in F,\ 2 \le t \in T \tag{29}$$

$$\sum_{v=1}^{t-1} \sum_{s=t}^{|T|+1} \mu_{ivs} = 1 \qquad \forall\, i \in F,\ 2 \le t \in T \tag{30}$$

$$\sum_{i\in F} \sum_{t=2}^{|T|+1} \sum_{v=1}^{t-1} c_{ivt}\mu_{ivt} \le \sum_{\omega\in\Omega} \sum_{r\in R} \sum_{t\in T} z_{rt\omega}. \tag{31}$$

Constraints (29) link the new μ variables to the original binary setup variables. Constraints (30) indicate that there must be one replenishment plan chosen for each period. Finally, constraints (31) give a lower bound on the sum of the artificial variable $z_{rt\omega}$ for the master problem.

4.4.2 Optimality cuts based on fractional solutions

In the initial iterations of the Benders algorithm, there are too few optimality cuts (24) to have a good approximation of the cost linked to each subproblem. We thus add some optimality cuts based on fractional solutions, at the root node only. The main advantage of this addition is that, when added at the root node, the optimality cuts generated are valid for the whole search tree.

To generate these optimality cuts, we work in a B&C framework with the use of callbacks. When we obtain a fractional solution to the master problem we pass this solution to the different subproblems $DSP_{rt\omega}$ and solve them as usual, giving optimality cuts for the master problem. If we solve each $DSP_{rt\omega}$ by means of the procedure described in Section 4.2, there are some changes to be made. Indeed, when we start from a fractional solution, we no longer have shortest path problems to solve as the primal subproblems. The reason is that some arcs now have capacities which are the fractional values of the setup variables. Therefore, we solve minimum cost flow problems in the same networks as the one depicted in Figure 2. To solve the different minimum cost flow problems, we designed a specific procedure, based on the structure of the holding costs we impose at each level. Indeed, we consider that the holding costs are higher when we go downstream in the supply chain, i.e., $hc_{pt} \leq hc_{\delta_w(r)t} \leq hc_{rt}$ for any retailer r and any period t. The dual values for each node are then obtained by solving shortest path problems in the residual graph of the network once the minimum cost flow problem has been solved (see Ahuja et al. [3]). In this residual graph, the costs are left unchanged.

When solving the minimum cost flow problem for commodity $d_{rt\omega}$, we have to find the cheapest way to have a flow of $d_{rt\omega}$ units going out of the network. The main idea of the procedure is to produce the demand as late as possible, and to send it to the lowest levels as late as possible. Indeed, as we have no negative costs on the arcs, a late production will give us savings on the inventory holding costs. Recall also that in the PSP there are no setup costs. In a similar spirit, as the holding costs are lower if we are more upstream in the supply chain, we should keep inventory at these levels as long as possible before sending the goods to the downstream levels. In the following paragraphs, we consider that we want to solve the subproblem associated with commodity $d_{rt\omega}$.

The idea of the procedure is to work backward, starting from period t at the retailer level and to obtain a flow equal to $d_{rt\omega}$ as late as possible. Therefore, we push the flow as late as possible, depending on the available capacities, i.e., the values of the setup variables obtained from the master problem. For each flow that we can push at the retailer level, we must also find an available path to obtain it at the warehouse level, while respecting the capacity requirements. In the same vein, for each flow that we push at the warehouse level we must find an available path to obtain it at the production plant level. The complete procedure is given in Algorithm 1, where $\text{cost}_{rt\omega}$ represents the cost of the minimum cost flow problem for retailer r in period t under scenario ω, and ϵ is a small value. Throughout the algorithm, the notations $a+=b$ and $a-=b$ are used to denote the operations $a=a+b$ and $a=a-b$, respectively.

Algorithm 1 Solution of the min cost flow problem for retailer r in period t under scenario w

for $1 \leq l \leq t$ **do**
  $\text{capa}_{pl} = y_{pl} d_{rt\omega},\ \text{capa}_{\delta_w(r)l} = y_{\delta_w(r)l} d_{rt\omega}$
end for
$\text{flow} = y_{rt} d_{rt\omega}, t' = t-1, \text{cost}_{rt} = 0$
while $\text{flow} < d_{rt} + \epsilon$ **do**
  $\text{addedFlowR} = \min\{y_{rt'} d_{rt\omega}, d_{rt\omega} - \text{flow}\}$
  $\text{flow} \mathrel{+}= \text{addedFlowR},\ \text{cost}_{rt\omega} \mathrel{+}= \sum_{l=t'}^{t-1} hc_{rl}\text{addedFlowR},\ \text{flow}_w = \text{capa}_{\delta_w(r)t'},\ t'' = t'-1$
  while $\text{flow}_w < \text{addedFlowR} + \epsilon$ **do**
    $\text{addedFlowW} = \min\{\text{capa}_{wt''}, \text{addedFlowR} - \text{flow}_w\}$
    $\text{flow}_w \mathrel{+}= \text{addedFlowW},\ \text{cost}_{rt\omega} \mathrel{+}= \sum_{l=t''}^{t'-1} hc_{\delta_w(r)l}\text{addedFlowW},\ \text{capa}_{wt''} \mathrel{-}= \text{addedFlowW},$
    $\text{flow}_p = \text{capa}_{pt''},\ t''' = t''-1$
    while $\text{flow}_p < \text{addedFlowW} + \epsilon$ **do**
      $\text{addedFlowP} = \min\{\text{capa}_{pt'''}, \text{addedFlowW} - \text{flow}_p\}$
      $\text{flow}_p \mathrel{+}= \text{addedFlowP},\ \text{cost}_{rt\omega} \mathrel{+}= \sum_{l=t'''}^{t''-1} hc_{pl}\text{addedFlowP},\ \text{capa}_{pt'''} \mathrel{-}= \text{addedFlowP},\ t''' = t''-1$
    end while
    $t'' = t'-1$
  end while
  $t' = t'-1$
end while

An example of a solution obtained by Algorithm 1 is given in Figure 3 for a retailer r under scenario ω and for $t=4$. In Figure 3, the value of the flow going through each horizontal arc, i.e., the inventory on hand at the end of a particular time period, is directly written below the arcs. For the flow between the facilities, the first number in parentheses represents the value obtained from the master problem for the setup variables. The second number represents the actual flow between two facilities, i.e., the solution for the ψ variables for $DSP_{rt\omega}$. For ease of representation, we have assumed a demand of one unit in Figure 3.

4.4.3 Addition of MIR inequalities

Bodur and Luedtke [10] mention that Benders decomposition usually does not take advantage of the integrality requirements on the master problem variables when deriving Benders cuts. They use

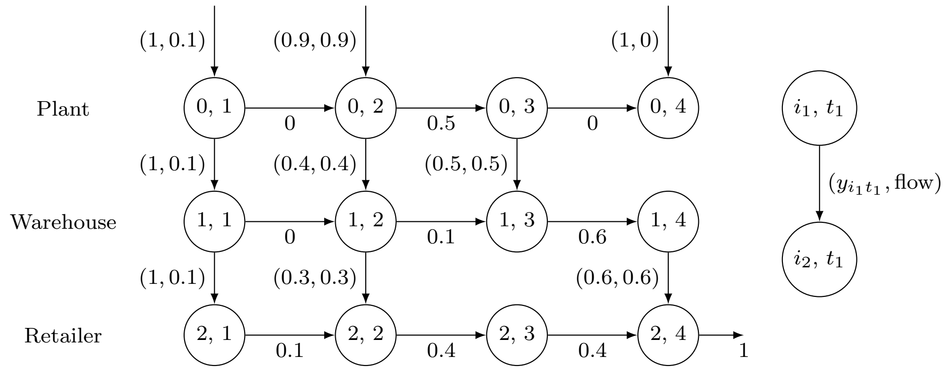

Figure 3: Graphical representation of the solution procedure for one subproblem

this observation to develop valid inequalities which lead to better LP relaxation values for the master problem. Their idea is to use a more elaborate mixed-integer rounding procedure than the one originally proposed by Wolsey [38]. They assume that there is a current set $\mathcal{V}_{rt\omega}$ of valid inequalities that describe the solution space of each subproblem, and which at least contains the Benders optimality cuts generated so far in the solution process. Then, given an optimal solution to the restricted master problem and given the associated Benders optimality cut, they develop a valid inequality that combines this cut and another cut already present in $\mathcal{V}_{rt\omega}$.

In more detail, let us assume that the subproblem corresponding to commodity $d_{rt\omega}$ has generated Benders optimality cuts of the form:

$$z_{rt\omega} \geq b_{rt\omega} - \sum_{i \in F} a_{it\omega} y_{it}, \tag{32}$$

where $b_{rt\omega}$ and $a_{it\omega}$ are scalar coefficients. Let $z_{rt\omega} \geq b'_{rt\omega} - \sum_{i \in F} a'_{it\omega} y_{it}$ be the Benders cut obtained in the last iteration performed. Given this new Benders cut, for each Benders optimality cut of the form (32), we derive the following valid inequality for BD-MC:

$$z_{rt\omega} \geq c^0_{rt\omega} - \sum_{i \in F} c^1_{it\omega} y_{it}, \tag{33}$$

where $c^0_{rt\omega} = b'_{rt\omega} + \frac{f_0 \lceil \beta (b_{rt\omega} - b'_{rt\omega}) \rceil}{\beta}$, $c^1_{it\omega} = \frac{\min\{f_0 \lceil \beta (a_{it\omega} - a'_{it\omega}) \rceil, f_i + f_0 \lfloor \beta (a_{it\omega} - a'_{it\omega}) \rfloor\}}{\beta} + a'_{it\omega}$, $f_i = \beta\,(a_{it\omega} - a'_{it\omega})$ - $\lfloor \beta\,(a_{it\omega} - a'_{it\omega}) \rfloor$, $f_0 = \beta\,(b_{it\omega} - b'_{itw}) - \lfloor \beta\,(b_{it\omega} - b'_{itw}) \rfloor$ and β is a scalar parameter, $0 < \beta \leq 1$. We then compute the scaled violation of each possible valid inequality of the form (33). If we denote by $c = (c^0, c^1)$, this scaled violation is defined as

$$\frac{\max\{c^0_{rt\omega} - \sum_{i \in F} c^1_{it\omega} \widehat{y}_{it} - \widehat{z}_{rt\omega}, 0\}}{\| (1, c) \|_2}. \tag{34}$$

We finally add to the master problem the valid inequality of the form (33) that yields the largest scaled violation. In our experiments, we apply this MIR procedure only at the root node.

4.4.4 Pareto-optimal cuts

One drawback when using Benders decomposition is that the primal subproblem can be highly degenerate, leading to numerous possible optimal dual solutions, each defining a different Benders cut. This problem has been first observed by Magnanti and Wong [24]. This is the case for our problem since we solve shortest path problems or minimum cost flow problems. To tackle this issue, it is possible to solve an auxiliary problem which returns, among the optimal solutions to the dual subproblem, the best one in terms of dominance of the cut generated. The dominance of a cut is defined as follows. Let $f(u) + yg(u) \leq z$ and $f(u_1) + yg(u_1) \leq z$ be two cuts obtained from dual solutions u and u_1, respectively. For a minimization problem, the cut obtained from the dual solution u dominates the one obtained from the dual solution u_1 if and only if $f(u) + yg(u) \geq f(u_1) + yg(u_1)$ with strict inequality holding for some solution y to the master problem. If the cut obtained from the dual solution u is not dominated by any other cut, it is said to be Pareto-optimal.

To obtain such Pareto-optimal cuts, one must solve an auxiliary problem which chooses, among the optimal solutions to the DSP, one that is undominated. This auxiliary problem is a slightly modified version of the DSP. The first difference is that there is an additional constraint stating that the objective function must be equal to the optimal value found when solving the original DSP. The second difference is that, instead of using the solution obtained from the master problem in the coefficients of the objective function (15), we use a core point y^0 which is in the relative interior of the master problem solution space. Let DSP^* be the optimal value of DSP given a solution $\widehat{y}$ for the master problem. The auxiliary problem used to obtain Pareto-optimal cuts is given as follows:

$$\text{Max} \sum_{\omega\in\Omega}\sum_{t\in T}\sum_{r\in R}\left(d_{rt\omega}\psi^{2r}_{tt\omega} - \sum_{k\leq t} d_{rt\omega}\left(y^0_{pk}\phi^{0r}_{kt\omega} + y^0_{\delta_w(r)k}\phi^{1r}_{kt\omega} + y^0_{rk}\phi^{2r}_{kt\omega}\right)\right) \tag{35}$$

$$\text{s. t. } (16) - (22) \tag{36}$$

$$\sum_{\omega\in\Omega}\sum_{t\in T}\sum_{r\in R}\left(d_{rt\omega}\psi^{2r}_{tt\omega} - \sum_{k\leq t} d_{rt\omega}\left(\widehat{y}_{pk}\phi^{0r}_{kt\omega} + \widehat{y}_{\delta_w(r)k}\phi^{1r}_{kt\omega} + \widehat{y}_{rk}\phi^{2r}_{kt\omega}\right)\right) = DSP^*. \tag{37}$$

One drawback with this auxiliary problem is that constraint (37) breaks the separability that we originally had in the DSP. In the case of network design problems, Magnanti et al. [25] have shown that it is possible to obtain Pareto-optimal cuts by solving a parametric minimum cost flow problem instead of solving both the DSP and the auxiliary problem. In our case, we can use a similar approach and solve a single problem to derive Pareto-optimal cuts. Let σ, x and μ be the dual variables linked to constraints (16)–(18), (19)–(21) and (37), respectively. We consider the dual of the auxiliary problem given by:

$$\text{Min} \sum_{\omega\in\Omega} p_\omega \sum_{t\in T}\sum_{r\in R}\sum_{k\leq t}\left(hc_{pk}\sigma^{0r}_{kt\omega} + hc_{\delta_w(r)k}\sigma^{1r}_{kt\omega} + hc_{rk}\sigma^{2r}_{kt\omega}\right) - DSP^*\mu \tag{38}$$

$$\sigma^{0r}_{k-1,t,\omega} + x^{0r}_{kt\omega} = x^{1r}_{kt\omega} + \sigma^{0r}_{kt\omega} \qquad \forall\ t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{39}$$

$$\sigma^{1r}_{k-1,t,\omega} + x^{1r}_{kt\omega} = x^{2r}_{kt\omega} + \sigma^{1r}_{kt\omega} \qquad \forall\ t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{40}$$

$$\sigma^{2r}_{k-1,t,\omega} + x^{2r}_{kt\omega} = \delta_{kt}d_{rt\omega}(1+\mu) + (1-\delta_{kt})\sigma^{2r}_{kt\omega} \qquad \forall\ t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{41}$$

$$x^{0r}_{kt\omega} \leq d_{rt\omega}\left(y^0_{pk} + \widehat{y}_{pk}\mu\right) \qquad \forall\ t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{42}$$

$$x^{1r}_{kt\omega} \leq d_{rt\omega}\left(y^0_{\delta_w(r)k} + \widehat{y}_{\delta_w(r)k}\mu\right) \qquad \forall\ t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{43}$$

$$x^{2r}_{kt\omega} \leq d_{rt\omega}\left(y^0_{rk} + \widehat{y}_{rk}\mu\right) \qquad \forall\ t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{44}$$

$$x^{0r}_{kt\omega}, x^{1r}_{kt\omega}, x^{2r}_{kt\omega}, \sigma^{0r}_{kt\omega}, \sigma^{1r}_{kt\omega}, \sigma^{2r}_{kt\omega} \geq 0 \qquad \forall\ t\in T, k\leq t\in T, r\in R, \omega\in\Omega. \tag{45}$$

This problem is exactly a parametric minimum cost flow problem where there is a rebate of DSP^* given for each extra unit of flow of the commodity routed in the network. For this problem, the demand is given by (41) and the capacities on the arcs are given by (42)–(44). In Magnanti et al. [25], the authors show that any fixed value $\mu \geq \sum_{i \in F} \sum_{t \in T} y^0_{it}$ is optimal for the problem. This leads to a minimum cost flow problem to be solved for each commodity $d_{rt\omega}$. Note that this minimum cost flow problem can be solved by means of the procedure described in Algorithm 1.

Finally, it has been seen in the literature that the core point selection leads to different Pareto-optimal cuts, see in particular Magnanti and Wong [24]. In our experiments, we tested two strategies for the core point selection. In the first strategy, for any facility i, the core point is fixed during the whole process to $y^0_{i1} = 1$ and $y^0_{it} = 0.5$ if $t \geq 2$. The second strategy is similar to the one used in Papadakos [27]. In the second strategy, for any facility i, the core point is initialized to $y^0_{it} = 1$ and dynamically updated as $y^0_{it} = 0.5 y^0_{it} + 0.5 \widehat{y}_{it}$.

4.4.5 Cut aggregation

As mentioned in the previous sections, each $DSP_{rt\omega}$ gives one possible optimality cut to be added to the master problem. Therefore, we could add one optimality cut per supbroblem to accelerate the convergence of our algorithm (see Birge and Louveaux [9]). However, this addition of a large number of cuts at the same time may worsen the performance of the algorithm because of the time taken to solve the master problem at each iteration (see de Camargo et al. [13]). We tested eight ways of adding cuts to the master problem at each iteration: adding one cut per subproblem, adding one cut per retailer, adding one cut per period, adding one cut per scenario, adding one cut per retailer and scenario, adding one cut per retailer and time period, adding one cut per scenario and time period, or adding one single cut. The effect of these strategies is discussed in Section 5.

5 Numerical experiments

In order to assess the performance of our decomposition approach to solve the 2S-3LSPD, we conducted numerical experiments based on the instances used in Gruson et al. [17]. In their experiments, the authors set the number of retailers $|R|$ equal to 50, 100 or 200, and the length of the time horizon $|T|$ is equal to 15 or 30. The number of warehouses $|W|$ is set equal to 5, 10, 15 or 20. Note that we just consider a number of warehouses equal to 5, 10 or 20. The demand at the retailers is generated both in a static and dynamic way from U$[5, 100]$. The fixed costs at all levels are generated in a static and in a dynamic way. The fixed costs are generated from U$[30000, 45000]$, from U$[1500, 4500]$, and from U$[5, 100]$ for the production plant, the warehouses and the retailers, respectively. All the demands and fixed costs are generated as integer values. The unit inventory holding costs are static and are set to 0.25 and to 0.5 for the production plant and for the warehouses, respectively. For the retailers, the unit inventory holding costs are generated from U$[0.5, 1]$. The holding costs take continuous values. For each combination of settings, Gruson et al. [17] generated five instances leading to 480 different instances to be solved.

For the experiments, we used the CPLEX 12.8.1.0 C++ library and turned off CPLEX's parallel mode. We set the CPLEX MIP tolerance parameter to 10^{-6}. All the other CPLEX parameters are set to their default value. The computation time limit imposed to solve each instance is 6 hours. For the cuts added at the root node, we set a limit to 0, 50 or 100. This is done in order not to add too many cuts at the root node, which would remove only a relatively small portion of the search space. Finally, we use the MIR procedure every 1, 5 or 10 iterations, and only when cuts are added at the root node. For the MIR procedure, the scaled parameter β is set to 0.5. Note that the results reported in this section all take into account the addition of LBL inequalities. Indeed, initial experiments have shown that their addition has a huge impact on the performance of the decomposition approach. We further impose initial setups at each facility. Indeed, as we do not have initial inventory, there must be production and an order placed by each warehouse and retailer to satisfy the demand of the first period for each retailer.

The following two sections give the results for deterministic and stochastic instances, respectively. In the deterministic instances, there is one unique possible scenario, i.e., $|\Omega| = 1$. In the following tables, we assess the performance of our decomposition approach with respect to different indicators: the number of optimal solutions found (opt), the CPU time taken to solve the instance (Time) and the subproblems (Time-SP), the best lower bound obtained during the search (BLB), the objective function value of the MIP optimal solution when available, or the cost of the best solution found otherwise (BUB), the number of nodes in the search tree (Nodes), the number of times the master problem was solved (It), the gap reported at the end of the time limit (Gap) and, finally, the gap compared to the solution given by CPLEX (C-Gap). For a particular instance, the gap compared to the solution given by CPLEX is the gap between the best solution found by a particular version of the algorithm and the best solution given by CPLEX at the end of the CPU time limit. The gap reported at the end of the time limit, for a particular version of the algorithm, is the gap between the best lower and upper bounds found during the search.

For ease of reading, we only display a subset of the results obtained among all our tests. This subset uses the setting for the MIR procedure and aggregation of cuts which provided the best average results obtained over all instances. The interested reader is referred to the appendices of this report for detailed results. These results indicate, in particular, that the way cuts are aggregated has a high impact on the different indicators we look at. On the contrary, the number of cuts used at the root node and the interval between two iterations with the addition of MIR cuts have a less substantial impact.

In the tables, each line represents a particular version of the Benders-based B&C algorithm with or without the use of Pareto-optimal cuts, and with the subproblems solved by CPLEX or by the procedure we designed. When we solve a parametric minimum cost flow problem by means of our Algorithm 1 to derive Pareto-optimal cuts, this is denoted by MW in the Pareto column. In case we use Pareto-optimal cuts, we further specify if these cuts were obtained using a fixed core point (F), or using the procedure proposed by Papadakos [27] (P). In each table, the last line represents the results obtained by CPLEX directly.

5.1 Results for the deterministic 2S-3LSPD

To solve the deterministic version of the 2S-3LSPD, we use the Benders decomposition presented in Section 4. The instances we use to test the decomposition are similar to those used in Gruson et al. [17], except that we set the number of retailers equal to 200, 500 or 1000. The partitioning of the retailers among the warehouses is given in Table 1. The structure of both the fixed and variable costs is left unchanged, as well as the way the demand for each retailer is generated. For the deterministic instances, we performed our experiments on a 3.07 GHz Intel Xeon processor with only one thread.

Table 1: Assignment of the retailers to the warehouses

Number of warehouses	Number of retailers		
	200	500	1000
5	40 $\forall w \in W$	100 $\forall w \in W$	200 $\forall w \in W$
10	20 $\forall w \in W$	50 $\forall w \in W$	100 $\forall w \in W$
20	10 $\forall w \in W$	25 $\forall w \in W$	50 $\forall w \in W$

The best results obtained for the deterministic instances are presented in Tables 2 and 3. For the small instances, i.e., when $|T| = 15$, we obtained the best results with 50 cuts at the root node, an aggregation of cuts per retailer and no use of the MIR procedure. For the big instances, i.e., when $|T| = 30$, we obtained the best results with 100 cuts at the root node, an aggregation of cuts among all commodities and no use of the MIR procedure.

Several conclusions can be drawn from the analysis of the results obtained. First, one can see that there is a big difference in performance between the small and the large instances. The results of the proposed algorithms obtained over the large instances are much better than the ones obtained over the small instances in terms of the quality of the solution found compared to the one obtained by CPLEX. Indeed, one can see in Table 3 that the gap compared to CPLEX is always negative, between -34.67% and -52%. The optimality gap obtained at the end of the time horizon for these instances is, however,

Table 2: Results for an aggregation of cuts per retailer, $|T| = 15$, 50 iterations at the root node and no MIR procedure

Pareto	Core point	CPLEX	BLB	BUB	Time (s)	Time SP (s)	Nodes	It	Gap (%)	C-Gap (%)	Opt (%)
✓	F	✓	717405	789166	15256	395	7972	62	6.68	0.65	42.78
✓	P	✓	621911	1249075	15060	428	9619	64	17.49	13.84	45.56
✓	F	x	713465	786284	15420	316	8887	64	6.84	0.34	36.67
✓	P	x	647263	1206032	14745	339	12060	64	15.56	12.29	46.11
x	-	✓	708143	787213	16092	106	12755	65	7.2	0.42	31.11
x	-	x	689632	807149	21681	12	11364	71	10.65	2.39	0
MW	F	x	780419	880430	17771	14	16196	69	7.28	7.11	25.56
MW	P	x	779366	908941	20088	15	10773	75	9.45	9.17	10
	CPLEX		782467	782467	1495	-	2.4	-	0	0	100

Table 3: Results with an aggregation of cuts over all commodities, $|T| = 30$, 100 iterations at the root node and no MIR procedure

Pareto	Core point	CPLEX	BLB	BUB	Time (s)	Time SP (s)	Nodes	It	Gap (%)	C-Gap (%)	Opt (%)
✓	F	✓	1487268	1783253	21873	8190	2698	164	14.87	-48.88	2.22
✓	F	✓	1514855	1758072	21818	6765	2454	173	11.8	-50.38	1.11
✓	P	x	1491088	1781034	22046	7351	2972	170	14.55	-48.92	0.56
✓	P	x	1514845	1751745	21789	4994	2982	191	11.43	-50.76	1.11
x	-	✓	1500980	1743270	21753	1886	4640	199	12.02	-52	12.22
x	-	x	1438283	1972689	21858	89	16911	413	25.21	-34.67	0
MW	F	x	1500402	1792284	21743	153	2948	190	14.55	-47.64	1.67
MW	P	x	1493412	1796762	21797	213	4946	282	14.98	-47.05	0.56
	CPLEX		1268454	3323226	12287	-	5.5	-	13.7	0	62.78

large, usually around 15%. This illustrates the difficulty of solving these instances and shows that the use of a decomposition approach is successful in that case. Note finally that the number of nodes in the large instances is lower than the number of nodes explored for the small instances. This is explained by the increased difficulty of both the master and subproblems. On the contrary, for the small instances, the decomposition approach is dominated by CPLEX. The instances solved seem to be too easy and CPLEX is able to derive better cuts than the ones we generate during the search process.

As far as the computational enhancements we designed are concerned, they have different impacts on the performance of the decomposition approach. First, there is no clear impact of the use of Pareto-optimal cuts in the search tree. Indeed, one can see in Tables 2 and 3 that it can have contradictory effects on the performance measures. For instance, for the large instances, it seems to improve the upper bound when CPLEX is used to solve the subproblems while it can worsen it if CPLEX is not used. The same can be said about the impact on the lower bound. We note, however, that the use of these cuts decreases the number of nodes explored in the search tree. This indicates that they are able to reduce the size of the search tree since the CPU time taken to solve the subproblems, which includes the CPU time taken to solve the auxiliary problem, is similar whether we include Pareto-optimal cuts or not. When we derive Pareto-optimal cuts by means of the procedure proposed by Magnanti et al. [25] (MW), we observe a large decrease in the CPU time taken to solve the subproblems, showing the efficiency of the procedure we designed to directly obtain the Pareto-optimal cuts.

The specific procedure we designed to solve the different subproblems does not lead to any improvements in the results obtained. The CPU time spent on solving the different subproblems is in general lower when the subproblems are solved by our procedure but this brings no overall improvement. Indeed, our procedure is efficient in solving the primal subproblems and uses the primal solution to build an optimal dual solution. However, as already mentioned, the dual subproblems are highly degenerate. Therefore, the way we build the optimal dual solution seems to be unable to derive strong Benders cuts. On the contrary, the dual solution obtained by CPLEX gives dual values which allow to derive better cuts.

Finally, adding Benders cut at the root node has a positive impact on the results as illustrated by the fact that the best results are always obtained with cuts added at the root node. However, the use of a MIR procedure does not give the best possible results for the deterministic instances. A closer look at detailed results revealed that this procedure does not take too much time but produces cuts that remove a tiny portion of the search space, limiting their usefulness. Note also that the way we aggregate the different cuts led to completely different results. The aggregation of cuts per time period or per commodity gave poor results on all the different KPIs we look at.

In light of the results shown in Tables 2 and 3, we can draw the following conclusions. First, the Benders decomposition approach is able to find solutions of much better quality than CPLEX for very large instances. Second, the solution of the subproblems by CPLEX is more time consuming but leads to cuts of better quality because of the dual solution chosen. Finally, the addition of Benders cuts at the root node is beneficial and gives better results.

5.2 Results for the 2S-3LSPD

To solve the 2S-3LSPD, we use the Benders decomposition presented in Section 4. The instances we use to test the decomposition are similar to those used in Gruson et al. [17]. The way the demand for each retailer is generated in each scenario is left unchanged. The number of demand scenarios generated is set equal to 5, 50 or 100. The number of time periods is set equal to 15. For these instances, we performed our experiments on a 2.1 GHz Intel E5-2683 v4 processor with only one thread.

We further report the expected value of perfect information (EVPI) and the value of the stochastic solution (VSS). These indicators were proposed by Birge and Louveaux [8] in the context of two-stage linear programs. The EVPI represents the additional cost compared to a solution where the actual demand for the whole time horizon would have been known in the first period. The VSS represents the possible gain (in terms of lower expected costs) from solving the actual stochastic model, using the same set of scenarios, instead of using the expected value for the parameters and applying the resulting plan to each of these scenarios of demand. In the following tables, these indicators are computed as a percentage of the optimal solution cost.

Tables 4–6 present the results we obtained during the computational experiments. We only display the best results obtained among all the experiments we performed. For the 2S-3LSPD, these best results were obtained with one cut added per retailer and time period at each iteration. This finding is in line with the results obtained by Adulyasak et al. [2].

Table 4: Results with one cut added per retailer and time period, $|S| = 5, |T| = 15$, 100 iterations at the root node and MIR procedure every 10 iterations

Pareto	Core point	CPLEX	BLB	BUB	Time (s)	Time SP (s)	Nodes	It	Gap (%)	C-Gap (%)	Opt (%)	EVPI (%)	VSS (%)
✓	F	✓	321270	321271	1532	215	1752	57	0	0	97.22	0.6	-0.2
✓	P	✓	321268	321271	1498	738	3800	52	0.01	0	94.44	0.6	-0.18
✓	F	x	321270	321271	1436	113	3154	56	0	0	94.44	0.6	-0.2
✓	P	x	321270	321271	1026	137	2715	49	0	0	97.22	0.6	-0.19
x	-	✓	321270	321271	1565	52	1583	48	0	0	94.44	0.58	-0.21
x	-	x	321045	321271	2335	14	1695	63	0.02	0	91.67	0.58	-0.21
MW	F	x	321265	321271	1348	14	1269	75	0	0	94.44	0.62	-0.17
MW	P	x	321219	321271	1419	15	304	64	0.01	0	94.44	0.62	-0.17
	CPLEX		321271	321271	1320	-	0.97	-	0	0	100	-	-

Table 4 indicates that for the instances with 5 scenarios, our proposed algorithms have a similar performance compared to CPLEX. Tables 5 and 6 show the superiority of our Benders decomposition approach compared to the use of a general-purpose solver for the instances with more scenarios. First, the CPU time taken to solve the instances is much lower with our approach. Second, the solutions found are of much better quality as illustrated by the negative values obtained in the C-Gap columns. These negative values indicate that CPLEX struggles to find solutions of good quality compared to our approach. Finally, the number of optimal solutions obtained is much higher. These three elements indicate that the use of a Benders decomposition approach is well suited for this problem and outperforms CPLEX both in terms of quality of solution and efficiency to obtain this high quality

Table 5: Results with one cut added per retailer and time period, $|S| = 50, |T| = 15$, 50 iterations at the root node and MIR procedure every 5 iterations

Pareto	Core point	CPLEX	BLB	BUB	Time (s)	Time SP (s)	Nodes	It	Gap (%)	C-Gap (%)	Opt (%)	EVPI (%)	VSS (%)
✓	F	✓	325334	325339	3320	2591	661	38	1.35×10^{-5}	-45.82	97.22	0.82	-0.02
✓	P	✓	325322	325340	2856	2150	254	37	4.29×10^{-5}	-45.82	97.22	0.82	-0.02
✓	F	x	325285	325350	3142	2698	104	41	0.02	-45.81	97.22	0.82	-0.02
✓	P	x	312531	1480571	3181	2618	347	35	2.78	-40.32	94.44	3.54	2.72
x	P	✓	325304	325341	1421	536	236	38	0.01	-45.82	97.22	0.82	-0.02
x	P	x	325288	325342	2061	93	495	43	0.01	-45.82	97.22	0.82	-0.02
MW	F	x	325338	325339	993	90	3272	48	2.16×10^{-6}	-45.82	97.22	0.82	-0.02
MW	P	x	325300	325341	948	87	165	42	0.01	-45.82	97.22	0.82	-0.02
	CPLEX		268866	509431	13602	-	0.22	-	47.22	0	50	-	-

Table 6: Results with one cut added per retailer and time period, $|S| = 100, |T| = 15$, 100 iterations at the root node and MIR procedure every 10 iterations

Pareto	Core point	CPLEX	BLB	BUB	Time (s)	Time SP (s)	Nodes	It	Gap (%)	C-Gap (%)	Opt (%)	EVPI (%)	VSS (%)
✓	F	✓	295841	4850361	21631	5457	96	50	5.56	-43.97	91.67	6.25	5.51
✓	P	✓	298997	4815522	6119	5844	42	53	5.56	-42.98	94.44	6.26	5.51
✓	F	x	295674	4811660	5610	4981	119	55	5.57	-44.27	91.67	6.27	5.51
✓	P	x	309570	2548929	5022	4536	109	52	2.8	-49.32	94.44	3.52	2.75
x	P	✓	321120	321217	2474	1157	360	55	0.02	-55.13	94.44	0.79	-0.01
x	P	x	321084	321213	2449	371	237	67	0.03	-55.13	94.44	0.79	-0.01
MW	F	x	321130	321210	1834	351	988	88	0.02	-55.13	97.22	0.79	-0.01
MW	P	x	321158	321211	1688	277	351	59	0.01	-55.13	97.22	0.79	-0.01
	CPLEX		227051	528880	17199	-	0	-	33.33	0	53.33	-	-

solution. One can note in the results reported here that the enhancements have a bigger influence in the experiments with the stochastic instances than in the deterministic case. Indeed, the best results are always obtained with the use of the MIR procedure, indicating the benefits of using such a procedure for the stochastic problem. Here, the use of our procedure to derive Pareto-optimal cuts gives excellent results both in terms of CPU time to solve the instances, on the number of optimal solutions found and on the quality of the solution.

In light of the results shown in Tables 4–6, we can draw the following conclusions. First, the Benders decomposition approach is able to find solutions of much better quality than CPLEX for a large number of instances and the aggregation of cuts among all scenarios gives excellent results. Then, the use of Pareto-optimal cuts is beneficial especially when we derive cuts without solving an auxiliary problem. Finally, the use of a MIR procedure helps accelerate the solution process.

6 Conclusions and future research

We have tackled the 2S-3LSPD and have applied a Benders decomposition to solve it, starting from the multi-commodity formulation of the problem. This decomposition yields numerous subproblems, one for each commodity. The use of such a decomposition naturally leads to the development of a Benders-based branch-and-cut algorithm where the solution of the different subproblems acts as a separation algorithm. We have further proposed some improvements to the initial algorithm. These improvements include the generation of Pareto-optimal cuts, the generation of cuts at the root node of the tree, the use of a MIR procedure and the addition of lower bound lifting inequalities. We have also developed a specific procedure to efficiently solve the different subproblems and the Pareto problem instead of using a general-purpose solver.

We have performed extensive numerical experiments to assess the performance of our decomposition technique on the solution of the problem. The use of a Benders-based branch-and-cut algorithm to solve the deterministic version of the 2S-3LSPD gave good results, especially for the large instances. The cuts added at the root node and the LBL inequalities also speed up the solution process, along with the use of CPLEX to solve the subproblems. The results obtained on the stochastic instances show the superiority of the Benders decomposition approach over CPLEX. The further use of our specific procedure to derive Pareto-optimal cuts without solving an auxiliary problem is in particular very useful to speed up the solution process. Finally, for the stochastic instances, using a MIR procedure also improves the CPU time taken to solve the instances.

In future research we want to introduce routing decisions in the problem between the warehouse and the retailers instead of having direct shipments. The approach introduced here could be used in a heuristic if the routing decisions are relaxed.

Appendices

See G–2019–51's supplementary material for Tables 7–222.

References

[1] S. Abdullah, A. Shamayleh, and M. Ndiaye. Three stage dynamic heuristic for multiple plants capacitated lot sizing with sequence-dependent transient costs. Computers & Industrial Engineering, 127:1024–1036, 2019.

[2] Y. Adulyasak, J.-F. Cordeau, and R. Jans. Benders decomposition for production routing under demand uncertainty. Operations Research, 63:851–867, 2015.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. Network Flows: Theory, Algorithms, and Applications. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1993.

[4] M. A. Aloulou, A. Dolgui, and M. Y. Kovalyov. A bibliography of non-deterministic lot-sizing models. International Journal of Production Research, 52:2293–2310, 2014.

[5] H. C. Bahl and S. Zionts. Multi-item scheduling by Benders' decomposition. The Journal of the Operational Research Society, 38(12):1141–1148, 1987.

[6] T. Bayley, H. Süral, and J. H. Bookbinder. A hybrid Benders approach for coordinated capacitated lot-sizing of multiple product families with set-up times. International Journal of Production Research, 56(3):1326–1344, 2018.

[7] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik, 4:238–252, 1962.

[8] J. R. Birge and F. Louveaux. Introduction to Stochastic Programming. Springer-Verlag, New-York, 1997.

[9] J. R. Birge and F. V. Louveaux. A multicut algorithm for two-stage stochastic linear programs. European Journal of Operational Research, 34(3):384–392, 1988.

[10] M. Bodur and J. R. Luedtke. Mixed-integer rounding enhanced benders decomposition for multiclass service-system staffing and scheduling with arrival rate uncertainty. Management Science, 63(7):2073–2091, 2016.

[11] J. H. Bookbinder and J.-Y. Tan. Strategies for the probabilistic lot-sizing problem with service-level constraints. Management Science, 34:1037–1156, 1988.

[12] N. Brahimi, N. Absi, S. Dauzère-Pérès, and A. Nordli. Single-item dynamic lot-sizing problems: An updated survey. European Journal of Operational Research, 263(3):838–863, 2017.

[13] R. S. de Camargo, J. G. de Mirande, and H. P. Luna. Benders decomposition for the uncapacitated multiple allocation hub location problem. Computers & Operations Research, 35:1047–1064, 2008.

[14] C. Dhaenens-Flipo and G. Finke. An integrated model for an industrial production-distribution problem. IIE Transactions, 33:705–715, 2001.

[15] M. Di Summa and L. A. Wolsey. Lot-sizing on a tree. Operations Research Letters, 36(1):7–13, 2008.

[16] P. S. Dixon and E. A. Silver. A heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing problem. Journal of Operations Management, 2:23–39, 1981.

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

[18] M. Gruson, J.-F. Cordeau, and R. Jans. The impact of service level constraints in deterministic lot sizing with backlogging. Omega, 79:91–103, 2018.

[19] S. Guan, Y. AMD Ahmed, G. L. Nemhauser, and A. J. Miller. A branch-and-cut algorithm for the stochastic uncapacitated lot-sizing problem. Mathematical Programming, 105(1):55–84, 2006.

[20] J. Gutiérrez, J. Puerto, and J. Sicilia. The multiscenario lot size problem with concave costs. European Journal of Operatinal Research, 156:162–182, 2004.

[21] K. K. Haugen, A. Løkketange, and D. L. Woodruff. Progressive hedging as a meta-heuristic applied to stochastic lot-sizing. European Journal of Operatinal Research, 132:116–122, 2001.

[22] S. Helber, F. Sahling, and K. Schimmelpfeng. Dynamic capacitated lot sizing with random demand and dynamic safety stocks. OR Spectrum, 35:75–105, 2013.

[23] J. Maes, J. O. McClain, and L. N. Van Wassenhove. Multilevel capacitated lot sizing complexity and LP-based heuristics. European Journal of Operational Research, 53(2):131–148, 1991.

[24] T. Magnanti and R. Wong. Accelerating Benders decomposition: algorithmic enhacement and model selection criteria. Operations Research, 23:464–484, 1981.

[25] T. L. Magnanti, P. Mireault, and R. T. Wong. Tailoring Benders decomposition for uncapacitated network design. Mathematical Programming Study, 26:112–154, 1986.

[26] R. A. Melo and L. A. Wolsey. Uncapacitated two-level lot-sizing. Operations Research Letters, 38(4):241–245, 2010.

[27] N. Papadakos. Practical enhancements to the Magnanti-Wong method. Operations Research Letters, 36(4):444–449, 2008.

[28] Y. Pochet and L. A. Wolsey. Production Planning by Mixed Integer Programming. Springer, New York, NY, USA, 2006.

[29] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The Benders decomposition algorithm: A literature review. European Journal of Operational Research, 259(3):801–817, 2017.

[30] R. T. Rockafellar and R. J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. Mathematics of Operations Research, pages 119–147, 1991.

[31] F. Sahling, L. Buschkühl, H. Tempelmeier, and S. Helber. Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. Computers & Operations Research, 36(9):2546–2553, 2009.

[32] S. Taskin and E. J. Lodree Jr. Inventory decisions for emergency supplies based on hurricane count predictions. International Journal of Production Economics, 126:66–75, 2010.

[33] H. Tempelmeier. On the stochastic uncapacitated dynamic single-item lot sizing problem with service level constraints. European Journal of Operational Research, 181(1):184–194, 2007.

[34] H. Tempelmeier. Stochastic lot sizing problems. In J. M. Smith and B. Tan, editors, Handbook of Stochastic Models and Analysis of Manufacturing System Operations, pages 313–344. Springer New York, New York, NY, 2013.

[35] H. Tempelmeier and S. Helber. A heuristic for dynamic multi-item multi-level capacitated lot sizing for general product structures. European Journal of Operational Research, 75(2):296–311, 1994.

[36] H. Tunc, O. A. Kilic, S. A. Tarim, and R. Rossi. An extended mixed-integer programming formulation anddynamic cut generation approach for the stochastic lot-sizing problem. INFORMS Journal on Computing, 30(3):492–506, 2018.

[37] H. M. Wagner and T. M. Whitin. Dynamic version of the economic lot size model. Management Science, 5:89–96, 1958.

[38] L. A. Wolsey. Integer Programming. Wiley, New York, 1998.

[39] S. Zhang and H. Song. Production and distribution planning in Danone waters China division. INFORMS Journal on Applied Analytics, 48:578–590, 2018.