HEC MONTRÉAL
École affiliée à l'Université de Montréal

Modélisation et résolution de problèmes de planification de production et de distribution à trois niveaux

par
Matthieu Gruson

Thèse présentée en vue de l'obtention du grade de Ph. D. en administration
(option Gestion des opérations et de la logistique)

Octobre 2020

HEC MONTRÉAL

École affiliée à l'Université de Montréal

Cette thèse intitulée :

Modélisation et résolution de problèmes de planification de production et de distribution à trois niveaux

Présentée par :

Matthieu Gruson

a été évaluée par un jury composé des personnes suivantes :

Yossiri Adulyasak
HEC Montréal
Président-rapporteur

Jean-François Cordeau
HEC Montréal
Codirecteur de recherche

Raf Jans
HEC Montréal
Codirecteur de recherche

Sanjay Dominik Jena
ESG-UQAM
Membre du jury

Stéphane Dauzère-Pérès
École des Mines de Saint-Étienne
Examinateur externe

Chantal Labbé
HEC Montréal
Représentante du directeur de HEC Montréal

Résumé

Durant les dernières décennies, l'importance de l'intégration des décisions opérationnelles dans les entreprises manufacturières est apparue de manière criante. La principale raison derrière cette intégration, que ce soit au sein d'une entreprise ou à travers la chaîne d'approvisionnement, peut s'expliquer facilement grâce aux bénéfices potentiels, tel un meilleur niveau de service pour les clients, une réduction des coûts ou encore une plus grande flexibilité. Plusieurs études se sont appuyées sur cette observation en proposant avec succès des modèles mathématiques qui prennent en compte l'intégration des décisions opérationnelles, telles les décisions de production et de distribution. Malgré les succès remportés par ce type de travaux, l'intégration des décisions opérationnelles au sens large n'est pas encore poussée à son plein potentiel. L'objectif de cette thèse est d'utiliser les techniques et outils de la recherche opérationnelle pour faciliter l'intégration des décisions opérationnelles au sein d'une chaîne d'approvisionnement à trois niveaux. Le premier niveau est composé d'une usine de production, le second d'entrepôts et le troisième de détaillants. Le problème étudié dans cette thèse est donc un problème intégré de lotissement et réapprovisionnement à trois niveaux (3LSPD). La principale contribution émane des modèles mathématiques proposés et des algorithmes développés pour la résolution de différentes versions du problème, versions qui incluent toutes les décisions de production et de réapprovisionnement, tout en minimisant les coûts opérationnels à travers toute la chaîne d'approvisionnement. La thèse est divisée en quatre projets distincts présentés ci-après.

Nous commençons par comparer 13 formulations de programmation mixte en nombres

entiers (PMNE) pour résoudre une version capacitaire et une version non capacitaire du problème. Dans la version capacitaire, les contraintes de capacité sont imposées uniquement au niveau de l'usine de production. Les formulations proposées sont soit adaptées de formulations de PMNE existantes dans le cadre du problème *one-warehouse multi-retailer*, soit nouvellement introduites dans notre contexte. Dans ce premier travail, nous considérons que notre chaîne d'approvisionnement a une structure de distribution avec des livraisons directes entre les différents sites : les produits passent de l'usine de production à un entrepôt avant de finalement arriver chez le détaillant. De plus, chaque détaillant est desservi par un seul et unique entrepôt. Enfin, un seul item est pris en compte. Des expériences numériques sont menées pour attester des performances pratiques des formulations, comparativement aux performances théoriques que nous avons prouvées. Ce premier travail sur les formulations de PMNE sert de base pour les trois autres projets que la thèse comporte.

Dans les deuxième et troisième projets, nous utilisons des méthodes de décomposition pour respectivement résoudre une version capacitaire, et une version stochastique non capacitaire du 3LSPD. En effet, les résultats des expériences numériques menées dans le cadre du premier projet ont mis en lumière une déficience des solveurs pour résoudre adéquatement la version capacitaire du problème. À l'inverse, la résolution de la version non capacitaire du problème est tout à fait acceptable, indiquant la possibilité de s'attaquer à une version plus complexe du problème, en particulier avec ajout d'incertitude. Pour résoudre la version capacitaire du problème, nous utilisons une décomposition de Dantzig-Wolfe et développons un algorithme de séparation et génération de colonnes. Les contraintes de capacité sont imposées au niveau de l'usine de production uniquement. Elles limitent les quantités qui peuvent être produites durant chaque période. Pour résoudre la version stochastique non capacitaire du problème, nous appliquons une décomposition de Benders. Dans ce cas, la stochasticité vient de l'incertitude entourant la demande des détaillants. Cette incertitude est modélisée sous forme de scénarios de demande dont la probabilité de réalisation est connue. Nous nous plaçons dans un contexte de processus de décision en deux étapes, où les décisions prises dans la première étape

se retrouvent dans le problème maître de Benders, alors que les décisions prises dans la deuxième étape se retrouvent dans les sous-problèmes de Benders. Les décisions prises dans la première étape sont les décisions de mise en route ou de commande, alors que les décisions prises dans la seconde étape sont les décisions relatives aux quantités produites, commandées ou gardées en stock. Pour cette version stochastique, nous étudions également une extension du problème en autorisant les ventes perdues. Dans ce cas, la demande des détaillants pourrait être partiellement satisfaite, moyennant des pénalités à payer. Dans ces deux projets, des améliorations sont apportées aux algorithmes de base dans le but d'accélérer le processus de résolution du problème. Des expériences numériques sont menées pour tester nos algorithmes sur de nombreuses instances. Pour ces deux projets, nous considérons à nouveau une structure de distribution pour notre chaîne d'approvisionnement et un seul item.

Enfin, dans le quatrième projet, nous incorporons de nombreuses contraintes supplémentaires et n'imposons plus une structure de distribution dans notre chaîne d'approvisionnement. Pour ce dernier projet, nous relâchons donc une des hypothèses principales du 3LSPD et ajoutons des contraintes opérationnelles. Les produits passent toujours de l'usine aux entrepôts et des entrepôts aux détaillants, mais chaque détaillant n'est plus rattaché à un seul et unique entrepôt. Nous considérons ici le cas où les livraisons entre l'usine de production et les entrepôts, et entre les entrepôts et les détaillants, sont réalisées par des camions ayant une capacité limitée. Nous prenons aussi des décisions quant aux tournées que les camions disponibles aux entrepôts vont devoir effectuer pour réapprovisionner les différents détaillants. Enfin, nous considérons plusieurs items. Nous développons deux méthodes heuristiques pour résoudre cette extension du problème. Ces deux heuristiques décomposent le problème en plusieurs sous-problèmes résolus de manière itérative. La première méthode est une approche *top-down* où les décisions de production dictent le reste des décisions opérationnelles, soit les décisions de stockage à chaque niveau et les décisions de distribution. La seconde heuristique est une approche *bottom-up* où ce sont cette fois les décisions de distribution auprès des détaillants qui dictent le reste des décisions opérationnelles à prendre. De plus, nous analysons les possibilités offertes

par le fait de scinder les livraisons aux détaillants. La demande des détaillants de chaque période peut ainsi être livrée de plusieurs façons : en une période par un camion, en une période par plusieurs camions, en plusieurs périodes par un camion, ou en plusieurs périodes par plusieurs camions. Pour valider les résultats obtenus par ces heuristiques, nous avons également développé un algorithme de séparation et coupes donnant ainsi des indications quant au coût minimal des plans intégrés.

Mots-clés

Production, lotissement, gestion de la chaîne d'approvisionnement, méthodes de décomposition, multi-niveaux, réapprovisionnement, programmation mixte en nombres entiers, intégration, tournées de véhicules, distribution

Méthodes de recherche

Recherche opérationnelle, programmation linéaire mixte en nombres entiers, méthodes de décomposition, heuristiques

Abstract

Over the last decades, the value of integrating operational decisions has become obvious to manufacturing companies. The main reason behind this integration, whether it is within a company or across a supply chain, can be easily explained by the potential benefits, such as cost reduction, increased flexibility or a higher customer service level. Several studies have built on this observation and successfully proposed mathematical models that take into account the integration of operational decisions, such as the production and distribution decisions. Despite these success stories, the integration of operational decisions at a broader level is still not fully exploited. The objective of this thesis is to use operations research techniques to optimize the integration of operational decisions within a supply chain with three levels and a distribution structure. The first, second and third level comprise a unique production plant, several warehouses and several retailers, respectively. The problem under study is therefore an integrated three-level lot sizing and replenishment problem (3LSPD). The main contributions lie in the mathematical models proposed along with the efficient algorithms developed to solve different versions of the problem, all of which integrate the production and replenishment decisions, while minimizing the operational costs across the supply chain. The thesis is split into four projects as follows.

We first compare 13 different mixed integer programming (MIP) formulations to solve a capacitated version and an uncapacitated version of the problem. In the capacitated version, we impose production capacity constraints at the plant level. The proposed formulations are adapted from existing MIP formulations found in the one-warehouse multi-

retailer literature, or are newly introduced in our context. In this work, the supply chain considered has a distribution structure where there are direct shipments between the different facilities : the products go from the production plant to the warehouses, and finally to the retailers. We further consider that each retailer is served by a unique warehouse. We finally consider one unique item. We perform numerical experiments to assess the computational performance of the proposed formulations. This computational performance is compared to the theoretical performance obtained through the comparison of the linear relaxations. The work on MIP formulations is used as a basis for the three other projects this thesis comprises.

In the second and third project, we apply decomposition methods to solve a deterministic capacitated and a stochastic uncapacitated version of the 3LSPD, respectively. Indeed, the results of the numerical experiments performed in the first project show the poor performance of the solver to solve capacitated instances. On the contrary, solving the uncapacitated version was very efficient, indicating the possibility to tackle a more complex variant of the problem, in particuler with demand uncertainty. To solve the deterministic capacitated version of the problem, we apply a Dantizg-Wolfe decomposition and develop a branch-and-price algorithm. The capacity requirements are imposed at the production plant level only, and limit the quantities produced in each time period. To solve the stochastic uncapacitated version of the problem, we apply a Benders decomposition. In that case, stochasticity comes from uncertainty in the demand at the retailer level and is modelled through demand scenarios with a known realization probability. We further consider a two-stage decision process, where the first stage decisions, which are the setup decisions, are made in the master problem and the second stage decisions, which are the production, replenishment and inventory level decisions, are made in the different subproblems. For this stochastic version, we also study an extension with lost sales. In such a case, the demand of the retailers may not be entirely satisfied, but this is penalized in the objective function. We finally develop a Benders-based branch-and-cut algorithm to efficiently solve the problem. For those two projects, we once again consider a distribution structure for the supply chain and a unique item. Several computational enhancements are

proposed to speed up both the branch-and-price and the branch-and-cut algorithms, and the proposed algorithms are tested on numerous instances.

Finally, in the fourth project, we add practical constraints and relax the assumption that our supply chain has a distribution structure. For this last project, we relax the main hypothesis of the 3LSPD and add operational constraints. The products still go from the plant to the warehouses, and finally to the retailers, but each retailer is not linked to a unique warehouse anymore. We consider that shipments between the plant and the warehouses, and between the warehouses and the retailers are performed by capacitated trucks. In this case, we also need to decide on the routes that the trucks available at the warehouses will follow, i.e., which retailers will be visited by each vehicle. We further consider a multi-item setting. We develop two heuristics to solve this version of the problem. The two heuristics decompose the whole problem in different subproblems that are iteratively solved. The first heuristic uses a top-down approach where the production decisions lead the rest of the operational decisions, i.e., the inventory and distribution decisions. The second heuristic uses a bottom-up approach where the distribution decisions at the retailer level lead the rest of the operational decisions. We additionally explore the possibility to have demand or delivery splitting. Demand splitting means that the demand of a retailer in a specific time period can be shipped over several periods. Delivery splitting means that the demand of a retailer in a specific time period can be shipped using several trucks. In this project, we compare the performance of the heuristics to the results obtained by a branch-and-cut algorithm that we also develop.

Keywords

Production; lot sizing; supply chain management; decomposition methods; multi-level; replenishment; mixed integer programming formulations; integration; routing; distribution

Research methods

Operations research; mixed integer linear programming; decomposition methods; heuristics

Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations

2S-3LSPD two-stage three-level lot sizing and replenishment problem with a distribution structure

3LSPD three-level lot sizing and replenishment problem with a distribution structure

3LSRP three-level lot sizing and replenishment problem

B&C branch-and-cut

CLSP capacitated lot sizing problem

DSP dual subproblem

FTL full truckload

GA genetic algorithm

GHG greenhouse gas

IRP inventory routing problem

LBL lower bound lifting

LNS large neighbourhood search

LR Lagrangian relaxation

LSP lot sizing problem

LTL less than truckload

MIP mixed integer programming

MIR mixed integer rounding

OWMR one warehouse multi retailer problem

PRP production routing problem

PSP primal subproblem

RMP restricted master problem

SI-CLSP single-item capacitated lot sizing problem

SI-ULSP single-item uncapacitated lot sizing problem

ULSP uncapacitated lot sizing problem

VNS variable neighbourhood search

Bonne lecture papy.

Remerciements

Alors que l'été s'offre une petite pause sur l'île de Montréal, l'heure est venue de terminer un chapitre entamé il y a près de quatre ans et demi. J'ai eu la chance de croiser le chemin de très nombreuses personnes durant ces années, ces quelques lignes sont pour les remercier.

Les premiers remerciements sont bien entendus adressés à mes deux directeurs de recherche, les professeurs Jean-François Cordeau et Raf Jans. Ils m'ont guidé tout au long de mon parcours et même avant lors d'un premier stage de recherche avec eux. Je me considère extrêmement chanceux d'avoir pu travailler avec ce duo magique. Leurs conseils avisés ont été d'une profonde aide et m'ont aidé à me construire en tant que chercheur. Ils m'ont toujours laissé une grande liberté quant aux orientations à donner à ma recherche et m'ont apporté le support nécessaire pour présenter mes travaux à plusieurs reprises lors de conférences. Je leur suis extrêmement reconnaissant pour leur aide et les remercie grandement pour leur temps passé à me superviser. J'espère être en mesure désormais de raconter les bonnes histoires, sans aucune erreurs dans les tableaux de résultats.

Je voudrais également remercier le professeur Sanjay Dominik Jena pour ses commentaires avisés aux moments charnières de ma thèse. Ses questions ont été une belle opportunité pour moi de remettre en question les hypothèses de ce travail et de mieux motiver les travaux effectués. Tous mes remerciements également aux autres membres du jury de cette thèse pour leur précieux temps : un grand merci à Stéphane Dauzère-Pérès et à Yossiri Adulyasak.

L'encadrement sans faille de mes directeurs est une chose, mais au quotidien ce sont de nombreuses personnes au Centre Interuniversitaire sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT) et au Groupe d'Études et de Recherche en Analyse des Décisions (GERAD) qui m'ont accompagné et ont fait de ce voyage un très beau voyage. Des joueurs de tarot aux buveurs de café, un grand merci à Narges, Alexis, Florian, Gislaine, Hani, David, Luciano, Mohammad, Aldair, Alfredo, Rosemarie, Nurit, Quentin, Khalid, Mehdi, Karim, Pedro, Masoud, Samuël, Marve et Okan. Une petite mention spéciale à Serge qui n'a jamais dit non à "*est-ce que je peux te déranger pour une question CPLEX ?*".

Durant ce doctorat j'ai eu la chance d'être élu représentant des étudiants du CIRRELT. C'est un immense privilège d'être élu par ses pairs ! J'en profite pour remercier les professeurs Yan Cimon et Martin Trépanier qui ont toujours apporté un soutien sans faille aux étudiants. Ils sont une des raisons qui font du CIRRELT un centre où il fait bon étudier. Les autres raisons sont Lucie, Nathalie, Catherine, Michèle, Pierre et Guillaume ! Dans un même ordre d'idée, un grand merci aux anges gardiens du GERAD pour leur aide dans l'organisation des séminaires étudiants. En particulier, merci à Marie, Edoh, Pierre, Marilyne, Guy et Karine.

Si j'ai passé la plupart de mon temps au CIRRELT, je n'en oublie pas moins l'environnement offert par HEC Montréal. C'est une école où il fait bon étudier, surtout au département de gestion des opérations et de la logistique ! Un grand merci en particulier à Claire Poitras pour son encadrement durant mes expériences d'enseignement, et aux professeurs Julie Paquette, Jorge Mendoza, Marie-Ève Rancourt, Claudia Rebolledo et Olivier Lourdel pour les marques de confiance qu'ils ont pu m'accorder à certains moments. Mille mercis aussi à mos anges gardiens Line, Michèle, Suzanne et Louise.

Cette thèse n'aurait surement pas été la même sans le soutien financier du CIRRELT, du Gouvernement du Canada via son programme de bourses Vanier du Centre de Recherche en Sciences Naturelles et Génie, de la fondation J. A. DeSève, du GERAD et de HEC Montréal. Des remerciements particuliers à Marie-France Courtemanche-Bell pour

son aide inestimable dans l'élaboration des demandes de bourse.

Et maintenant, le meilleur pour la fin. Papa, Maman, Élise, Alex, merci de m'avoir laissé mettre 6000km entre nous pour m'épanouir pleinement. Enfin, un grand merci à ma douce moitié. Tu m'as toujours soutenu et ne m'a jamais arrêté dans mes explications, quitte à devenir une experte en génération de colonnes et décomposition de Benders, à ton insu.

Chapitre 1

Introduction

Dans les dernières décennies, de nombreux chercheurs ont porté leur attention sur les problèmes de planification de production. Dans la littérature, le problème de planification de production le plus simple est le problème de lotissement pour un item. Ce problème de lotissement de base consiste à déterminer, pour chaque période d'un horizon de temps fini et discret, la taille des lots à produire dans le but de satisfaire une demande dynamique, tout en minimisant les coûts de mise en route et de stockage. Ce problème de lotissement de base, connu dans la littérature sous le nom de *single-item uncapacitated lot sizing problem* (SI-ULSP), se retrouve dans de nombreuses problématiques relatives à la gestion de production, de stocks ou encore de distribution.

Dans le contexte de la planification pour la chaîne d'approvisionnement, on retrouve également de nombreuses extensions du problème de lotissement de base. Traditionnellement, les clients d'une entreprise, dont on cherche à satisfaire la demande pour un ou plusieurs produits, sont situés dans une zone géographique qui diffère de celle de l'usine de production, là où les produits sont réellements fabriqués et où les décisions de lotissement sont prises. Cela mène tout naturellement à un problème de réapprovisionnement où l'entreprise doit décider du moment opportun pour réapprovisionner ses clients, entrepôts et sites intermédiaires éventuels, dans le but de minimiser ses coûts opérationnels totaux (coûts de stockage, coûts de mise en route, coûts de commande et coûts de transport). Les

entreprises qui font face à cette situation prennent souvent leurs décisions en série : soit les décisions de réapprovisionnement sont dictées par les décisions de production, soit les décisions de production sont dictées par les décisions de réapprovisionnement. Malheureusement, cela débouche sur des solutions, c'est-à-dire, des plans de production et de réapprovisionnement, dont les coûts peuvent être relativement éloignés des coûts des solutions optimales d'un problème intégré de lotissement et de réapprovisionnement.

Pourtant, les bénéfices de l'intégration des décisions opérationnelles ne sont pas chose nouvelle, comme mentionné il y a plus de 25 ans déjà par Chandra et Fisher (1994). Ils prouvent qu'une coordination des plans de production et de distribution peut déboucher sur des réductions allant de 3 à 20% des coûts opérationnels. Dans un même ordre d'idée, les cas de Kellogg et Frito-Lay, respectivement reportés par Brown et collab. (2001) et Çetinkaya et collab. (2009), sont deux histoires à succès relatives à l'intégration des décisions opérationnelles. Ces deux entreprises intègrent leurs décisions de production et de distribution vers leurs clients, rapportant ainsi des économies de plusieurs millions de dollars. En ce qui concerne les bénéfices liés à l'intégration des décisions opérationnelles, on recense aussi des améliorations en termes de niveau de service (Gopal et Cypress, 1993), d'amélioration de la performance organisationnelle (Vickery et collab., 2003) ou encore un plus grand avantage compétitif (Li et collab., 2006; Flynn et collab., 2010).

Malgré tous ces bénéfices potentiels, l'intégration des décisions opérationnelles reste un défi pour les entreprises manufacturières, tant sur le papier que sur le terrain. D'un point de vue pratique, l'intégration des décisions opérationnelles d'une entreprise est un défi puisque cela implique la destruction de la culture de silo qui existe encore parfois, ainsi que rapporté par Normandin (2016). Cela sous-entend également une meilleure communication entre les différents acteurs de la chaîne d'approvisionnement, qui peuvent être réticents à l'idée de partager des informations avec des entreprises externes, même partenaires. Sur le papier, du point de vue de la recherche opérationnelle, l'intégration des décisions opérationnelles rend les problèmes intégrés de planification de production et de distribution plus difficiles à résoudre. On assiste alors au développement de méthodes

heuristiques pour résoudre ces problèmes, comme c'est le cas dans les travaux de Darvish et Coelho (2018), par exemple.

Dans les dernières décennies, deux principaux problèmes intégrant des décisions opérationnelles ont vu le jour, soit le problème de production et tournées de véhicules (*production routing problem*, PRP) et le problème de gestion des stocks et de tournées de véhicules (*inventory routing problem*, IRP). Dans le PRP, une usine (ou dépôt) produit des items et réapprovisionne ensuite plusieurs clients sur un horizon temporel fini et discret. Le réapprovisionnement auprès des différents clients est réalisé par des véhicules suivant des tournées définies à la suite de l'optimisation intégrée des décisions de production et de réapprovisionnement. Ce problème a été introduit dans la littérature par Chandra (1993). Dans le cadre de l'IRP, un entrepôt est responsable du réapprovisionnement de plusieurs clients, également via des tournées de véhicules. Ici, il n'y a pas de décisions à prendre quant à la production des biens dont on suppose la disponibilité au niveau de l'entrepôt. Dans l'IRP, l'intégration est au niveau des décisions de stockage et de distribution. Ce problème est apparu pour la première fois dans Bell et collab. (1983) dans le cas d'un problème de livraison de gaz industriels.

Dans le PRP comme dans l'IRP, seulement deux niveaux de la chaîne d'approvisionnement sont pris en compte. Dans le PRP l'usine et les clients sont pris en compte, alors que dans l'IRP ce sont l'entrepôt et les clients qui sont pris en compte. Or, avec la mondialisation grandissante des opérations et avec la complexification des structures des chaînes d'approvisionnement, il y a un besoin pour un cadre plus général qui incluerait plusieurs niveaux de la chaîne d'approvisionnement, et qui couvrirait ainsi l'ensemble des décisions opérationnelles des entreprises manufacturières.

L'objectif de cette these est, *in fine*, de développer des modèles mathématiques et des algorithmes de résolution pour atteindre l'intégration des décisions opérationnelles dans une chaîne d'approvisionnement à trois niveaux. Dans les modèles et algorithmes, l'objectif est la minimisation des coûts opérationnels (coûts de mise en route, coût de commande, coûts de stockage et coûts de transport). La chaîne d'approvisionnement consi-

dérée comporte une usine de production (niveau zéro), plusieurs entrepôts (niveau un), et plusieurs détaillants (niveau deux). Les détaillants ont chacun une demande dynamique pour un ou plusieurs produits pour chacune des périodes de l'horizon de temps discret considéré. Le fait de ne prendre en compte qu'une seule usine de production est cohérent avec les pratiques de l'industrie pour les entreprises qui cherchent à améliorer leur efficacité opérationnelle par le développement de politiques multi-sites, voir Martel et Klibi (2016). Ces politiques font en sorte que les usines de production d'une entreprise sont spécialisées dans la production d'une famille de produits, amenant ainsi ladite efficacité opérationnelle, mais aussi des économies d'échelle.

La chaîne d'approvisionnement considérée dans cette thèse a une structure semblable à une structure de distribution : les entrepôts sont tous reliés à l'unique usine de production et les détaillants sont reliés aux entrepôts. La figure 1.1 illustre le flux des produits dans un tel réseau de distribution. Dans les trois premiers travaux de cette thèse, on considèrera de plus que chaque détaillant est relié à un unique entrepôt. Dans ce cas, on a une structure de distribution telle que définie par Pochet et Wolsey (2006). La figure 1.2 représente le flux des produits dans une telle chaîne, quand chaque détaillant est relié à un unique entrepôt. La structure de distribution est utilisée comme hypothèse de base dans les travaux de cette thèse. Ainsi, nous appelons le problème étudié dans le cadre de cette thèse le problème de lotissement et réapprovisionnement à trois niveaux avec structure de distribution (3LSPD). L'objectif du 3LSPD est de déterminer, pour chaque période de temps, les flux des produits à transférer entre les différents sites tout en cherchant à minimiser les coûts opérationnels.

Dans cette thèse, nous voulons effectuer une étude approfondie du 3LSPD. Pour ce faire, nous avons dans un premier temps développé de nombreux modèles mathématiques pour modéliser le problème. Dans un second temps, nous avons développé différents algorithmes capables de résoudre exactement des versions capacitaires et stochastique non capacitaire du problème. Pour ces algorithmes, on recherche l'efficacité : les temps de résolution doivent être suffisemment petits pour être utilisables en pratique. Enfin, dans

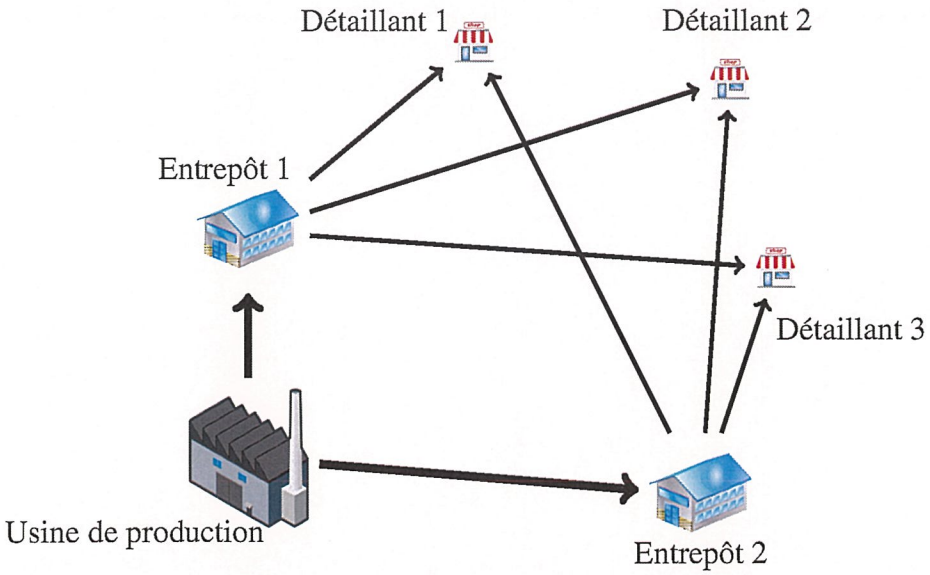

FIGURE 1.1 – Représentation graphique du problème à l'étude

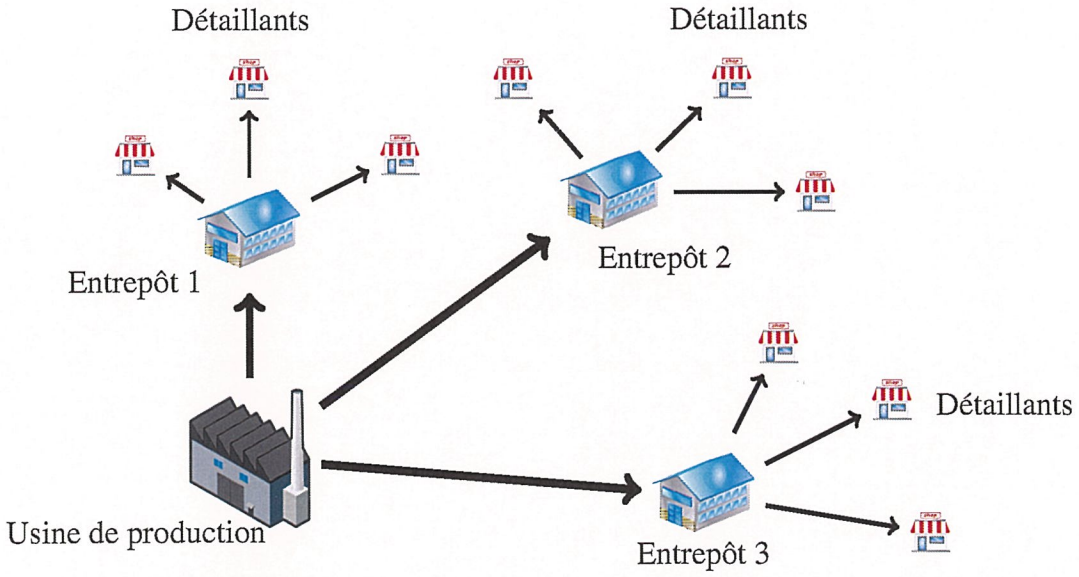

FIGURE 1.2 – Représentation graphique du problème à l'étude avec un entrepôt par détaillant

un troisième temps, nous proposons deux heuristiques pour résoudre une version du problème qui comprend plusieurs contraintes opérationnelles supplémentaires, telles des capacités de production et transport. Dans ce dernier projet, on considère également que les livraisons entre les entrepôts et les détaillants ne sont plus effectuées par du transport direct, mais plutôt via des tournées de véhicules. Nous relâchons également l'hypothèse d'une structure de distribution, permettant ainsi à chaque détaillant d'être réapprovisionné par n'importe quel entrepôt. Si on a pu trouver plusieurs cas industriels dans la littérature, il n'existe par contre pas de cadre général disponible pour l'étude du 3LSPD.

Le 3LSPD étudié est une extension du *one-warehouse multi-retailer problem* (OWMR). Dans l'OWMR, un entrepôt central réapprovisionne plusieurs détaillants ayant une demande dynamique pour un ou plusieurs produits à chaque période de l'horizon de temps considéré. L'objectif de l'OWMR est de déterminer les quantités qui seront livrées à chaque période aux différents détaillants dans le but de minimiser les coûts de mise en route et de stockage pour le système au complet. Arkin et collab. (1989) ont montré que ce problème est *NP*-difficile. Il apparaît également comme un sous problème dans le PRP, voir Adulyasak et collab. (2015). Contrairement au PRP et à l'IRP, il n'y a pas de tournées de véhicules dans l'OWMR. Le 3LSPD est une extension de l'OWMR dans la mesure où on garde une structure de distribution.

Dans le reste de ce chapitre, nous allons effectuer une revue de la littérature reliée à notre problème et donnerons plus de détails quant aux différents projets qui constituent cette thèse.

1.1 Revue de la littérature

Cette section passe en revue la littérature reliée aux travaux de recherche effectués dans cette thèse. En particulier, elle passe au travers des principaux cas industriels qui font état de bénéfices grâce à l'intégration des décisions opérationnelles, des études sur l'OWMR, des études qui ont une structure de chaîne d'approvisionnement similaire à la notre, et enfin, au travers des principales études sur les problèmes de lotissement multi-

niveaux.

1.1.1 Cas industriels

Grâce aux bénéfices de l'intégration identifiés plus haut, il est possible de trouver plusieurs histoires à succès quant à l'intégration des décisions opérationnelles au sein de la chaîne d'approvisionnement. Haq et collab. (1991) utilisent un solveur pour résoudre un cas industriel de fabrication d'urée. Ils proposent un modèle de programmation mixte en nombres entiers (PMNE) qui contient un délai de transport et autorise les commandes en souffrance. Ces possibilités de commandes en souffrance sont néanmoins mises de côté dans les expériences numériques. Dhaenens-Flipo et Finke (2001) étudient le cas d'une entreprise de métallurgie qui a des usines opérant tant en Europe qu'en Amérique du Nord. Dans cette étude, les coûts de transport et de production sont reliés, ce qui débouche naturellement sur un problème intégré où l'objectif est de minimiser les coûts opérationnels. Les auteurs développent un modèle mathématique résolu en faisant appel à un solveur. Des instances de taille réalistes sont résolues en un court temps de calcul. Leur modèle a par la suite été incoporé dans l'outil d'aide à la décision de l'entreprise. Kopanos et collab. (2012) s'attaquent au cas d'une entreprise grecque d'agro-alimentaire. Ils considèrent la disponibilité de différents modes de transport et associent un coût fixe par véhicule utilisé pour effectuer les livraisons entre les différentes sites de la chaîne d'approvisionnement. Les auteurs proposent un modèle de PMNE pour cette extension et le résolvent en faisant appel à un solveur. Ils proposent finalement d'étendre leur modèle au cas avec plusieurs usines de production. Dans un autre domaine, Bouchard et collab. (2017) proposent un modèle intégré pour planifier l'ensemble des décisions opérationnelles présentes dans la chaîne de valeur de l'industrie forestière (récolte, culture, maintenance, production et distribution). Les auteurs proposent de décomposer le problème en une partie stratégique et une partie tactique qui échangent des informations pour planifier les opérations sur un horizon de 150 ans. Les relations entre les parties stratégiques et tactiques sont formalisées dans un algorithme de génération de colonnes utilisé pour résoudre le problème intégré.

Les auteurs rapportent une augmentation de profit de 13% grâce à leur approche intégrée comparativement à une approche séquentielle. Zhang et Song (2018) étudient le cas de Danone Waters en Chine, avec une chaîne d'approvisionnement qui comporte plusieurs usines, centres de distribution et entrepôts plus locaux. Ils développent un outil d'aide à la décision fondé sur la programmation mathématique pour aider les gestionnaires dans leurs processus de production et distribution. Des économies minimales de 3.5% par an sont rapportées grâce à l'intégration de ces deux problèmes.

Les cas industriels ont également été attaqués via des heuristiques. Blumenfeld et collab. (1987) étudient le cas de General Motors. Au moment de l'étude, General Motors a un vaste réseau avec des milliers de fournisseurs, une centaine d'usines d'assemblage et des milliers de détaillants. Les auteurs proposent d'attaquer le problème en fixant la taille des lots livrés entre les usines "Delco", qui fabriquent des composants automobiles électroniques, et les usines d'assemblage de General Motors. Cette taille est obtenue comme étant la quantité économique à commander, voir Harris (1990). Avec cette approximation sur les flux entre les différents sites, les auteurs décomposent le problème global en plusieurs sous-problèmes faciles à résoudre. Les résultats obtenus par leur approche indiquent une réduction des coûts opérationnels de 26%. Özdamar et Yazgaç (1999) étudient le cas d'une entreprise de détergents en Turquie et proposent de planifier la production et la distribution sur une année. Ils imposent des contraintes de capacité de transport et proposent deux modèles de PMNE : un agrégé et un non agrégé. Dans le modèle agrégé l'horizon de temps considéré est l'année et chaque période de temps représente deux mois. De plus, les produits sont regroupés en famille. Dans le modèle désagrégé l'horizon de temps considéré est de deux mois et chaque période de temps représente une semaine. Dans ce modèle désagrégé, chaque produit est considéré individuellement. Dans leur modèle désagrégé, les auteurs considèrent un horizon roulant pour planifier l'année au complet par tranches de deux mois. Ils développent une heuristique fondée sur une approche hiérarchique itérative. Cette approche part du plan obtenu via le modèle agrégé pour ensuite définir un plan détaillé de production via le modèle désagrégé. Les auteurs indiquent que les plans de production et distribution obtenus par cette approche sont au plus 4% plus

coûteux que des plans entièrement optimisés et intégrés. Lejeune (2006) considère un 3LSPD avec des coûts de transport fixes et variables. La partie fixe vient de l'utilisation de chaque véhicule alors que la partie variable vient de la quantité transportée. L'auteur impose également des restrictions sur la capacité de transport et prend en compte la disponibilité des transporteurs. La méthode heuristique proposée combine les techniques de séparation et évaluation et de recherche à voisinage variable. Des expériences numériques sont réalisées à partir des données d'une entreprise américaine de produits chimiques. Les résultats indiqués montrent que la méthode heuristique dépasse les performances du solveur CPLEX. Dans un contexte de chaîne d'approvisionnement de l'industrie forestière, Sanei Bajgiran et collab. (2016) proposent un modèle de PMNE qui intègre des décisions de récolte, achat, production, distribution et vente. En raison de la taille des données, le modèle est toutefois inutilisable en pratique pour être résolu directement par un solveur. Une heuristique fondée sur la relaxation lagrangienne est alors proposée par les auteurs. Ils comparent ensuite leur approche intégrée à une approche séquentielle et rapportent des économies allant de 11 à 84%. Plus récemment, Abdullah et collab. (2019) étudient le cas d'une entreprise pétrochimique qui a une chaîne d'approvisionnement à quatre niveaux. Dans cette étude, l'intégration prend en compte les décisions de lotissement, ordonnancement, transport et entreposage, décisions qui apparaissent aux quatre niveaux de la chaîne d'approvisionnement. Les auteurs développent une méthode heuristique en trois étapes pour résoudre le problème. Leur méthode est en mesure de trouver des solutions de bonne qualité dans un court laps de temps.

1.1.2 L'OWMR

Le 3LSPD étudié dans cette thèse est une généralisation de l'OWMR à trois niveaux. Ces deux problèmes ont chacun une structure de distribution et on y retrouve des décisions similaires concernant la production, le stockage et le réapprovisionnement. Chacune de ces décisions doit être prise à chaque période de temps pour satisfaire la demande des détaillants. La principale différence entre notre problème et l'OWMR est le fait que

l'OWMR ne prend en compte que deux niveaux dans sa structure de distribution, soient les détaillants et l'entrepôt central.

La version de base de l'OWMR est un problème *NP*-difficile comme l'ont prouvé Arkin et collab. (1989). Par conséquent, de nombreuses méthodes heuristiques ont été développées pour résoudre la version basique de l'OWMR ainsi que plusieurs extensions. C'est le cas de Federgruen et Tzur (1999) qui proposent une heuristique fondée sur une partition de l'horizon temporel. Les périodes de temps initialement considérées sont découpées en sous-intervalles de plus courte durée. Des contraintes sont ajoutées aux bornes de chaque sous-intervalle pour lier lesdits sous-intervalles entre eux. Une relaxation lagrangienne est également utilisée pour obtenir des bornes inférieures sur la valeur de la fonction objectif. L'heuristique se décompose en quatre étapes : définition des intervalles, ajout des contraintes aux bornes des intervalles, résolution des problèmes sur chaque intervalle et construction d'une solution réalisable qui minimise les coûts variables tout en maintenant les décisions reliées aux coûts fixes. Ils considèrent des coûts de transport fixes et unitaires. Des expériences numériques sont réalisées pour analyser la performance de leur heuristique. Levi et collab. (2008) proposent quant à eux une méthode heuristique avec un rapport maximal de 1,8 entre la solution proposée par la méthode heuristique et la solution optimale. Les auteurs utilisent la relaxation linéaire de la formulation en transport de l'OWMR (voir la Section 2.3). Les solutions entières sont obtenues à partir d'arrondis de la solution fractionnaire courante. Enfin, Chen et Li (2011) développent une heuristique pour résoudre un OWMR de base. Ils travaillent séparément sur des problèmes de planification de production et de distribution. Ces deux problèmes échangent des informations de manière itérative jusqu'à ce qu'un critère d'arrêt prédéfini soit atteint.

Certains articles considèrent des structures de coût spécifiques, toujours dans le contexte de l'OWMR de base. C'est le cas de Yang et collab. (2012) qui développent un algorithme génétique. Dans cet article, les auteurs optent pour une structure de coût *all-units discount*. Dans cette structure de coûts, le prix unitaire au niveau des détaillants est réduit dès que la quantité commandée à l'entrepôt dépasse un certain seuil. L'algorithme génétique uti-

lise un croisement à un point et la mutation est réalisée via une inversion aléatoire. Les expériences numériques réalisées prouvent la bonne performance de leur algorithme génétique. Plus récemment, Gayon et collab. (2017) ont proposé un algorithme avec un ratio maximal de 2 entre la solution retournée par leur algorithme et la solution optimale. Leur méthode procède en deux phases distinctes. Dans un premier temps, l'OWMR est décomposé en plusieurs problèmes à un niveau. Dans un second temps, les solutions obtenues sur ces problèmes sont combinées pour construire une solution réalisable pour l'OWMR initial. Dans le cas où les coûts suivent une structure non spéculative, la complexité de leur algorithme est de $O(|R||T|)$, où R représente l'ensemble des détaillants et T représente l'ensemble des périodes de temps. La structure de coût dite non-spéculative signifie qu'il est bénéfique d'effectuer la production le plus tard possible étant donné un plan de mise en route. Ils étendent par la suite leur algorithme au cas où on ajoute une capacité de transport. Dans ce cas, les coûts suivent des structures dites de chargement complet (*full truck-load*, FTL) et de chargement partiel (*less than truck-load*, LTL). Ces structures incorporent toutes deux des coûts fixes sans égard à la quantité transportée. Dans le cas de la structure FTL, chaque camion utilisé entraîne le même coût fixe alors qu'avec la structure LTL, seuls les camions utilisés à pleine capacité entraînent un coût fixe. Un coût variable en fonction de la quantité transportée est ajouté pour les camions dont le chargement n'est pas égal à la capacité de transport.

Plusieurs articles incorporent des éléments additionnels à l'OWMR de base. C'est le cas de Chand et collab. (2007) qui ajoutent des possibilités de commandes en souffrance dans le cas de l'OWMR. Les commandes en souffrance sont calculées comme étant la somme de toutes les commandes non satisfaites à temps, pour une période donnée. Des coûts de transports unitaires sont considérés et, à partir de la structure des solutions au problème de lotissement sans capacité à un item, les auteurs développent un algorithme de programmation dynamique ayant une complexité de $O(|R||T|^2)$. Ici encore, R représente l'ensemble des détaillants et T représente l'ensemble des périodes de temps. Initialement, les auteurs empêchent les détaillants de conserver un inventaire d'une période à l'autre. Cette contrainte est par la suite enlevée. Pour autant, la complexité de l'algorithme de

programmation dynamique reste la même. Monthatipkul et Yenradee (2008) ajoutent une variable de décision pour la prise en compte de stocks de sécurité. Ils imposent des coûts de transport fixes et unitaires et considèrent un environnement stochastique. En raison de cet environnement incertain, une contrainte de niveau de service est ajoutée. Le niveau de service utilisé est le *fill rate*, qui impose le fait qu'une certainte proportion de la demande soit satisfaite à temps. Le modèle de PMNE qu'ils proposent est comparé à une politique *(R, s, S)*. Dans cette politique, un réapprovisionnement est effectué toutes les s périodes ou quand le niveau de stock atteint le point de commande R. La quantité commandée dans les deux cas est alors S. Les résultats des expériences numériques indiquent que le modèle de PMNE est plus adapté au problème. Toujours dans un contexte stochastique, Meng et collab. (2014) utilisent des règles de décision linéaires pour approximer les stratégies optimales de réapprovisionnement des détaillants. Les auteurs prennent en compte un scénario où chaque détaillant est indépendant des autres et un second scénario où tous les détaillants travaillent de concert pour obtenir une livraison gratuite. Cette gratuité est obtenue seulement si la commande totale est supérieure à un certain seuil prédéterminé. Les expériences numériques montrent que les détaillants joignent leurs commandes pour bénéficier de cette gratuité. Une analyse de sensibilité est ensuite menée sur le seuil de gratuité. Solyalı et collab. (2010) incluent une politique où le niveau d'inventaire doit atteindre un certain niveau cible dès qu'une commande est passée par un détaillant. Ils proposent une formulation dont la relaxation linéaire est proche du coût de la solution optimale. Des expériences numériques sont réalisées avec un ou plusieurs détaillants, et avec ou sans stock initial. Enfin, Li et Hai (2019) incorporent des coûts relatifs aux émissions de carbone induites par les activités de transport, réapprovisionnement et stockage. Les auteurs obtiennent une politique de réapprovisionnement en considérant les intervalles de réapprovisionnement comme étant les seules et uniques variables de décisions. Ils comparent ensuite leur solution respectueuse de l'environnement à une solution sans coûts reliés aux émissions de carbone.

1.1.3 Revue de la littérature sur le 3LSPD

Dans la littérature sur les problèmes de lotissement à trois niveaux, il est possible de rencontrer une grande variété de structures au niveau des chaînes d'approvisionnement. Cette variété est due tant à la diversité des sites qui constituent chacun des trois niveaux de la chaîne qu'à la diversité des décisions opérationnelles. Malgré tout, la littérature sur les problèmes de lotissement à trois niveaux ayant la même structure de distribution que la nôtre est peu fournie, et encore plus si on exclut les cas industriels (voir la section 1.1.1). Nous avons seulement été en mesure de trouver quelques articles qui s'attaquent tous à des extensions du 3LSPD considéré dans cette thèse. Gebennini et collab. (2009) proposent une heuristique pour résoudre le 3LSPD avec ajout de stocks de sécurité, ajout de dates butoires pour les livraisons aux détaillants, et ajout d'une possibilité d'avoir des commandes en souffrance. Le modèle intial proposé est non linéaire à cause de la structure des coûts associés au stock de sécurité. Pour remédier à cela, la fonction objectif est approximée par une fonction linéaire. Les auteurs développent finalement une procédure pour obtenir une solution réalisable au problème. Ben Mohamed et collab. (2020) s'attaquent aussi à un problème de lotissement à trois niveaux mais avec une structure générale pour leur chaîne d'approvisionnement. Ils ajoutent également une décision quant aux entrepôts qui doivent être ouverts (niveau un de la chaîne d'approvisionnement), et incluent un temps d'attente avant que les plateformes de distribution ne puissent réellement traiter les commandes des consommateurs finaux. Ils incluent finalement de l'incertitude autour de la demande des clients via un arbre de scénarios. Le problème obtenu est résolu par un processus en deux étapes qui repose sur une approximation du problème multi étapes auquel ils font face en réalité.

1.1.4 Lotissement multi-niveaux

Dans cette section nous passons en revue les travaux reliés aux problèmes de lotissement multi-niveaux. Dans les problèmes de lotissement multi-niveaux, la production d'un ou de plusieurs produits finaux découle de la production d'un ou de plusieurs compo-

sants présents dans la nomenclature des produits finaux. Dans Pochet et Wolsey (2006), on retrouve quatre structures de produit : assemblage, où chaque composant a un unique successeur ; en série, où chaque composant a un unique prédecesseur et successeur ; distribution, où chaque composant a un unique prédecesseur ; et générale. Ce problème a principalement été abordé par le développement de méthodes heuristiques en raison de sa difficulté. Dans le cadre de cette thèse, nous traitons un cas particulier du problème de lotissement multi-niveaux avec structure de distribution au sens défini par Pochet et Wolsey (2006). Dans le cas où chaque détaillant n'est plus relié à un unique entrepôt, nous traitons un cas particulier du problème général de lotissement multi-niveaux.

Maes et collab. (1991) s'attaquent à un problème de lotissement multi-niveaux avec contraintes de capacité. Les capacités sont sur les quantités disponibles pour chacun des composants de la nomenclature des produits finaux. Les auteurs proposent une heuristique fondée sur la relaxation linéaire du problème. Ils prennent comme point de départ la solution de la relaxation linéaire et utilisent des informations venant de la structure du problème pour arrondir les variables entières. Grâce à leur méthode, ils sont en mesure de résoudre à l'optimalité des petites instances du problème. Tempelmeier et Helber (1994) proposent une heuristique générale pour le même problème que Maes et collab. (1991). Leur heuristique procède à la résolution d'une série de problèmes de lotissement avec contrainte de capacité en utilisant une version modifiée de l'heuristique de Dixon-Silver (voir, Dixon et Silver, 1981). Ils proposent quatre versions de leur heuristique générale. Les différences émanent de l'ordre dans lequel les niveaux sont considérés. Plus tard, Sahling et collab. (2009) ont proposé une heuristique dite *fix and optimize* pour résoudre un problème similaire. La différence vient du fait que les mises en route peuvent être utilisées sur plusieurs périodes consécutives et n'ont pas à être recommencées s'il n'y a pas de changement de production de produit. L'idée principale de leur heuristique est de résoudre de manière séquentielle une série de petits problèmes de PMNE et d'utiliser les solutions obtenues pour fixer à une certaine valeur une grande proportion des variables entières. Des itérations ont lieu par la suite jusqu'à l'obtention d'une solution réalisable. Chen (2015) utilise également une heuristique dite *fix and optimize* pour résoudre un problème de lotis-

sement à plusieurs niveaux avec contraintes de capacité de production. L'auteur distingue une version où les mises en routes doivent être réalisées à chaque période de production, et une version où les mises en route peuvent être utiles pour plusieurs périodes. Dans le cas où les mises en route doivent être réalisées à chaque période de production, une recherche à voisinage variable est utilisée.

Plusieurs extensions du problème de lotissement multi-niveaux ont également été étudiées dans la littérature. Billington et collab. (1986) traitent le cas d'un problème de lotissement multi-niveaux avec un goulot d'étranglement à un certain niveau. Ils développent un algorithme de séparation et évaluation avec des heuristiques incorporées dans l'algorithme de séparation et évaluation. Les heuristiques, fondées sur des relaxations lagrangiennes, permettent d'obtenir les quantités à produire pour chaque item, à chaque noeud de l'arbre de recherche. Ces plans de production sont évalués selon des coûts artificiels qui découragent la production réelle au niveau du goulot d'étranglement pour les périodes de temps ayant une sur-utilisation de capacité. Plusieurs itérations sont ainsi réalisées pour obtenir un plan de production satisfaisant. Diaby et Martel (1993) incorporent des délais de production et des rabais en fonction des quantités produites. Pour résoudre leur problème, les auteurs utilisent aussi un algorithme de séparation et évaluation fondé sur la relaxation lagrangienne. Dans cette procédure, les contraintes liant les quantités reçues et les quantités produites sont relachées. Ils obtiennent ainsi deux sous-problèmes dont ils prouvent l'équivalence avec d'autres problèmes plus faciles à résoudre (problèmes de flot à coût minimal et problèmes de plus court chemin). Wu et collab. (2011) considèrent la possibilité d'avoir des commandes en souffrance. Ils proposent deux formulations dont la relaxation linéaire est proche de la valeur de la solution optimale. Ils proposent également un cadre général pour résoudre le problème, cadre appelé méthode LugNP. Cette méthode cherche un moyen efficace pour fixer à une certaine valeur un sous-ensemble des variables binaires. L'objectif est de trouver un sous-ensemble de variables qui va rapidement mener à l'obtention d'une solution admissible de grande qualité.

Plus récemment, Wei et collab. (2019) ont étudié un cas de lotissement multi-niveaux

où la nomenclature des produits peut être modifiée. Les auteurs effectuent également une revue de littérature sur des applications spécifiques des problèmes de lotissement multi-niveaux. Le lecteur intéressé est renvoyé aux références dans Wei et collab. (2019) pour plus de détails sur les problèmes de lotissement multi-niveaux.

1.2 Travail de recherche effectué dans le cadre de cette thèse

Dans cette section, les différents projets qui constituent cette thèse sont brièvement décrits. Le premier projet sert de fondation à l'ensemble de la thèse en adoptant une perspective de modélisation du problème. Pour ce premier projet, une version capacitaire et une version non-capacitaire du problème sont étudiées. Ces deux versions considèrent néamoins que la demande des détaillants est déterministe. Dans la version capacitaire, des contraintes de capacité de production sont imposées au niveau de l'usine. L'analyse des résultats numérique de ce premier projet justifie le développement d'une méthode de résolution spécifique pour la version capacitaire du problème. C'est ce que nous faisons dans le deuxième projet de cette thèse, avec l'utilisation d'une méthode de décomposition prenant appui sur une formulation proposée dans le premier projet. L'analyse des résultats numériques du premier projet ayant fait également ressortir de bonnes performances sur une version déterministe non-capacitaire, le troisième projet explore cette voie en ajoutant une dose d'incertitude entourant la demande des détaillants. Avec cette incertitude, il y a un besoin pour développer une méthode de résolution spécifique. Comme pour le deuxième projet, nous nous sommes tournés vers une méthode de décomposition qui utilise les travaux de modélisation effectués dans le premier projet comme point de départ. Pour ce troisième projet aussi, l'emphase est au niveau de la résolution. Enfin, le quatrième projet remet en question plusieurs hypothèses faites dans les trois premiers projets : les détaillants peuvent être servis par n'importe quel entrepôt, il y a des contraintes de capacité de production et transport, et il y a des routes à construire pour les livraisons

entre les entrepôts et les détaillants. Le quatrième projet aborde donc une version plus flexible du problème initial, avec ajout de nombreuses contraintes opérationnelles.

1.2.1 Premier projet : modélisation du problème

Dans le premier projet, nous avons développé 13 formulations de PMNE pour le 3LSPD. L'objectif de ce premier projet était de modéliser le problème sous différents points de vue. En effet, nous avons réalisé que les modèles trouvés dans la littérature ne prenaient pas forcément appui sur la même formulation initiale de PMNE (voir section 1.1). La présence de ces formulations différentes peut être expliquée par plusieurs facteurs. Parmi ceux-ci, on retrouve la nécessité pour les auteurs d'avoir des formulations qui soient cohérentes avec les besoins qui émanent des méthodes utilisées pour résoudre les cas industriels. Ainsi, certaines formulations mettent en exergue des sous-structures exploitables efficacement dans certains algorithmes de résolution. Avec un grand éventail de méthodes disponibles en recherche opérationnelle, nous avions la volonté de proposer de nombreuses formulations différentes qui peuvent s'adapter aux besoins de l'une ou l'autre de ces méthodes de résolution. Pour ce premier projet, nous considérons un produit et une assignation unique des détaillants aux entrepôts.

Nous avons donc développé 13 formulations différentes de PMNE. Ces formulations peuvent être regroupées en trois familles distinctes. La première famille regroupe les formulations classiques qui proviennent de la littérature sur les problèmes de lotissement. La deuxième famille de formulations prend appui sur le concept d'inventaire échelon, concept développé dans le contexte des problèmes de lotissement multi-niveaux. Enfin, la troisième famille de formulations est formée de formulations "riches", dans le sens où les variables de décision utilisées dans ces formulations contiennent plus d'informations. Pour ce premier projet, nous avons prouvé des relations d'ordre entre les relaxations linéaires de chaque formulation, pour la version sans capacité du 3LSPD. Nous avons également réalisé de nombreuses expériences numériques à partir de plusieurs instances que nous avons nous même générées. L'objectif des expériences numériques était d'identifier

en pratique les forces et faiblesses de chacune des formulations proposées. Ces forces et faiblesses obtenues en pratique ont été comparées aux forces et faiblesses prouvées de manière théorique.

Ce premier projet sert de fondation pour les trois autres projets de la thèse. Le travail sur les formulations est utilisé comme point de départ pour l'utilisation de méthodes de décomposition dans les deuxième et troisième projets. Ce projet a également été utilisé pour tester plusieurs formulations dans le quatrième projet.

1.2.2 Deuxième projet : décomposition de Dantzig-Wolfe appliquée à une version capacitaire du 3LSPD

Les expériences numériques réalisées dans le cadre du premier projet ont permis de mettre en exergue les grands temps de calcul requis pour résoudre les instances capacitaires du problème, c'est à dire quand il y a des contraintes de capacité de production imposées à l'usine. En particulier, il y a un faible nombre de solutions faisables trouvées, et un encore plus petit nombre de solutions optimales prouvées. Ces performances ne sont pas acceptables en l'état pour faire usage d'un solveur directement : il y a un besoin pour développer des méthodes efficaces de résolution qui viendront améliorer les temps de calcul et augmenter le nombre de solutions faisables trouvées, et le nombre de solutions optimales prouvées. Ainsi, dans le deuxième projet de la thèse nous nous sommes attaqués spécifiquement à cette version capacitaire du 3LSPD. Dans ce cas, la contrainte de capacité est imposée au niveau de l'usine de production : la production est limitée à chaque période. Ici encore, on considère un item et une assignation unique des détaillants aux entrepôts. La nécessité d'attaquer le problème capacitaire est venue des conclusions tirées du premier projet. En effet, dans le premier projet, nous avons remarqué que les temps de calcul utilisés pour résoudre les instances capacitaires étaient trop longs pour rendre nos modèles utilisables en pratique avec un solveur seul. Le travail effectué sur les différentes formulations a été utilisé comme point de départ pour identifier des sous-structures exploitables en utilisant des méthodes de décomposition. L'utilisation d'une décomposition

de Dantzig-Wolfe étant relativement rare dans la littérature sur les problèmes de lotissement, nous avons décidé d'utiliser cette méthode et de l'appliquer sur la formulation de stock échelon proposée dans le premier projet. Cela nous a mené à une reformulation de Dantzig-Wolfe. À partir de cette reformulation, nous avons développé un algorithme de séparation et génération de colonnes pour résoudre le problème. Nous avons également proposé plusieurs améliorations à l'algorithme de séparation et génération de colonnes pour accélérer le processus de résolution. Dans les améliorations proposées, on retrouve l'utilisation de la relaxation lagrangienne, la stabilisation des valeurs duales et l'ajout d'inégalités valides. Malgré ces améliorations, les résultats des expériences numériques n'ont pas été concluants. La performance de notre algorithme est en effet bien moindre que celle obtenue par l'utilisation du solveur CPLEX seul.

1.2.3 Troisième projet : décomposition de Benders pour une version stochastique du 3LSPD

Que ce soit au travers d'une chaîne d'approvisionnement ou au sein même d'une entreprise, la présence d'incertitude dans certains paramètres rend la planification des opérations complexe. La principale source d'incertitude apparaît dans la demande des détaillants et dans les délais de livraison. Cette incertitude n'a pas été prise en compte dans les premier et deuxième projets. Avec cette prise en compte d'incertitude, on se retrouve dans une situation totalement différente qui demande de repenser la modélisation effectuée dans le premier projet, et de développer des méthodes de résolution efficace pour cette variante stochastique du problème. Comme pour le premier projet, nous avons choisi ici de nous intéresser à une version non capacitaire du problème. L'objectif du troisième projet était donc d'intégrer l'incertitude entourant la demande des détaillants dans le 3LSPD, et de trouver des méthodes efficaces pour résoudre cette version stochastique du problème. La prise en compte d'incertitude entourant la demande s'est faite en utilisant des scénarios de demande, chaque scénario ayant la même probabilité de réalisation. On aurait pu prendre en compte l'incertitude entourant d'autres paramètres du

problème, par exemple sur les coûts opérationnels. Comme on considère une entreprise qui détient l'usine, les entrepôts et les détaillants, ladite usine a un meilleur contrôle sur les coûts opérationnels, comparativement à un contrôle sur la demande des clients. Cela nous a amené à considérer la demande comme étant la principale source d'incertitude, la demande émanant, *in fine*, du consommateur final.

L'introduction de ces scénarios de demande nous a orienté vers un processus de décision en deux étapes. Dans la première étape, on prend des décisions relatives aux variables de mise en route, soit le déclenchement de la production ou le déclenchement d'un réapprovisionnement de la part des entrepôts ou détaillants. Dans la seconde étape, après la réalisation de la demande, on prend les décisions relatives aux flux des produits à travers la chaîne d'approvisionnement, mais également les décisions relatives aux quantités gardées en stock. Ce processus en deux étapes peut se retrouver dans des situations où la production et la réalisation des opérations de réception de la marchandise sont effectuées par des opérateurs spécialisés en nombres limités, ou avec des disponibilités limitées. Ainsi, il est important de définir et fixer leur horaire en amont car l'entreprise est tributaire de ces personnes pour la bonne marche des opérations. En considérant que nos décisions de mise en route sont les mêmes pour tous les scénarios de demande, nous nous sommes naturellement tournés vers une décomposition de Benders pour résoudre le problème. Les variables de mise en route sont alors les variables liantes que l'on retrouve traditionnellement dans la décomposition de Benders. Nous avons également considéré une extension où on peut avoir des ventes perdues au lieu de satisfaire entièrement la demande des détaillants. Dans ce cas, les ventes perdues sont pénalisées dans la fonction objectif.

Nous avons par la suite développé un algorithme de séparation et coupes fondé sur cette décomposition de Benders. Nous avons également incorporé plusieurs améliorations à l'algorithme de séparation et coupes qui se sont avérées bénéfiques en pratique. Ces améliorations incluent des coupes de Pareto, des inégalités utilisées pour améliorer la borne inférieure dans le problème maître, et l'utilisation de coupes fractionnaires. Les sous-problèmes sont résolus par des algorithmes que nous avons nous-même développés.

Les expériences numériques menées sur des instances différentes ont mené à des performances bien meilleures que celles obtenues par le solveur CPLEX, notamment au niveau des temps de calcul et de la qualité des solutions obtenues.

1.2.4 Quatrième projet : méthodes heuristiques pour résoudre une extension 3LSPD avec capacités de production et transport, et tournées de véhicules

Le 3LSPD étudié dans les premier, deuxième et troisième projet est intéressant et complexe, mais les hypothèses des trois premiers projets sont assez fortes. En particulier, on a une affectation fixe des détaillants aux entrepôts, et on n'a pas de capacité de transport pour les livraisons entre l'usine et les entrepôts, et entre les entrepôts et les détaillants. Ainsi, dans le but de s'attaquer à une version plus complexe et réaliste du 3LSPD, nous avons décidé d'ajouter des contraintes de capacité au problème. Nous avons ajouté des contraintes de capacité de production au niveau de l'usine de production, mais aussi des contraintes de capacité de transport au niveau des livraisons effectuées entre l'usine de production et les entrepôts, et entre les entrepôts et les détaillants. Entre l'usine de production et les entrepôts, on considère qu'on a une flotte de camions homogènes ayant une certaine capacité. On considère également que le nombre de camions disponibles n'est pas limité. Cette hypothèse revient à considérer qu'un prestataire de logistique tierce partie peut effectuer les livraisons pour nous, avec un coût fixe par camion utilisé. À l'inverse, pour les livraisons entre les entrepôts et détaillants, on considère qu'on a un nombre prédéfini de camions disponibles pour effectuer les livraisons. Nous supposons que chaque entrepôt a le même nombre de camions disponibles à chaque période et que ces camions ont tous la même capacité de transport. Pour les livraisons entre les entrepôts et détaillants, nous avons également ajouté des décisions quant à la séquence des détaillants que chaque camion va visiter : cela revient donc à décider des routes de livraison de chaque camion. Par contre, entre l'usine de production et les entrepôts, les livraisons sont toujours directes. Enfin, nous ne considérons non plus un produit mais plusieurs produits, et ne

figeons plus les assignations des détaillants aux entrepôts : les entrepôts peuvent réapprovisionner n'importe quel détaillant. Nous perdons alors la structure de distribution au sens défini par Pochet et Wolsey (2006). Pour les livraisons entre les entrepôts et détaillants, nous explorons également la possibilité de scinder les demandes des détaillants ou les livraisons. Par le fait de scinder une livraison on entend le fait d'effectuer une livraison à l'aide de plusieurs camions, alors que par le fait de scinder une demande on entend le fait de livrer la demande d'une certaine période de manière étalée sur plusieurs périodes. Ces possibilités peuvent aussi être jumelées, donnant ainsi quatre situations potentielles. Pour ce dernier projet, nous avons donc des différences majeures avec les projets précédents en raison de la présence de capacités de transport, de tournées de véhicule entre les entrepôts et les détaillants, et d'une liberté d'affectation des détaillants aux entrepôts.

La prise en compte des capacités de transport et des décisions quant aux tournées de véhicules amène le problème à un autre niveau de difficulté. C'est pourquoi, nous avons développé deux méthodes heuristiques pour résoudre le problème en un temps raisonnable. La première heuristique part des décisions de production pour finalement décider du réapprovisionnement des détaillants alors que la seconde heuristique part des décisions de réapprovisionnement au niveau des détaillants pour finalement décider des quantités produites à l'usine. Les deux heuristiques décomposent ainsi le problème global en plusieurs sous-problèmes qui échangent des informations et sont résolus de manière itérative. Chaque heuristique comprend des phases d'intensification et de diversification. Les résultats obtenus par ces heuristiques sont comparés aux résultats obtenus par un algorithme de séparation et coupes que nous avons également développé pour résoudre le problème de manière exacte.

Chapitre 2

A comparison of formulations for a three-level lot sizing and replenishment problem with a distribution structure

Informations sur le chapitre

Un article fondé sur ce chapitre a été publié dans la revue *Computers & Operations Research* : Gruson, M., Bazrafshan, M., Cordeau, J.-F., and Jans, R. (2019). A comparison of formulations for a three-level lot sizing and replenishment problem with a distribution structure. *Computers & Operations Research*, 111 :297-310.

Abstract

We address a three-level lot sizing and replenishment problem with a distribution structure (3LSPD), which is an extension of the one-warehouse multi-retailer problem (OWMR). We consider one production plant that produces one type of item over a discrete and finite planning horizon. The items produced are used to replenish warehouses and then retailers using direct shipments. Each retailer is linked to a unique warehouse and there are no transfers between warehouses nor between retailers. We also assume that

transportation is uncapacitated. However, we consider the possibility of imposing production capacity constraints at the production plant level. The objective is to minimize the sum of the fixed production and replenishment costs and of the variable inventory holding costs at all three levels. We compare 13 different MIP formulations to solve the problem. All of these formulations are adapted from existing MIP formulations found in the one-warehouse multi-retailer literature, but most formulations are new in the context of the 3LSPD. We run experiments on both balanced and unbalanced networks. Our results indicate that the multi-commodity formulation is well suited for uncapacititated instances and that the echelon stock reformulations are better for capacitated instances. They also show that the richer formulations are not necessarily the best ones and that the unbalanced instances are harder to solve.

2.1 Introduction

Over the last decades, lot sizing problems have drawn the attention of many researchers, mainly because of their numerous applications in production, distribution and inventory management problems. Extensions of the basic lot sizing problem (LSP) are often encountered in the context of supply chain planning. Usually, the customers of a company, which have a certain demand, are located in a different area from the production plant where the items are actually produced and where lot sizing decisions are made. This leads to a replenishment problem where the company needs to determine when to replenish its customers so as to minimize the replenishment costs. Companies facing these two operational problems often make decisions in sequence. This leads, however, to solutions that can be far from the optimal solution of an integrated lot sizing and replenishment problem.

The integration of these two operational decisions has proven to be very effective for several industrial cases. Zhang and Song (2018) study the case of Danone Waters in China, where the supply chain comprises several factories, distribution centers and local warehouses. They develop a decision support system based on mathematical pro-

gramming to help managers in their production and distribution process. They report cost savings of at least 3.5% per year thanks to the integration of the two problems. In the same vein, Dhaenens-Flipo and Finke (2001) study the case of a metal manufacturer which has facilities operating both in Europe and North America and whose transportation and production costs are inter-related. This naturally leads to an integrated problem where the objective is to simultaneously minimize these operational costs. They develop a mathematical model which is solved by means of a general purpose solver. They are able to solve instances of practical size in a short amount of CPU time and their model has been further incorporated in the decision system used at the company. Thanks to this tool, the company was able to reduce its operational costs. In a recent paper, Abdullah et al. (2019) study the case of a petrochemical company having a four-level supply chain. Here, the integration combines lot-sizing, scheduling, transportation and warehousing decisions made at the different layers of the supply chain. The authors develop a three-stage heuristic to solve this integrated problem. Their heuristic is able to find solutions of good quality in a short amount of CPU time.

Following this line of research, we address here an integrated three-level lot sizing and replenishment problem with a distribution structure (3LSPD). We consider a general manufacturing company that has one production plant (level zero), several warehouses (level one) and multiple retailers (level two) facing a dynamic and known demand for one item over a discrete and finite time horizon. Considering only one production plant and one item is in line with industrial practice for companies developing multi-site policies to improve their operational efficiency. With such policies, each production site is focused on one particular item (see Dhaenens-Flipo and Finke (2001)). The supply chain considered has a distribution structure: the warehouses are all linked to the single plant and all retailers are linked to exactly one warehouse. Figure 2.1 illustrates the flow of goods in such a distribution network. The objective of the problem is to determine the optimal timing and flows of goods between the different facilities while minimizing the operational and replenishment costs in the whole network.

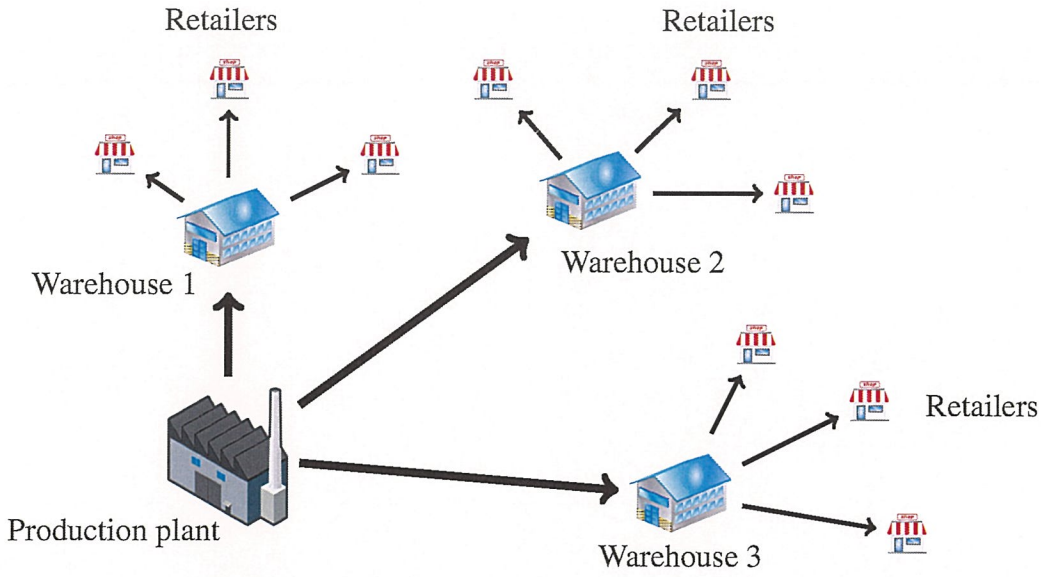

Figure 2.1 – Graphical representation of the problem considered

More specifically, given the set T of time periods, we face an integrated problem where decisions are made at all facilities for each time period. The optimal solution will indicate, for each time period, the optimal quantities to be produced and to be ordered from their predecessor for the production plant and for the warehouses and retailers, respectively, so that the final demand at each retailer is satisfied. The objective is to minimize the sum over all periods t of the fixed setup costs sc_t^p at the production plant, the fixed replenishment costs sc_t^w and sc_t^r of the warehouses and of the retailers, and the unit inventory holding costs hc_t^i of all facilities i. We do not include any unit production cost at the plant since the total production cost is a constant when all the demand is satisfied and when the unit production cost is constant over time. The same holds for the unit replenishment cost at the warehouses and retailers. Transfers of goods between the warehouses and between the retailers are not allowed. Finally, we only consider uncapacitated direct shipments and do not incorporate any routing in the transportation decisions. This is done in order to have a first step towards the study of the 3LSPD. Note that in a disaggregated context, the problem faced by any facility can be seen as the basic LSP. This basic LSP has attracted a lot of research since the seminal paper of Wagner and Whitin (1958) 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. (2017) and to Pochet and Wolsey (2006) for a review of the work done on the SI-ULSP and its extensions. We consider both a capacitated and an uncapacitated version of the 3LSPD. In the capacitated version, the capacity constraints are imposed at the production plant level. There are no capacities on the flows between the facilities nor on the inventory level. Note that with the addition of the capacity constraints at the production plant level, the problem faced by the production plant can be seen as a basic capacitated lot sizing problem (CLSP). The reader is referred to Karimi et al. (2003) for a review of models and algorithms used to solve the CLSP, and to Jans and Degraeve (2006) for a review of industrial applications.

The 3LSPD we study is an extension of the one-warehouse multi-retailer problem (OWMR). In the OWMR, a central warehouse replenishes several retailers that face a dynamic demand for one or several items over a discrete and finite time horizon. The objective of the problem is to jointly determine the optimal timing and quantities that are shipped between the warehouse and the retailers to minimize the sum of setup costs and inventory holding costs for the whole system. This problem has been shown to be *NP*-hard by Arkin et al. (1989) and appears as a substructure in the production routing problem (PRP, Adulyasak et al. (2015)).

The motivation to consider MIP formulations for the 3LSPD is twofold. First we want to work on modelling by developing or extending MIP formulations that are able to efficiently solve instances of practical size. A similar motivation can be found in the works of Solyalı and Süral (2012) and Cunha and Melo (2016) who compare several MIP formulations for the OWMR. Solyalı and Süral (2012) compare four MIP formulations and Cunha and Melo (2016) compare eight different MIP formulations for the OWMR. They provide results concerning the LP bounds of each formulation and numerical experiments are performed. Our second aim is to verify if the theoretical and computational results obtained on the two-level OWMR still stand for our three-level problem.

This chapter makes two main contributions. First, we fill a gap by adapting several

MIP formulations that have been proposed in the context of the two-level OWMR (Solyalı and Süral (2012), Cunha and Melo (2016)) to the 3LSPD. To the best of our knowledge, this is the first attempt to provide strong formulations for the 3LSPD that can solve instances of large scale. We also analyze the relationships between the linear relaxations of these formulations. Second, we report the results of extensive numerical experiments using a general-purpose solver to assess the strengths and weaknesses of the different formulations. Indeed, we perform experiments for different structures of the main parameters (fixed or dynamic demand, fixed or dynamic setup costs) and for two distribution structures of the supply chain network. In one case we consider a balanced distribution network in which each warehouse is responsible for the same number of retailers. In the other case, we consider an unbalanced distribution network where 20% of the warehouses replenish 80% of the retailers. The results obtained highlight the importance of properly choosing a formulation depending on the characteristics of the problem.

The remainder of this chapter is organized as follows. First, we survey the work related to our study in Section 2.2. We then present thirteen different MIP formulations for the problem in Section 2.3. These MIP formulations can be divided into three groups of formulations: the classical formulation, which uses the standard MIP formulation of the basic LSP, the echelon stock based formulations, inspired from the echelon stock concept for the multi-level LSP, and the richer formulations, containing more information in the decision variables, inspired from the work on the polyhedral structure of the solutions of both the SI-ULSP and the two-level lot sizing problems. Section 2.4 presents computational results to determine the strengths and weaknesses of the different formulations that we propose, and to analyze the impact of the different parameters. This is followed by the conclusion in Section 2.5.

2.2 Literature review

When reviewing the literature on the three-level LSP (3L-LSP), one can find several supply chain structures depending on the type of decisions made at each level and the con-

figuration of the links between the different levels. The following section only reviews the literature for which the supply chain structure is the same as in our problem: one production plant, several warehouses and several retailers. When not explicitely mentionned, the supply network structure considered in the papers reviewed in this section is a distribution structure as in our problem.

Only a few papers address a three-level lot sizing problem with a number of facilities per level which is the same as in our problem. The ones that we found all address extensions of the 3LSPD considered in this paper. Gebennini et al. (2009) propose a heuristic to solve a problem where they consider safety stocks and allow backorders. The basic model they propose is non-linear because of the safety stock cost but is linearized with an approximation of the objective function. There are also due dates for the deliveries to the customers. The authors design a procedure to solve the approximate problem.

Barbarasoglu and Özgür (1999) address the 3L-LSP where each retailer is linked to every warehouse. They thus do not have a distribution structure in their network but a general one instead. They also work in a just-in-time (JIT) environment which translates into a constraint that prevents retailers from keeping inventory. The model contains both fixed and unit transportation costs. The authors propose a transportation based MIP model and use Lagrangean relaxation to solve the problem. They relax the constraints linking the production and distribution components to obtain a production subproblem which can be decomposed into knapsack problems, and a distribution subproblem that can be easily solved for each item-customer pair. A customized procedure is then used to build feasible solutions from the solutions obtained in these two sub-problems.

Several extensions relate to applications for industrial cases. Kopanos et al. (2012) address an industrial case in Greece in the food industry. They have a fixed cost per vehicle used for the deliveries between the facilities and there are several transportation modes available. They consider restrictions on the vehicles that can make the deliveries between facilities. They extend their MIP model to consider several production plants and use a general-purpose solver to solve their instances. Haq et al. (1991) also use a general-

purpose solver to solve an industrial case of urea manufacturing. They propose a MIP model that contains transportation lead time and backlog but these features are discarded in the numerical experiments performed.

Heuristics have also been proposed to solve extensions of the 3LSPD applied to industrial cases. Lejeune (2006) proposes to solve a problem with a fixed cost per truck used and unit transportation costs. The author also considers transportation capacities and time availability of the carriers. A combination of branch-and-bound (B&B) and variable neighborhood search (VNS) is used to solve the problem. A computational experiment using data of a US chemical company indicates that this method outperforms CPLEX. In the same vein, Özdamar and Yazgaç (1999) treat the case study of a detergent company in Turkey. They design an algorithm to approximately solve the problem. The authors consider transportation capacities and propose an aggregate and a disaggregate MIP model. The algorithm is based on an iterative hierarchical approach as well as on a rolling horizon.

Note that in the works mentioned in this section, only three different types of MIP formulation have been used: Haq et al. (1991), Lejeune (2006), Gebennini et al. (2009) and Özdamar and Yazgaç (1999) use a classical formulation, Barbarasoglu and Özgür (1999) use a combined classical and transportation formulation, and Kopanos et al. (2012) use a transportation formulation. The classical formulation will be presented in Section 2.3.1 while the transportation formulation will be given in Section 2.3.4. The combined transportation and classical formulation is not presented in this paper because of its poor performances compared to other formulations. The interested reader is referred to Gruson et al. (2017).

2.3 Formulations

Let $G = (F, A)$ be a graph where F is the set of nodes (facilities in our problem) and A is the set of arcs. Let $P = \{p\} \subset F$ be the set containing the unique production plant, $W \subset F$ be the set of warehouses and $R \subset F$ be the set of retailers. Following the problem

description in Section 2.1, 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 $r \in R$. Let d_{rt} be the demand for retailer r in period t. The notion of the demand faced by any retailer is extended to the warehouses and to the production plant in the following fashion:

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

Using the notion of the demand faced by any facility, we introduce D_{it}, the total demand between period t and the end of the time horizon computed as $D_{it} = \sum_{k \geq t} d_{ik}$. Similarly, we introduce, for any facility i, the demand between periods k and t as $d_{ikt} = \sum_{k \leq l \leq t} d_{il}$. In the following sections, all the MIP formulations are presented in their capacitated version, with C_t representing the available capacity in period t.

2.3.1 Classical formulations

We first present a simple MIP formulation that extends the basic MIP formulation for the ULSP as used by Pochet and Wolsey (2006). We call this formulation the classical formulation (C). This formulation is based on three sets of decisions variables: x_{it} represents the production quantity in period t if $i = p$ and the quantity ordered from the predecessor if $i \in W \cup R$, s_{it} is the inventory held at the end of period t in facility i, and y_{it} is a boolean setup variable taking value 1 iff $x_{it} > 0$. The formulation is as follows:

$$\text{Min} \sum_{t \in T} \left(\sum_{i \in F} sc_{it} y_{it} + \sum_{i \in F} hc_{it} s_{it} \right) \tag{2.1}$$

$$\text{s.t. } x_{it} \leq D_{it} y_{it} \qquad \forall\, t \in T, i \in F \tag{2.2}$$

$$s_{i,t-1} + x_{it} = \sum_{j \in \delta(i)} x_{it} + s_{it} \qquad \forall\, t \in T, i \in P \cup W \tag{2.3}$$

$$s_{r,t-1} + x_{rt} = d_{rt} + s_{rt} \qquad \forall\, t \in T, r \in R \tag{2.4}$$

$$x_{pt} \leq \min\{C_t, D_{pt}\} y_{pt} \qquad \forall t \in T \tag{2.5}$$

$$x_{it}, s_{it} \geq 0 \qquad \forall\, t \in T, i \in F \tag{2.6}$$

$$y_{it} \in \{0,1\} \qquad \forall\, t \in T, i \in F. \tag{2.7}$$

The objective function minimizes the sum of the fixed setup and replenishment costs and of the unit inventory holding costs. Constraints (2.2) are the setup forcing constraints for all facilities. Constraints (2.3) are the inventory balance equations for the production plant and the warehouses whereas (2.4) are the inventory balance equations for the retailers. Constraints (2.5) are the capacity constraints at the production plant.

The classical formulation C has a rather poor linear relaxation which can be improved by using some ideas coming from the ULSP literature. We observe that when we only consider the inventory balance equations (2.4) and the setup constraints (2.2) specifically for the retailers, we have a single item lot sizing structure for each retailer since the inventory balance equations (2.4) only incorporate the independent demand for each retailer. This means that we can use the existing strong reformulations of the ULSP for each of the retailers. These reformulations are not directly applicable at the warehouse or plant level, since at these levels the inventory balance constraints contain dependent demand in the form of decision variables related to the ordering decisions at the direct successors. Despite these improvements, the results we obtained for the classical formulation using a strong formulation at the retailer level are still poor compared to the results obtained with other formulations presented hereafter and therefore, we do not present them. The interested reader is referred to Gruson et al. (2017). Note, however, that these reformulations can be applied at all levels using the echelon stock formulation (see Section 2.3.2).

2.3.2 Echelon stock formulations

Employing the idea of an echelon stock presented in Federgruen and Tzur (1999), the 3LSPD can be decomposed into several independent SI-ULSPs. To do so, the inventory variables of the classical formulation C are replaced with echelon stock variables representing the total inventory at all descendents of a particular facility. This idea has proven successful in the context of the OWMR to derive strong lower bounds within a Lagrangian relaxation scheme (see Federgruen and Tzur (1999)). We define the echelon stock I_t^i for

facility i in period t as:

$$I_{it} = \begin{cases} s_{it} + \sum_{w \in W} s_{wt} + \sum_{r \in R} s_{rt} & \text{if } i = p \\ s_{it} + \sum_{r \in \delta(i)} s_{rt} & \text{if } i \in W \\ s_{it} & \text{if } i \in R. \end{cases}$$

The echelon stock formulation (ES) is then as follows:

$$\text{Min} \sum_{t \in T} \left(\sum_{i \in F} sc_{it} y_{it} + \sum_{p \in P} hc_{pt} I_{pt} + \sum_{w \in W} \left(hc_{wt} - hc_{pt} \right) I_{wt} + \sum_{r \in R} \left(hc_{rt} - hc_{\delta_w(r)t} \right) I_{rt} \right) \tag{2.8}$$

$$\text{s.t. (2.2), (2.5), (2.7)}$$

$$I_{it} + x_{it} = d_{it} + I_{it} \qquad \forall\, t \in T, i \in F \tag{2.9}$$

$$I_{it} \geq \sum_{j \in \delta(i)} I_{jt} \qquad \forall\, t \in T, i \in P \cup W \tag{2.10}$$

$$x_{it}, I_{it} \geq 0 \qquad \forall\, t \in T, i \in F. \tag{2.11}$$

The objective function (2.8) is written in terms of echelon stock variables. Constraints (2.9) are the inventory balance constraints using the new echelon stock variables. Constraints (2.10) are the echelon stock constraints ensuring that the echelon stock at a specific facility is greater than the sum of the echelon stocks at all its direct successors. These constraints come from the non-negativity constraints (2.6) imposed on the stock variables in the classical formulation C. Note that with the introduction of the echelon stock variables, the problem has an uncapacitated lot sizing structure (in constraints (2.2) and (2.9)) with independent demand at each level. This means that we can apply the known reformulation techniques for the ULSP at each level.

First, we use the network reformulation proposed by Eppen and Martin (1987) for the SI-CLSP. This reformulation is based on the property of extreme flows in a network as applied by Zangwill (1969) to the SI-ULSP. This property, also known as the zero inventory ordering property, states that if there is a positive entering stock at any period in the SI-ULSP, then the flow coming from production is equal to zero. Conversely, if the production is positive at any period, then the entering stock for this period is equal to zero.

Although this property does not hold for the capacitated case, Eppen and Martin (1987) show that their proposed reformulation is valid for the capacitated case. Both for the SI-ULSP and SI-CLSP this reformulation drastically improves the linear programming relaxation. In particular, this leads to the integrality property in the case of the SI-ULSP. We define Z_{ikt} to be positive variables representing the proportion of d_{ikt} that is produced in period k for $i = p$, and to be the proportion of d_{ikt} that is ordered in period k for $i \in W \cup R$. The echelon stock network formulation (ES-N) is obtained by substituting constraints (2.2), (2.5) and (2.9) by constraints (2.12), (2.13) and (2.14)-(2.16), respectively:

$$\sum_{k=t:d_{itk}>0}^{|T|} Z_{itk} \leq y_{it} \qquad \forall\, t \in T, i \in F \tag{2.12}$$

$$\sum_{k=t}^{|T|} Z_{ptk} d_{ptk} \leq \min\{C_t, D_{pt}\} y_{pt} \qquad \forall\, t \in T \tag{2.13}$$

$$\sum_{k=1}^{|T|} Z_{i1k} = 1 \qquad \forall\, i \in F \tag{2.14}$$

$$\sum_{l=1}^{t-1} Z_{i,l,t-1} = \sum_{l=t}^{|T|} Z_{itl} \qquad \forall\, t \geq 2, i \in F \tag{2.15}$$

$$I_{it} = \left(\sum_{l=1}^{t} \sum_{k=l}^{|T|} d_{ilk} Z_{ilk} \right) - d_{i1t} \qquad \forall\, t \in T, i \in F. \tag{2.16}$$

Constraints (2.14) are the initial flow constraints for each facility and constraints (2.15) are the flow conservation constraints. Constraints (2.16) link the flow variables and the echelon stock variables.

Second, one can use the transportation reformulation of the ULSP proposed by Krarup and Bilde (1977) to obtain another formulation for the problem. This reformulation, when applied to the SI-ULSP, also has the integrality property. We define X_{ikt} to be the quantity that is produced in period k and used to satisfy d_{it} for $i = p$, and to be the quantity that is ordered in period k for $i \in W \cup R$ and used to satisfy d_{it}. The echelon stock transportation formulation (ES-TP) is obtained by substituting constraints (2.2), (2.5) and (2.9) by constraints (2.17), (2.18) and (2.19)-(2.20), respectively:

$$X_{itk} \leq d_{ik} y_{it} \qquad \forall\, k \in T, t \leq k \in T, i \in F \tag{2.17}$$

$$\sum_{k=t}^{|T|} X_{ptk} \leq \min\{C_t, D_{pt}\} y_{pt} \qquad \forall\, t \in T \tag{2.18}$$

$$I_{i,t-1} + \sum_{k=t}^{|T|} X^i_{itk} = d_{it} + I_{it} \qquad \forall\, t \in T, i \in F \tag{2.19}$$

$$\sum_{k=1}^{t} X^i_{ikt} = d_{it} \qquad \forall\, t \in T, i \in F. \tag{2.20}$$

Constraints (2.19) are the inventory balance constraints. These are included in order to correctly calculate the inventory levels. Constraints (2.20) are the demand satisfaction constraints.

Finally, one can also use the polyhedral results for the SI-ULSP to improve the echelon stock formulation at each of the 3 levels. In particular, Barany et al. (1984) propose the (l,S) valid inequalities that describe the polyhedron of solutions of the SI-ULSP. Besides, if the SI-ULSP has Wagner-Whitin costs (i.e., $pc_t + hc_t \geq pc_{t+1}$, $\forall\, t \in T$, where pc_t is the unit production cost in period t), Pochet and Wolsey (2006) propose the (l,S,WW) valid inequalities. When adapted to our problem, these (l,S,WW) inequalities are defined as follows:

$$I_{i,k-1} \geq \sum_{j=k}^{l} d_{ij} \left(1 - \sum_{u=k}^{j} y_{iu}\right) \quad \forall\, l \in T, k \leq l \in T, i \in F. \tag{2.21}$$

These inequalities are added to ES to form the echelon stock-(l,S) formulation (ES-LS). These inequalities are always valid, even if the costs do not satisfy the Wagner-Whitin condition. However, in case the Wagner-Whitin cost condition holds, they are sufficient to describe the convex hull of the SI-ULSP.

Following the model proposed in Federgruen and Tzur (1999), another change can be made to the echelon stock formulation ES. Indeed, one can alternatively write the echelon stock constraints (2.10) using the production variables of the ES, ES-N or ES-TP formulation, respectively:

$$\sum_{k=1}^{t} x_{ik} \geq \sum_{j \in \delta(i)} \sum_{k=1}^{t} x_{jk} \qquad \forall\, t \in T, i \in P \cup W \tag{2.22}$$

$$\sum_{k=1}^{t} \sum_{l \geq k} d_{ikl} Z_{ikl} \geq \sum_{j \in \delta(i)} \sum_{k=1}^{t} \sum_{l \geq k} d_{jkl} Z_{jkl} \qquad \forall\, t \in T, i \in P \cup W \tag{2.23}$$

$$\sum_{k=1}^{t}\sum_{l\geq k}X_{ikl} \geq \sum_{j\in\delta(i)}\sum_{k=1}^{t}\sum_{l\geq k}X_{jkl} \qquad \forall\, t\in T, i\in P\cup W. \tag{2.24}$$

If we substitute (2.10) by (2.22), (2.23) and (2.24) in formulations ES or ES-LS, ES-N and ES-TP, respectively, we obtain the echelon stock Federgruen formulations ES-F or ES-F-LS, ES-F-N and ES-F-TP, respectively.

2.3.3 Network formulation

The following formulation uses the network reformulation as proposed by Eppen and Martin (1987) for the SI-ULSP to rewrite the variables and constraints of the problem. Such a reformulation has also been applied by Solyalı and Süral (2012) and Cunha and Melo (2016) for the OWMR. Both Solyalı and Süral (2012) and Cunha and Melo (2016) showed that this reformulation gives a strong linear relaxation for the OWMR compared to numerous other formulations. For any retailer r, let ψ_{rklst} be the proportion of d_{rst} that is produced by the production plant in period k, transported to the warehouse of retailer r in period l and to retailer r in period s. Let also nc_{rklst} be the cost linked to the variable ψ_{rklst}: $nc_{rklst} = \sum_{j=k}^{l-1} hc_{pj}d_{rst} + \sum_{j=l}^{s-1} hc_{\delta_w(r)j}d_{rst} + \sum_{j=s}^{t-1} hc_{rj}d_{r,j+1,t}$. The network formulation (N) is given as follows:

$$\text{Min} \sum_{t\in T}\left(\sum_{i\in F} sc_{it}y_{it} + \sum_{r\in R}\sum_{k=1}^{t}\sum_{l=k}^{t}\sum_{s=l}^{t} nc_{rklst}\psi_{rklst}\right) \tag{2.25}$$

$$\text{s.t.} \sum_{t=1}^{|T|}\psi_{r111t} = 1 \qquad \forall\, r\in R \tag{2.26}$$

$$\sum_{k=1}^{t-1}\sum_{l=k}^{t-1}\sum_{s=l}^{t-1}\psi_{i,k,l,s,t-1} = \sum_{k=1}^{t}\sum_{l=k}^{t}\sum_{s=t}^{|T|}\psi_{iklts} \qquad \forall\, t\geq 2, r\in R \tag{2.27}$$

$$\sum_{l=k}^{t}\sum_{s=l}^{t}\sum_{j=t:d_{rsj}>0}^{|T|}\psi_{rklsj} \leq y_{pk} \qquad \forall\, t\in T, k\leq t\in T, r\in R \tag{2.28}$$

$$\sum_{k=1}^{l}\sum_{s=l}^{t}\sum_{j=t:d_{rsj}>0}^{|T|}\psi_{rklsj} \leq y_{\delta_w(r)l} \qquad \forall\, t\in T, l\leq t\in T, r\in R \tag{2.29}$$

$$\sum_{k=1}^{s}\sum_{l=k}^{s}\sum_{j=t:d_{rsj}>0}^{|T|}\psi_{rklsj} \leq y_{rs} \qquad \forall\, t \in T, s \leq t \in T, r \in R \tag{2.30}$$

$$\sum_{i\in R}\sum_{l=k}^{|T|}\sum_{s=l}^{|T|}\sum_{t=s}^{|T|}\psi_{iklst}d_{ist} \leq \min\{C_k, D_{pk}\}y_{pk} \qquad \forall\, k \in T \tag{2.31}$$

$$\psi_{rklst} \geq 0 \qquad \forall\, k \leq l \leq s \leq t \in T, r \in R \tag{2.32}$$

$$y_{it} \in \{0,1\} \qquad \forall\, t \in T, i \in F. \tag{2.33}$$

Constraints (2.26) are the demand satisfaction constraints written as initial flow constraints. Constraints (2.27) are the flow conservation constraints. Constraints (2.28), (2.29) and (2.30) are the setup forcing constraints for the production plant, the warehouses and the retailers, respectively. Constraints (2.31) are the capacity constraints at the production plant.

2.3.4 Transportation formulation

In the following formulation, the interactions between the facilities are modeled based on the transportation formulation of Krarup and Bilde (1977) for the SI-ULSP. Levi et al. (2008) propose such a transportation formulation for the OWMR. For the OWMR, Solyalı and Süral (2012) have proven that its linear relaxation is weaker than the one of the network formulation but their results also indicate that the transportation formulation has a better performance for solving the MIP compared to the N formulation. For any retailer r, let θ_{rklst} be the quantity that is produced by the production plant in period k, transported to the warehouse of retailer r in period l, transported to retailer r in period s and used to satisfy d_{rt}. Let also H_{rklst} be the cost linked to θ_{rklst}: $H_{rklst} = \sum_{j=k}^{l-1} hc_{pj} + \sum_{j=l}^{s-1} hc_{\delta_w(r)j} + \sum_{j=s}^{t-1} hc_{rj}$. The transportation formulation (TP) is given as follows:

$$\text{Min} \sum_{t\in T}\left(\sum_{i\in F} sc_{it}y_{it} + \sum_{r\in R}\sum_{k=1}^{t}\sum_{l=k}^{t}\sum_{s=l}^{t} H_{rklst}\theta_{rklst}\right) \tag{2.34}$$

$$\text{s.t.} \sum_{k=1}^{t}\sum_{l=k}^{t}\sum_{s=l}^{t}\theta_{rklst} = d_{rt}^{r} \qquad \forall\, t \in T, r \in R \tag{2.35}$$

$$\sum_{l=k}^{t}\sum_{s=l}^{t}\theta_{rklst} \leq d_{rt}y_{pk} \qquad \forall\, t \in T, k \leq t \in T, r \in R \tag{2.36}$$

$$\sum_{k=1}^{l}\sum_{s=l}^{t}\theta_{rklst} \leq d_{rt}y_{\delta_w(r)l} \qquad \forall\, t \in T, l \leq t \in T, r \in R \tag{2.37}$$

$$\sum_{k=1}^{s}\sum_{l=k}^{s}\theta_{rklst} \leq d_{rt}y_{rs} \qquad \forall\, t \in T, s \leq t \in T, r \in R \tag{2.38}$$

$$\sum_{i\in R}\sum_{l=k}^{|T|}\sum_{s=l}^{|T|}\sum_{t=s}^{|T|}\theta_{iklst} \leq \min\{C_k, D_{pk}\}y_{pk} \qquad \forall\, k \in T \tag{2.39}$$

$$\theta_{rklst} \geq 0 \qquad \forall\, k \leq l \leq s \leq t \in T, r \in R \tag{2.40}$$

$$y_{it} \in \{0,1\} \qquad \forall\, t \in T, i \in F. \tag{2.41}$$

Constraints (2.35) are the demand satisfaction constraints. Constraints (2.36), (2.37) and (2.38) are the setup forcing constraints for the production plant, the warehouses and the retailers, respectively. Constraints (2.39) are the capacity constraints at the production plant.

2.3.5 Multi-commodity formulation

The next formulation proposed is based on the distinction of each retailer-period pair (i.e., each d_{rt} is viewed as a distinct commodity). In the context of a two-level lot sizing problem in series, this formulation has proven to be very effective both in terms of the CPU time taken to solve instances and of the strength of the linear programming relaxation bound, see Melo and Wolsey (2010). Similar results have been observed by Cunha and Melo (2016) for the OWMR. In particular, this formulation has proven to be very efficient to solve large scale MIP instances for the OWMR despite a linear relaxation which is weaker than the one provided by the N formulation. For this formulation, for any retailer r, let X^0_{rkt} be the amount produced at the production plant in period k to satisfy d_{rt}, let X^1_{rkt} be the amount transported from the production plant to the warehouse of retailer r in period k to satisfy d_{rt} and let X^2_{rkt} be the amount transported from the warehouse of retailer r to retailer r in period k to satisfy d_{rt}. Let also σ^0_{rkt} be the amount stocked at the production plant at the end of period k to satisfy d_{rt}, let σ^1_{rkt} be the amount stocked at

the warehouse of retailer r at the end of period k to satisfy d_{rt} and let σ^2_{rkt} be the amount stocked at retailer r at the end of period k to satisfy d_{rt}. In the following formulation, we denote by δ_{kt} the Kronecker delta which takes the value 1 if $k = t$ and 0 otherwise. The multi-commodity formulation (MC) is as follows:

$$\text{Min} \sum_{t\in T}\left(\sum_{i\in F} sc_{it}y_{it} + \sum_{r\in R}\sum_{k\le t} hc_{pk}\sigma^0_{rkt} + \sum_{r\in R}\sum_{k\le t} hc_{\delta_w(r)k}\sigma^1_{rkt} + \sum_{r\in R}\sum_{k\le t} hc_{rk}\sigma^2_{rkt}\right) \tag{2.42}$$

$$\text{s.t. } \sigma^0_{r,k-1,t} + X^0_{rkt} = X^1_{rkt} + \sigma^0_{rkt} \qquad \forall\, t\in T, k\le t\in T, r\in R \tag{2.43}$$

$$\sigma^1_{r,k-1,t} + X^1_{rkt} = X^2_{rkt} + \sigma^1_{rkt} \qquad \forall\, t\in T, k\le t\in T, r\in R \tag{2.44}$$

$$\sigma^2_{r,k-1,t} + X^2_{rkt} = \delta_{kt}d_{rt} + (1-\delta_{kt})\sigma^2_{rkt} \qquad \forall\, t\in T, k\le t\in T, r\in R \tag{2.45}$$

$$X^0_{rkt} \le d_{rt}y_{pk} \qquad \forall\, t\in T, k\le t\in T, r\in R \tag{2.46}$$

$$X^1_{rkt} \le d_{rt}y_{\delta_w(r)k} \qquad \forall\, t\in T, k\le t\in T, r\in R \tag{2.47}$$

$$X^2_{rkt} \le d_{rt}y_{rk} \qquad \forall\, t\in T, k\le t\in T, r\in R \tag{2.48}$$

$$\sum_{r\in R}\sum_{t=k}^{|T|} X^0_{rkt} \le \min\{C_k, D_{pk}\}y_{pk} \qquad \forall k\in T \tag{2.49}$$

$$X^0_{rkt}, X^1_{rkt}, X^2_{rkt} \ge 0 \qquad \forall\, t\in T, k\le t\in T, r\in R \tag{2.50}$$

$$\sigma^0_{rkt}, \sigma^1_{rkt}, \sigma^2_{rkt} \ge 0 \qquad \forall\, t\in T, k\le t\in T, r\in R \tag{2.51}$$

$$y_{it} \in \{0;1\} \qquad \forall t\in T, i\in F. \tag{2.52}$$

Constraints (2.43), (2.44) and (2.45) are the balance constraints for each commodity at the production plant, at the warehouses and at the retailers, respectively. Constraints (2.46), (2.47) and (2.48) are the setup forcing constraints for the production plant, the warehouses and the retailers, respectively. Constraints (2.49) are the capacity constraints at the production plant.

The last formulation combines the idea of an echelon stock presented in Federgruen and Tzur (1999) and the MC formulation. To the best of our knowledge, it is the first time that such a formulation is proposed. We call it the multi-commodity echelon formulation (MCE). To get this formulation, the inventory variables of the MC formulation are

replaced with multi-commodity echelon variables E^l_{rkt} representing the amount stocked at the end of period k at all predecessors of retailer r which are in level l or more, and which will be used to fulfill the specific demand d_{rt}. We define the multi-commodity echelon variables E^l_{rkt} as:

$$E^l_{rkt} = \begin{cases} \sigma^0_{rkt} + \sigma^1_{rkt} + \sigma^2_{rkt} & \text{if } l = 0 \\ \sigma^1_{rkt} + \sigma^2_{rkt} & \text{if } l = 1 \\ \sigma^2_{rkt} & \text{if } l = 2. \end{cases}$$

The multi-commodity echelon formulation (MCE) is then as follows:

$$\text{Min} \sum_{t\in T}\left(\sum_{i\in F} sc_{it}y_{it} + \sum_{r\in R}\sum_{k\le t} hc^p_{pk}E^0_{rkt} + \sum_{r\in R}\sum_{k\le t}\left(hc_{\delta_w(r)k} - hc_{pk}\right)E^1_{rkt}\right) + \sum_{t\in T}\left(\sum_{r\in R}\sum_{k\le t}\left(hc_{rk} - hc_{\delta_w(r)k}\right)E^2_{rkt}\right) \quad (2.53)$$

$$\text{s.t. } (2.46) - (2.52)$$

$$E^0_{r,k-1,t} + w^0_{rkt} = \delta_{kt}d_{rt} + (1-\delta_{kt})E^0_{rkt} \qquad \forall\, t\in T, k\le t\in T, r\in R \quad (2.54)$$

$$E^1_{r,k-1,t} + w^1_{rkt} = \delta_{kt}d_{rt} + (1-\delta_{kt})E^1_{rkt} \qquad \forall\, t\in T, k\le t\in T, r\in R \quad (2.55)$$

$$E^2_{r,k-1,t} + w^2_{rkt} = \delta_{kt}d_{rt} + (1-\delta_{kt})E^2_{rkt} \qquad \forall\, t\in T, k\le t\in T, r\in R \quad (2.56)$$

$$E^0_{rkt} \ge E^1_{rkt} \qquad \forall\, t\in T, k\le t\in T, r\in R \quad (2.57)$$

$$E^1_{rkt} \ge E^2_{rkt} \qquad \forall\, t\in T, k\le t\in T, r\in R \quad (2.58)$$

$$E^0_{rkt}, E^1_{rkt}, E^2_{rkt} \ge 0 \qquad \forall\, t\in T, k\le t\in T, r\in R. \quad (2.59)$$

Constraints (2.54), (2.55) and (2.56) are the balance constraints for each commodity at the production plant, at the warehouses and at the retailers respectively. Constraints (2.57) and (2.58) are the echelon constraints ensuring that the multi-echelon stock at a specific facility for a specific commodity is greater than or equal to the sum of the multi-echelon stocks at all its direct successors for the same commodity.

2.3.6 Summary

The formulations previously introduced are extensions of the MIP formulations proposed for the OWMR. For all the formulations presented, the adaptation of the original

decision variables naturally leads to an increase in their number. For the N and TP formulations, this increase translates into an additionnal dimension with the new subscript k in the decision variables ψ_{rklst} and θ_{rklst} to reflect the third level. For all the other formulations, the increase in the number of decision variables is just the result of the increase in the number of facilities due to the added third level. Thus, the increase in the number of decision variables for the N and TP formulations is much higher than for the other formulations when going from a two-level LSP to a three-level LSP.

Table 2.1 gives a summary of the number of variables (binary and in total) and constraints for each formulation, and the paper from which the formulation has been adapted to our problem. Recall that these papers present a one-level or two-level problem whereas we consider a three-level problem. Note that, to the best of our knowledge, the ES-N, ES-F-N, ES-F-TP, ES-F-LS and MCE formulations we propose are completely new. In Table 2.1, one can see that the richer formulations, i.e., the ones that have more information in the decision variables, are the largest ones.

Table 2.1 – Summary of the sizes of all formulations for the 3LSPD

Formulation	Variables		Constraints	Reference
	Binary	Total		
C	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|)$	Pochet and Wolsey (2006)
ES	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|)$	Pochet and Wolsey (2006)
ES-N	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|^2)$	$O(\|F\| \times \|T\|)$	
ES-TP	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|^2)$	$O(\|F\| \times \|T\|^2)$	Solyalı and Süral (2012)
ES-LS	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|^2)$	Barany et al. (1984)
ES-F	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|)$	Federgruen and Tzur (1999)
ES-F-N	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|^2)$	$O(\|F\| \times \|T\|)$	
ES-F-TP	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|^2)$	$O(\|F\| \times \|T\|^2)$	
ES-F-LS	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|)$	$O(\|F\| \times \|T\|^2)$	
N	$O(\|F\| \times \|T\|)$	$O(\|R\| \times \|T\|^4)$	$O(\|R\| \times \|T\|^2)$	Solyalı and Süral (2012)
TP	$O(\|F\| \times \|T\|)$	$O(\|R\| \times \|T\|^4)$	$O(\|R\| \times \|T\|^2)$	Levi et al. (2008)
MC	$O(\|F\| \times \|T\|)$	$O(\|R\| \times \|T\|^2)$	$O(\|R\| \times \|T\|^2)$	Melo and Wolsey (2010)
MCE	$O(\|F\| \times \|T\|)$	$O(\|R\| \times \|T\|^2)$	$O(\|R\| \times \|T\|^2)$	

2.3.7 Analysis of the LP relaxation of formulations

We analyse the strength of the MIP formulations in terms of the objective function value of their LP relaxation, without considering the production capacity constraint (2.5). We denote by z_{LP}^{X} the objective function value of the LP relaxation of formulation X. The following example is used to illustrate most of the strict dominance relations between the formulations. The strict dominance relation between formulations MC and N cannot be observed empirically on small instances such as the one presented hereafter. However, we have observed it for large instances in our computational study, for example with $|R| = 200$ and $|T| = 30$.

Example 1. Consider an instance of the 3LSPD with $T = 4$, $|W| = 2$ and $|R| = 4$. Each warehouse is responsible for two retailers. The first warehouse is responsible for the first two retailers and the second warehouse is responsible for the other two. We have, for any $t \in T$, $hc_{pt} = 30$, $hc_{w_1t} = 50$, $hc_{w_2t} = 60$, $hc_{r_1t} = 10$, $hc_{r_2t} = 20$, $hc_{r_3t} = 100$, $hc_{r_4t} = 10$, $sc_{pt} = 100$, $sc_{w_1t} = 500$, $sc_{w_2t} = 600$, $sc_{r_1t} = 100$, $sc_{r_2t} = 200$, $sc_{r_3t} = 300$, $sc_{r_4t} = 50$ and $d_{r_1} = (10, 20, 15, 10)$, $d_{r_2} = (5, 30, 10, 10)$, $d_{r_3} = (45, 20, 20, 10)$, $d_{r_4} = (10, 20, 15, 20)$. For this instance, the optimal LP solutions values for five of the formulations are $z_{LP}^{C} = 3903.56$, $z_{LP}^{ES-LS} = 6017.25$, $z_{LP}^{ES-N} = 6096.343$, $z_{LP}^{MC} = 6750.00$ and $z_{LP}^{N} = 6750.00$.

Proposition 1 establishes dominance or equality relations between the linear programming relaxation of the different formulations presented in this section. These theoretical results show the superiority of the richer formulations (MC, MCE, TP and N) with respect to the strength of the linear relaxation but are to be compared with the computational results on the MIP problems obtained during the numerical experiments in Section 2.4.

Proposition 1.

$$\begin{aligned} z_{LP}^{C} = z_{LP}^{ES} = z_{LP}^{ES-F} \leq z_{LP}^{ES-LS} = z_{LP}^{ES-F-LS} \leq z_{LP}^{ES-N} = z_{LP}^{ES-TP} \\ = z_{LP}^{ES-F-N} = z_{LP}^{ES-F-TP} \leq z_{LP}^{TP} = z_{LP}^{MC} = z_{LP}^{MCE} \leq z_{LP}^{N} \end{aligned} \quad (2.60)$$

Proof. The reader is referred to Gruson et al. (2017) for detailed proofs. □

2.4 Numerical experiments

In order to assess the strengths and weaknesses of the different formulations, we conducted computational experiments based on the instances used in Solyalı and Süral (2012). As we have one more level than in Solyalı and Süral (2012), we slightly adapted these instances. In our instances, the number of retailers $|R|$ is set equal to 50, 100 or 200. The number of warehouses $|W|$ is set equal to 5, 10, 15 or 20. We used two different horizon lengths: $|T| = 15$ and 30. The demand at the retailers is generated both in a static and dynamic way from U[5, 100]. In the case of a static demand, we have $d_{rt} = d_r \; \forall \, t \in T, r \in R$. 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. For each combination of settings, we generate five different instances leading to 480 different instances to be solved for each formulation.

In order to test our formulations, we additionally define two structures for the distribution network represented in Figure 4.1. In the first structure, we consider a balanced network where each warehouse has the same number of retailers, except when the number of retailers is not a multiple of the number of warehouses. In the second structure, we consider an unbalanced network where 80% of the retailers are assigned to 20% of the warehouses. For each pair $(|W|, |R|)$, Tables 2.2 and 2.3 give the number of retailers assigned to each warehouse for the balanced and unbalanced networks, respectively. Each structure is tested on the 480 instances we generated.

For the experiments, we used the CPLEX 12.6.1.0 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 is 6 hours.

Table 2.2 – Number of retailers assigned to the warehouses for the balanced network

Number of warehouses	Number of retailers		
	50	100	200
5	10 $\forall w \in W$	20 $\forall w \in W$	40 $\forall w \in W$
10	5 $\forall w \in W$	10 $\forall w \in W$	20 $\forall w \in W$
15	3 if $w \in [\![1,10]\!]$ 4 if $w \in [\![11,15]\!]$	6 if $w \in [\![1,5]\!]$ 7 if $w \in [\![6,15]\!]$	14 if $w \in [\![1,10]\!]$ 12 if $w \in [\![11,15]\!]$
20	3 if $w \in [\![1,10]\!]$ 2 if $w \in [\![11,20]\!]$	5 $\forall w \in W$	10 $\forall w \in W$

Table 2.3 – Number of retailers assigned to the warehouses for the unbalanced network

Number of warehouses	Number of retailers		
	50	100	200
5	40 if $w = 1$ 3 if $w \in [\![2,3]\!]$ 2 if $w \in [\![4,5]\!]$	80 if $w = 1$ 5 if $w \in [\![2,5]\!]$	160 if $w = 1$ 10 if $w \in [\![2,5]\!]$
10	17 if $w \in [\![1,2]\!]$ 2 if $w \in [\![3,10]\!]$	38 if $w \in [\![1,2]\!]$ 3 if $w \in [\![3,10]\!]$	80 if $w \in [\![1,2]\!]$ 5 if $w \in [\![3,10]\!]$
15	9 if $w \in [\![1,2]\!]$ 8 if $w = 3$ 2 if $w \in [\![4,15]\!]$	25 if $w \in [\![1,2]\!]$ 26 if $w = 3$ 2 if $w \in [\![4,15]\!]$	54 if $w \in [\![1,2]\!]$ 56 if $w = 3$ 3 if $w \in [\![4,15]\!]$
20	5 if $w \in [\![1,2]\!]$ 4 if $w \in [\![3,4]\!]$ 2 if $w \in [\![5,20]\!]$	17 if $w \in [\![1,4]\!]$ 2 if $w \in [\![5,20]\!]$	38 if $w \in [\![1,4]\!]$ 3 if $w \in [\![5,20]\!]$

We compare the formulations with respect to different indicators which are (1) the number of instances for which the MIP is solved to optimality, (2) the CPU time (s) taken to solve the LP relaxation, (3) the CPU time (s) taken to solve the MIP, (4) the objective function value of the LP relaxation, (5) the objective function value of the MIP optimal solution when available, cost of the best solution found otherwise, (6) the number of nodes in the branch-and-cut tree (7) the integrality gap (%) and (8) the optimality gap (%).

For a particular instance, if we denote by z_{LP}^X the objective function value of the LP relaxation with formulation X and by z^* the optimal objective function value of this instance when available (or the best objective function value obtained among all formulations for this instance otherwise), the integrality gap is computed as $(z^* - z_{LP}^X) \ / \ z^*$. The optimality gap is the gap between the best solution found and the best lower bound given by CPLEX at the end of the CPU time limit.

In the following sections, results will be reported in two tables. The first table illustrates the aggregated results obtained for $|T| = 15$ while the second table displays the aggregated results obtained for $|T| = 30$. In each table, each row represents the results obtained for a particular formulation while each column refers to the different indicators previously defined. In the tables, MIP-opt denotes the number of MIP optimal solutions obtained (out of 240 instances in each table); LP-CPU and MIP-CPU represent the CPU time taken to solve the LP and MIP instances, respectively; LP-cost and MIP-cost represent the cost of the LP and MIP optimal solutions (or best solution found at the end of the time limit for the MIP solutions), respectively; I-gap gives the integrality gap and O-gap indicates the optimality gap. In Sections 2.4.1 and 2.4.2 we will report the results for the uncapacitated and capacitated instances, respectively. In Section 2.4.3, we will perform an analysis of the influence of the parameters in our experiments. For more detailed results, the interested reader is referred to Gruson et al. (2017).

2.4.1 Uncapacitated instances

We first report the results for the balanced network in Section 2.4.1, followed by the unbalanced network in Section 2.4.1. For the uncapacitated instances, we performed our experiments on a 3.07 GHz Intel Xeon processor with only one thread. For these instances, CPLEX was able to find a feasible MIP solution for all uncapacitated instances with a balanced network and with an unbalanced network. The LP relaxation values are calculated separately. Note that we do not impose any time limit to solve the LP relaxations.

Balanced network

In the balanced network, each warehouse is responsible for approximately the same number of retailers (see Table 2.2). Tables 2.4 and 2.5 illustrate the performance of the different MIP formulations for $|T| = 15$ and $|T| = 30$, respectively. In Table 2.4, which presents the results for the small instances, one can see that the formulations MC, MCE, N

Table 2.4 – Performance of the formulations for the uncapacitated balanced network - $|T| = 15$

Formulation	LP-cost	LP-CPU (s)	MIP-cost	MIP-CPU (s)	Nodes	MIP-opt	I-gap (%)	O-gap (%)
C	186156	0.03	327484	8291	71832	157	40.94	2.94
ES	186156	0.02	326906	601	29725	240	40.94	0
ES-N	320903	0.47	326906	117	4253	240	1.62	0
ES-TP	320903	1.69	326906	177	2652	240	1.62	0
ES-LS	320897	1.51	326906	298	1760	240	1.62	0
ES-F	186156	0.03	326906	875	29628	238	40.94	0
ES-F-N	320903	0.7	326906	121	3401	240	1.62	0
ES-F-TP	320903	1.3	326906	214	3673	240	1.62	0
ES-F-LS	320897	1.12	326906	209	3110	240	1.62	0
N	326887	121.27	326906	74	0.3	240	4.7×10^{-3}	0
TP	326832	80.21	326906	82	0.8	240	0.02	0
MC	326832	26.45	326906	36	0.7	240	0.02	0
MCE	326832	37.8	326906	40	0.7	240	0.02	0

Table 2.5 – Performance of the formulations for the uncapacitated balanced network - $|T| = 30$

Formulation	LP-cost	LP-CPU (s)	MIP-cost	MIP-CPU (s)	Nodes	MIP-opt	I-gap (%)	O-gap (%)
C	240367	0.07	664638	21600	35357	0	60.86	24.58
ES	240367	0.05	645908	15252	91231	84	60.86	4.03
ES-N	624974	6.77	643306	6070	53463	186	2.77	0.09
ES-TP	624974	30.3	643714	7744	14847	175	2.77	0.64
ES-LS	624935	4.09	644312	9035	4786	160	2.77	0.78
ES-F	240367	0.14	644747	14404	24791	90	60.86	2
ES-F-N	624974	11.14	643863	6271	32941	181	2.77	0.1
ES-F-TP	624974	26.5	643385	8174	23708	173	2.77	0.4
ES-F-LS	624935	5.04	643843	9931	17482	155	2.77	0.76
N	643057	27969.13	1068367	9209	0.9	188	0.04	16.86
TP	642779	1901.78	693483	5773.58	2.4	211	0.08	3.62
MC	642779	826.09	643303	1021.77	5.1	240	0.08	0
MCE	642779	996.56	643303	1276.72	5.2	240	0.08	0

and TP obtain the best performance in general, with all MIP optimal solutions found, the lowest MIP-CPU and a value of the LP relaxation which is very close to the optimal MIP cost. Yet, the LP relaxation for these three formulations is not the same as the MIP optimal cost, as witnessed by the small but positive values for the I-gap. The impact of the strength of the LP relaxation can also be seen on the small number of nodes in the branch-and-cut tree, less than 1 on average. Besides, the MC formulation has the lowest MIP-CPU time among all formulations. However, the CPU time needed to solve the LP relaxation of these formulations is much higher than with the other formulations. Note that for the N

formulation, the LP-CPU is higher than the MIP-CPU because of the efficiency of the heuristic used by CPLEX at the root node before going in the branch-and-cut tree. In general, the high performance of these formulations is also expected because of the rich information which is contained in the decision variables used for each formulation.

For the small instances, the C formulation obtains the worst results, mainly because of a poor LP relaxation as shown by the integrality gap reported in Table 2.4. The echelon stock based formulations can be divided into two groups with formulations ES and ES-F on one side, and formulations ES-N, ES-TP, ES-LS, ES-F-N, ES-F-TP and ES-F-LS on the other side. The last six formulations are much stronger than the first two formulations, as indicated by the integrality gap reported in Table 2.4. They were able to solve all instances to optimality, which is not the case for the ES and ES-F formulations. This better performance of these six formulations is easily explained by the use of a reformulation of the uncapacitated lot sizing structure found in the ES formulation, and the resulting improved LP bound. The effect of this improved LP bound can also be seen in the number of nodes in the branch-and-cut tree, which is lower than for formulations C, ES and ES-F, and in the low I-gap obtained, around 1.6%.

Table 2.5 reports the performance of each formulation for the large instances, with $|T| = 30$. The performance of the richer formulations N, TP, MC and MCE is more contrasted than for the small instances. The number of instances solved to optimality for the N formulation is much lower than for the three other rich formulations. This can be explained by the inability of the N formulation to solve the LP relaxation of the instances in a short time. One can see a similar behavior, but to a lesser extent, for the TP formulation. This difficulty for the formulations N and TP to even solve the LP relaxations of many large instances can be explained by the huge number of variables used in the models when $|T| = 30$, which is a major drawback of these two formulations. This practical drawback is the price one has to pay for the strong LP relaxation given by these two formulations, as stated by the theoretical results presented in Section 2.3.7. Finally, the MC formulation still provides the best performances for these large instances,

both in terms of CPU time to solve the MIP instances and in terms of number of optimal solutions found within the time limit.

In light of the results provided in Tables 2.4 and 2.5, we can draw the following conclusions about the performance of our formulations on an uncapacitated balanced network: (a) the C formulation is the poorest, mainly because of a bad LP relaxation; (b) applying the echelon stock reformulation to the classical formulation does not have any impact on the LP relaxation value (as we also theoretically proved), but the results nevertheless show a substantial improvement in CPU time, optimality gap and number of instances solved to optimality (the conjecture is that because the echelon stock reformulation exposes the single item lot sizing structure at the three different levels, CPLEX is able to derive better cuts); (c) the echelon stock reformulation can still be improved by explicitly using one of the lot sizing reformulations at each level, i.e., using formulations ES-N, ES-TP, ES-LS, ES-F-N, ES-F-TP and ES-F-LS, with ES-N generally having the best performance among these six formulations; (d) when comparing the various echelon stock reformulations with the traditional echelon stock constraints (2.10) to their counterpart using the constraint (2.22) proposed in Federgruen and Tzur (1999), we observe individual differences, but overall no general tendencies appear and the formulations provide fairly similar results; (e) the N and TP formulations have difficulty to solve the LP relaxations of some large instances because of the huge size of the model resulting in an overall substantially weaker performance compared to the best formulation; (f) the MC formulation performs the best for the balanced network; (g) the results we obtained here are in line with the ones obtained by Solyalı and Süral (2012) and Cunha and Melo (2016) for the OWMR.

Unbalanced network

We performed the same experiments as in Section 2.4.1 but considering an unbalanced distribution network. In the unbalanced network, 20% of the warehouses are responsible for 80% of the retailers (see Table 2.3). Tables 2.6 and 2.7 illustrate the performance of

Table 2.6 – Performance of the formulations for the uncapacitated unbalanced network - $|T| = 15$

Formulation	LP-cost	LP-CPU (s)	MIP-cost	MIP-CPU (s)	Nodes	MIP-opt	I-gap (%)	O-gap (%)
C	177633	0.02	310925	5669	21108	197	40.78	0.66
ES	177633	0.02	310871	602	14711	239	40.78	0
ES-N	300182	0.55	310871	182	4046	240	2.99	0
ES-TP	300182	1.68	310871	262	3084	240	2.99	0
ES-LS	300178	1.6	310871	414	2917	240	2.99	0
ES-F	177633	0.04	310871	1166	17758	240	40.78	0
ES-F-N	300182	0.89	310871	187	3731	240	2.99	0
ES-F-TP	300182	1.92	310871	303	3815	240	2.99	0
ES-F-LS	300178	1.24	310871	372	3509	240	2.99	0
N	310832	125.33	310871	112	1	240	0.01	0
TP	310750	58.37	310871	94	2.7	240	0.03	0
MC	310750	20.33	310871	40	1.6	240	0.03	0
MCE	310750	41.06	310871	48	1.6	240	0.03	0

Table 2.7 – Performance of the formulations for the uncapacitated unbalanced network - $|T| = 30$

Formulation	LP-cost	LP-CPU (s)	MIP-cost	MIP-CPU (s)	Nodes	MIP-opt	I-gap (%)	O-gap (%)
C	231785	0.06	624878	21104	36752	10	60.35	19.39
ES	231785	0.05	613737	14748	26617	91	60.35	4.89
ES-N	583375	8	610963	8690	30169	164	4.14	0.37
ES-TP	583375	29.78	611589	10271	15351	149	4.14	1.36
ES-LS	583349	6.53	612763	12295	7218	128	4.14	1.63
ES-F	231785	0.17	613421	14547	14335	90	60.35	2.99
ES-F-N	583375	20.82	611004	9512	24100	157	4.14	0.45
ES-F-TP	583375	45.58	611424	10510	18438	147	4.14	1.11
ES-F-LS	583349	10.48	612275	11767	10525	130	4.14	1.63
N	610542	11473.94	828581	8019	3.7	204	0.04	9.05
TP	610109	1700.49	705844	6356	14.7	201	0.1	5.55
MC	610109	460.85	610908	1364	19.2	240	0.1	0
MCE	610109	994.48	610908	1476	18.7	239	0.1	0

our formulations for the small and large instances, respectively. In Table 2.6, one can see that, compared to Table 2.4 and except for the formulation, there is an increase in CPU time to solve the instances as MIPs. This increase ranges between 0.16% and 78.5% for the ES formulation and for the ES-F-LS formulation, respectively. conclusions drawn in Section 2.4.1 for the small instances with a balanced network still hold for an unbalanced structure of the supply network.

In Table 2.7, one can see that, for the formulation, the performance is worse than in the case of a balanced network. This difficulty is in particular reflected in the number of

optimal MIP solutions found, which decreases by a number ranging from 0 for the MC formulation and up to 32 for the ES-LS formulation. This indicates that the unbalanced instances are harder to solve than the balanced instances. This difficulty can be explained by the fact that, in the network, the warehouses that are responsible for many retailers represent a much larger MIP to solve. Compared to the balanced instances, we have thus several big distribution channels to cope with, which makes the instances harder to solve. Note, however, that formulations C, ES and N were able to find more optimal MIP solutions for the unbalanced instances.

In light of the results provided in Tables 2.6 and 2.7, we can draw the following conclusions about the performance of our formulations on an unbalanced network: (a) the unbalanced instances are generally harder to solve than the balanced instances; (b) the C, N and ES formulations have a better performance on the unbalanced instances than on the balanced ones in terms of number of instances solved to optimality; (c) the other formulations have a worse performance on the unbalanced instances compared to the balanced ones; (d) the N and TP formulations have a large O-gap for many large instances; (e) the MC formulation is the best suited for the unbalanced instances since it is able to solve all instances to optimality with the lowest CPU time.

2.4.2 Capacitated instances

For the capacitated instances, we set the production capacity as a given factor C of the average total demand. The production capacity imposed is thus $C_t = C \sum_{i \in R} \sum_{t \in T} d_t^i \; / \; |T|$. We additionally consider three different values for the capacity factor $C : C \in \{2, 1.75, 1.5\}$. We performed these experiments on a 6.67 GHz Intel Xeon X5650 Westmere processor with one thread. Because of the bad performance of the formulations C, ES and ES-F in the previous section, and based on preliminary results, we decided not to run experiments using these formulations. For the sake of brevity, we also do not display the results obtained with formulations ES-F-N, ES-F-TP and ES-F-LS as these results were similar to the ones obtained by formulations ES-N, ES-TP and ES-LS. For the capacitated instances,

we impose a time limit of 6 hours even to solve the LP instances. Note that we did not get any infeasible instances during our experiments.

The results of this section will be reported in tables having the same columns as the tables in Section 2.4.1 plus three additional columns indicating the value of the capacity factor, the number of LP optimal solutions found within the time limit and the number of instances for which a MIP solution was found, in columns Capacity, LP-opt and MIP-sol, respectively. For the columns LP-cost and I-gap, we only report the average cost and integrality gap obtained, respectively, over instances for which all formulations have both solved the LP relaxation to optimality and have found a MIP solution within the time limit. In the same vein, for the columns MIP-cost, Nodes and O-gap, we only report the average MIP cost, number of nodes and optimality gap obtained, respectively, over instances for which all formulations have found a MIP solution within the time limit. We first report the results for the balanced network in Section 2.4.2, followed by the unbalanced network in Section 2.4.2. Note also that we performed some additional experiments, for which we present only the general results. Experiments using two and four threads instead of one indicated that the CPU time needed to solve the instances decreases by a factor of up to 4 for the best performing formulations and for small problems ($|T| = 15$). We also performed additional experiments with time-varying capacity on the balanced network with $|T| = 15$. The general conclusions on the relative performance of the presented formulations still hold, but the performance of formulations N and TP deteriorated substantially as indicated by the bad quality of the upper bounds found and the fact that none of the instances were solved to optimality within the time limit.

Balanced network

Tables 2.8 and 2.9 illustrate the performance of the different MIP formulations for the different values of the time horizon. When comparing the results with those obtained for the uncapacitated instances on the balanced network, we can see that they are completely different. Indeed, the richer formulations have more trouble achieving a good

Table 2.8 – Performance of the formulations for the capacitated balanced network - $|T| = 15$

Capacity factor	Formulation	LP-cost	LP-CPU (s)	LP-opt	MIP-cost	MIP-CPU (s)	Nodes	MIP opt	MIP sol	I-gap (%)	O-gap (%)
	ES-N	486642	1.06	240	510641	9517	141103	161	207	4.76	0.06
	ES-TP	486642	2.39	240	510677	10368	38086	151	240	4.76	0.1
	ES-LS	486634	1.71	240	510675	9030	37634	164	233	4.76	0.1
2.0	N	490899	173.84	240	511024	17181	4453	89	240	3.93	0.92
	TP	490800	398.16	240	511809	18104	3372	76	240	3.95	1.14
	MC	490800	185.39	240	511242	16582	9398	92	240	3.95	1.02
	MCE	490800	192.17	240	511042	14988	6901	113	240	3.95	0.77
	ES-N	549589	1.18	240	572268	11059	118612	142	231	3.99	0.31
	ES-TP	549589	2.48	240	572377	15231	47566	80	161	3.99	0.32
	ES-LS	549583	1.88	240	572395	14929	43996	83	159	3.99	0.32
1.75	N	554709	113.09	240	573023	17532	10690	77	240	3.14	0.72
	TP	554597	283.63	240	573307	18183	10586	67	240	3.16	1.06
	MC	554597	144.67	240	573036	17254	15059	83	240	3.16	0.9
	MCE	554597	153.53	240	572939	17060	12081	83	240	3.16	0.8
	ES-N	574785	1.7	240	583124	13304	184360	116	198	1.41	0.15
	ES-TP	574785	2.71	240	583221	14764	74318	99	210	1.41	0.22
	ES-LS	574779	3.1	240	583280	13069	52541	112	190	1.41	0.22
1.5	N	579363	124.95	240	583690	18349	12262	62	212	0.64	0.24
	TP	579287	360.09	240	584047	18712	11320	55	227	0.66	0.32
	MC	579287	136.62	240	583888	17126	19018	75	238	0.66	0.23
	MCE	579287	156.63	240	583528	16941	20167	78	234	0.66	0.21

performance in terms of CPU time, MIP cost, number of MIP optimal solutions found and optimality gap. On the contrary, the echelon stock formulations have a better performance than the richer formulations on these indicators. This difference in performance is even more pronounced when the capacity level gets tighter. This indicates that the capacity constraint has a major impact on the performance of the formulations. Despite the properties related to the strength of their LP relaxation for the uncapacitated case, the richer formulations seem to be less adequate to handle capacitated instances.

We also see that the MC formulation does not perform the best for the capacitated instances on the balanced network. The best performance, in terms of MIP-CPU time, number of optimal solutions found and optimality gap, is obtained by one of the echelon stock formulations, depending on the capacity level. Within the richer formulations, our newly introduced MCE formulation performs the best on average. Note also that the addition of the capacity constraint makes the problem harder, as stated by the increase in CPU

Table 2.9 – Performance of the formulations for the capacitated balanced network - $|T| = 30$

Capacity factor	Formulation	LP-cost	LP-CPU (s)	LP-opt	MIP-cost	MIP-CPU (s)	Nodes	MIP opt	MIP sol	I-gap (%)	O-gap (%)
	ES-N	907807	18.72	240	922301	21544	113942	2	240	1.5	0.9
	ES-TP	907807	24.31	240	923143	21576	26371	1	240	1.5	1.05
	ES-LS	907778	7.91	240	923508	21538	27768	2	231	1.5	0.98
2.0	N	913381	7689.24	191	1112999	21706	94	0	141	0.89	14.28
	TP	913184	8999.21	193	1288961	21867	66	0	191	0.91	21.71
	MC	913184	2626.04	240	934216	21601	1441	0	240	0.91	1.43
	MCE	913184	2522.05	240	928742	21603	1622	0	240	0.91	1.43
	ES-N	1014806	17.06	240	1056456	21600	42768	0	234	3.79	3.39
	ES-TP	1014806	21.94	240	1055172	21563	28240	2	236	3.79	0.95
	ES-LS	1014785	8.75	240	1055736	20914	36048	13	128	3.79	0.59
1.75	N	1019421	6634.88	202	1273297	21673	56	0	131	3.35	17.42
	TP	1019195	7936.58	203	1321777	21737	44	0	210	3.37	19.58
	MC	1019195	2460.44	240	1063381	21602	1281	0	240	3.37	4.04
	MCE	1019195	2196.98	240	1063581	21473	2136	6	239	3.37	2.75
	ES-N	1149701	18.96	240	1172770	21562	68917	1	194	1.74	1.39
	ES-TP	1149701	25.97	240	1174267	21588	26223	1	169	1.74	1.43
	ES-LS	1149660	10.5	240	1175080	21500	32601	2	193	1.74	1.3
1.5	N	1160244	5867.03	218	1300323	21701	208	0	181	0.91	9.07
	TP	1160088	6773.7	222	1298666	21782	182	0	213	0.92	8.97
	MC	1160088	2257.01	240	1199898	21600	1940	0	240	0.92	2.74
	MCE	1160088	1947.3	240	1191768	21600	1825	0	240	0.92	2.28

time to solve both the MIP and LP instances. This difficulty is also apparent by observing that the number of MIP solutions found is not equal to the number of instances present in the data set used for the experiments. Finally, note that in Table 2.9, for formulations N, TP, MC and MCE, the values obtained for O-gap is higher than the values obtained for I-gap. Since the I-gap is calculated relative to the optimal or best solution found among all formulations this indicates that these formulations have a good LP relaxation but are unable to provide a MIP solution with a low objective function value.

Unbalanced network

Tables 2.10 and 2.11 illustrate the performance of the different MIP formulations on the unbalanced instances for the different values of the time horizon and capacity level. If we compare the results with those obtained for the uncapacitated instances on the unbalanced network, we can see similar differences as the ones observed in Section 2.4.2.

Table 2.10 – Performance of the formulations for the capacitated unbalanced network - $|T| = 15$

Capacity factor	Formulation	LP-cost	LP-CPU (s)	LP-opt	MIP-cost	MIP-CPU (s)	Nodes	MIP opt	MIP sol	I-gap (%)	O-gap (%)
	ES-N	478953	1	240	504744	1264	28050	238	238	5.15	0
	ES-TP	478953	1.96	240	504744	2107	22230	236	240	5.15	0
	ES-LS	478950	1.28	240	504744	1531	13069	239	240	5.15	0
2.0	N	485877	118.19	240	504904	13804	5406	144	239	3.82	0.31
	TP	485819	302.27	240	505029	13626	4473	142	240	3.83	0.46
	MC	485819	142.82	240	504920	12935	6093	130	239	3.83	0.42
	MCE	485819	129.69	240	504851	11099	6894	155	237	3.83	0.24
	ES-N	526251	1.06	240	549514	2225	45995	238	240	4.27	0
	ES-TP	526251	2.01	240	549524	3193	28369	227	240	4.27	0.01
	ES-LS	526248	1.46	240	549516	2434	25884	236	239	4.27	0
1.75	N	532584	106.24	240	549853	15214	7213	119	240	3.14	0.56
	TP	532528	391.23	240	550351	16144	5282	96	240	3.15	0.95
	MC	532528	125.55	240	549761	14359	10031	113	240	3.15	0.56
	MCE	532528	122.02	240	549700	12874	11765	141	240	3.15	0.41
	ES-N	568713	1.19	240	577246	5628	129453	203	222	1.48	0
	ES-TP	568713	2.03	240	577254	8292	82476	174	204	1.48	0.02
	ES-LS	568710	1.54	240	577264	7253	102830	185	215	1.48	0.02
1.5	N	573416	105.81	240	577546	17308	14994	77	238	0.68	0.14
	TP	573355	321.95	240	577558	17069	15284	81	239	0.69	0.14
	MC	573355	130.26	240	577413	14334	26345	110	234	0.69	0.09
	MCE	573355	134.1	240	577386	14169	34757	114	236	0.69	0.08

The richer formulations also have more trouble obtaining a good performance than on the uncapacitated instances. They have a worse performance than the echelon stock formulations on numerous performance indicators, such as the number of best solutions found, which is generally much higher for the echelon stock formulations. Within the richer formulations, the MCE formulation still has the best performance on average. Note finally that, compared to the balanced structure, the unbalanced structure of the supply network combined with the production capacity restriction results in general in better values for the number of MIP solutions found and for the number of MIP optimal solutions found.

In light of the results provided in Tables 2.8 - 2.11, we can draw the following conclusions about the performance of our formulations on capacitated instances: (a) the capacitated instances are harder to solve than the uncapacitated instances; (b) the richer formulations have a relative worse performance than on uncapacitated instances compared to the echelon stock formulations; (c) the echelon stock formulations are better than the

Table 2.11 – Performance of the formulations for the capacitated unbalanced network - $|T| = 30$

Capacity factor	Formulation	LP-cost	LP-CPU (s)	LP-opt	MIP-cost	MIP-CPU (s)	Nodes	MIP opt	MIP sol	I-gap (%)	O-gap (%)
	ES-N	886871	13.45	240	908524	19921	62720	27	239	2.24	1
	ES-TP	886871	16.8	240	909227	20267	30783	22	240	2.24	1.24
	ES-LS	886856	9.8	240	910133	20163	22140	24	219	2.24	1.24
2.0	N	897248	6767.62	200	1095597	21673	108	0	155	1.14	14.13
	TP	897096	8348.99	203	1077364	21706	124	0	194	1.15	12.51
	MC	897096	2226.9	240	919258	21481	1558	3	240	1.15	1.92
	MCE	897096	1806.59	240	919104	21523	1994	4	240	1.15	1.86
	ES-N	983612	17.08	240	1029626	21465	32601	2	240	4.34	3.56
	ES-TP	983612	18.61	240	1028846	20596	25641	18	240	4.34	1.09
	ES-LS	983594	10.65	240	1029850	20653	29067	16	236	4.34	1.12
1.75	N	993223	6127.87	212	1197855	21691	98	0	178	3.43	14.45
	TP	993072	7664.68	208	1182565	21788	116	0	201	3.45	13.25
	MC	993072	2052.43	240	1041875	21600	1604	0	239	3.45	4.34
	MCE	993072	1608.27	240	1037816	21229	2236	10	240	3.45	2.58
	ES-N	1078552	18.64	240	1104468	20795	45636	14	239	2.17	1.31
	ES-TP	1078552	21.21	240	1105210	20908	31171	10	236	2.17	1.4
	ES-LS	1078537	12.15	240	1105338	20800	25449	13	159	2.17	1.28
1.5	N	1091771	5397.68	218	1260172	21811	173	0	203	1.04	11.44
	TP	1076552	6687.1	217	1268325	21839	142	0	182	2.11	11.96
	MC	1091655	1982.83	240	1120214	21526	2426	2	240	1.05	2.15
	MCE	1091655	1472.36	240	1114113	21552	3599	1	240	1.05	1.64

richer formulations; (d) within the richer formulations, the MCE formulation has the best performances.

2.4.3 Influence of the parameters

Table 2.12 reports the performance of the MC formulation for all experiments with a balanced uncapacitated network and with $|T| = 30$. The first two columns indicate the parameter that varies and the respective values taken by the parameter. We only report here the results for the MC formulation with a balanced network. Since most of the following conclusions also apply for the other formulations and for the experiments with an unbalanced network. The findings that are specific to this formulation are discussed at the end of this section. All the other results are available in Gruson et al. (2017).

In Table 2.12, one can see that when $|R|$ increases, the problems gets harder and the CPU time taken to solve both the LP and MIP instances increases. On the contrary, when

Table 2.12 – Performances of the MC formulation for the uncapacitated balanced network - $|T| = 30$

Parameter	Value	LP-cost	LP-CPU (s)	MIP-cost	MIP-CPU (s)	Nodes	MIP-opt	I-gap (%)	O-gap (%)
$\|R\|$	50	423630	60.36	423765	88.07	1.9	80	0.04	0
	100	609655	414.54	610096	643.15	4.5	80	0.08	0
	200	895053	2003.37	896048	2334.08	8.8	80	0.12	0
$\|W\|$	5	540034	587.33	541416	1451.26	14.8	60	0.23	0
	10	621960	912.23	622489	1095.86	3.5	60	0.07	0
	15	678045	1023.23	678196	827.76	1.5	60	0.02	0
	20	731078	781.58	731111	712.18	0.5	60	0	0
Costs	SF	658846	1040.45	659632	1508.85	8.6	120	0.12	0
	DF	626713	611.74	626974	534.68	1.6	120	0.04	0
Demand	SD	644294	840.35	644921	1077.51	6.4	120	0.1	0
	DD	641265	811.84	641685	966.03	3.7	120	0.06	0

$|W|$ increases, the CPU time taken to solve the MIP instances decreases. Indeed, with the same number of retailers, if the number of warehouses increases, the supply network has a smaller number of channels linked to each warehouse. This leads to a smaller problem per warehouse and makes the global problem easier to solve, thus reducing the CPU time and the number of nodes. The integrality gap is also lower but less significantly. Table 2.12 indicates that for the MC formulation, generally the instances with dynamic fixed costs are much easier to solve compared to the instances with a static fixed cost. We further note that the dynamic demand case is generally slightly easier to solve than the static demand case.

Finally, the detailed results provided in Gruson et al. (2017) illustrate the fact that the impact of the setting of the parameters (static or dynamic demand, static or dynamic fixed cost), depends on the kind of formulation used. For the C formulation, apart from the very small instances where $|R| = 50$ and $|T| = 15$, the instances with a dynamic fixed cost are harder to solve, thus requiring a higher CPU time. For the ES-N, ES-TP and ES-LS formulations, the instances with a dynamic fixed cost are also harder to solve. On the contrary, for the N, TP and MC formulations, the instances with a static fixed cost are harder to solve in terms of CPU time required. For the ES and ES-F formulations, there is no clear impact of the setting of the parameters on the CPU time required to solve the instances. Note, however, that this result does not question the higher global performance

of the MC formulation stated in the previous sections.

2.5 Conclusions

We have extended eight MIP formulations proposed in the context of the OWMR and have applied them to the 3LSPD. We also introduced the ES-N, ES-F-N, ES-F-TP, ES-F-LS and MCE formulations that had not been tested before in the context of the OWMR. For our numerical experiments, we have considered two network structures (a balanced one and an unbalanced one) and have assessed the performance of the formulations proposed using several indicators. We have also considered the possibility of having production capacities at the plant level. The results indicate that, for the uncapacitated case, the unbalanced instances are harder to solve than the balanced instances and lead to a worse performance of all formulations, except for the C formulation. On the contrary, for the capacitated case, the unbalanced instances give better values for our different performance indicators compared to the balanced instances. The MC formulation obtains the best performance on the uncapacitated instances and is able to solve all instances for both network structures. This result is similar to the conclusion of Cunha and Melo (2016) for the OWMR. The other formulations obtain results that are not entirely satisfactory for the uncapacitated instances. In particular, for the rich formulations TP and N, the non-satisfactory performances on the large instances, in terms of number of MIP optimal solutions found and CPU time, are due to the huge size of the model. As a consequence, it is already very time-consuming to solve the LP relaxation of these formulations. When we impose capacity restrictions for production at the plant level, the performance of the formulations are reversed: the rich formulations have a worse performance and the echelon formulations have the best performance. Within the rich formulations, for the capacitated instances, our newly introduced MCE formulation generally has the best performance.

In light of the results obtained, we recommend that researchers and decision-makers interested in solving integrated lot sizing and replenishment problems with a distribution structure carefully choose the formulation they will use to model their problem. Despite

theoretical advantages, richer formulations are not necessarily the ones who obtain the best performances, both in terms of CPU time taken to solve the problem and of the quality of the solution obtained. On the contrary, this choice should be guided by (1) the structure of the distribution network (balanced or unbalanced), (2) the presence of production capacity restrictions and (3) the settings of the parameters of the problem.

Acknowledgements

We would like to thank the three anonymous referees for their valuable comments. 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).

References

Abdullah, S., A. Shamayleh and M. Ndiaye. 2019, «Three stage dynamic heuristic for multiple plants capacitated lot sizing with sequence-dependent transient costs», *Computers & Industrial Engineering*, vol. 127, p. 1024–1036.

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

Arkin, E., D. Joneja and R. Roundy. 1989, «Computational complexity of uncapacitated multi-echelon production planning problems», *Operations Research Letters*, vol. 8, n° 2, p. 61–66.

Barany, I., T. Van Roy and L. A. Wolsey. 1984, «Uncapacitated lot-sizing: The convex hull of solutions», *Mathematical Programming*, vol. 22, p. 32–43.

Barbarasoglu, G. and D. Özgür. 1999, «Hierarchical design of an integrated production and 2-echelon distribution system», *European Journal of Operational Research*, vol. 118, p. 464–484.

Brahimi, N., N. Absi, S. Dauzère-Pérès and A. Nordli. 2017, «Single-item dynamic lot-sizing problems: An updated survey», *European Journal of Operational Research*, vol. 263, p. 838–863.

Cunha, J. O. and R. A. Melo. 2016, «On reformulations for the one-warehouse multi-retailer problem», *Annals of Operations Research*, vol. 238, n° 1, p. 99–122, ISSN 1572-9338.

Dhaenens-Flipo, C. and G. Finke. 2001, «An integrated model for an industrial production-distribution problem», *IIE Transactions*, vol. 33, p. 705–715.

Eppen, G. and R. Martin. 1987, «Solving multi-item capacitated lot-sizing problems with variable definition», *Operations Research*, vol. 35, p. 832–848.

Federgruen, A. and M. Tzur. 1999, «Time-partitioning heuristics: Application to one warehouse, multiitem, multiretailer lot-sizing problems», *Naval Research Logistics*, vol. 46, n° 5, p. 463–486.

Gebennini, E., R. Gamberini and R. Manzini. 2009, «An integrated production – distribution model for the dynamic location and allocation problem with safety stock optimization», *International Journal of Production Economics*, vol. 122, n° 1, p. 286–304, ISSN 0925-5273.

Gruson, M., M. Bazrafshan, J.-F. Cordeau and R. Jans. 2017, «A comparison of formulations for a three-level lot sizing and replenishment problem with a distribution structure», *Cahier du GERAD, HEC Montréal*, vol. G-2017-59.

Haq, A. N., P. Vrat and A. Kanda. 1991, «An integrated production-inventory-distribution model for manufacture of urea: a case», *International Journal of Production Economics*, vol. 39, p. 39–49.

Jans, R. and Z. Degraeve. 2006, «Modeling industrial lot sizing problems: a review», *International Journal of Production Research*, vol. 46, p. 1619–1643.

Karimi, B., G. Fatemi and J. Wilson. 2003, «The capacitated lot sizing problem: a review of models and algorithms», *Omega*, vol. 31, p. 365–378.

Kopanos, G. M., L. Puigjaner and M. C. Georgiadis. 2012, «Simultaneous production and logistics operations planning in semicontinuous food industries», *Omega*, vol. 40, nº 5, p. 634–650, ISSN 0305-0483.

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

Lejeune, M. A. 2006, «A variable neighborhood decomposition search method for supply chain management planning problems», *European Journal of Operational Research*, vol. 175, p. 959–976.

Levi, R., R. Roundy, D. Shmoys and M. Sviridenko. 2008, «A constant approximation algorithm for the one-warehouse multiretailer problem», *Management Science*, vol. 54, nº 4, p. 763–776.

Melo, R. A. and L. A. Wolsey. 2010, «Uncapacitated two-level lot-sizing», *Operations Research Letters*, vol. 38, nº 4, p. 241–245, ISSN 0167-6377.

Özdamar, L. and T. Yazgaç. 1999, «A hierarchical planning approach for a production-distribution system», *International Journal of Production Research*, vol. 37, nº 16, p. 3759–3772.

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

Solyalı, O. and H. Süral. 2012, «The one-warehouse multi-retailer problem: reformulation, classification, and computational results», *Annals of Operations Research*, vol. 196, p. 517–541.

Wagner, H. M. and T. M. Whitin. 1958, «Dynamic version of the economic lot size model», *Management Science*, vol. 5, p. 89–96.

Zangwill, W. 1969, «A backlogging model and a multi-echelon model of a dynamic economic lot size production system - a network approach», *Management Science*, vol. 15(9), p. 506–527.

Zhang, S. and H. Song. 2018, «Production and distribution planning in Danone waters China division», *INFORMS Journal on Applied Analytics*, vol. 48, p. 578–590.

Chapitre 3

Dantzig-Wolfe decomposition for a capacitated three-level lot sizing and replenishment problem with a distribution structure

Abstract

We address in this chapter a capacitated version of the three-level lot sizing and replenishment problem with a distribution structure (3LSPD). We propose to use a Dantzig-Wolfe decomposition to efficiently solve this version of the problem, using a branch-and-price algorithm we develop. We take advantage of the substructures highlighted in the decomposition and design efficient procedures to solve the different subproblems obtained. We also propose several computational enhancements to speed up the solution process. We finally perform numerous computational experiments to assess the performance of our decomposition approach and see the impact of our enhancements. The branch-and-price algorithm we designed did not obtain better performance than the CPLEX solver only.

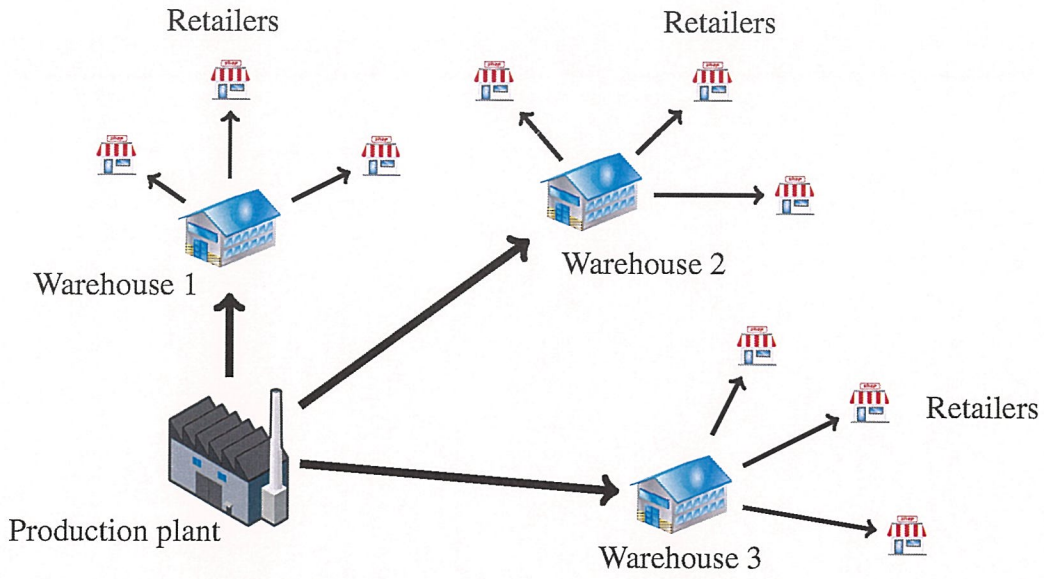

Figure 3.1 – Graphical representation of the problem considered

3.1 Introduction

We address here an integrated three-level lot sizing and replenishment problem with a distribution structure (3LSPD) as introduced in Gruson et al. (2019). We consider a general manufacturing company that has one production plant (level one), several warehouses (level two) and multiple retailers (level three) facing a dynamic and known demand for one item over a discrete and finite time horizon. The supply chain considered has a distribution structure: the warehouses are all linked to the single plant and all retailers are linked to exactly one warehouse. When we consider the demand of a particular retailer, the flow of goods in the supply chain network is hence as follows: an item is produced at the production plant, then sent to the warehouse for storage and finally sent to the retailer to satisfy its demand. Figure 3.1 illustrates this flow of goods in a distribution network which consists of one production plant, three warehouses and three retailers linked to each warehouse.

The objective of the problem is to determine the optimal timing and flows of goods between the different facilities while minimizing the operational costs in the whole network

(sum of the fixed setup and transportation costs and unit inventory holding costs). Note that the problem considered here can be viewed as an extension of the one-warehouse multi-retailer problem (OWMR) to three levels. In the OWMR, a central warehouse replenishes several retailers that face a dynamic demand for one or several items over a discrete and finite time horizon. The objective of the OWMR is to jointly determine the optimal timing and quantities that are shipped between the warehouse and the retailers to minimize the sum of setup costs and unit inventory holding costs of the whole system. In the 3LSPD, transfers of goods between the warehouses and between the retailers are not allowed. Finally, we only consider direct shipments and do not incorporate any routing in the transportation decisions. Note that in a disaggregated context, the problem faced by any facility can be seen as the basic lot sizing problem (LSP). This basic LSP has stimulated a lot of research since the seminal paper of Wagner and Whitin (1958) 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. (2017) and to Pochet and Wolsey (2006) for a review of the work done on the SI-ULSP and its extensions, and to Jans and Degraeve (2006) for a review of industrial applications.

We consider here that there are production capacity requirements at the plant. The production capacity is constant through time. We do not consider any other capacity restrictions. Note that with the addition of the capacity constraints at the production plant level, the problem faced by the production plant can be seen as a basic capacitated lot sizing problem (CLSP). The reader is referred to Karimi et al. (2003) for a review of models and algorithms used to solve the CLSP.

In a prior work, we compare thirteen MIP formulations for the problem and run experiments both on balanced and unbalanced supply chain networks (in the unbalanced network, 80% of the retailers are served by 20% of the warehouses while in the balanced network the same number of retailers is linked to each warehouse) using a general-purpose solver. The experiments are done both with and without production capacity restrictions at the plant. The results indicate that the strongest formulation, in terms of the value

of the linear relaxation, are not necessarily the best one in practice, especially when we solve large instances. Indeed, it takes too much CPU time to solve the different instances when the strong formulations are used. On the contrary, other formulations, less strong in theory, manage to obtain solutions of better quality in the same amount of time. This is mainly explained by the size of the strongest formulations compared to the size of the less strong ones. Results also indicate that the unbalanced instances are harder to solve.

The motivation to apply decomposition methods on the capacitated 3LSPD is to improve the results obtained in Gruson et al. (2019). In particular, we want to improve the CPU time taken to solve the different instances and improve the optimality gap obtained in their experiments. To do so, we apply a Dantzig-Wolfe reformulation to the capacitated version of the 3LSPD and solve it using a branch-and-price algorithm we develop.

This chapter makes two main contributions. First, we extend the work done in Gruson et al. (2019) by applying a decomposition method to the capacitated version of the problem. The decomposition method highlights the substructures that appear in the echelon stock formulation proposed in Gruson et al. (2019). We also develop a branch-and-price algorithm to solve this capacitated version of the 3LSPD. This algorithm takes advantage of the substructures revealed by the use of Dantzig-Wolfe decomposition. 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 branch-and-price algorithms.

The remainder of this chapter is organized as follows. First, we survey the work linked to our study in Section 3.2. Then, we give a formal model of the Dantzig-Wolfe reformulation in Section 3.3. In this section, we also present the original formulation on which we apply Dantzig-Wolfe reformulation. Section 3.4 gives the details of the branch-and-price algorithm we designed, along with computational enhancements. Section 3.5 details the computational results obtained to assess the performance of the branch-and-price algorithm, and to analyze the impact of the improvements we add to the initial branch-and-price algorithm we develop. This is followed by the conclusion in Section 3.6.

3.2 Literature review

To the best of our knowledge, the application of Dantzig-Wolfe decomposition in pure lot sizing problems is relatively scarce. Degraeve and Jans (2007) address the CLSP with setup times and propose a Dantzig-Wolfe reformulation. Although Manne (1958) had already proposed such a reformulation, they highlight the fact that, because the integrality requirements are imposed at the master level, an optimal integer solution may not be found. They overcome this drawback by giving an exact Dantzig-Wolfe reformulation to the original CLSP with setup times. They finally design a branch-and-price algorithm to solve the problem. This work was used as a starting point in Caserta and Voß (2013) where the authors develop a matheuristic exploiting the structures that were highlighted in the Dantzig-Wolfe reformulation proposed in Degraeve and Jans (2007). De Araujo et al. (2015) also study the multi-item CLSP with setup times. They develop a branch-and-price heuristic based on two strong reformulations of the multi-item CLSP with setup times. Those two reformulations are based on the shortest path and the facility location reformulations for the single item lot sizing problem. The authors develop period Dantzig-Wolfe decompositions and design a specific algorithm to solve the subproblems. They further use a lagrangian relaxation scheme to solve the restricted master problem by dualizing the linking constraints, and to obtain new columns for their column generation algorithm. They finally use a heuristic rule to fix to a certain value the setup variables that are above or beyond a certain threshold. In the same vein, Fragkos et al. (2016) propose horizon decomposition. They also study the multi-item CLSP with setup times and apply a Dantzig-Wolfe decomposition to a model that includes the idea of horizon decomposition. The principle of horizon decomposition is to decompose the problem in several subproblems of identical structure but with a shorter time horizon than the original problem. In their decomposition, the horizon of the different subproblems have overlaps. This is done to increase the exchange of information between the different subproblems. The reformulation is solved using a branch-and-price algorithm. While the works of Degraeve and Jans (2007) and Caserta and Voß (2013), and the works of De Araujo et al.

(2015) and Fragkos et al. (2016) decompose the subproblem per item and per period, respectively, Pimentel et al. (2010) had proposed earlier to decompose the subproblem per item and period, still using Dantzig-Wolfe decomposition. They develop a branch-and-price algorithm based on such decomposition and report better lower bounds than the ones obtained when the subproblem is decomposed per period or item only.

In the context of stochastic lot sizing, Tempelmeier (2011) uses a set partitioning formulation for a dynamic CLSP with random demand. The author imposes a fill rate constraint and uses column generation to solve the problem. The Dantzig-Wolfe decomposition is not explicitly mentioned in this work but the model proposed (set partitioning) can easily be related to it.

Regarding integrated problems such as the inventory routing problem (IRP), Le et al. (2013) use a column generation based heuristic for a case with perishability constraints. In a tactical planning setting, Michel and Vanderbeck (2012) develop a branch-and-price-and-cut algorithm after applying Dantzig-Wolfe decomposition to their IRP, where their main objective is to cluster the customers rather than defining the actual sequence of visits, i.e., the routes. In the same vein but in the context of the production routing problem (PRP), Mourgaya and Vanderbeck (2007) also use Dantzig-Wolfe decomposition and develop a column generation algorithm to choose clusters of customers that are then used as starting points to build routes. Finally, still in the context of the PRP, Bard and Nananukul (2010) also apply Dantzig-Wolfe decomposition and develop a branch-and-price algorithm that combines both exact and heuristic procedures. They further incorporate efficient branching strategies to escape the degeneracy issue often encountered in column generation. Regarding integrated problems, the use of Dantzig-Wolfe reformulation can also be encountered in integrated lot sizing and scheduling problem, see, e. g., Duarte and de Carvalho (2013).

3.3 Dantzig-Wolfe reformulation

In this section we apply a Dantzig-Wolfe reformulation to the echelon stock formulation for the 3LSPD as introduced in Gruson et al. (2019). We start by giving some general notation for the problem. Let $G = (F, A)$ be a graph where F is the set of nodes (facilities in our problem) and A is the set of arcs. Let $P = \{p\} \subset F$ be the set containing the unique production plant, $W \subset F$ be the set of warehouses and $R \subset F$ be the set of retailers. Following the problem description in Section 3.1, 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 $r \in R$. Let d_{rt} be the demand for retailer r in period t. The notion of the demand faced by any retailer is extended to the warehouses and to the production plant in the following fashion:

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

Using the notion of the demand faced by any facility, we introduce D_{it}, the total demand between period t and the end of the time horizon computed as $D_{it} = \sum_{k \geq t} d_{ik}$. Similarly, we introduce, for any facility i, the demand between periods k and t as $d_{ikt} = \sum_{k \leq l \leq t} d_{il}$. Let sc_{pt}, sc_{wt} and sc_{rt} be the fixed production costs, the fixed transportation cost between the plant and warehouse w, and the fixed transportation cost between retailer r and its warehouse, respectively. Let hc_{it} be the unit inventory holding cost of facility i in period t. In the following sections, we do not include any unit production cost since the total production cost is a constant when all the demand is satisfied. The same holds for the unit transportation cost. Finally, let C be the available production capacity in any period t.

3.3.1 The echelon stock formulation

In this section we present the formulation that will be used as a basis for the use of Dantzig-Wolfe reformulation. The formulation we take as a basis is the echelon stock formulation proposed by Gruson et al. (2019) for the 3LSPD. For each period, the echelon stock of a specific facility represents the total inventory present at this facility and at all its

descendents. This concept has been used, among others, by Federgruen and Tzur (1999).

Let I_{pt}, I_{wt} and I_{rt} be the echelon stock of the production plant, the warehouses and the retailers in period t, respectively. Let x_{it} represents the production quantity in period t if $i = p$ and the quantity ordered from the predecessor if $i \in W \cup R$. Finally, let y_{it} be a boolean setup variable taking value 1 iff $x_{it} > 0$. The echelon stock formulation (ES) is as follows:

$$\text{Min} \sum_{t \in T} \left(\sum_{i \in F} sc_{it} y_{it} + \sum_{p \in P} hc_{pt} I_{pt} + \sum_{w \in W} \left(hc_{wt} - hc_{pt} \right) I_{wt} + \sum_{r \in R} \left(hc_{rt} - hc_{\delta_w(r)t} \right) I_{rt} \right) \tag{3.1}$$

$$\text{s.t. } I_{it} + x_{it} = d_{it} + I_{it} \qquad \forall\, t \in T, i \in F \tag{3.2}$$

$$x_{it} \leq D_{it} y_{it} \qquad \forall\, t \in T, i \in F \tag{3.3}$$

$$x_{pt} \leq \min\{C, D_{pt}\} y_{pt} \qquad \forall t \in T \tag{3.4}$$

$$I_{it} \geq \sum_{j \in \delta(i)} I_{jt} \qquad \forall\, t \in T, i \in P \cup W \tag{3.5}$$

$$x_{it}, I_{it} \geq 0 \qquad \forall\, t \in T, i \in F \tag{3.6}$$

$$y_{it} \in \{0,1\} \qquad \forall\, t \in T, i \in F. \tag{3.7}$$

The objective function (3.1) minimizes the sum of the fixed setup and replenishment costs and of the unit inventory holding costs. Constraints (3.2) are the inventory balance constraints using the echelon stock variables. Constraints (3.3) are the setup forcing constraints for all facilities. Constraints (3.4) are the capacity constraints at the production plant. Constraints (3.5) are the echelon stock constraints ensuring that the echelon stock at a specific facility is greater than the sum of the echelon stocks at all its direct successors. These constraints come from the non-negativity of the stock on hand at the end of any period. Finally, constraints (3.6)-(3.7) define the bounds and domains of the decision variables.

3.3.2 The reformulation

With the introduction of the echelon stock variables, the ES formulation presented before has the advantage of containing a SI-ULSP substructure in constraints (3.2), (3.3) and

a SI-CLSP substructure in constraints (3.2), (3.4). We want to exploit these substructures by applying a Dantzig-Wolfe decomposition to the ES formulation presented before. The Dantzig-Wolfe decomposition, proposed by Dantzig and Wolfe (1960), can be seen as a special variable redefinition where the original variables are replaced by a convex combination of the extreme points of several subsystems, in case the subsystems are bounded. In our case, the subsystems are the SI-CLSP for the production plant and one SI-ULSP for any other facility. Therefore, we face bounded polyhedrons because we have finite demand and no negative costs. Besides, the LP relaxation of the Dantzig-Wolfe reformulation has a better bound than the original LP relaxation, unless the subsystems have the integrality property.

Let X^i be the set of solutions to the SI-CLSP for the production plant or to the SI-ULSP for any other facility $i \in W \cup R$. Let also $conv(X^i)$ be its convex hull. We define $c = (x_i^c, I_i^c, y_i^c)$ to be an extreme point of $conv(X^i)$ where $x_i^c = \{x_{i1}^c, x_{i2}^c, ..., x_{i|T|}^c\}$, $I_i^c = \{I_{i1}^c, I_{i2}^c, ..., I_{i|T|}^c\}$ and $y_i^c = \{y_{i1}^c, y_{i2}^c, ..., y_{i|T|}^c\}$. With each extreme point of $conv(X^i)$, we associate a variable θ_i^c. If we denote by $C(i)$ the set of all extreme points of $conv(X^i)$, the Dantzig-Wolfe reformulation, which we denote as ES-DW, gives:

$$\text{Min} \sum_{t\in T}\left(\sum_{i\in F}\sum_{c\in C(i)} sc_{it} y_{it}^c \theta_i^c + \sum_{c\in C(p)} hc_{pt} I_{pt}^c \theta_p^c + \sum_{w\in W}\sum_{c\in C(w)} (hc_{wt} - hc_{pt})\, I_{wt}^c \theta_w^c \right.$$
$$\left. + \sum_{r\in R}\sum_{c\in C(r)} \left(hc_{rt} - hc_{\delta_w(r)t}\right) I_{rt}^c \theta_r^c \right) \quad (3.8)$$

$$\text{s.t.} \sum_{c\in C(i)} I_{it}^c \theta_i^c \geq \sum_{j\in\delta(i)}\sum_{c\in C(j)} I_{jt}^c \theta_j^c \qquad \forall\, t \in T, i \in P \cup W \quad (3.9)$$

$$\sum_{c\in C(i)} \theta_i^c = 1 \qquad \forall\, i \in F \quad (3.10)$$

$$y_{it} = \sum_{c\in C(i)} y_{it}^c \theta_i^c \qquad \forall\, t \in T, i \in F \quad (3.11)$$

$$x_{it} = \sum_{c\in C(i)} x_{it}^c \theta_i^c \qquad \forall\, t \in T, i \in F \quad (3.12)$$

$$I_{it} = \sum_{c\in C(i)} I_{it}^c \theta_i^c \qquad \forall\, t \in T, i \in F \quad (3.13)$$

$$y_{it} \in \{0,1\} \qquad \forall\, t \in T, i \in F \tag{3.14}$$

$$\theta_i^c \geq 0 \qquad \forall\, i \in F, c \in C(i). \tag{3.15}$$

Constraints (3.9) are the echelon stock constraints and (3.10) are the convexity constraints. Constraints (3.11)-(3.13) express the links between the extreme points of the subsystems and the original setup, production and echelon stock variables, respectively. Note that here the capacity requirements (3.4) are hidden in the θ_p^c variables for the production plant. We could have explicitly put the capacity requirements in the reformulation but initial experiments have shown that it is better to put them in the production plant subsystem. Note that when one wants to solve the LP relaxation of this problem, it is also possible to drop constraints (3.11)-(3.13).

The main challenge with this reformulation is that the sets $C(i)$ for any $i \in F$ may contain a huge number of extreme points, leading to a huge number of variables θ_i^c. The idea is therefore to use an iterative procedure which will allow us to work with a subset of the θ_i^c variables in a so called restricted master problem (RMP) and generate new ones as needed. This is done thanks to a column generation algorithm. The column generation procedure allows to generate dynamically the columns, i.e., the θ_i^c variables, whose reduced costs are negative, through the resolution of different subproblems. Each subproblem represents either the SI-CLSP for the production plant or a SI-ULSP for any other facility, and there is one column generated per subproblem. The column generation procedure stops when there are no more columns of negative reduced costs that can be generated for any subproblem.

Let $\delta_b(i)$ be the direct predecessor of facility $i \in W \cup R$. If we denote by π_{it} and λ^i the dual variables linked to constraints (3.9) and (3.10), respectively, a subproblem for facility $i \in W \cup R$ is defined by:

$$\text{Min} \sum_{t \in T} \left(sc_{it} y_{it} + \left(hc_{it} - hc_{\delta_b(i)t} + \pi_{\delta_b(i)t} - \pi_{it} \right) I_{it} \right) - \lambda^i \tag{3.16}$$

$$\text{s.t. } (3.2) - (3.3), (3.6) - (3.7).$$

Note that for each facility $i \in W \cup R$, only the constraints (3.2) - (3.3) and (3.6) - (3.7) for this specific facility apply. This subproblem can be efficiently solved in $O(|T|\log|T|)$ by a dynamic programming algorithm as described in Wagelmans et al. (1992).

For the production plant, the subproblem is defined by:

$$\text{Min} \sum_{t \in T} (sc_{it} y_{it} + (hc_{it} - \pi_{it}) I_{it}) - \lambda^i \tag{3.17}$$

$$\text{s.t. } (3.2) - (3.4), (3.6) - (3.7).$$

As we consider a constant capacity C available in each time period, this subproblem can be efficiently solved in $O(|T|^3)$ by a dynamic programming algorithm as described in van Hoesel and Wagelmans (1996).

3.4 A branch-and-price algorithm

This section presents the details of the branch-and-price algorithm we developed to solve the problem. The column generation procedure only allows us to solve the LP relaxation of the formulation ES-DW. Therefore, to fully solve the problem, we developed a branch-and-price algorithm. A branch-and-price algorithm is similar to a branch-and-bound algorithm in the sense that there is a tree with nodes whose LP relaxation is solved, and there are branching decisions linking the different nodes. In a branch-and-price algorithm, each node represents a restricted master problem whose LP relaxation is solved by a column generation procedure. This restricted master problem contains only the columns generated so far in the tree. Moreover, the branching decisions are transposed not only in the tree but also in the different subproblems. The remainder of this section gives details about the different elements of the branch-and-price algorithm we developed.

3.4.1 Initialization of the algorithm

To start the algorithm, we need some initial columns to feed the solver, i.e., we need some initial θ_i^c variables. The quality of the initial columns is an issue that has been raised

in the column generation literature, see in particular Lübbecke and Desrosiers (2005b). In our algorithm, we generate several initial columns depending on the subproblem considered. For the production plant, we generate a unique initial column where production takes place at full capacity C from period 1 up to period t' where t' is defined as $\lfloor d_{p1|T|}/C \rfloor$. We then impose production in period $t'+1$ and the quantity produced is $d_{p1|T|} - Ct'$. This ensures a feasible solution since the production capacity is never exceeded. Note that if $d_{p1|T|} = Ct'$ we do not impose any production nor setup in period $t'+1$.

For the other facilities $i \in W \cup R$, we generate two sets of initial columns. The first column is the Wagner-Within plan linked to the considered facility. To obtain this column, we solve the basic SI-ULSP linked to the facility by means of a dynamic programming algorithm as described in Wagelmans et al. (1992). The second column generated is obtained by ordering $d_{i1|T|}$ in the first period. This column is feasible for the warehouses and retailers, since they are not constrained by any ordering capacities, but not overall feasible, because of the production capacity requirements.

3.4.2 Branching decisions

In the branch-and-price tree, branching decisions must be made. These branching decisions are made on the original setup variables y_{it} to keep a more balanced search tree. Having a more balanced search tree has proven better for branch-and-price algorithms as stated in Barnhart et al. (1998) and Vanderbeck (2000). The branching decisions translate into slight changes in the different subproblems we solve. When the branching decision impose a setup, i.e., when we impose $y_{it} = 1$ for some facility i in some period t, we temporarily set the setup cost to 0: $sc_{it} = 0$. We then solve the subproblem as usual and obtain a column c representing an extreme point of the polytope $conv(X^i)$. We then check if there is actually production/an order placed in period t, i.e., we check if $y^c_{it} > 0$. If not, we enforce $y^c_{it} = 1$ and adjust the cost of the column. This way, we are not misleading the algorithm towards production/order in period t.

When the branching decision imposes that there must be no production/order placed in period t, we temporarily put a prohibitive setup cost: $sc_{it} = M$, where M is a very large number. This way, we ensure that there will never be any production/order placed in period t.

3.4.3 Improvements

Numerous authors have highlighted several main drawbacks of the column generation procedure, see Lübbecke and Desrosiers (2005a) for instance. The first drawback is the tailing-off effect which can be defined as the fact the lower and upper bounds slowly converge towards each other. The second main drawback is the big variations of the dual values π_{it} and λ^i during the iterations of the column generation procedure. Indeed, it has been observed that the dual variables do not converge smoothly towards their optimal values, see Lübbecke and Desrosiers (2005a). We added several improvements to our basic branch-and-price algorithm to tackle these two difficulties.

Lagrangian relaxation

We first tackle the tailing-off effect by adding columns with a different structure than the ones generated through the resolution of our basic subproblems. Indeed, if, in formulation ES, we relax the echelon-stock constraints (3.5) in a lagrangian fashion with positive multipliers α_{it}, we obtain the following model:

$$\text{Min} \sum_{t\in T}\left(\sum_{i\in F} sc_{it}y_{it} + \sum_{p\in P}(hc_{pt}-\alpha_{pt})I_{pt} + \sum_{w\in W}(\alpha_{pt}+hc_{wt}-hc_{pt}-\alpha_{wt})I_{wt}\right) + \sum_{t\in T}\left(\sum_{r\in R}\left(\alpha_{\delta_w(r)t}+hc_{rt}-hc_{\delta_w(r)t}\right)I_{rt}\right) \quad (3.18)$$

$$\text{s.t. } I_{i,t-1}+x_{it} = d_{it}+I_{it} \qquad \forall\, t\in T, i\in F \quad (3.19)$$

$$x_{it} \le D_{it}y_{it} \qquad \forall\, t\in T, i\in F \quad (3.20)$$

$$x_{pt} \le Cy_{it} \qquad \forall\, t\in T \quad (3.21)$$

$$x_{it}, I_{it} \geq 0 \qquad \forall\, t \in T, i \in F \tag{3.22}$$

$$y_{it} \in \{0,1\} \qquad \forall\, t \in T, i \in F. \tag{3.23}$$

In this formulation, we obtain the same subproblems as the ones we identified when we applied the Dantzig-Wolfe reformulation to the ES formulation. This property has been observed and successfully used in the column generation literature, see Huisman et al. (2005). It has also been shown that the optimal dual variables of the linking constraints in the master problem and the optimal lagrangian multipliers used to penalize the linking constraints are the same, see Magnanti et al. (1976). Therefore, if we use the lagrangian multipliers in our subproblems, we will likely obtain columns that have a different structure than the ones obtained with the use of the simplex dual values.

When we use lagrangian relaxation to generate new columns, we first solve the restricted master problem to obtain the dual variables π_{it} and λ^i. We then do several iterations of lagrangian relaxation to obtain the new columns. The lagrangian multipliers are initialized with the actual dual values, i.e., initially, $\alpha_{it} = \pi_{it}$. At each iteration, we update the multipliers using subgradient optimization, see Fisher (1985) for more information. This idea of using lagrangian relaxation in a branch-and-price algorithm has been successfully applied by Degraeve and Jans (2007) in the context of a multi-item CLSP with setup times. A sketch of the procedure used to generate new columns through lagrangian relaxation is given in Algorithm 1. In Algorithm 1, we define a limit on the maximum number of lagrangian iterations that we allow. We do this since we may add too many columns at the same time, thus increasing the size of the restricted master problem and the CPU time taken to solve it. We further have a gap tolerance as stopping criterion. If the gap between the upper bound and the lower bound is below a threshold ε, we stop the procedure.

Stabilization of the dual values

The second improvement we tried works on the stabilization of the dual values. Instead of using the actual values of the dual variables π_{it} and λ^i, we use a weighted sum

Algorithm 1 Lagrangian relaxation procedure to generate new columns

```
Solve the RMP and get dual values π_it
for 1 ≤ i ≤ |F| do
   Solve the subproblem i
   if column of negative reduced cost found then
      Initialize lagrange multipliers α_it with current dual values
      it = 0
      while it < limit and gap < ε do
         Solve subproblem i using lagrange multipliers α_it as dual values
         Update lagrange multipliers α_it by subgradient optimization
         it = it + 1
      end while
      Add columns with negative reduced cost to the RMP
   end if
end for
```

between these values and the values obtained in the previous iterations of the column generation procedure. A similar idea was successfully used in Wentges (1997) where the author uses a weighted sum between the current dual values and the dual values that led to the best lower bound obtained so far. The weights are computed based on the iteration number and on the number of improvements of the lower bound.

Let β be a weight linked to the actual dual values ($0 \leq \beta \leq 1$). If we denote by π_{itk} and λ_k^i the dual values obtained at iteration k of the column generation procedure, the values π'_{itk} and $\lambda_k'^i$ used for the dual variables are defined as:

$$\pi'_{itk} = \beta \pi_{itk} + (1-\beta)\, \pi'_{i,t,k-1} \qquad \forall\, i \in F,\ t \in T \tag{3.24}$$

$$\lambda_k'^i = \beta \lambda_k^i + (1-\beta)\, \lambda_{k-1}'^i \qquad \forall\, i \in F. \tag{3.25}$$

These values are used in the different subproblems we solve. Note that if we do not get a column of negative reduced cost with these values, we must use the actual dual values to check if no more columns of negative reduced costs can actually be generated. This ensures the validity of the column generation procedure.

Duplication of the columns

The third improvement is used to speed up the solution process in general. When we make branching decisions in the search tree, we adapt some existing columns. Indeed, some of the columns we have generated so far may fit with all branching decisions, except the last one. Therefore, in order not to generate columns that are very similar to the ones generated so far, we tried to duplicate existing columns to fit with the current branching decisions in the search tree. This idea has been proposed in the context of the CLSP with setup times by Degraeve and Jans (2007). Suppose that we impose a setup for facility i in period t, i.e., the branching decision gives $y_{it} = 1$. We duplicate the existing columns c for which $y_{it}^c = 0$ and denote by c' the copy of column c. In column c', we set $y_{it}^{c'} = 1$. We also adjust the cost of column c' to reflect this change. Note that when we enforce a setup in a duplicated column, we do not impose positive production or order quantities for the period whose setup value has been enforced. The duplication of the column is only done at the plant and warehouse level. Indeed, for these facilities there will be extreme points in the optimal solution that are most probably not Wagner-Whitin plans.

In initial experiments, this improvement did not give good results. We suspect that the reason behind this poor performance is the increase of the size of the RMP. We therefore discarded it from the results reported in Section 3.5.

Valid inequalitiy for the RMP

The last improvement we tried is the addition of valid inequalities for the RMP, based on the columns generated. We add one set of valid inequalities to improve the lower bound obtained throughout the tree:

$$(1 - y_{it}^c)\,\theta_i^c + y_{it} \leq 1 \qquad \forall\, i \in F, c \in C(i), t \in T. \tag{3.26}$$

Inequalities (3.26) indicate that if the column c is chosen and if $y_{it}^c = 0$, the setup variable y_{it} must be equal to 0. Inequalities (3.26) give new dual values that must be added to the objective function of each subproblem. In initial experiments, the addition of valid

inequalities did not give good results. The number of those inequalities was too big for the RMP to actually help improve the CPU time taken to solve the RMP. We therefore discarded it from the results reported in Section 3.5.

3.5 Numerical experiments

In order to assess the performance of our decomposition approach to solve the capacitated 3LSPD, we conducted numerical experiments based on the instances presented in Gruson et al. (2019). In the experiments, the number of retailers $|R|$ is set equal to 50, 100 or 200. The number of warehouses $|W|$ is set equal to 5, 10, 15 or 20. We used two different horizon lengths: $|T| = 15$ and 30. The demand at the retailers is generated both in a static and dynamic way from U[5, 100]. In the case of a static demand, we have $d_{rt} = d_r \, \forall \, t \in T, r \in R$. 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. For each combination of settings, we generate five different instances leading to 480 different instances to be solved for each formulation. We set the production capacity as a given factor C' of the average total demand. The production capacity imposed for any period t is thus $C = C' \sum_{i \in R} \sum_{t \in T} d_t^i \; / \; |T|$. The value of the capacity factor C' is set equal to 2. In the experiments, the computation time limit imposed to solve each instance is 6 hours.

We assess the performance of our decomposition approach with respect to different indicators:

— number of instances which are solved to optimality (%);

— CPU time (s) taken to solve the instance;

— CPU time (s) taken to solve the subproblems;

— best lower bound obtained during the search;

— objective function value of the MIP optimal solution when available, cost of the best solution found otherwise;

— number of nodes in the search tree;

— gap reported at the end of the time limit (%);

— gap compared to the solution given by CPLEX (%).

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 this particular version of the algorithm.

The following section gives the results for the capacitated instances. In the following tables, Opt denotes the percentage of MIP optimal solutions obtained; Time and Time-SP represent the CPU time taken to solve the MIP instances and the subproblems, respectively; BLB and BUB represent the best values found for the lower and upper bound, respectively; Gap gives the gap between the lower and upper bounds; C-gap indicates the gap with the best solution found by CPLEX; Nodes gives the number of nodes explored in the branch-and-price tree and Cols indicate the average number of columns generated.

The branch-and-price algorithm we developed was coded in C++ within the SCIP 5.0.1 framework, using CPLEX 12.8 as LP solver. To improve the efficiency of our algorithm, we impose that columns that have been non basic for the last ten iterations are removed from the RMP. We do so in order to reduce the size of the RMP which must be solved at each iteration. For the other improvements described in Section 3.4.3, the value of β, used in the stabilization of the dual values, is set equal to 0.25, 0.5 and 0.75. The maximum number of Lagrangian iterations done is set to 10.

The results obtained will be reported in two tables. The first table illustrates the aggregated results obtained for $|T| = 15$ (over all possible values for the capacity factor) while

Table 3.1 – Results obtained with the B&P algorithm for the capacitated instances, $|T| = 15$

Version	BLB	BUB	Time (s)	Time SP (s)	Nodes	Cols	Gap (%)	C-gap (%)	Opt (%)
B&P	525956	543295	15035	234.18	177439	21630	2.4	0.51	50
WS-0.25	536738	544933	14871	337.28	174373	22165	1.26	0.78	48
WS-0.5	536271	543531	14897	352.94	187586	22045	1.12	0.58	49
WS-0.75	535722	542793	14805	355.74	184283	21739	1.1	0.46	52
LR	536924	541362	15338	232.04	157629	23798	0.73	0.25	50
WS+LR-0.25	535976	542417	14904	344.84	178071	22462	1.02	0.43	48
WS+LR-0.5	536363	542832	14834	334.58	177975	23144	1.07	0.5	49
WS+LR-0.75	537059	542133	15294	364.49	158471	21918	0.84	0.38	48
CPX-ES	538875	539828	10367	-	46000	-	0.15	-	78

the second table displays the aggregated results obtained for $|T| = 30$. In each table, each row represents the results obtained for a particular version of the branch-and-price algorithm while each column refers to the different indicators previously defined. Regarding the different versions of the algorithm, the line B&P reports the results obtained for the basic version of the branch-and-price algorithm we developed. The lines WS-β give the results when we stabilize the dual values with different values of the parameter β. The line LR gives the results when we do some iterations of Lagrangian relaxation after solving the different subproblems. The lines WS+LR-β report the results obtained when we use both a stabilization of the dual values (with different values for the parameter β) and Lagrangian relaxation. Finally the line CPX-ES reports the results obtained by CPLEX directly.

Table 3.1 illustrates the performance of our branch-and-price algorithm for instances where the time horizon is set to 15. In Table 3.1, one can see that the number of optimal solutions found by our different algorithms is really low, at most 52%. This number is to be compared with the number of optimal solutions found using CPLEX only, which is 78%. Besides, the CPU time taken is also large, representing around 150% of the time taken by CPLEX to solve the different instances. Despite our wish to reduce the size of the RMP to speed up the solution process, it seems that the different versions of the branch-and-price algorithm spend too much time solving the RMP, as indicated by the difference between the total solving time and the time taken to solve the subproblems.

Table 3.2 – Results obtained with the B&P algorithm for the capacitated instances, $|T| = 30$

Version	BLB	BUB	Time (s)	Time-SP (s)	Nodes	Cols	Gap (%)	C-gap (%)	Opt (%)
B&P	1007693	1102475	21642	291.25	19609	78370	7.1	5.41	0
WS-0.25	1007520	1107808	21638	497.49	15627	65070	7.41	5.71	0
WS-0.5	1007760	1086964	21650	438.19	18407	73705	6.31	4.61	0
WS-0.75	1007849	1077942	21653	380.61	20631	79852	5.77	4.06	0
LR	1007558	1125106	21641	287.69	15387	77850	8.25	6.59	0
WS+LR-0.25	1007460	1101404	21648	419.75	16669	72067	7.09	5.38	0
WS+LR-0.5	1007587	1114474	21652	373.97	18324	75057	7.6	5.91	0
WS+LR-0.75	1007628	1102231	21644	378.16	13677	76066	7.12	5.43	0
CPX-ES	1010072	1034074	21576	-	19062	-	2.1	-	12.5

This is mostly explained by the number of calls to solve the RMP. Indeed, one can see that the number of nodes explored in the search tree is large, and there are several calls to solve the RMP at each node in the tree.

If we analyze the results obtained for our improvements, one can see that they lead to various results on our different indicators. Depending on the value used for the parameter β, the stabilization of the dual values leads to improvements or worsens the results compared to the basic branch-and-price algorithm: the higher the value of the parameter β, the better the results. Indeed, with a value of 0.75 for the parameter β, i.e., when we put more emphasis on the current dual values, we are able to find more optimal solutions, obtain a better optimality gap and spend less time solving the instances. The number of columns generated also decreases. The use of Lagragian relaxation also has positive impacts compared to the basic branch-and-price. It improves both bounds and reduces the different gaps. However, this is done at the cost of a slight increase in CPU time. Indeed, as we add much more columns per iterations, the size of the RMP is bigger and it is therefore harder to solve this RMP at each iteration. Finally, note that the combination of the stabilization of the dual values with Lagrangian relaxation has positive impacts on the gaps obtained, compared to the results of the use of stabilization alone. However, these improvements also come at the cost of an increase in the total CPU time taken to solve the instances.

Table 3.2 reports the aggregated results obtained for $|T| = 30$. In Table 3.2, one can

see that the size of the instances to be solved has a major impact on the performance of our different algorithms. First, the number of optimal solutions obtained is equal to zero for all versions. Then, the CPU time taken to solve the instances always reaches the CPU time limit imposed of 6 hours. Once again most of the time is spent on solving the RMP as stated by the low value of the time taken to solve the subproblems. A comparison of Tables 3.1 and 3.2 indicates that the number of nodes explored is much lower for the instances where $|T| = 30$. This shows that the RMP gets harder to solve and it also explains why the algorithm spends that much time solving the RMPs.

The gaps obtained are high, between 5.77 and 8.25%, and between 4.06 and 6.9% for the gap at the end of the time limit and gap compared to CPLEX, respectively. These values illustrate a relatively bad performance of our algorithms if we compare them to the gaps obtained by CPLEX alone. The better performance of CPLEX could be explained by the fact that CPLEX is able to exploit the structure of the entire problem to derive strong cuts during its search process. On the contrary, with our branch-and-price algorithm, we have a restricted number of variables available, thus limiting the possible exploitation of information.

As far as our improvements are concerned, we observe similar results for the use of dual value stabilization as in Table 3.1. However, the use of Lagrangian relaxation worsens the results compared to the basic branch-and-price algorithm. Indeed, as mentioned before, the RMP gets harder to solve. With the higher number of columns generated with Lagrangian relaxation, it also increases the size of the RMP, making it even harder to solve. This conclusion is also supported by the relatively low number of nodes explored with the use of Lagrangian relaxation. In the same spirit, the combination of Lagrangian relaxation and dual value stabilization worsens the performance of the use of dual value stabilization alone.

In light of the results presented in Tables 3.1 and 3.2, one can see that the use of a Dantzig-Wolfe decomposition is not efficient to solve the capacitated 3LSPD. The branch-and-price algorithm developed to solve the problem still suffers from the tailing-off effect,

even when we add several improvements. We were able to exploit a nice substructure to efficiently solve the different subproblems but it appears that it is not enough to develop an efficient branch-and-price algorithm. In particular, the master problem is still hard to solve as illustrated by the CPU time taken to solve the problem in Tables 3.1-3.2, compared to the CPU time taken to solve the subproblems. This happens despite our attempt to reduce the size of the master problem by only keeping in the master problem the columns that have been basic at least once in the last ten iterations. However, such attempt may lead to the generation of the same column several times, which would slow down the solution process. This hypothesis is supported by the large number of columns generated as reported in Tables 3.1-3.2. This high number of columns generated could also be explained by a low quality of information contained in the dual values. Finally, the number of nodes explored is also large compared to CPLEX. This indicates that the enhancements we proposed did not achieve the goal of reducing the tailing-off effect.

3.6 Conclusion

We have tackled the capacitated 3LSPD and have applied a Dantzig-Wolfe reformulation to the ES formulation of the problem. The use of such a reformulation allowed us to decompose the problem in one subproblem per facility, which then can be easily solved by means of a dynamic programming algorithm. Such a reformulation naturally led to the development of a branch-and-price algorithm, for which we also proposed several improvements. The first one uses Lagrangian relaxation to generate columns with a different structure during the search process, while the second one works on the stabilization of the dual values. We tried two other enhancements that did not prove useful in initial experiments.

We have performed numerous numerical experiments to assess the performance of our decomposition technique on the resolution of the problem. The use of a branch-and-price algorithm to solve the capacitated version of the 3LSPD led to poor results in terms of the number of optimal solutions found, of the quality of the bounds, and of the CPU time

taken to solve the different instances. This is mainly explained by the numerous calls to the RMP, illustrating the presence of a big tailing-off effect.

In future research we want to introduce scenarios of demand to represent the uncertainty that appears at the retailer's level in the uncapacitated case.

References

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

Barnhart, C., E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh and P. H. Vance. 1998, «Branch-and-price: Column generation for solving huge integer programs», *Operations Research*, vol. 46, p. 293–432.

Brahimi, N., N. Absi, S. Dauzère-Pérès and A. Nordli. 2017, «Single-item dynamic lot-sizing problems: An updated survey», *European Journal of Operational Research*, vol. 263, n° 3, p. 838–863.

Caserta, M. and S. Voß. 2013, «A math-heuristic Dantzig-Wolfe algorithm for capacitated lot sizing», *Annals of Mathematics and Artificial Intelligence*, vol. 69, p. 207–224.

Dantzig, G. B. and P. Wolfe. 1960, «Decomposition principle for linear programming», *Operations Research*, vol. 8, p. 101–111.

De Araujo, S. A., B. De Reyck, Z. Degraeve, I. Fragkos and R. Jans. 2015, «Period decompositions for the capacitated lot sizingproblem with setup times», *INFORMS Journal on Computing*, vol. 27, p. 431–448.

Degraeve, Z. and R. Jans. 2007, «A new Dantzig-Wolfe reformulation and branch-and-price algorithm for the capacitated lot-sizing problem with setup times», *Operations Research*, vol. 55, n° 5, p. 909–920.

Duarte, A. J. S. T. and J. M. V. V. de Carvalho. 2013, «A column generation approach to the discrete lot sizing and scheduling problem on parallel machines», in *Operational Research*, edited by J. Almeida, J. F. Oliviera and A. A. Pinto, Springer, p. 367–377.

Federgruen, A. and M. Tzur. 1999, «Time-partitioning heuristics: Application to one warehouse, multiitem, multiretailer lot-sizing problems», *Naval Research Logistics*, vol. 46, n° 5, p. 463–486.

Fisher, M. L. 1985, «An applications oriented guide to lagrangian relaxation», *Interfaces*, vol. 15, n° 2, p. 10–21.

Fragkos, I., Z. Degraeve and B. De Reyck. 2016, «A horizon decomposition approach for the capacitated lot-sizing problem with setup times», *INFORMS Journal on Computing*, vol. 28, p. 465–482.

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

van Hoesel, C. P. M. and A. P. M. Wagelmans. 1996, «An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities», *Management Science*, vol. 42, n° 1, p. 142–150.

Huisman, D., R. Jans, M. Peeters and A. P. M. Wagelamns. 2005, «Combining column generation and lagrangian relaxation», in *Column Generation*, edited by G. Desaulniers, J. Desrosiers and M. M. Solomon, chap. 9, Springer, New York, p. 247–270.

Jans, R. and Z. Degraeve. 2006, «Modeling industrial lot sizing problems: a review», *International Journal of Production Research*, vol. 46, p. 1619–1643.

Karimi, B., G. Fatemi and J. Wilson. 2003, «The capacitated lot sizing problem: a review of models and algorithms», *Omega*, vol. 31, p. 365–378.

Le, T., A. Diabat, J.-P. Richard and Y. Yih. 2013, «A column generation-based heuristic algorithm for an inventory routing problem with perishable goods», *Optimization Letters*, vol. 7, p. 1481–1502.

Lübbecke, M. E. and J. Desrosiers. 2005a, «A primer in column generation», in *Column Generation*, edited by G. Desaulniers, J. Desrosiers and M. M. Solomon, chap. 1, Springer, New York, p. 1–32.

Lübbecke, M. E. and J. Desrosiers. 2005b, «Selected topics in column generation», *Operations Research*, vol. 53, n° 6, p. 1007–1023.

Magnanti, T. L., J. F. Shapiro and M. H. Wagner. 1976, «Generalized linear programming solves the dual», *Management Science*, vol. 22, p. 1195–1203.

Manne, A. S. 1958, «Programming of the economic lot sizes», *Management Science*, vol. 4, p. 115–135.

Michel, S. and F. Vanderbeck. 2012, «A column-generation based tactical planning method for inventory routing», *Operations Research*, vol. 60, p. 382–397.

Mourgaya, M. and F. Vanderbeck. 2007, «Column generation based heuristic for tactical planning in multi-period vehicle routing», *European Journal of Operational Research*, vol. 183, p. 1028–1041.

Pimentel, C. M. O., F. P. Alvelos and J. M. Valèrio de Carvalho. 2010, «Comparing Dantzig-Wolfe decompositions and branch-and-price algorithms for the multi-item capacitated lot sizing problem», *Optimization Methods & Software*, vol. 25, p. 299–319.

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

Tempelmeier, H. 2011, «A column generation heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint», *Omega*, vol. 39, p. 627–633.

Vanderbeck, F. 2000, «On Dantzig–Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm», *Operations Research*, vol. 48, p. 111–412 832.

Wagelmans, A., S. van Hoesel and A. Kolen. 1992, «Economic lot sizing: An o(n log n) algorithm that runs in linear time in the wagner-whitin case», *Operations Research*, vol. 40, nº 1-supplement-1, p. S145–S156.

Wagner, H. M. and T. M. Whitin. 1958, «Dynamic version of the economic lot size model», *Management Science*, vol. 5, p. 89–96.

Wentges, P. 1997, «Weighted Dantzig-Wolfe decomposition for linear mixed-integer programming», *International Transactions in Operational Research*, vol. 4, p. 151–162.

Chapitre 4

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

Information sur le chapitre

Un rapport de recherche fondé sur ce chapitre a été publié dans la série des Cahiers du GERAD : Gruson, M., Cordeau, J.-F., et Jans, R. (2019). Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure. *Les Cahiers du GERAD*, G-2019-51. Ce chapitre a également mené à la publication d'un article dans la revue *European Journal of Operational Research* début septembre 2020.

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 also study an extension where we allow for lost sales at the retailer level. 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 the enhancements. The Benders-based branch-and-cut algorithm we propose clearly outperforms CPLEX.

4.1 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 (2001), Zhang and Song (2018) and Abdullah et al. (2019).

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. (2015), if these fore-

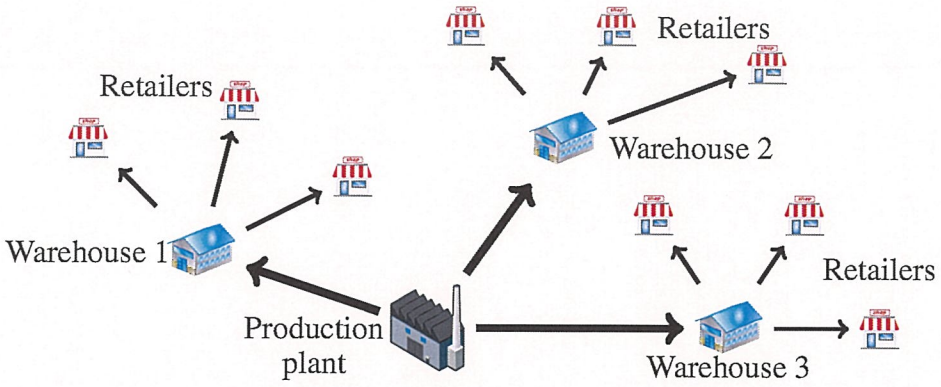

Figure 4.1 – Graphical representation of the problem considered

casts 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 4.1 illustrates this flow of goods in a distribution network composed by one production plant, three warehouses and three retailers linked to each warehouse.

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. Given a finite set T of time periods, indexed by t, we denote by sc_{it} the

setup costs at any facility i belonging to any of the three levels. They represent 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 Bookbinder and Tan (1988) 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 in the literature. The interested reader is referred to Brahimi et al. (2017) and to Pochet and Wolsey (2006) 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 (2013) and Aloulou et al. (2014) for reviews on stochastic lot sizing.

Our paper makes four main contributions. First, we extend the work of Gruson et al. (2019a) by studying a stochastic version of the three-level lot sizing problem with a distribution structure (3LSPD). We also propose an extension where we allow for lost sales at the retailer level. 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 4.2. Then, we give a formal problem description in Section 4.3 along with one mathematical formulation for the problem. This formulation is used as a basis for the application of a Benders decomposition in Section 4.4. In this section, we further present a Benders-based branch-and-cut algorithm along with computational enhancements. Section 4.5 details the computational experiments performed to assess the performance of the algorithm we developed, and to analyze the impact of the enhancements we proposed. This is followed by the conclusion in Section 4.6.

4.2 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.

4.2.1 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 (1988): 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 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. (2001), who study the stochastic single item lot sizing problem with backlogging and embed the progressive hedging algorithm of Rockafellar and Wets (1991) in a metaheuristic. Gutiérrez et al. (2004) 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 Jr (2010) 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. (2013) 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. (2018) 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. (2006), still with demand scenarios. They adapt the (l,S) valid inequalities to the stochastic case and call them $(\mathscr{Q},S_{\mathscr{Q}})$ valid inequalities. They further establish necessary and sufficient conditions for these $(\mathscr{Q},S_{\mathscr{Q}})$ inequalities to be facet defining. These inequalities have been later simplified by Di Summa and Wolsey (2008) 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 (2007) for a comparison of various service level constraints and their implications in a stochastic setting and to Gruson et al. (2018) for a discussion of inventory service levels in deterministic lot sizing problems.

4.2.2 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 Pochet and Wolsey (2006): 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 (1994) propose a general heuristic for the multi-item multi-level capacitated lot sizing problem which solves a series of CLSPs using a modified Dixon-Silver heuristic (see Dixon and Silver (1981)). They propose four variants of this heuristic where the differences come from the ordering at the different levels. Sahling

et al. (2009) propose a fix-and-optimize heuristic to solve this problem while also considering setup carry-over. Their heuristic sequentially solves a series of small mixed-integer programs and use the solutions to fix a large proportion of the integer variables to a specific value in the next iterations. Chen (2015) also uses a fix-and-optimize heuristic to solve a multi-level CLSP with or without setup carryover. In the case without setup carryover, a variable neighbourhood search is used. Furlan and Santos (2017) propose a bees-and-fix-and optimize algorithm to solve a multi-level CLSP. The bees algorithm helps partition the set of decision variables to be fixed. More recently, Wei et al. (2019) study a case where the bill of materials can be replaced. They propose a matching-induced search algorithm to solve the problem. For studies on specific applications of multi-level LSPs, the interested reader is referred to the references in Wei et al. (2019).

4.2.3 Benders decomposition in lot sizing

The use of Benders decomposition in the lot sizing literature is very scarce. Bahl and Zionts (1987) 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. (2018) 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. (2015) propose a Benders decomposition algorithm to tackle a combined production planning and routing problem with demand uncertainty. Caserta and Voß (2020) propose an accelerating technique, called the corridor method, which is used in the Benders decomposition algorithm. This method consists in solving the master problem around the incumbent solution. They observe improvements compared to the classical Benders decomposition, when applied to a multi-item CLSP.

Table 4.1 – Sets, parameters and decision variables used in the mathematical model

Sets		**Parameters**	
P	set containing the unique production plant, $P = \{p\} \subset F$	$W(r)$	warehouse linked to the retailer $r \in R$
W	set containing the warehouses, $W \subset F$	d_{rt}	demand for retailer r in period t
R	set containing the retailers, $R \subset F$	δ_{kt}	Kronecker delta that takes the value 1 if $k = t$ and 0 otherwise
$S(i)$	set of all direct successors of facility i		
Decision variables			
x_{kt}^{lr}	quantities produced or ordered in level l in period k to satisfy d_{rt}		
σ_{kt}^{lr}	stock at level l at the end of period k to satisfy d_{rt}		
y_{it}	1 iff there is production or an order placed by facility i in period t		

4.3 Mathematical formulation for the 2S-3LSPD

In this section we first present a stochastic programming model for the 2S-3LSPD. We then present a scenario-based reformulation 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. Table 4.1 lists all the sets, parameters and decision variables used which have not been defined earlier. In the stochastic model, the demand $\tilde{d}_{rt}$ is a random variable. The two-stage stochastic programming model is given as follows:

$$\text{Min} \sum_{t \in T} \sum_{i \in F} sc_{it} y_{it} + E_{\tilde{d}} \left[Q(y, \tilde{d}) \right] \tag{4.1}$$

$$\text{s.t. } y_{it} \in \{0,1\} \forall\, t \in T, i \in F, \tag{4.2}$$

where, for a specific realization d of $\tilde{d}$, $Q(y,d)$ is the optimal value of the following second stage problem:

$$\text{Min} \sum_{t \in T} \sum_{r \in R} \sum_{k \leq t} hc_{pk} \sigma_{kt}^{0r} + hc_{W(r)k} \sigma_{kt}^{1r} + hc_{rk} \sigma_{kt}^{2r} \tag{4.3}$$

$$\text{s. t. } x_{kt}^{1r} + \sigma_{kt}^{0r} = \sigma_{k-1,t}^{0r} + x_{kt}^{0r} \qquad \forall\, t \in T, k \leq t \in T, r \in R \tag{4.4}$$

$$x_{kt}^{2r} + \sigma_{kt}^{1r} = \sigma_{k-1,t}^{1r} + x_{kt}^{1r} \qquad \forall\, t \in T, k \leq t \in T, r \in R \tag{4.5}$$

$$\delta_{kt} d_{rt} + (1 - \delta_{kt}) \sigma_{kt}^{2r} = \sigma_{k-1,t}^{2r} + x_{kt\omega}^{2r} \qquad \forall\, t \in T, k \leq t \in T, r \in R \tag{4.6}$$

$$x_{kt}^{0r} \leq d_{rt} y_{pk} \qquad \forall\, t \in T, k \leq t \in T, r \in R \tag{4.7}$$

$$x_{kt}^{1r} \leq d_{rt} y_{W(r)k} \qquad \forall\, t \in T, k \leq t \in T, r \in R \tag{4.8}$$

$$x_{kt}^{2r} \leq d_{rt} y_{rk} \qquad \forall\, t \in T, k \leq t \in T, r \in R \tag{4.9}$$

$$x^{0r}_{kt}, x^{1r}_{kt}, x^{2r}_{kt}, \sigma^{0r}_{kt}, \sigma^{1r}_{kt}, \sigma^{2r}_{kt} \geq 0 \qquad \forall\, t \in T, k \leq t \in T, r \in R. \tag{4.10}$$

The objective function (4.1) minimizes, for the first stage problem, the sum of the setup costs and of the expected inventory holding costs at each facility with respect to the random demand $\tilde{d}$. The second stage problem (4.3)-(4.10) is based on the multi-commodity (MC) formulation proposed by Melo and Wolsey (2010) for a two-level lot sizing problem, and later extended by Gruson et al. (2019a) to the deterministic 3LSPD. The idea of this formulation is to disaggregate distinct commodities d_{rt}. In more detail, the decision variables used describe the flow of each commodity in the supply chain, both in terms of time and space. The objective function (4.3) minimizes, for the second stage problem, the total inventory holding costs for a particular realization of the random demand. Constraints (4.4)-(4.6) represent the inventory balance equations at the production plant, the warehouse and the retailer level, respectively. Constraints (4.7)-(4.9) are the setup forcing constraints for the production plant, the warehouses and the retailers, respectively.

Because it contains random variables, the two-stage stochastic model (4.1)-(4.2) is intractable. To make it tractable, we assume that there is a finite number of possible demand scenarios. We denote by Ω the set of all possible scenarios. Let p_ω be the probability of realization of scenario ω and let $d_{rt\omega}$ be the demand of retailer r in period t under scenario ω. Compared to the above model, we add a subscript ω to the second stage variables, i.e., the production and inventory variables x and σ. The MC formulation for the 2S-3LSDP is as follows:

$$\text{Min} \sum_{t \in T} \left(\sum_{i \in F} sc_{it} y_{it} + \sum_{\omega \in \Omega} p_\omega \sum_{r \in R} \sum_{k \leq t} \left(hc_{pk} \sigma^{0r}_{kt\omega} + hc_{W(r)k} \sigma^{1r}_{kt\omega} + hc_{rk} \sigma^{2r}_{kt\omega} \right) \right) \tag{4.11}$$

$$x^{1r}_{kt\omega} + \sigma^{0r}_{kt\omega} = \sigma^{0r}_{k-1,t,\omega} + x^{0r}_{kt\omega} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{4.12}$$

$$x^{2r}_{kt\omega} + \sigma^{1r}_{kt\omega} = \sigma^{1r}_{k-1,t,\omega} + x^{1r}_{kt\omega} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{4.13}$$

$$\delta_{kt} d_{rt\omega} + (1 - \delta_{kt}) \sigma^{2r}_{kt\omega} = \sigma^{2r}_{k-1,t,\omega} + x^{2r}_{kt\omega} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{4.14}$$

$$x^{0r}_{kt\omega} \leq d_{rt\omega} y_{pk} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \qquad (4.15)$$

$$x^{1r}_{kt\omega} \leq d_{rt\omega} y_{W(r)k} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \qquad (4.16)$$

$$x^{2r}_{kt\omega} \leq d_{rt\omega} y_{rk} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \qquad (4.17)$$

$$x^{0r}_{kt\omega}, x^{1r}_{kt\omega}, x^{2r}_{kt\omega} \geq 0 \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \qquad (4.18)$$

$$\sigma^{0r}_{kt\omega}, \sigma^{1r}_{kt\omega}, \sigma^{2r}_{kt\omega} \geq 0 \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \qquad (4.19)$$

$$y_{it} \in \{0,1\} \qquad \forall\, t \in T, i \in F. \qquad (4.20)$$

This formulation can be slightly modified to allow lost sales. Let $ls_{rt\omega}$ be a positive and continuous variable that represents the amount of lost sales among $d_{rt\omega}$. This amount is penalized in the objective function by lsc_{rt}, a unit penalty cost. Besides, the inventory balance constraint (4.14) will be replaced by:

$$\sigma^{2r}_{k-1,t,\omega} + x^{2r}_{kt\omega} = \delta_{kt}\,(d_{rt\omega} - ls_{rt\omega}) + (1 - \delta_{kt})\sigma^{2r}_{kt\omega} \quad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega. \qquad (4.21)$$

4.4 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.

4.4.1 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 (see, Benders, 1962) 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 (4.12)-(4.18). For a recent review on Benders decomposition, the reader is referred to Rahmaniani et al. (2017).

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. In the following formulation, the dual variables linked to each constraint have been put into brackets. The primal subproblem PSP is defined by:

$$\text{Min} \sum_{\omega\in\Omega} p_\omega \sum_{t\in T}\sum_{r\in R}\left(\sum_{k\leq t} hc_{pk}\sigma^{0r}_{kt\omega} + \sum_{k\leq t} hc_{W(r)k}\sigma^{1r}_{kt\omega} + \sum_{k\leq t} hc_{rk}\sigma^{2r}_{kt\omega}\right) \tag{4.22}$$

$$\text{s. t. } x^{1r}_{kt\omega} + \sigma^{0r}_{kt\omega} = \sigma^{0r}_{k-1,t,\omega} + x^{0r}_{kt\omega} \quad [\psi^{0r}_{kt\omega}] \quad \forall\, t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{4.23}$$

$$x^{2r}_{kt\omega} + \sigma^{1r}_{kt\omega} = \sigma^{1r}_{k-1,t,\omega} + x^{1r}_{kt\omega} \quad [\psi^{1r}_{kt\omega}] \quad \forall\, t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{4.24}$$

$$\delta_{kt} d_{rt\omega} + (1-\delta_{kt})\sigma^{2r}_{kt\omega} = \sigma^{2r}_{k-1,t,\omega} + x^{2r}_{kt\omega} \quad [\psi^{2r}_{kt\omega}] \quad \forall\, t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{4.25}$$

$$x^{0r}_{kt\omega} \leq d_{rt\omega}\widehat{y}_{pk} \quad [\phi^{0r}_{kt\omega}] \quad \forall\, t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{4.26}$$

$$x^{1r}_{kt\omega} \leq d_{rt\omega}\widehat{y}_{W(r)k} \quad [\phi^{1r}_{kt\omega}] \quad \forall\, t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{4.27}$$

$$x^{2r}_{kt\omega} \leq d_{rt\omega}\widehat{y}_{rk} \quad [\phi^{2r}_{kt\omega}] \quad \forall\, t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{4.28}$$

$$x^{0r}_{kt\omega}, x^{1r}_{kt\omega}, x^{2r}_{kt\omega} \geq 0 \quad \forall\, t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{4.29}$$

$$\sigma^{0r}_{kt\omega}, \sigma^{1r}_{kt\omega}, \sigma^{2r}_{kt\omega} \geq 0 \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega. \tag{4.30}$$

The dual subproblem DSP corresponding to PSP is given as follows:

$$\text{Max} \sum_{\omega\in\Omega}\sum_{t\in T}\sum_{r\in R}\left(d_{rt\omega}\psi^{2r}_{tt\omega} - \sum_{k\leq t} d_{rt\omega}\left(\widehat{y}_{pk}\phi^{0r}_{kt\omega} + \widehat{y}_{W(r)k}\phi^{1r}_{kt\omega} + \widehat{y}_{rk}\phi^{2r}_{kt\omega}\right)\right) \tag{4.31}$$

$$\text{s. t. } \psi^{0r}_{k+1,t,\omega} - \psi^{0r}_{kt\omega} \leq p_\omega hc_{pk} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{4.32}$$

$$\psi^{1r}_{k+1,t,\omega} - \psi^{1r}_{kt\omega} \leq p_\omega hc_{W(r)k} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{4.33}$$

$$\psi^{2r}_{k+1,t,\omega} - (1-\delta_{kt})\psi^{2r}_{kt\omega} \leq p_\omega hc_{rk} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{4.34}$$

$$\psi^{0r}_{kt\omega} - \phi^{0r}_{kt\omega} \leq 0 \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{4.35}$$

$$\psi^{1r}_{kt\omega} - \psi^{0r}_{kt\omega} - \phi^{1r}_{kt\omega} \leq 0 \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{4.36}$$

$$\psi^{2r}_{kt\omega} - \psi^{1r}_{kt\omega} - \phi^{2r}_{kt\omega} \leq 0 \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \tag{4.37}$$

$$\phi^{0r}_{kt\omega}, \phi^{1r}_{kt\omega}, \phi^{2r}_{kt\omega} \geq 0 \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega. \tag{4.38}$$

Constraints (4.32)-(4.34) are the constraints linked to the original stock variables σ. Constraints (4.35)-(4.37) are the constraints linked to the original production and ordering variables x.

The DSP can be decomposed into $|R||T||\Omega|$ subproblems, denoted by $DSP_{rt\omega}$, one for each commodity $d_{rt\omega}$. Note that if we consider a multi-item setting, we still have this separability and the algorithm presented in Section 4.4.2 is still valid. In that case, a commodity would be the demand of a particular retailer for a specific item, in a certain time period, under a specific scenario. On the contrary, if we include production capacity constraints, we lose the separability. Indeed, such capacity constraints would link the ordering variables for each commodity. In that case, the solution method proposed in Section 4.4.2 is not valid anymore.

Let $\Delta_{SP}(r,t,\omega)$ represent the polyhedron defined by constraints (4.32)-(4.38) for 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 $d_{rt\omega}$, i.e., $\Delta_{SP}(r,t,\omega)$. The Benders reformulation, denoted as BD-MC, is defined by:

$$\text{Min} \sum_{\omega\in\Omega}\sum_{r\in R}\sum_{t\in T} z_{rt\omega} + \sum_{t\in T}\sum_{i\in F} sc_{it}y_{it} \tag{4.39}$$

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

$$\forall\, t\in T, r\in R, \omega\in\Omega,\ \forall\ \left(\phi^{0r}_{kt\omega}, \phi^{1r}_{kt\omega}, \phi^{2r}_{kt\omega}, \psi^{0r}_{kt\omega}, \psi^{1r}_{kt\omega}, \psi^{2r}_{kt\omega}\right) \in \Delta_{SP}(r,t,\omega) \tag{4.40}$$

$$y_{it}\in\{0,1\}\ \forall\, t\in T, i\in F. \tag{4.41}$$

The objective function (4.39) minimizes the sum of the setup costs and of the lower bound on the cost of the subproblems. Constraints (4.40) are the optimality cuts for each subproblem.

4.4.2 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 (4.40) 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 an optimality cut if the dual subproblem is feasible, or a feasibility cut if 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, both a lower and an upper bound can be derived from the solutions obtained for the master problem and the subproblems. Note

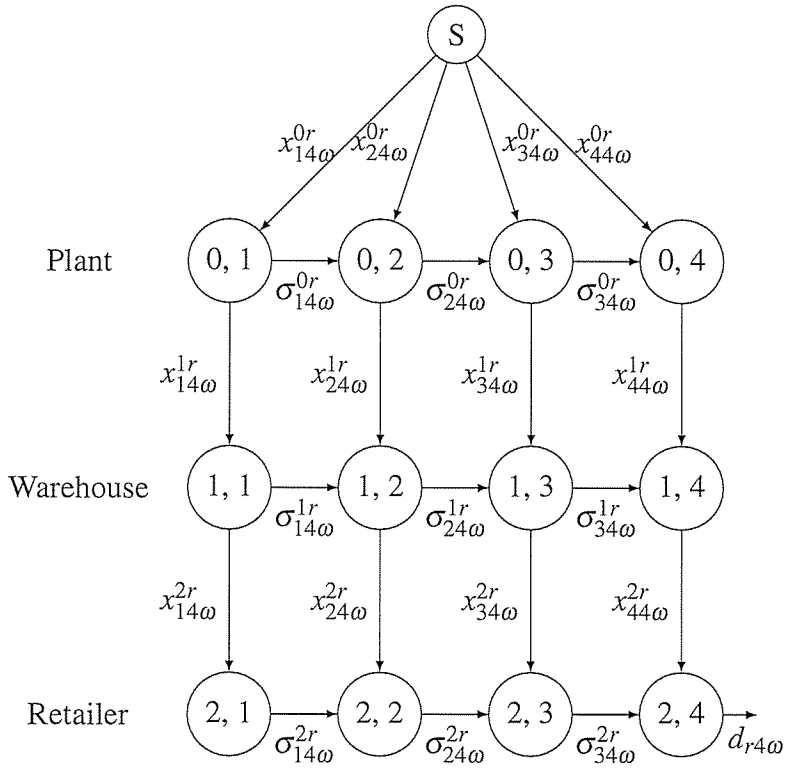

Figure 4.2 – Graphical representation of one subproblem ($t = 4$)

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 (4.23)-(4.25) for each commodity $d_{rt\omega}$. This substructure is illustrated in Figure 4.2, where we consider an example for which $t = 4$, one retailer r and one scenario ω.

In Figure 4.2, 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 4.2 we consider 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 still solve a shortest path problem 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 4.2 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 go from the source node to this particular node (see Ahuja et al. (1993)). These dual values correspond to the optimal values of the ψ variables in DSP. Using the structure of (4.35)-(4.37), the other dual values related to subproblem $DSP_{rt\omega}$ are computed as follows:

$$\phi^{0r}_{kt\omega} = \psi^{0r}_{kt\omega} \qquad \forall\, k \leq t \in T \tag{4.42}$$

$$\phi^{1r}_{kt\omega} = \max\{\psi^{1r}_{kt\omega} - \psi^{0r}_{kt\omega}; 0\} \qquad \forall\, k \leq t \in T \tag{4.43}$$

$$\phi^{2r}_{kt\omega} = \max\{\psi^{2r}_{kt\omega} - \psi^{1r}_{kt\omega}; 0\} \qquad \forall\, k \leq t \in T. \tag{4.44}$$

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 4.2. 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 $O(|R||T|^2|\Omega|)$.

4.4.3 A Benders-based branch-and-cut algorithm

It is well known that the optimality cuts (4.40) can be generated from any solution and not only from an optimal integer solution to the master problem. Therefore, we 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. (2015).

4.4.4 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.

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. (2015) 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:

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

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:

$$\sum_{v=1}^{t-1} \mu_{ivt} = y_{it} \qquad \forall\, i \in F,\ 2 \leq t \in T \tag{4.45}$$

$$\sum_{v=1}^{t-1} \sum_{s=t}^{|T|+1} \mu_{ivs} = 1 \qquad \forall\, i \in F,\ 2 \leq t \in T \tag{4.46}$$

$$\sum_{i\in F} \sum_{t=2}^{|T|+1} \sum_{v=1}^{t-1} c_{ivt} \mu_{ivt} \leq \sum_{\omega\in\Omega} \sum_{r\in R} \sum_{t\in T} z_{rt\omega}. \tag{4.47}$$

Constraints (4.45) link the new μ variables to the original binary setup variables. Constraints (4.46) indicate that there must be one replenishment plan chosen for each period. Finally, constraints (4.47) give a lower bound on the sum of the artificial variable $z_{rt\omega}$ for the master problem.

Optimality cuts based on fractional solutions

In the initial iterations of the Benders algorithm, there are too few optimality cuts (4.40) 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. The fractional solution obtained for the master problem is passed to the different

subproblems $DSP_{rt\omega}$ and we 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 4.4.2, 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 4.2. 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., $hc_{pt} \leq hc_{W(r)t} \leq hc_{rt}$ 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. (1993)). 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

Algorithm 2 Solution of the min cost flow problem for retailer r in period t under scenario w

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

warehouse level we must find an available path to obtain it at the production plant level. The complete procedure is given in Algorithm 2, 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.

An example of a solution obtained by Algorithm 2 is given in Figure 4.3 for a retailer r under scenario ω and for $t = 4$. In Figure 4.3, 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

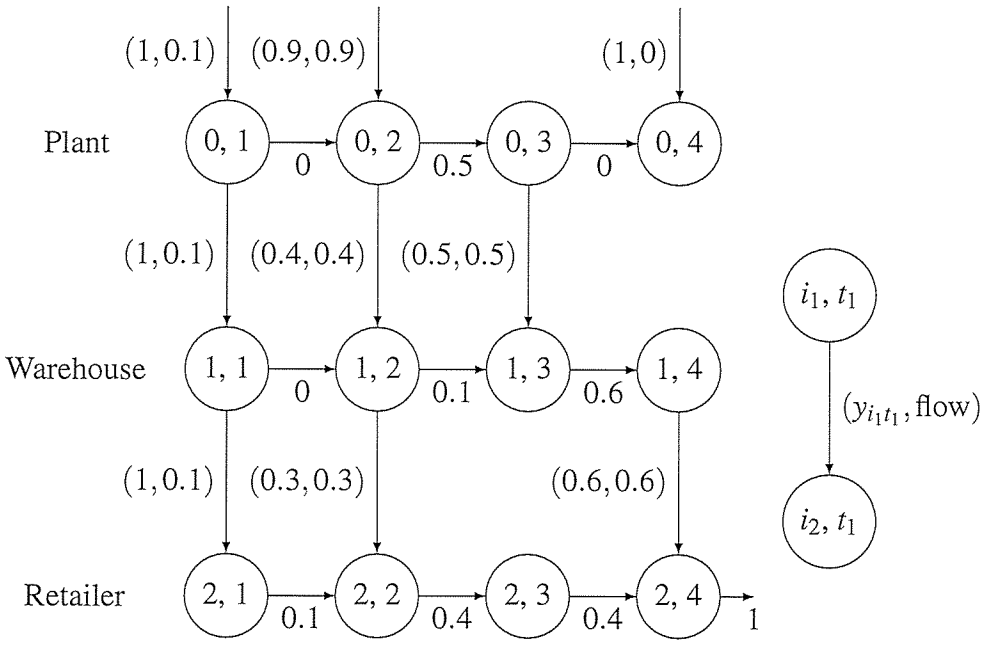

Figure 4.3 – Graphical representation of the solution procedure for one subproblem

the ψ variables for $DSP_{rt\omega}$. For ease of representation, we have assumed a demand of one unit in Figure 4.3.

Addition of MIR inequalities

Bodur and Luedtke (2016) mention that Benders decomposition usually does not take advantage of the integrality requirements on the master problem variables when deriving Benders cuts. They use 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 (1998). They assume that there is a current set $\mathscr{V}_{rt\omega}$ 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 $\mathscr{V}_{rt\omega}$. In more detail, let us

assume that the subproblem corresponding to commodity $d_{rt\omega}$ has generated Benders optimality cuts of the form:

$$z_{rt\omega} \geq b_{rt\omega} - \sum_{i \in F} a_{it\omega} y_{it}, \tag{4.48}$$

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 (4.48), we derive the following valid inequality for BD-MC:

$$z_{rt\omega} \geq c^0_{rt\omega} - \sum_{i \in F} c^1_{it\omega} y_{it}, \tag{4.49}$$

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 (4.49). If we denote by $c = (c^0, c^1)$, this scaled violation is defined as

$$\frac{\max\{c^0_{rt\omega} - \sum_{i \in F} c^1_{it\omega} \widehat{y}_{it} - \widehat{z}_{rt\omega}, 0\}}{\| (1, c) \|_2}. \tag{4.50}$$

We finally add to the master problem the valid inequality of the form (4.49) that yields the largest scaled violation. In our experiments, we apply this MIR procedure only at the root node.

Pareto-optimal cuts

When using Benders decomposition, the primal subproblem can be highly degenerate, leading to numerous possible optimal dual solutions, each defining a different Benders cut. This drawback has been first observed by Magnanti and Wong (1981). 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 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 (4.31), 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:

$$\text{Max} \sum_{\omega\in\Omega}\sum_{t\in T}\sum_{r\in R}\left(d_{rt\omega}\psi^{2r}_{tt\omega} - \sum_{k\leq t} d_{rt\omega}\left(y^0_{pk}\phi^{0r}_{kt\omega} + y^0_{W(r)k}\phi^{1r}_{kt\omega} + y^0_{rk}\phi^{2r}_{kt\omega}\right)\right) \tag{4.51}$$

$$\text{s. t. } (4.32)-(4.38) \tag{4.52}$$

$$\sum_{\omega\in\Omega}\sum_{t\in T}\sum_{r\in R}\left(d_{rt\omega}\psi^{2r}_{tt\omega} - \sum_{k\leq t} d_{rt\omega}\left(\widehat{y}_{pk}\phi^{0r}_{kt\omega} + \widehat{y}_{W(r)k}\phi^{1r}_{kt\omega} + \widehat{y}_{rk}\phi^{2r}_{kt\omega}\right)\right) = DSP^*. \tag{4.53}$$

The main drawback is that constraint (4.53) breaks the separability that we originally had in the DSP. In the case of network design problems, Magnanti et al. (1986) 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 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 (4.32)-(4.34), (4.35)-(4.37) and (4.53), respectively. We consider the dual of the auxiliary problem given by:

$$\text{Min} \sum_{\omega\in\Omega} p_\omega \sum_{t\in T}\sum_{r\in R}\sum_{k\leq t}\left(hc_{pk}\sigma^{0r}_{kt\omega} + hc_{W(r)k}\sigma^{1r}_{kt\omega} + hc_{rk}\sigma^{2r}_{kt\omega}\right) - DSP^*\mu \tag{4.54}$$

$$\text{s. t. } \sigma^{0r}_{k-1,t,\omega} + x^{0r}_{kt\omega} = x^{1r}_{kt\omega} + \sigma^{0r}_{kt\omega} \qquad \forall\, t\in T, k\leq t\in T, r\in R, \omega\in\Omega \tag{4.55}$$

$$\sigma^{1r}_{k-1,t,\omega} + x^{1r}_{kt\omega} = x^{2r}_{kt\omega} + \sigma^{1r}_{kt\omega} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \quad (4.56)$$

$$\sigma^{2r}_{k-1,t,\omega} + x^{2r}_{kt\omega} = \delta_{kt} d_{rt\omega}(1+\mu) \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega$$
$$+ (1-\delta_{kt})\sigma^{2r}_{kt\omega} \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \quad (4.57)$$

$$x^{0r}_{kt\omega} \leq d_{rt\omega}\left(y^0_{pk} + \widehat{y}_{pk}\mu\right) \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \quad (4.58)$$

$$x^{1r}_{kt\omega} \leq d_{rt\omega}\left(y^0_{W(r)k} + \widehat{y}_{W(r)k}\mu\right) \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \quad (4.59)$$

$$x^{2r}_{kt\omega} \leq d_{rt\omega}\left(y^0_{rk} + \widehat{y}_{rk}\mu\right) \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \quad (4.60)$$

$$x^{0r}_{kt\omega}, x^{1r}_{kt\omega}, x^{2r}_{kt\omega} \geq 0 \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega \quad (4.61)$$

$$\sigma^{0r}_{kt\omega}, \sigma^{1r}_{kt\omega}, \sigma^{2r}_{kt\omega} \geq 0 \qquad \forall\, t \in T, k \leq t \in T, r \in R, \omega \in \Omega. \quad (4.62)$$

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 (4.57) and the capacities on the arcs are given by (4.58)-(4.60). In Magnanti et al. (1986), 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 2.

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 (1981). 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 (2008). In the second strategy, for any facility i, the core point is initialized to $y^0_{it} = 1$ and dynamically updated as $y^0_{it} = 0.5y^0_{it} + 0.5\widehat{y}_{it}$.

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 subproblem to accelerate the convergence of our algorithm (see Birge and Louveaux (1988)).

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. (2008)). 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 4.5.

4.5 Numerical experiments

The following three sections give the results of the numerical experiments we conducted. We first present results for stochastic instances following a uniform demand distribution, with data based on the instances used in Gruson et al. Gruson et al. (2019a). We then present results where the demand at the retailer level is generated using other demand distributions. Finally, we present results for an extension where we allow lost sales at the retailer level. 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 instances (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.

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 (1997) in the context of two-stage linear programs. In case the optimal solution for the 2S-3LSPD was not found, the best upper bound was used to calculate these indicators, which leads to an approximation of these values. The EVPI represents the additional cost of the stochastic model compared to the case where the actual demand for the whole time horizon would have been known in the first stage. The VSS represents the possible gain (in terms of lower expected costs) from solving the actual stochastic model, using a specific set of scenarios, instead of using the expected value for the parameters and applying the resulting plan to each of these demand scenarios. In the following tables, both the VSS and EVPI are computed as a percentage of the optimal solution cost.

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}. The other CPLEX parameters are set to their default value. The computation time limit imposed to solve each instance is 6 hours. The results reported all take into account the addition of LBL inequalities since initial experiments have shown that their addition has a huge impact on the performance of the decomposition approach. We performed our experiments on a 2.1 GHz Intel E5-2683 v4 processor with only one thread.

4.5.1 Results for the 2S-3LSPD

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. Gruson et al. (2019a). 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. 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 each combination of settings, Gruson et al. Gruson et al. (2019a) generated five instances leading to 480 different instances to be solved. In our case, we have 540 instances to be solved.

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. For the cuts added at the root node, we set a limit to 0, 50 or 100 in order not to add too many cuts at the root node, which would remove only a small portion of the search space. 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. Finally, we impose initial setups at each facility. Indeed, as we have no 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 interested reader is referred to Gruson et al. Gruson et al. (2019b) for detailed results. These results indicate, in particular, that the way cuts are aggregated has a high impact on the different indicators. 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 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. In the Pareto column, we denote by MW the use of Algorithm 2 to solve a parametric minimum cost flow problem and derive Pareto-optimal cuts. In case we use Pareto-optimal cuts, we specify if these cuts were obtained using a fixed core point (F), or using the procedure proposed by Papadakos Papadakos (2008) (P).

Table 4.2 – 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 (%)
✓	F	✓	321270	321271	1532	215	1752	57	0	0	97.22	0.6
✓	P	✓	321268	321271	1498	738	3800	52	0.01	0	94.44	0.6
✓	F	x	321270	321271	1436	113	3154	56	0	0	94.44	0.6
✓	P	x	321270	321271	1026	137	2715	49	0	0	97.22	0.6
x	-	✓	321270	321271	1565	52	1583	48	0	0	94.44	0.58
x	-	x	321045	321271	2335	14	1695	63	0.02	0	91.67	0.58
MW	F	x	321265	321271	1348	14	1269	75	0	0	94.44	0.62
MW	P	x	321219	321271	1419	15	304	64	0.01	0	94.44	0.62
CPLEX			321271	321271	1320	-	0.97	-	0	0	100	-

Table 4.3 – 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 (%)
✓	F	✓	325334	325339	3320	2591	661	38	1.35×10^{-5}	-45.82	97.22	0.82
✓	P	✓	325322	325340	2856	2150	254	37	4.29×10^{-5}	-45.82	97.22	0.82
✓	F	x	325285	325350	3142	2698	104	41	0.02	-45.81	97.22	0.82
✓	P	x	312531	325339	3181	2618	347	35	2.78	-45.82	94.44	0.82
x	P	✓	325304	325341	1421	536	236	38	0.01	-45.82	97.22	0.82
x	P	x	325288	325342	2061	93	495	43	0.01	-45.82	97.22	0.82
MW	F	x	325338	325339	993	90	3272	48	2.16×10^{-6}	-45.82	97.22	0.82
MW	P	x	325300	325341	948	87	165	42	0.01	-45.82	97.22	0.82
CPLEX			268866	509431	13602	-	0.22	-	47.22	0	50	-

The last line represents the results obtained by CPLEX directly.

Tables 4.2-4.4 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. Adulyasak et al. (2015).

Table 4.2 indicates that for the instances with 5 scenarios, our proposed algorithms have a similar performance compared to CPLEX. Tables 4.3 and 4.4 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

Table 4.4 – 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 (%)	V (
✓	F	✓	295841	321212	21631	5457	96	50	5.56	-55.13	91.67	0.79	0
✓	P	✓	298997	321210	6119	5844	42	53	4.64	-55.13	94.44	0.79	0
✓	F	x	295674	321213	5610	4981	119	55	5.57	-55.13	91.67	0.79	0
✓	P	x	309570	321212	5022	4536	109	52	2.8	-55.13	94.44	0.79	0
x	P	✓	321120	321217	2474	1157	360	55	0.02	-55.13	94.44	0.79	0
x	P	x	321084	321213	2449	371	237	67	0.03	-55.13	94.44	0.79	0
MW	F	x	321130	321210	1834	351	988	88	0.02	-55.13	97.22	0.79	0
MW	P	x	321158	321211	1688	277	351	59	0.01	-55.13	97.22	0.79	0
CPLEX			227051	528880	17199	-	0	-	33.33	0	53.33	-	

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 solution. One can note in the results reported here that the enhancements have a big influence in the experiments. 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, on the number of optimal solutions found and on the quality of the solution.

In Tables 4.2-4.4, one can see that the values obtained for EVPI are relatively low, ranging between 0.58 and 0.82%. This is explained by the way we generated the demands for the retailers. Indeed, the demands for each retailer are generated in an independent fashion, which gives, on average, a stable demand for the plant despite the differences that can exist between the demands of the different retailers. To validate this hypothesis, we conducted additional experiments with correlated demands for the retailers. We initially generate the demand for the first retailer and the demands for the other retailers range between 0.5 and 3 times the demand for the first retailer. In these additional experiments, we kept the same possible numbers of retailers, warehouses and scenarios, and the same

possible demand and cost structure. On average, the correlation coefficient we obtain for the demands between the retailers is 0.58. The value of EVPI we obtained on these instances is 15.88%, 6.33% and 3.55% for a number of scenarios equal to 5, 50 and 100, respectively. These additional experiments were made with 100 cuts at the root node, an aggregation of cuts per retailer and time period, the MIR procedure done every 10 iterations and the use of Pareto optimal cuts obtained through our specialized with a core point updated as in Papadakos Papadakos (2008).

In light of the results shown in Tables 4.2-4.4, 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.

4.5.2 Results for the 2S-3LSPD with different demand distributions

We conducted additional experiments with different demand distributions at the retailer level. The demand distributions are the same as the ones used in Tunc et al. Tunc et al. (2018). In their experiments, Tunc et al. Tunc et al. (2018) consider that the demands are normally distributed with a fixed coefficient of variation α equal to 0.1, 0.2 and 0.3. They further consider an erratic and a lumpy pattern for the mean demands. For the erratic pattern, the mean demands are drawn from U[0, 100]. In the case of the lumpy pattern, the mean demands are drawn from U[0,420] with probability 0.2 and from U[0,20] with probability 0.8. Table 4.5 presents the results we obtained for each demand distribution setting. In Table 4.5, each line represents 540 instances solved, using the same settings as in the previous section. Based on the results obtained from the experiments in Section 4.5.1, we only conducted experiments with the use of Pareto-optimal cuts obtained through our specialized algorithm and with a dynamic core point as in Papadakos Papadakos (2008).

In Table 4.5, the first two columns indicate the demand pattern used to generate the

Table 4.5 – Results with different demand distributions

Pattern	α	BLB	BUB	Time (s)	Time SP (s)	Nodes	It	Gap (%)	C-Gap (%)	Opt (%)	EVPI (%)	VSS (%)
Erratic	0.1	320225	320279	1481	165	364	64	0.01	-31.86	95.4	0.53	0.35
Erratic	0.2	320115	320154	1323	143	493	64	0.01	-31.84	96.3	0.64	0.33
Erratic	0.3	320103	320122	1318	146	603	63	0.005	-32.78	96.3	2.31	0.11
Lumpy	0.1	312240	312244	904	140	682	62	0.001	-25.78	98.1	5.59	0.82
Lumpy	0.2	312124	312141	941	138	376	63	0.004	-26.33	97.2	5.66	0.83
Lumpy	0.3	311987	311998	827	139	223	61	0.003	-26.53	98.1	6.31	0.96

demand, along with the coefficient of variation used. The results displayed in Table 4.5 show once again the superiority of our specialized algorithm over CPLEX. In particular, we have obtained the optimal solution in almost all cases. When we are not able to prove optimality, the cost of the solution is close to the optimal one, as stated by the Gap column. This very good performance of our algorithm on different demand distributions is another strength of our Benders-based branch-and-cut.

4.5.3 Results for the 2S-3LSPD with lost sales

We conducted experiments on an extension of the 2S-3LSPD where we allow for lost sales. In that case, constraints (4.13) are replaced by (4.21). We penalize the lost sales in the objective function through a unit penalty cost lsc_{rt}. We set the value of this penalty cost as a multiple ρ of the retailer holding cost. We set ρ equal to 0, 5, 10, 20 and 100. For these experiments, we use the same set of instances as in Section 4.5.1. Table 4.6 presents the results we obtained for each unit penalty cost, among all instances. Similar to the experiments in Section 4.5.2, we only conducted experiments with the use of Pareto-optimal cuts obtained through our specialized algorithm and with a dynamic core point as in Papadakos Papadakos (2008). In Table 4.6, the column Extra time indicates the extra time taken by our algorithm compared to CPLEX. The columns All LS, Part LS and No LS report the proportion of instances for which all the demand went to lost sales, part of the demand went to lost sales and no demand went to lost sales in the optimal solution, respectively.

The results reported in Table 4.6 once again illustrate the superiority of our algorithm

Table 4.6 – Results with the possibility of having lost sales

ρ	BLB	BUB	Time (s)	Time SP (s)	Extra time (s)	Nodes	It	Gap (%)	C-Gap (%)	Opt (%)	EVPI (%)	VSS (%)	No L (%)
0	0	0	0.54	0	0.31	0	0.66	0	0	100	0	0	0
5	259753	259753	1931	1320	-8203	2385	37	0	-39.51	93.81	0.28	0.11	16
10	294398	294410	2843	1992	-8910	2860	63	0	-18.99	87.42	1.21	0.36	59
20	302193	302203	2736	1718	-6216	4277	71	0	-113.93	84.72	1.02	0.21	93
100	322559	322608	1352	126	-9355	273	55	0.01	-33.65	96.29	0.74	0.07	100

compared to CPLEX alone, both in terms of CPU time and quality of the solution obtained. One can also see that when the lost sales cost are low there is a tendency to have a lot of lost sales, because the setup costs would be too high compared to the lost sales costs. On the contrary, when the lost sales costs become high, it is worth paying for a setup and actually satisfying some part of the demand.

4.6 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. In these experiments, we varied the way the demand is generated and we also allow for possible lost sales. The use of a Benders-based branch-and-cut algorithm to solve 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 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, 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.

Acknowledgements

We would like to thank to the three anonymous referees for their valuable comments. 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).

References

Abdullah, S., A. Shamayleh and M. Ndiaye. 2019, «Three stage dynamic heuristic for multiple plants capacitated lot sizing with sequence-dependent transient costs», *Computers & Industrial Engineering*, vol. 127, p. 1024–1036.

Adulyasak, Y., J.-F. Cordeau and R. Jans. 2015, «Benders decomposition for production routing under demand uncertainty», *Operations Research*, vol. 63, p. 851–867.

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

Aloulou, M. A., A. Dolgui and M. Y. Kovalyov. 2014, «A bibliography of non-deterministic lot-sizing models», *International Journal of Production Research*, vol. 52, p. 2293–2310.

Bahl, H. C. and S. Zionts. 1987, «Multi-item scheduling by Benders' decomposition», *The Journal of the Operational Research Society*, vol. 38, n° 12, p. 1141–1148.

Bayley, T., H. Süral and J. H. Bookbinder. 2018, «A hybrid Benders approach for coordinated capacitated lot-sizing of multiple product families with set-up times», *International Journal of Production Research*, vol. 56, n° 3, p. 1326–1344.

Benders, J. F. 1962, «Partitioning procedures for solving mixed-variables programming problems», *Numerische Mathematik*, vol. 4, p. 238–252.

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

Birge, J. R. and F. V. Louveaux. 1988, «A multicut algorithm for two-stage stochastic linear programs», *European Journal of Operational Research*, vol. 34, n° 3, p. 384–392.

Bodur, M. and J. R. Luedtke. 2016, «Mixed-integer rounding enhanced benders decomposition for multiclass service-system staffing and scheduling with arrival rate uncertainty», *Management Science*, vol. 63, n° 7, p. 2073–2091.

Bookbinder, J. H. and J.-Y. Tan. 1988, «Strategies for the probabilistic lot-sizing problem with service-level constraints», *Management Science*, vol. 34, p. 1037–1156.

Brahimi, N., N. Absi, S. Dauzère-Pérès and A. Nordli. 2017, «Single-item dynamic lot-sizing problems: An updated survey», *European Journal of Operational Research*, vol. 263, n° 3, p. 838–863.

de Camargo, R. S., J. G. de Mirande and H. P. Luna. 2008, «Benders decomposition for the uncapacitated multiple allocation hub location problem», *Computers & Operations Research*, vol. 35, p. 1047–1064.

Caserta, M. and S. Voß. 2020, «Accelerating mathematical programming techniques with the corridor method», *International Journal of Production Research.*

Chen, H. 2015, «Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems», *Omega*, vol. 56, p. 25–36.

Dhaenens-Flipo, C. and G. Finke. 2001, «An integrated model for an industrial production-distribution problem», *IIE Transactions*, vol. 33, p. 705–715.

Di Summa, M. and L. A. Wolsey. 2008, «Lot-sizing on a tree», *Operations Research Letters*, vol. 36, nº 1, p. 7–13.

Dixon, P. S. and E. A. Silver. 1981, «A heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing problem», *Journal of Operations Management*, vol. 2, p. 23–39.

Furlan, M. M. and M. O. Santos. 2017, «BFO: a hybrid bees algorithm for the multi-level capacitated lot-sizing problem», *Journal of Intelligent Manufacturing*, vol. 28, p. 924–944.

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

Gruson, M., J.-F. Cordeau and R. Jans. 2018, «The impact of service level constraints in deterministic lot sizing with backlogging», *Omega*, vol. 79, p. 91–103.

Gruson, M., J.-F. Cordeau and R. Jans. 2019b, «Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure», *Cahier du GERAD, HEC Montréal*, vol. G-2019-51.

Guan, Y., S. Ahmed, G. L. Nemhauser and A. J. Miller. 2006, «A branch-and-cut algorithm for the stochastic uncapacitated lot-sizing problem», *Mathematical Programming*, vol. 105, nº 1, p. 55–84.

Gutiérrez, J., J. Puerto and J. Sicilia. 2004, «The multiscenario lot size problem with concave costs», *European Journal of Operatinal Research*, vol. 156, p. 162–182.

Haugen, K. K., A. Løkketange and D. L. Woodruff. 2001, «Progressive hedging as a metaheuristic applied to stochastic lot-sizing», *European Journal of Operatinal Research*, vol. 132, p. 116–122.

Helber, S., F. Sahling and K. Schimmelpfeng. 2013, «Dynamic capacitated lot sizing with random demand and dynamic safety stocks», *OR Spectrum*, vol. 35, p. 75–105.

Magnanti, T. and R. Wong. 1981, «Accelerating Benders decomposition: algorithmic enhacement and model selection criteria», *Operations Research*, vol. 23, p. 464–484.

Magnanti, T. L., P. Mireault and R. T. Wong. 1986, «Tailoring Benders decomposition for uncapacitated network design», *Mathematical Programming Study*, vol. 26, p. 112–154.

Melo, R. A. and L. A. Wolsey. 2010, «Uncapacitated two-level lot-sizing», *Operations Research Letters*, vol. 38, nº 4, p. 241–245, ISSN 0167-6377.

Papadakos, N. 2008, «Practical enhancements to the Magnanti-Wong method», *Operations Research Letters*, vol. 36, nº 4, p. 444–449.

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

Rahmaniani, R., T. G. Crainic, M. Gendreau and W. Rei. 2017, «The Benders decomposition algorithm: A literature review», *European Journal of Operational Research*, vol. 259, nº 3, p. 801–817.

Rockafellar, R. T. and R. J.-B. Wets. 1991, «Scenarios and policy aggregation in optimization under uncertainty», *Mathematics of Operations Research*, p. 119–147.

Sahling, F., L. Buschkühl, H. Tempelmeier and S. Helber. 2009, «Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic», *Computers & Operations Research*, vol. 36, nº 9, p. 2546–2553.

Taskin, S. and E. J. Lodree Jr. 2010, «Inventory decisions for emergency supplies based on hurricane count predictions», *International Journal of Production Economics*, vol. 126, p. 66–75.

Tempelmeier, H. 2007, «On the stochastic uncapacitated dynamic single-item lot sizing problem with service level constraints», *European Journal of Operational Research*, vol. 181, nº 1, p. 184–194.

Tempelmeier, H. 2013, «Stochastic lot sizing problems», in *Handbook of Stochastic Models and Analysis of Manufacturing System Operations*, edited by J. M. Smith and B. Tan, Springer New York, New York, NY, p. 313–344.

Tempelmeier, H. and S. Helber. 1994, «A heuristic for dynamic multi-item multi-level capacitated lot sizing for general product structures», *European Journal of Operational Research*, vol. 75, nº 2, p. 296–311.

Tunc, H., O. A. Kilic, S. A. Tarim and R. Rossi. 2018, «An extended mixed-integer programming formulation and dynamic cut generation approach for the stochastic lot-sizing problem», *INFORMS Journal on Computing*, vol. 30, nº 3, p. 492–506.

Wei, M., M. Qi, T. Wu and C. Zhang. 2019, «Distance and matching-induced search algorithm for the multi-level lot-sizing problem with substitutable bill of materials», *European Journal of Operational Research*, vol. 277, p. 521–541.

Wolsey, L. A. 1998, *Integer Programming*, Wiley, New York.

Zhang, S. and H. Song. 2018, «Production and distribution planning in Danone waters China division», *INFORMS Journal on Applied Analytics*, vol. 48, p. 578–590.

Chapitre 5

Top-down and bottom-up heuristics for an integrated three-level lot sizing and replenishment problem

Information sur le chapitre

Ce chapitre est un travail en cours.

Abstract

We address a three-level lot sizing and replenishment problem (3LSRP), which is an extension of the classical two-level production routing problem. We consider one production plant that produces several items over a discrete and finite planning horizon. The items produced are used to replenish warehouses and then retailers. The items are sent from the plant to the warehouses using direct shipments, and routes are built to deliver the goods from the warehouses to the retailers. The shipments between the plant and the warehouses, and between each warehouse and the retailers are performed by a homogeneous fleet of capacitated vehicles. We also impose production capacity restrictions at the plant. The objective is to minimize the sum of the fixed production and replenish-

ment costs, of the variable inventory holding costs at all three levels, and of the routing costs. We develop a branch-and-cut algorithm to solve this problem and compare it to two heuristics we design. The first heuristic uses a top-down approach with the production decisions being the leading decisions. In the second heuristic, we use a bottom-up approach representing a situation where the replenishment decisions at the retailer level are the leading decisions. We run numerical experiments to assess the performance of each heuristic. We further analyze the impact of allowing split demands or split deliveries on the performance of each heuristic.

5.1 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, Vogel et al., 2017; Absi et al., 2018). Other potential benefits of integration, both in terms of money and customer service, explain why this area has attracted a lot of research over the last decades. Early studies include those of Chandra and 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. The interested reader is referred to Adulyasak et al. (2015b) for a review of models and solutions algorithms for the PRP.

With a 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 routing part between the plant and the DCs is not always relevant.

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 done between the plant and DCs are more likely to be direct shipments. Unlike the problem we address in this work, the 2E-VRP does not incorporate any production decision. Note finally that the 2E-VRP considers just one time period compared to our case where we consider several time periods.

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) 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 direct and subject to capacity restrictions. In a similar spirit, the shipments between each warehouse and the retailers are performed using homogeneous capacitated vehicles. We must therefore decide on the set of retailers to be visited in each time period by each vehicle, and the sequence of the visits to the retailers. We also consider production capacity restrictions at the plant. The objective is to minimize the sum of the fixed production and replenishment costs, of the variable inventory holding costs at all three levels, and of the routing costs. We do not consider unit production costs nor unit transportation costs. Indeed, if we consider static unit production and transportation costs, it will just result in a constant added to the objective function since the complete demand must be satisfied.

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 distinctions between that prior work and the work presented here. First, we add transportation capacity requirements. Second, compared to Gruson et al. (2019a), 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 do 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 big trucks. This is not the case for deliveries between larger facilities, the production plant and the warehouses in our case. We also integrate realistic constraints such as the transportation capacity constraints. Note that the assumption of having an unlimited fleet of vehicles to make the deliveries between the plant and the warehouses corresponds to the possible use of third party logistics providers to make these deliveries.

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 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 and 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 we develop to solve the problem exactly. Third, we also evaluate the gains obtained from the possibility of having split demands and split deliveries. The literature on the PRP and related problems such as the inventory routing problem (IRP) usually assumes that each customer can be visited by one truck only in each time

period, but that the demand of a specific time period t can be spread over several periods before period t. From an operations perspective, 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 5.2. We then present a mathematical formulation along with valid inequalities for the problem in Section 5.3. 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 we designed to exactly solve the problem. The top-down and bottom-up heuristics are presented in Sections 5.4 and 5.5, respectively. Section 5.6 reports the results of the computational experiments performed to assess the performance of our heuristics. This is followed by the conclusion in Section 5.7.

5.2 Literature review

We review here the literature on several multi-echelon routing problems that relate to the 3LSRP we study here. First, we review the literature on the 2E-VRP. Second, we briefly review the literature on the PRP. Note that the 3LSRP studied here is also 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 works similar to the study of the 3LSPD.

5.2.1 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 second set of depots (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. Note that we only quickly 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 linked to two-echelon routing problems.

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 also define the different 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. They propose an adaptive large neighbourhood search algorithm to solve the problem while considering synchronization for the routes that end and start at level one. In the same vein, Dellaert et al. (2019) propose four MIP formulations for the problem without synchronization and develop a branch-and-price algorithm to solve it.

Darvish et al. (2019) study a two-echelon VRP and incorporate flexibility in the decisions. In their case, the intermediate facilities, i.e., the facilities of level one (called distribution centers), can be rented and one must decide on such possibilities. They also consider flexibility in satisfying the due dates of the customer orders. When the due dates

are not met, a penalty is paid. They propose a MIP formulation for the problem together with valid inequalities. The problem is solved using an enhanced parallel exact method.

Several works incorporate environmental considerations. Soysal et al. (2015) consider fuel consumption in the objective function. This cost is dependent on the speed of the trucks and the authors therefore also incorporate different speed possibilities depending on the time of the delivery. The inclusion of this feature is done with the aim to represent a realistic case where one wants also to minimize the environmental impact of the routes. They develop a MIP formulation and propose valid inequalities for their problem. They use a general purpose solver to address the real case of a supermarket chain in the Netherlands. 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) study a similar problem but 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. The assignment of customers to each kind of depots is included in the decision variables of the proposed model. They further incorporate greenhouse gas (GHG) and disturbance in their objective function, along with transportation costs. Disturbance is defined as the external effect of noise and congestion to local citizens. They develop a metaheuristic that combines large neighbourhood search and multi-objective methods to find a Pareto front for their problem.

5.2.2 Production routing problem

In the 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 an extensive review of the PRP.

The PRP has attracted a lot of research since the early work of Chandra and 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, the works on the PRP deal with the merge between 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 and Whitin, 1958) or transportation (Krarup and Bilde, 1977) formulations have been mostly used. This last formulation is in particular useful in the case of perishable products (see, Dayarian and Desaulniers, 2019). On the VRP side, there has been formulations both with and without vehicle indices. In the VRP, the vehicle index is used when the fleet of vehicle 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). In case a vehicle index is used, the reason is to avoid having fractional subtour elimination constraints.

From a methodological perspective, because of its complexity, the PRP has been mainly solved by heuristics or metaheuristics. We present the most recent approaches. Adulyasak et al. (2015b) develop an adaptive large neighbourhood search algorithm (ALNS). Initially, a pool of different solutions is built by iteratively solving production-distribution and routing subproblems. Each initial solution is then improved in the 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 and Süral (2017) design a heuristic that comprises five phases. First, a TSP is solved to determine the best tour among all clients and plant. The tour obtained, called a priori tour, is used as an input in the second phase to decide on the production, inventory and delivery quantities. The third phase actually constructs the routes by solving capacitated VRPs or TSPs. The solution obtained is improved in the fourth phase by allowing some insertion and removal of clients in the routes. The fifth phase improves the solution obtained after the fourth phase by solving a TSP on each route built. 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. The first phase of the matheuristic solves a special LSP to obtain a production setup plan. In this first phase, the cost to dispatch a vehicle from the plant is approximated. The second phase consists in deciding which suppliers to visit and how much to pick-up from them. In this second phase, as in Absi et al. (2015), the routing costs are approximated. Finally, the third phase solves a series of VRPs to obtain a solution for the initial problem. This solution is also used to update the transportation costs in the second phase.

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 and Yildiz, 2019), or of routes extending to several periods (Miranda et al., 2018), among others.

5.3 A transportation based formulation

In this section we present the mathematical formulation for the problem. Table 5.1 lists all the sets, parameters and decision variables used in the mathematical model. This formulation can be seen 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 pro-

Table 5.1 – Sets, parameters and decision variables used in the combined classical and transportation model

Sets

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 $S \subseteq R \cup W$
F	set of all facilities, $F = P \cup W \cup R$	$E(i)$	set of edges $(j,j') \in E$ such that $j = i$ or $j' = i$

Parameters

sc_{it}^{p}	setup cost at the plant for item i in period t	d_{it}^{r}	demand of retailer r for item i in period t
sc_{t}^{w}	setup cost for warehouse w in period t	d_{t}^{r}	total demand for retailer r in period t
sc_{t}^{r}	setup cost for retailer r in period t	d_{kt}^{r}	total demand for retailer r between periods k and t
hc_{it}^{j}	holding cost for facility j in period t for item i	d_{it}^{p}	demand at the plant for item i in period t
c_{ij}	cost to go from facility i to facility j	d_{ikt}^{p}	demand at the plant for item i between periods k and t
C^{P}	production capacity available in any period t	st_i	setup time for item i
v_i	volume of item i	vt_i	variable production time for item i
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_{it}^{w}	quantity of item i ordered by warehouse w in period t
q_{iktw}^{r}	quantity of item i ordered by retailer r in period k from warehouse w to satisfy d_{it}^{r}
I_{it}^{j}	inventory of item i in facility j at the end of period t
y_{it}^{p}	binary setup variable that takes the value 1 if there is a setup for item i in period t
y_{t}^{w}	number of trucks used to do deliveries between the plant and warehouse w in period t
y_{t}^{r}	binary setup variable that takes the value 1 if there is an order placed by retailer r in period t
z_{t}^{w}	number of vehicles used by warehouse w in period t to make deliveries to retailers
z_{tw}^{r}	total number of times retailer r is visited by the vehicles of warehouse w in period t
Z_{tw}^{r}	binary variable that takes the 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

posed by Alvarez et al. (2020) in the context of the IRP with perishable products. The formulation is as follows:

$$\text{Min} \sum_{t \in T} \left(\sum_{i \in I} sc_{it}^{p} y_{it}^{p} + \sum_{w \in W} sc_{t}^{w} y_{t}^{w} + \sum_{r \in R} sc_{t}^{r} y_{t}^{r} \right) + \sum_{t \in T} \sum_{i \in I} \left(hc_{it}^{p} I_{it}^{p} + hc_{it}^{w} I_{it}^{w} + \sum_{r \in R} \sum_{w \in W} \sum_{k \geq t}^{k-1} \sum_{j=t} hc_{ij}^{r} q_{itkw}^{r} \right) + \sum_{t \in T} \sum_{w \in W} \sum_{(i,j) \in E(W \cup R)} c_{ij} x_{ijtw} \tag{5.1}$$

$$\text{s. t. } I_{i,t-1}^{p} + p_{it} = \sum_{w \in W} q_{it}^{w} + I_{it}^{p} \qquad \forall\, i \in I, t \in T \tag{5.2}$$

$$I^w_{i,t-1} + q^w_{it} = \sum_{k \geq t} \sum_{r \in R} q^r_{itkw} + I^w_{it} \qquad \forall\, w \in W, i \in I, t \in T \qquad (5.3)$$

$$\sum_{w \in W} \sum_{k \leq t} q^r_{iktw} = d^r_{it} \qquad \forall\, r \in R, i \in I, t \in T \qquad (5.4)$$

$$p_{it} \leq \min\left\{ \frac{C^p - st_i}{vt_i}; d^p_{it|T|} \right\} y^p_{it} \qquad \forall\, i \in I, t \in T \qquad (5.5)$$

$$q^w_{it} \leq \frac{C^{PW}}{v_i} y^w_t \qquad \forall\, w \in W, i \in I, t \in T \qquad (5.6)$$

$$q^r_{iktw} \leq d^r_{it} y^r_k \qquad \forall\, r \in R, i \in I, k \leq t \in T \qquad (5.7)$$

$$\sum_{i \in I} \left(vt_i p_{it} + st_i y^p_{it} \right) \leq C^P \qquad \forall\, t \in T \qquad (5.8)$$

$$\sum_{i \in I} v_i q^w_{it} \leq C^{PW} y^w_t \qquad \forall\, w \in W, t \in T \qquad (5.9)$$

$$\sum_{r \in R} \sum_{i \in I} \sum_{k \geq t} v_i q^r_{itkw} \leq C^{WR} z^w_t \qquad \forall\, w \in W, t \in T \qquad (5.10)$$

$$\sum_{k \geq t} \sum_{i \in I} v_i q^r_{itkw} \leq C^{WR} z^r_{tw} \qquad \forall\, r \in R, w \in W, t \in T \qquad (5.11)$$

$$z^r_{tw} \leq |V(w)| Z^r_{tw} \qquad \forall\, r \in R, w \in W, t \in T \qquad (5.12)$$

$$\sum_{w \in W} Z^r_{tw} \leq 1 \qquad \forall\, r \in R, t \in T \qquad (5.13)$$

$$z^w_t \leq |V(w)| \qquad \forall\, w \in W, t \in T \qquad (5.14)$$

$$\sum_{(j,j') \in E(r)} x_{jj'tw} = 2 z^r_{tw} \qquad \forall\, r \in R, w \in W, t \in T \qquad (5.15)$$

$$C^{WR} \sum_{(i,j) \in E(S)} x_{ijtw} \leq \sum_{r \in S} \left(C^{WR} z^r_{tw} - \sum_{k \geq t} \sum_{i \in I} v_i q^r_{itkw} \right) \qquad \forall\, w \in W, S \subseteq R, |S| \geq 2, t \in T \qquad (5.16)$$

$$I^p_{it} \geq 0 \qquad \forall\, i \in I, t \in T \qquad (5.17)$$

$$I^w_{it} \geq 0 \qquad \forall\, w \in W, i \in I, t \in T \qquad (5.18)$$

$$p_{it} \geq 0 \qquad \forall\, i \in I, t \in T \qquad (5.19)$$

$$q^w_{it} \geq 0 \qquad \forall\, w \in W, i \in I, t \in T \qquad (5.20)$$

$$q^r_{iktw} \geq 0 \qquad \forall\, r \in R, w \in W, i \in I, k \leq t \in T \qquad (5.21)$$

$$y_{it}^{p} \in \{0;1\} \qquad \forall\, i \in I, t \in T \tag{5.22}$$

$$y_{t}^{w} \in \mathbb{N} \qquad \forall\, t \in T \tag{5.23}$$

$$y_{t}^{r} \in \{0;1\} \qquad \forall\, t \in T \tag{5.24}$$

$$z_{t}^{w} \in \mathbb{N} \qquad \forall\, w \in W, t \in T \tag{5.25}$$

$$z_{tw}^{r} \in \mathbb{N} \qquad \forall\, r \in R, w \in W, t \in T \tag{5.26}$$

$$Z_{tw}^{r} \in \{0;1\} \qquad \forall\, r \in R, w \in W, t \in T \tag{5.27}$$

$$x_{ijtw} \in \mathbb{N} \qquad \forall\, (i,j) \in E(W \cup R), t \in T, w \in W. \tag{5.28}$$

The objective function (5.1) minimizes the sum of the setup costs, of the inventory holding costs at each facility, and of the routing costs. We consider a fixed cost whenever there is a truck used for delivery between the plant and any warehouse. The routing costs are computed based on the distance the distance travelled by the trucks available at the warehouses. Constraints (5.2)-(5.3) are the inventory balance constraints for the plant and the warehouses, respectively. Constraints (5.4) are the demand satisfaction constraints at the retailer level. Constraints (5.5)-(5.7) are the setup constraints for the plant, the warehouses and the retailers, respectively. Constraints (5.6) further compute the number of trucks used for the deliveries between the plant and each warehouse. Constraints (5.8)-(5.10) 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 (5.11) link the order and visit variables for the retailers. Constraints (5.12) link the different visit variables. Constraints (5.13) state that each retailer can be visited by one warehouse only in each time period, possibly by several trucks. Constraints (5.14) limit the number of trucks available at each warehouse. Constraints (5.15) are the degree constraints. Constraints (5.16) are the subtour elimination constraints. These can be found in the literature as general fractional subtour elimination constraints (GFSEC). Finally, constraints (5.17)-(5.28) define the domains of the decision variables.

5.3.1 Valid inequalities

We propose several valid inequalities to strengthen the proposed formulation. We start by presenting valid inequalities related to the lot sizing part of the model, and then to the routing part of the model. The valid inequalities presented here are added a priori to the model.

Valid inequalities for the lot sizing part

To strengthen the lot sizing part of the problem, we add the following valid inequalities:

$$s^p_{i,k-1} + \sum_{w\in W} s^w_{i,k-1} + \sum_{r\in R}\sum_{w\in W}\sum_{u=1}^{k-1}\sum_{l=u}^{|T|}\left(q^r_{iulw} - d^r_{i,1,k-1}\right) \geq d^p_{ikt} - \sum_{j=k}^{t} d^p_{ijt} y^p_{ij} \qquad \forall\, i\in I, k\leq t\in T \quad (5.29)$$

$$\sum_{k\leq t} y^p_{kt} \geq \left\lceil \frac{d^p_{i1t}}{C_t - st_i} \right\rceil \qquad \forall\, i\in I, t\in T \quad (5.30)$$

$$\sum_{w\in W}\sum_{k\leq t} y^w_k \geq \left\lceil \frac{\sum_{i\in I} v_i d^p_{i1t}}{C^{PW}} \right\rceil \qquad \forall\, t\in T. \quad (5.31)$$

Inequalities (5.29) are an adaptation of the (l,S,WW) valid inequalities. They impose a setup if the total inventory in the whole supply chain is not sufficient to satisfy the demand. The left hand side of inequalities (5.29) represents the notion of echelon stock, which is the total stock of item i in the supply chain at the end of period $k-1$. Inequalities (5.30) compute the minimal number of setups required at the plant to satisfy the demand of a specific item, given the production capacity available. Inequalities (5.31) define the minimum number of trucks that will be used to make the deliveries between the plant and the warehouses, given the total demand over all the items since the first period of the planning horizon.

Valid inequalities for the routing part

To strengthen the routing part of the problem, we add the following valid inequalities:

$$x_{ijtw} \leq z^r_{tw} \qquad \forall\, r \in R, (i,j) \in E(r), w \in W, t \in T \tag{5.32}$$

$$x_{wrtw} \leq 2z^w_t \qquad \forall\, r \in R, w \in W, t \in T \tag{5.33}$$

$$z^r_{tw} \leq z^w_t \qquad \forall\, r \in R, w \in W, t \in T \tag{5.34}$$

$$z^r_{tw} \leq |V(w)| y^r_t \qquad \forall\, r \in R, w \in W, t \in T \tag{5.35}$$

$$\sum_{w \in W} \sum_{u=1}^{t-s-1} \sum_{l=u}^{|T|} q^r_{iulw} - d^r_{i,1,t-s-1} \leq \left(\sum_{j=0}^{s} d^r_{i,t-j} \right) - \left(\sum_{j=0}^{s} d^r_{i,t-j} \right) \left(\sum_{w \in W} \sum_{j=0}^{s} Z^r_{t-j,w} \right) \qquad \forall\, r \in R, i \in I, s < t \in T, t > 2. \tag{5.36}$$

Inequalities (5.32) impose that there must be a visit if a vehicle visits a particular retailer. Inequalities (5.33) are similar and impose a visit to warehouse w when one of its vehicle is used. Inequalities (5.34) impose a minimum number of vehicles to be dispatched from a warehouse if there is any retailer visited by this warehouse. Inequalities (5.35) impose a setup at a retailer if it is visited by any warehouse in a certain time period. Finally, inequalities (5.36) impose retailer replenishment when the inventory is not sufficient. These last inequalities have been initially proposed by Archetti et al. (2011).

Valid inequalities that combine the lot sizing and routing parts

We can also add valid inequalities that link the replenishment quantities and the visit variables, making a link between the lot sizing and routing parts. These are as follows:

$$q^r_{iktw} \leq d^r_{it} z^r_{kw} \qquad \forall\, r \in R, w \in W, i \in I, k \leq t \in T. \tag{5.37}$$

Note that in the numerical experiments, we added these inequalities in the case where delivery splitting was prohibited only.

5.3.2 Split demands and deliveries

We further want to explore the possibilities of having split demands or split deliveries to ship the goods from the warehouses to the retailers. Recall that delivery splitting means that a retailer can be visited by several trucks 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.

The proposed model (5.1)-(5.28) 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:

$$\sum_{w \in W} z_{tw}^r \leq 1 \qquad \forall r \in R, t \in T. \tag{5.38}$$

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

$$q_{iktw}^r \leq d_{it}^r Y_{ktw}^r \qquad \forall r \in R, w \in W, i \in I, k \leq t \in T \tag{5.39}$$

$$\sum_{w \in W} \sum_{k \leq t} Y_{ktw}^r \leq 1 \qquad \forall r \in R, t \in T \tag{5.40}$$

$$Y_{ktw}^r \in \{0;1\} \qquad \forall r \in R, w \in W, k \leq t \in T. \tag{5.41}$$

Constraints (5.39) link the ordering variables and the newly defined Y_{ktw}^r variables. Constraints (5.40) impose that the deliveries for all items must be done in at most one period. The inequality sign allows for the demand to be equal to zero. Table 5.2 summarizes the constraints that must be added to the original model (5.1)-(5.28), depending on the splitting possibilities. Note that this table does not take into account the valid inequalities introduced previously.

5.3.3 Branch-and-cut algorithm

The number of GFSECs (5.16) 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

Table 5.2 – Constraints to be added to (5.1)-(5.28) based on the splitting possibilities

	No demand splitting	Demand splitting
No delivery splitting	(5.38)-(5.41)	(5.38)
Delivery splitting	(5.39)-(5.41)	-

exactly, using the callback features of solvers. 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 and each time period, 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 definied as follows:

$$\sum_{i\in S}\sum_{j\in S, j\neq i} x_{ijtw} \leq \sum_{i\in S} z^i_{tw} - z^e_{tw} \qquad \forall\, S\subseteq R, |S|\geq 2, w\in W, t\in T, e\in S. \tag{5.42}$$

To separate the SEC, we use an exact separation algorithm that solves a series of minimum $s-t$ cut problems. It allows us to detect, for each warehouse and for each time period, the violated SECs. We build a graph from the set of retailers such that $\overline{z}^r_{tw} > 0$ and set the arc values to $\overline{x}_{ijtw}$. Then, for each retailer node of the constructed graph, a minimum $s-t$ cut problem is solved, setting the warehouse node as the source node and the retailer node as the sink node. Let $(S, \mathscr{T})$ be the minimum cut. Whenever the capacity of the minimum cut found is less than $2\overline{z}^r_{tw}$, we check whether the subtour elimination constraint with $e = \text{argmax}_{i\in S} z^i_{tw}$ is violated. If yes, a violated SEC is found and is added to the problem (see, Solyalı and Süral, 2011). To solve the minimum $s-t$ cut problems, the Concorde solver of Applegate et al. (2011) is used.

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.

5.4 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. Then, the first problem determines the production quantities at the plant along with the ordering decisions at the warehouse level. The solution of this first problem is used as an input to the second problem, which determines the ordering quantities at the retailer level. 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 and Melo (2016). We do so in light of both the theoretical and practical observations made in Cunha and Melo (2016) and in prior work by the authors (Gruson et al., 2019a). After a certain number of iterations for each of these OWMR, we diversify the search by changing the retailer assignment to the warehouses, for each time period. Between two iterations of the resolution of the first OWMR, 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, we update an approximation of the routing costs. The following sections present in more detail the different steps of the top-down heuristic.

5.4.1 Step 1: initial assignment of the retailers

We start the heuristic by assigning the retailers to the different warehouses, for each time period. In other words, the assignment defines from which warehouse the demand of each retailer in each time period will be satisfied. Initially, we assign each retailer to

its closest warehouse, for each time period. We then check that this initial assignment is consistent with the transportation capacity requirements. Otherwise, we heuristically reassign retailers so as to satisfy these requirements. 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 ordering variables. This idea of solving a TSP beforehand has been proposed in the context of the IRP by Solyalı and 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.

5.4.2 Step 2: solution of a first OWMR

In the second step of the heuristic, we solve a special 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. 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 $d_{it}^w = \sum_{r \in R(w,t)} d_{it}^r$. Let X_{wikt}^p be the amount produced at the production plant in period k to satisfy d_{it}^w and let X_{ikt}^w be the amount transported from the production plant to the warehouse w in period k to satisfy d_{it}^w. Let also s_{wikt}^p be the amount stocked at the production plant at the end of period k to satisfy d_{it}^w, and let s_{ikt}^w be the amount stocked at warehouse w at the end of period k to satisfy d_{it}^w. We further use the same setup variables for the plant and the warehouses as the ones used in Section 5.3. 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 is as follows:

$$\text{Min} \sum_{t \in T} \left(\sum_{i \in I} sc_{it}^p y_{it}^p + \sum_{w \in W} sc_t^w y_t^w + \sum_{i \in I} \sum_{w \in W} \sum_{k \leq t} \left(hc_{ik}^p s_{wikt}^p + hc_{ik}^w s_{ikt}^w \right) \right) \tag{5.43}$$

$$\text{s. t. } s_{w,i,k-1,t}^p + X_{wikt}^p = X_{ikt}^w + s_{wikt}^p \qquad \forall\, w \in W, i \in I, t \in T, k \leq t \in T \tag{5.44}$$

$$s_{i,k-1,t}^w + X_{ikt}^w = \delta_{kt} d_{it}^w + (1 - \delta_{kt}) s_{ikt}^w \qquad \forall\, w \in W, i \in I, t \in T, k \leq t \in T \tag{5.45}$$

$$X_{wikt}^p \leq d_{it}^w y_{ik}^p \qquad \forall\, w \in W, i \in I, t \in T, k \leq t \in T \tag{5.46}$$

$$X^w_{ikt} \leq d^w_{it} y^w_k \qquad \forall\, w \in W, i \in I, t \in T, k \leq t \in T \qquad (5.47)$$

$$\sum_{w \in W} \sum_{i \in I} \sum_{k \geq t} vt_i X^p_{witk} + \sum_{i \in I} st_i y^p_{it} \leq C^P \qquad \forall\, t \in T \qquad (5.48)$$

$$\sum_{i \in I} \sum_{k \geq t} v_i X^w_{itk} \leq C^{PW} y^w_t \qquad \forall\, w \in W, t \in T \qquad (5.49)$$

$$X^p_{wikt}, X^w_{ikt}, s^p_{wikt}, s^w_{ikt} \geq 0 \qquad \forall\, w \in W, i \in I, t \in T, k \leq t \in T \qquad (5.50)$$

$$y^p_{it} \in \{0;1\} \qquad \forall\, i \in I, t \in T \qquad (5.51)$$

$$y^w_t \in \mathbb{N} \qquad \forall\, w \in W, t \in T. \qquad (5.52)$$

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 $V^w_{kt} = \sum_{i \in I} v_i d^w_{ikt}$ and let $C^{WR,w}_{kt} = (t-k+1)|V(w)|C^{WR}$. We add the following constraints:

$$\sum_{i \in I} \sum_{l < k} \sum_{u=k}^{t} v_i X^w_{ilu} \geq V^w_{kt} - C^{WR,w}_{kt} \qquad \forall\, w \in W, k \leq t \in T \mid V^w_{kt} > C^{WR,w}_{kt}. \qquad (5.53)$$

Considering a time interval from periods k to t, the left hand side of constraints (5.53) calculates the amount of space occupied by items sent before period k to satisfy demand in periods k to t. This amount must be at least as big as the additional transportation space needed (above the total available space) to transport the total demand from periods k to t. If we do not allow demand nor delivery splitting, constraints (5.53) are still valid but do not impose that there must be no splitting at all. Indeed, in the right hand side of constraints (5.53), there may be a split of demand for one or more retailers. This right hand side only represents a lower bound on the quantity that must be delivered before period k. Let V^r_t be the demand of retailer r in period t, converted into volume, i.e., $V^r_t = \sum_{i \in I} v_i d^r_{iu}$. Therefore, if we do not allow demand nor delivery splitting, we add the

following constraints:

$$\sum_{i\in I}\sum_{l<k}\sum_{u=k}^{t} v_i X^w_{ilu} \geq \max\{\min_{k\leq u\leq t, r\in R(w,u)}\{V^r_u\}; C^{min,w}_{kt}\} \quad \forall\, w\in W, k\leq t\in T | V^w_{kt} > C^{WR,w}_{kt}. \tag{5.54}$$

Constraints (5.54) indicate that, if the cumulative demand of warehouse w between periods k and t, converted into volume, exceeds the cumulative transportation capacity available at this warehouse between the same periods, then the quantities ordered strictly before period k to satisfy demand between periods k and t must be greater than the lowest demand, converted into volume, and $C^{min,w}_{kt}$, where $C^{min,w}_{kt}$ represents the minimum quantity that must be delivered to satisfy the non splitting constraint. Example 5.4.1 illustrates how we compute this minimum quantity on a small instance. Note that the addition of constraints (5.54) makes a link between the second and third steps of the heuristic, and helps the heuristic find good integrated plans. From an operational perspective, it links our operational decisions, and therefore differs from a sequential approach that would be totally myopic regarding the links between the operational decisions. In the second step of our algorithm, we solve the problem defined by (5.43)-(5.54) with a general purpose solver.

Example 5.4.1. *In case we do not allow delivery or demand splitting, we have introduced constraints* (5.54) *to define the 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 5.3 is used to illustrate the problem that arises when we do not allow for delivery and demand splitting. In Table 5.3, we display the demands for a four period example with one item and one warehouse being in charge of 5 retailers. We consider that the volume of each item is equal to 1. In Table 5.3 each line represents one period and each column one retailer.*

In this small example, we further 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 the missing*

Table 5.3 – 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

capacity, being $110-100=10$ *units. However, this would imply delivery splitting since the minimum demand of a retailer is 15 units for retailer 3. Therefore, before period 2, we must at least deliver 15 units and the right hand side of constraints* (5.54) *will be 15 in period 2.*

Let us consider now the case of period 4. In period 4, we have 120 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. A quick look at the numbers indicate that we would need to deliver at least 30 units in the previous periods. This represents the demand of retailer 2 and will allow us to satisfy both the non splitting and the transportation capacity requirements. To find this minimum quantity, we actually solve a knapsack problem. Let* k *and* $t, k \leq t$, *be two periods such that* $\sum_{i\in I} v_i d^w_{ikt} > (t-k+1)C^{WR}|V(w)|$. *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:*

$$\text{Min} \sum_{u=k}^{t} \sum_{r\in R(w,u)} \left(\sum_{i\in I} v_i d^r_{iu} \right) y_{ru} \tag{5.55}$$

$$\text{s. t.} \sum_{u=k}^{t} \sum_{r\in R(w,u)} \left(\sum_{i\in I} v_i d^r_{iu} \right) y_{ru} \geq \sum_{i\in I} V^w_{kt} - C^{WR,w}_{kt} \tag{5.56}$$

$$y_{ru} \in \{0;1\} \qquad \forall\, u \in [k;t], r \in R(w,u). \tag{5.57}$$

The objective (5.55) *minimizes the total volume that must be delivered strictly before*

period k and constraint (5.56) *defines a lower bound on this minimum volume, which is the missing capacity. The value of the objective function is what we defined as* $C_{kt}^{min,w}$.

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

$$\sum_{i\in I}\sum_{t|\overline{y}_{it}^{p}=1}\left(1-y_{it}^{p}\right)+\sum_{t|\overline{y}_{it}^{p}=0}y_{it}^{p}\geq 1 \tag{5.58}$$

$$\sum_{w\in W}\sum_{t|\overline{y}_{t}^{w}>0}\left(1-y_{t}^{w}\right)+\sum_{w\in W}\sum_{t|\overline{y}_{t}^{w}=0}y_{t}^{w}\geq 1. \tag{5.59}$$

In our heuristic, we start by adding diversification constraints linked to the warehouses, i.e., constraints (5.59). Once we have performed a certain number of iterations of Step 2, we add the diversification constraints linked to the plant, i.e., constraints (5.58), and continue with new iterations of Step 2. If we add a new diversification constraint (5.58) related to the plant, we remove all the current diversification constraints (5.59) present in the model. On the contrary, if we add new diversification constraints (5.59) related to the warehouses, we do not remove any other diversification constraints present in the model. Note that we remove all diversification constraints after Step 4.

5.4.3 Step 3: solution of a second OWMR

In the third step of our heuristic, we solve a second OWMR. This OWMR does not integrate the routing decisions. The routes will actually 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. Therefore, we denote by $W(r,t)$ the warehouse assigned to retailer r for its demand in period t. Let $X_{rikt}^{W(r,t)}$ be the amount ordered by warehouse $W(r,t)$ 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 also $s^{W(r,t)}_{rikt}$ be the amount stocked at warehouse $W(r,t)$ at the end of period k to satisfy d^r_{it} and let s^r_{ikt} be the amount stocked at retailer r at the end of period k to satisfy d^w_{it}. We use the same visit variables z^r_{tw} and retailer setup variables y^r_t as in Section 5.3, but impose binary conditions on the z^r_{tw} variables. 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 O^w_{it} be the quantity of item i available at warehouse w in period t. This quantity is computed from the solution obtained in Step 2. If we denote by $\overline{X}^w_{itk}$ the optimal value of the X^w_{itk} variables obtained after an execution of Step 2, the available quantities are 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 is as follows:

$$\text{Min} \sum_{t \in T} \left(\sum_{r \in R} sc^r_t y^r_t + \sum_{r \in R} tc^r_{tW(r,t)} z^r_{tW(r,t)} + \sum_{r \in R} \sum_{k \leq t} \sum_{i \in I} \left(hc^{W(r,t)}_{ik} s^{W(r,t)}_{rikt} + hc^r_{ik} s^r_{ikt} \right) \right) \quad (5.60)$$

$$\text{s. t. } s^{W(r,t)}_{r,i,k-1,t} + X^{W(r,t)}_{rikt} = \sum_{v \in V_{rt}} X^r_{iktv} + s^{W(r,t)}_{rikt} \qquad \forall\, r \in R, i \in I, t \in T, k \leq t \in T \quad (5.61)$$

$$s^r_{i,k-1,t} + \sum_{v \in V_{rt}} X^r_{iktv} = \delta_{kt} d^r_{it} + (1 - \delta_{kt}) s^r_{ikt} \qquad \forall\, r \in R, i \in I, t \in T, k \leq t \in T \quad (5.62)$$

$$\sum_{r \in R(w,t)} \sum_{t \geq k} X^w_{rikt} \leq O^w_{ik} \qquad \forall\, w \in W, i \in I, k \in T \quad (5.63)$$

$$\sum_{r \in R(w,t)} \sum_{t \geq k} X^r_{iktv} \leq C^{WR} \qquad \forall\, w \in W, k \in T, v \in V(w) \quad (5.64)$$

$$X^r_{iktv} \leq d^r_{it} y^r_k \qquad \forall\, r \in R, i \in I, t \in T, k \leq t \in T, v \in V_{rt} \quad (5.65)$$

$$X^r_{iktv} \leq d^r_{it} z^r_{kw} \qquad \forall\, w \in W, k \leq t \in T, r \in R(w,t), i \in I \quad (5.66)$$

$$\sum_{w \in W} z^r_{tw} \leq 1 \qquad \forall\, r \in R, t \in T \quad (5.67)$$

$$X^w_{rikt}, s^w_{rikt} \geq 0 \qquad \forall\, r \in R, w \in W, t \in T, k \leq t \in T \quad (5.68)$$

$$s^r_{ikt} \geq 0 \qquad \forall\, r \in R, t \in T, k \leq t \in T \quad (5.69)$$

$$X^r_{iktv} \geq 0 \qquad \forall\, r \in R, t \in T, k \leq t \in T, v \in V_{rt} \quad (5.70)$$

$$y^r_{tv} \in \{0;1\} \qquad \forall\, r \in R, t \in T, v \in V_{rt} \quad (5.71)$$

$$z^r_{tw} \in \{0;1\} \qquad \forall\, r \in R, w \in W, t \in T. \quad (5.72)$$

Constraints (5.61)-(5.62) are the inventory balance constraints at the warehouse and retailer level, respectively. Constraints (5.63) make the link between the second and third steps of the heuristic, where the orders made at the warehouse level are limited by the orders made in the second step of the heuristic. Constraints (5.64) are the transportation capacity constraints. Constraints (5.65) are the setup constraints at the retailer level. Constraints (5.66) link the ordering and visit variables. Constraints (5.67) state that each retailer can be visited by a unique warehouse in each time period. Finally, constraints (5.68)-(5.72) define the bounds and domains of the decision variables.

Regarding the splitting possibilities, 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:

$$X^r_{iktv} \leq d^r_{it} z'^r_{kt} \qquad \forall\, r \in R, i \in I, k \leq t \in T, v \in V_{rt} \quad (5.73)$$

$$\sum_{k \leq t} z'^r_{kt} \leq 1 \qquad \forall\, r \in R, t \in T \quad (5.74)$$

$$z'^r_{kt} \in \{0;1\} \qquad \forall\, r \in R, k \leq t \in T. \quad (5.75)$$

Constraints (5.73) link the order variables to the new binary variables. Constraints (5.74) 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:

$$X^r_{iktv} \leq d^r_{it} z''^r_{kv} \qquad \forall\, r \in R, i \in I, k \leq t \in T, v \in V_{rt} \quad (5.76)$$

$$\sum_{v \in V(W(r,t))} z''^r_{tv} \leq 1 \qquad \forall\, r \in R, t \in T \quad (5.77)$$

$$z''^r_{tv} \in \{0;1\} \qquad \forall\, r \in R, t \in T, v \in V_{rt}. \quad (5.78)$$

Constraints (5.76) link the order variables to the new binary variables. Constraints (5.77) 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:

$$X^r_{iktv} \leq d^r_{it} Y^r_{ktv} \qquad \forall\, r \in R, i \in I, k \leq t \in T, v \in V_{rt} \qquad (5.79)$$

$$\sum_{k \leq t} \sum_{v \in V_{rt}} Y^r_{ktv} \leq 1 \qquad \forall\, r \in R, t \in T \qquad (5.80)$$

$$Y^r_{ktv} \in \{0;1\} \qquad \forall\, r \in R, k \leq t \in T, v \in V_{rt}. \qquad (5.81)$$

Constraints (5.79) link the order variables to the new binary variables. Constraints (5.80) actually state that the deliveries per period and vehicle cannot be split.

In order to obtain a feasible solution, we first need to construct routes. To construct the routes, we do not solve a TSP for each vehicle of each warehouse after each iteration of Step 3, based on the values of the retailer visit variables. Instead, we initially solve TSPs a priori 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 and 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. In case we identify a retailer which is visited but not present in the a priori tour, we actually solve a TSP. Such situation may happen if a retailer r is visited in period k for its demand in period t ($k < t$) and that $W(r,k) \neq W(r,t)$. In our experiments, this rarely happens. Based on these a priori tours and on the solution obtained in Step 3, we can construct the routes. The routes we construct follow the sequence of retailers in the a priori tours, but we skip the retailers that are not visited, i.e., the ones for which the value of the setup variable y^r_{tv} is equal to zero. This allows to construct a feasible solution for the whole problem and to actually 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 numerous cost update mechanisms, among which we 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 actually 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.

5.4.4 Step 4: diversification of the search

Once we have performed Steps 2 and 3, we need to 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. We have tested three different strategies to reassign the retailers: based on the cost per unit of demand, based on the number of retailers linked to each warehouse and based on the cost per unit transportation capacity available. In each strategy, for each time period t, we construct a set $W^+(t)$ containing the warehouses that can accept more retailers, and a set $W^-(t)$ containing the 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 $W^-(t)$. Note that we also tried to change the retailer that was the most costly but initial experiments have shown that choosing the furthest retailer led to a better performance of the heuristic. We give more details on each of these strategies in the following sections. In particular, we will explain how we built the sets $W^-(t)$. We will then have $W^+(t) = W \setminus W^-(t)$.

Reassignment based on the cost per unit of demand

In this strategy, 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_t^w by $\sum_{i \in I} d_{it}^w$. 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_t^w greater than $\overline{C}_t$. In the sequel we refer to this strategy as the CD strategy.

Reassignment based on the number of retailers

In this strategy, we compute the average number of retailers linked to each warehouse, for each time period. We denote by $\overline{R}_t$ this average. For each period t, the set $W^-(t)$ contains the warehouses that have a number of retailers greater than $\overline{R}_t$. In the sequel we refer to this strategy as the BW strategy.

Reassignment based on the cost per unit of transportation capacity

In this strategy, we compute the total cost C_t^w for each warehouse, in each time period, in the same way as with the CD strategy. We then divide this cost by $|V(w)|C^{WR}$, which represents the total transportation capacity available. We finally compute the average cost per period $\overline{C}_t$, over all the warehouses. For each period t, the set $W^-(t)$ contains the warehouses that have a cost C_t^w greater than $\overline{C}_t$. In the sequel we refer to this strategy as the CC strategy.

5.4.5 Step 5: improving the solution

Once we have obtained a final 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, despite the a priori tour generated initially, the best tour is not necessarily the subtour we can extract from this a priori tour. This last step is again performed using the Concorde solver (Applegate et al., 2011). Note that this last step exactly corresponds to the fifth step of the heuristic proposed by Solyalı and Süral (2017) for the PRP.

5.4.6 Pseudo-code for the top-down heuristic

A sketch of the full top-down heuristic is presented hereafter. In Algorithm 3, 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 3 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 the OWMR (5.43)-(5.52)
      Update the available quantities for each warehouse
      it_3 = 0
      while it_3 < it_R do
        Step 3: solve the OWMR (5.60)-(5.71)
        Build routes and apply the cost update mechanism
        it_3 = it_3 + 1
      end while
      it_2 = it_2 + 1
      Add diversification constraint (5.59)
    end while
    it1 = it1 + 1
    Add diversification constraint (5.58) and remove all diversification constraints
    (5.59)
  end while
  Step 4: reassign the retailers to warehouses, build a priori tours and remove all di-
  versification constraints (5.58) and (5.59)
end while
Step 5: improve the best routes by solving TSPs
```

5.5 A bottom-up heuristic

In this section we present a bottom-up heuristic. This heuristic represents a situation where the distribution decisions to the retailers lead the operational decisions of the com-

pany. 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 then 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 OWMR with the plant playing the part of the warehouse, and the warehouses playing the parts of the retailers. 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.

5.5.1 Step 1: solution of SI-ULSPs

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 weighted demand over all items. The weights used are the values of the holding costs for each item, i.e., $d_t^r = \sum_{i \in I} hc_{it}^r d_{it}^r$. The setup costs considered are the same as previously, i.e., the sc_t^r values, and do not incorporate any routing cost. These SI-ULSPs are solved by a backward dynamic programming algorithm as described in Pochet and Wolsey (2006). Solving these SI-ULSPs gives us the best replenishment plan for each retailer individually. Before proceeding to Step 2, we make some adjustments to these plans, to take into account both transportation and production capacity requirements. As in the top-down heuristic, 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 4. Let $I(t)$ be the set of items that are ordered in period t. In Algorithm 4, "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 between periods t and $t+1$ is minimized. The shift includes a new setup in period $t+1$ and the quantity ordered in period $t+1$ is $\sum_{i\in I} D^{r_1}_{i,t+1,t'-1}$, 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^r_t. In the sequel, we denote by $Y(r)$ the set of periods with an order placed by retailer r, and denote by $\overline{o}^r_{it}$ the quantity of item i ordered by retailer r in period t in the solution obtained from step 1. Note that we first adjust the replenishment plans to satisfy the transportation capacity requirements, and then to satisfy the production capacity requirements, if needed.

Algorithm 4 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 = $\sum_{i\in I} \alpha_i D^{r_1}_{i,t+1,t'-1}$
    capacityUsed = capacityUsed - capacityGained
  end while
end for

5.5.2 Step 2: solution of a facility location problem

Once we have a replenishment plan for each retailer, we turn to the assignment of retailers to warehouses. This is done by solving a problem similar to a facility location problem. 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:

$$\text{Min} \sum_{r\in R}\sum_{w\in W}\sum_{t\in T}\left(vc^r_{wt}X^r_{wt}+\sum_{i\in I}\sum_{v\in V(w)}\sum_{k\le t}\sum_{u=k}^{t-1}h^r_u\overline{o}^r_{it}a^r_{iktwv}\right) \tag{5.82}$$

$$\text{s. t.} \sum_{k\le t|k\in Y(r)}\sum_{w\in W}\sum_{v\in V(w)}a^r_{iktwv}=\chi^r_{it} \qquad \forall\, r\in R, i\in I, t\in T \tag{5.83}$$

$$a^r_{iktwv}\le X^r_{tw} \qquad \forall\, r\in R, i\in I, w\in W, k\le t\in T, v\in V(w) \tag{5.84}$$

$$\sum_{r\in R}\sum_{t\ge k}\sum_{i\in I}v_i\overline{o}^r_{it}a^r_{iktwv}\le C^{WR} \qquad \forall\, w\in W, k\in T, v\in V(w) \tag{5.85}$$

$$\sum_{w\in W}X^r_{tw}\le 1 \qquad \forall\, r\in R, t\in T \tag{5.86}$$

$$a^r_{iktwv}\ge 0 \qquad \forall\, r\in R, w\in W, i\in I, k\le t\in T, v\in V(w) \tag{5.87}$$

$$X^r_{tw}\in\{0;1\} \qquad \forall\, r\in R, w\in W, t\in T. \tag{5.88}$$

The objective function (5.82) minimizes the sum of the visiting costs and of the inventory holding costs at the retailer level. Constraints (5.83) are the demand satisfaction constraints. Constraints (5.84) link the assignment variables and the delivery variables. Constraints (5.85) are the transportation capacity constraints. Constraints (5.86) state that a retailer can be visited by one warehouse only in every period. Finally constraints (5.87)-(5.88) define the bounds and domains of the decision variables.

We take the delivery splitting possibilities into account in this second step of the bottom-up heuristic. 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 if vehicle v belonging to warehouse w visits retailer r in period k. We then add the following constraints:

$$a^r_{iktwv}\le X'^r_{kwv} \qquad \forall\, r\in R, w\in W, i\in I, k\le t\in T, v\in V(w) \tag{5.89}$$

$$X'^{r}_{kwv} \leq X^{r}_{kw} \qquad \forall\, r \in R, w \in W, k \in T, v \in V(w) \tag{5.90}$$

$$\sum_{v \in V(w)} X'^{r}_{kwv} \leq 1 \qquad \forall\, r \in R, w \in W, k \in T \tag{5.91}$$

$$X'^{r}_{twv} \in \{0;1\} \qquad \forall\, r \in R, w \in W, t \in T, v \in V(w). \tag{5.92}$$

Constraints (5.89) link the delivery variables and the new binary variables. Constraints (5.90) link the new binary variables and the assignment variables. Finally constraints (5.91) 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:

$$a^{r}_{iktwv} \leq X''^{r}_{kt} \qquad \forall\, r \in R, w \in W, i \in I, k \leq t \in T, v \in V(w) \tag{5.93}$$

$$\sum_{k \leq t} X''^{r}_{kt} \leq 1 \qquad \forall\, r \in R, t \in T \tag{5.94}$$

$$X''^{r}_{kt} \in \{0;1\} \qquad \forall\, r \in R, k \leq t \in T. \tag{5.95}$$

Constraints (5.93) link the delivery variables to the new binary variables. Constraints (5.94) impose that the demand of a specific time period for a specific retailer is served in at most one period.

In the objective function (5.82), the costs vc^{r}_{wt} linked to the visit variables are updated based on the routes constructed at each iteration of the heuristic. This is done in order to intensify the search and obtain better routes, in terms of costs. The routes are constructed by solving a TSP for each vehicle of each warehouse. For any period t and for any retailer r visited by vehicle v of warehouse w, we set the cost vc^{r}_{tw} to $c_{r_p r} + c_{r r_s} - c_{r_p r_s}$, where r_p and r_s are, in the current route of vehicle v, the facilities that are visited just before and after retailer r, respectively. If a retailer is visited by several vehicles, we sum those costs. For any retailer r not visited by warehouse w, we identify the best insertion in the current routes. We then set the cost vc^{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 that visited just before and after retailer r, respectively.

5.5.3 Step 3: solution of a OWMR

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 therefore have to solve an OWMR problem very 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 $\sum_{r\in R|\overline{X}^r_{tw}=1} \sum_{k\leq t} \sum_{v\in V(w)} \overline{o}^r_{ik}\overline{a}^r_{iktwv}$. 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 (5.43)-(5.52).

5.5.4 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 therefore 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 proceed differently. 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}.

5.5.5 Pseudo-code for the bottom-up heuristic

A sketch of the full top-down heuristic is presented hereafter. In Algorithm 5, it_R and it_W are used to represent the maximum number of iterations done to diversify and intensify the search, respectively.

Algorithm 5 Sketch of the bottom-up heuristic

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 4
  $it_2 = 0$
  while $it_2 < it_W$ **do**
    Step 2: solve the facility location problem (5.82)-(5.87)
    Build the routes from the solution to the facility location problem
    Update the costs vc^r_{wt}
    $it_2 = it_2 + 1$
  end while
  Compute the demand at each warehouse
  Step 3: solve the OWMR (5.43)-(5.52)
  Change the replenishment plans of the retailers
  $it = it + 1$
end while

5.6 Numerical results

In order to assess the performance of our heuristics, we conducted numerical experiments on small and large instances. We use the small instances to measure the performance of our heuristics based on lower and upper bounds given by the branch-and-cut algorithm presented in Section 5.3.3.

In all the experiments, 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 equal to 2 or 1.75. We set the capacity of the trucks used to make the deliveries between the plant and the warehouses as a given factor C'^{PW} of the average volume that the total demand represents. This transportation capacity is therefore $C^{PW} = C'^{PW} \sum_{r \in R} \sum_{i \in I} \sum_{t \in T} v_i d^r_{it} \,/\, |T|$. We set the factor C'^{PW} equal to 1. We set the capacity of the trucks used to make the deliveries between the warehouses and the retailers 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 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. Finally, 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. The production time vt_i to produce one unit of item i is set equal to 1 for all items.

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 $d_{it}^r = d_i^r \; \forall \; t \in T, r \in R, i \in I$. 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.

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.

We performed the experiments on a 6.67 GHz Intel Xeon X5650 Westmere processor with one thread. For the experiments, 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 3 hours.

The CPU time limit imposed for the heuristics is set to one hour. 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. This choice of values is based on the results of initial experiments. 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 two consecutive integer solutions is too large compared to the improvement obtained. In more details, let z_i and

z_{i+1} be the objective value of the two consecutive integer solutions i and $i+1$, respectively. Let also cpu_i and cpu_{i+1} be the moments when solutions i and $i+1$ have been obtained, respectively. We finally 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 very long, even if we change CPLEX parameters or use multiple threads.

For the bottom-up heuristic, the number of iterations is set equal to 10 and 5 for Steps 2 and 4, respectively. This choice of values is based on the results of initial experiments.

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 the solution found by our branch-and-cut algorithm. These two measures are denoted by Total CPU and GAP 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 further 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 also report the CPU time taken for Steps 1, 2 and 3. These values are denoted by CPU_1, CPU_2, CPU_3, respectively. All the CPU times reported are expressed in seconds.

5.6.1 Results on small instances

In the small instances, 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. The results reported in this section are to be compared with the results obtained by our branch-and-cut algorithm directly, which are reported in Table 5.4. In Table 5.4, the first two columns indicate the splitting possibilities. The best upper bound and lower bound are given by BUB and BLB, respectively. The columns CPU time, Nodes and Optimality gap

Table 5.4 – Results obtained by CPLEX for the small instances

Delivery splitting	Demand splitting	BUB	BLB	CPU time	Nodes	Optimality gap
x	x	298031	296986	2249	18463	0.27
x	✓	297134	295759	3019	14316	0.42
✓	x	300911	296185	2523	21989	0.84
✓	✓	301457	295444	3258	14989	1.13

Table 5.5 – Results of the top-down heuristic for small instances

Delivery splitting	Demand splitting	Diversification strategy	BUB	CPU_2	CPU_3	CPU_{TSP}	Total CPU	GAP (%)
x	x	BW	311961	5.6	285	0.3	290.9	5.82
		CC	312276	5.7	256.2	0.3	262.2	5.84
		CD	311937	5.6	211.3	0.2	217.1	5.84
x	✓	BW	310589	5.6	883	0.5	889	4.52
		CC	312178	5.7	886.1	0.5	892.3	5.06
		CD	309593	5.5	860.3	0.4	866.3	4.19
✓	x	BW	317371	5.8	352.7	0.4	358.9	5.35
		CC	317374	5.8	287	0.3	293.2	5.36
		CD	317429	5.8	245.3	0.2	251.3	5.37
✓	✓	BW	312450	5.6	357.1	0.5	363.2	5.14
		CC	312516	5.7	300.2	0.4	306.4	5.17
		CD	312498	5.6	235.9	0.4	241.9	5.16

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.

Results for the top-down heuristic

Table 5.5 gives the results obtained on the small instances for the top-down heuristic. In Table 5.5, one can see that the solutions obtained by this heuristic are of acceptable quality, with a gap of around 5% compared to the solution found by the branch-and-cut algorithm. Given the optimality gaps obtained by CPLEX directly (see Table 5.4), this gives a gap of around 5% also, between the solution of our top-down heuristic and the lower bound obtained by CPLEX, on the small instances. Note, however, that these solutions are obtained in less time compared to the branch-and-cut algorithm.

We further note that the diversification strategy has little impact on the performance of the top-down heuristic. The quality of the solutions obtained is almost the same regardless

Table 5.6 – Results of the bottom-up heuristic for small instances

Delivery splitting	Demand splitting	BUB	CPU_1	CPU_2	CPU_3	Total CPU	GAP (%)
x	x	316182	0	811	1.4	813	7.09
x	✓	309468	0	472	1.4	474	4.17
✓	x	313820	0	1620	0.9	1622	4.29
✓	✓	309596	0	1726	1.1	1728	2.7

of the diversification mechanism used. We just note that when we diversify the search based on the cost per unit of demand, i.e., when we use strategy CD, the CPU time taken is slightly less than with the other two diversification strategies.

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. We finally note that the CPU time taken for the improvement of the routes in the best solution obtained is negligible.

Regarding the case where we allow for split demand only, we note that the second step takes substantially more CPU time. For this case, we further note that the BUB are more different for the three strategies, with CD providing the best results.

Results for the bottom-up heuristic

Table 5.6 gives the results obtained on the small instances for the bottom-up heuristic. In Table 5.6, 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 similar to that of the solutions given by the top-down heuristic if we have demand splitting, but worse if we do not. We still have a gap of around 4-7% 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 2.7 and to 4.29 % if we further allow demand splitting or not, respectively.

This increase in solution quality comes, however, with a large increase of CPU time.

When we allow delivery splitting, the CPU time now reaches around 1700 seconds, compared to around 300 seconds with the top-down approach. This change in CPU time can be explained by the complexity of the FL problem compared to the second OWMR solved in the top-down heuristic. Indeed, the FL problem in the bottom-up heuristic and the second OWMR in the top-down heuristic play the same role, which is the intensification to construct routes of good quality. However, the FL problem is bigger in size, due to a higher number of decision variables, in particular the assignment variables a^r_{iktwv}. For the bottom-up heuristic, the CPU time taken in Step 1 is too low to be measured in practice. 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.

Influence of the splitting possibilities

We now analyze the results obtained in general 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 smaller than in the other settings, as reported in Table 5.4. This may sound counter-intuitive because of the additional restrictions imposed. However, as there are fewer possibilities in such a case compared to the other settings with splitting possibilities, it seems to help the solver find a solution quickly. We also note that we cannot directly compare the costs of the solutions with demand splitting only compared to the costs of the solutions with delivery splitting only. When we have no splitting at all, the top-down approach obtains the best results. When we have delivery splitting (only or with demand splitting allowed too), the bottom-up approach obtains the best performance, in terms of quality of the solution. Finally, if we have demand splitting only, the two heuristics have similar performance. This indicates that the bottom-up approach is more suitable for a setting with delivery splitting.

As far as our heuristics are concerned, they have different performance depending on the splitting possibilities. As illustrated in Table 5.5, 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.

On the contrary, the bottom-up approach seems to better handle the splitting possibilities. Indeed, one can see in Table 5.6 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.

We further analyzed the proportion of time with demand and delivery splitting that actually occurs in the integrated plans. For the top-down heuristic, demand splitting occurs for the demand of retailer r in period t if $\sum_{k \leq t} z'^{r}_{kt} > 1$, and delivery splitting occurs for the demand of retailer r in period t if $\sum_{v \in V(W(r,t))} z''^{r}_{tv} > 1$. For the bottom-up heuristic, demand splitting occurs for the demand of retailer r in period t if $\sum_{k \leq t} X''^{r}_{kt} > 1$ and delivery splitting occurs for the demand of retailer r in period t if $\sum_{v \in V(w)} X'^{r}_{twv} > 1$. Table 5.7 illustrates the proportion of demands where delivery and/or demand splitting occurs, depending on the splitting possibilities. 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. For both the top-down and bottom-up heuristics, we define a sequential approach as an approach with no diversification iterations. There are, however, still intensification iterations to improves the routes constructed. 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 $\frac{z^{int} - z^{seq}}{z^{int}}$.

In Table 5.7, one can see that for both 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

Table 5.7 – Proportion of splitting possibilities for the small instances

Delivery splitting	Demand splitting	Top-down heuristic Del split (%)	Top-down heuristic Dem split (%)	Top-down heuristic Gap seq (%)	Bottom-up heuristic Del split (%)	Bottom-up heuristic Dem split (%)	Bottom-up heuristic Gap seq (%)	CPLEX Del split (%)	CPLEX Dem split (%)
x	x	0	0	-0.2	0	0	-1	0	0
x	✓	0	0.9	-0.1	0	2.2	-14.3	0	4.12
✓	x	31.7	0	-0.2	73.5	0	-5	2.45	0
✓	✓	32.9	0.8	-0.4	64.6	1.9	-12.7	2.43	3.89

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. We also note that 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 fact that the bottom-up approach first takes advantage of the splitting possibilities is one reason for these higher values compared to the top-down approach. The second reason is the fact 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.

Results with inventory restrictions at the retailer level

To better understand the settings that facilitates demand or delivery splitting, we further performed experiments with inventory limits at the retailer level. Let I' be a factor to limit the inventory kept on hand. We add the following constraints:

$$\sum_{i\in I}\sum_{w\in W}\sum_{u=1}^{t}\sum_{l=u}^{|T|} v_i\,(q^r_{iulw} - d^r_{i1t}) \leq I'\frac{d^r_{1|T|}}{|T|} \qquad \forall\, r\in R, t\in T \tag{5.96}$$

Table 5.8 – Results of the heuristics on small instances with inventory limit at the retailer's level

Delivery splitting	Demand splitting	Top-down heuristic						Bottom-up heuristic					CPLEX	
		BUB	CPU_2	CPU_3	CPU_{TSP}	Total CPU	GAP (%)	BUB	CPU_2	CPU_3	Total CPU	GAP (%)	CPU CPLEX	CPLEX GAP (%)
x	x	313036	5.7	271.9	0.9	280.2	4.8	310668	605.6	1.3	607.2	4.1	3967.9	1.5
x	✓	311162	5.6	621.6	0.3	628.5	5.2	310023	446.9	1.4	448.6	5	3077.3	1.3
✓	x	312864	5.6	172.0	0.3	179	4.7	299199	1073.4	1	1074.8	0.7	3832	1.3
✓	✓	312579	5.7	145.6	0.3	152.4	4.9	303507	1359.6	1	1360.9	0.5	3325	1.3

$$\sum_{i\in I}\sum_{k\leq t} v_i X^{W(r,t)}_{rikt} \leq I' \frac{d^r_{1|T|}}{|T|} \qquad \forall\, r \in R, t \in T \tag{5.97}$$

$$\sum_{i\in I}\sum_{k\leq t|k\in Y(r)}\sum_{w\in W}\sum_{v\in V(w)} v_i a^r_{iktwv} \leq I' \frac{d^r_{1|T|}}{|T|} \qquad \forall\, r \in R, t \in T. \tag{5.98}$$

Constraints (5.96), (5.97) and (5.98) 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. Tables 5.8 illustrates the average results obtained with an inventory factor equal to 1 and 2. Note that for the top-down approach, we only report results with the use of the CD strategy for the update of the assignment of retailers to warehouses since it obtained the best results in the previous experiments.

In Table 5.8, one can see that the values of the gap obtained with inventory limit are roughly the same compared to the values obtained without inventory limit. We only note that when delivery splitting is allowed, the bottom-up approach obtains solutions of high quality with a gap less than 0.7% compared to the solution obtained by CPLEX. The gap reported by CPLEX at the end of the time limit is always around 1.3%, indicating that the quality of the solutions obtained by our approaches is around 6%. Regarding the CPU time taken to solve the instances, the same conclusions stands as for the instances without inventory limit. However, for CPLEX, the CPU times are bigger, illustrating the higher difficulty of those instances. The fact that our approaches do not face such increase is a clear advantage of our approaches. Finally, regarding the cost of the solutions obtained, we can draw the same conclusions as for the instances without inventory limit. The top-

Table 5.9 – Analysis on small instances with inventory limit at the retailer's level

Delivery splitting	Demand splitting	Top-down heuristic Del split (%)	Top-down heuristic Dem split (%)	Top-down heuristic Gap seq (%)	Bottom-up heuristic Del split (%)	Bottom-up heuristic Dem split (%)	Bottom-up heuristic Gap seq (%)	CPLEX Del split (%)	CPLEX Dem split (%)
x	x	0	0	-0.2	0	0	-1.1	0	0
x	✓	0	1.3	-0.1	0	2.4	-1.4	0	4.4
✓	x	32.4	0	-0.4	73.4	0	-5	2.4	0
✓	✓	30.8	0.5	-0.4	64.6	1.9	-12.7	2.3	4.1

down approach still obtains solutions of similar cost regardless of the splitting possibilities while the bottom-up approach takes more advantage of the splitting possibilities.

We further analyzed the proportion of demands with delivery or demand splitting for those instances with inventory limit at the retailer level. Table 5.9 reports the results obtained. In Table 5.9, we observe similar findings compared to the instances with no inventory limit. There is always more delivery splitting than demand splitting and the gains compared to a sequential approach are higher for the bottom-up heuristic. We only note that for the bottom-up approach, the gain compared to a sequential approach with demand splitting only is much lower compared to instances with no inventory limit. Indeed, demand splitting induces an increase of inventory on hand at the retailer level whereas inventory requirements limit such possibility.

Sensitivity analysis

We finally performed a sensitivity analysis on the small instances. The numbers reported in Tables 5.10-5.17 illustrate the results obtained by our two heuristics depending on the values of the parameters we have in the problem data. We carry this sensitivity analysis depending on the splitting possibilities, the number of trucks, items, retailers and warehouses, the value of the production and transportation capacities, the way the demand and the setup costs are generated, and the potential inventory limit at the retailer level. In Tables 5.10-5.17, the first two columns indicate the parameter studied and its value used in the experiments with small instances, respectively. The rest of the columns are the same as the ones used in Sections 5.6.1 and 5.6.1.

Table 5.10 – Sensitivity analysis the top-down heuristic, no demand nor delivery splitting

Parameter	Value	BUB	CPU_2	CPU_3	CPU_{TSP}	Total CPU	GAP (%)	Del split (%)	Dem split (%)	Gap seq (%)
$\|V\|$	2	312426	5.6	260.4	0.9	268.6	2.87	0	0	-0.2
	5	314063	5.7	271	0.8	279.1	6.89	0	0	-0.2
$\|I\|$	1	179912	1.6	217.8	0.6	222	5.53	0	0	-0.2
	3	300249	4.3	227.9	0.9	234.4	5.63	0	0	-0.2
	5	459572	11	351.4	1.1	365.2	3.53	0	0	-0.1
$\|R\|$	10	310331	5.4	54.9	0.4	61.4	5.37	0	0	-0.1
	20	316158	5.8	476.6	1.3	486.3	4.41	0	0	-0.3
$\|W\|$	2	310519	2.9	88.4	0.8	92.6	3.29	0	0	-0.1
	4	315970	8.3	443	1	455.1	6.45	0	0	-0.2
C^{WR}	0.4	313405	5.7	301.6	0.8	309.8	3.36	0	0	-0.2
	0.5	313085	5.6	229.8	0.9	237.9	6.42	0	0	-0.2
C^P	1.75	329205	6.1	246.7	0.8	255.3	4.56	0	0	-0.2
	2	297284	5.1	284.7	1	292.4	5.22	0	0	-0.2
Demand	DD	314272	5.3	264.7	0.9	272.5	5.01	0	0	-0.2
	SD	312217	6	266.7	0.9	275.2	4.78	0	0	-0.2
Setup costs	DF	310215	5.7	268.5	0.9	276.7	5.08	0	0	-0.2
	SF	316274	5.6	262.9	0.9	271	4.7	0	0	-0.2
Inventory	1	313245	5.6	265.7	0.9	273.9	4.89	0	0	-0.2
	2	312827	5.7	278	0.9	286.5	4.76	0	0	-0.2
	inf	311937	5.6	211.3	0.2	219	4.78	0	0	-0.2
Average		313245	5.6	265.7	0.9	273.9	4.89	0	0	-0.2

Table 5.11 – Sensitivity analysis the top-down heuristic, demand splitting only

Parameter	Value	BUB	CPU_2	CPU_3	CPU_{TSP}	Total CPU	GAP (%)	Del split (%)	Dem split (%)	Gap seq (%)
$\|V\|$	2	311007	5.6	209.7	0.6	216.9	4.47	0	2.9	-0.2
	5	311094	5.6	1217.3	0.5	1224.3	5.94	0	0.1	-0.1
$\|I\|$	1	178830	1.6	825.5	0.3	828.5	5.33	0	0.7	-0.2
	3	297749	4.3	507.5	0.5	513.2	6.04	0	2.4	-0.2
	5	456573	10.9	807.4	0.9	820.1	4.26	0	1.5	-0.1
$\|R\|$	10	308940	5.4	276.2	0.3	282.2	5.08	0	1.7	-0.1
	20	313162	5.8	1150.7	0.8	1159.1	5.33	0	1.3	-0.2
$\|W\|$	2	307835	2.8	898.6	0.5	902.1	2.79	0	2.5	-0.1
	4	314267	8.4	528.3	0.6	539.1	7.61	0	0.5	-0.2
C^{WR}	0.4	311096	5.6	768.3	0.5	775.4	4.6	0	2	-0.1
	0.5	311005	5.6	658.6	0.6	665.8	5.81	0	1	-0.1
C^P	1.75	327039	6.1	672.7	0.6	680.5	5.07	0	1.5	-0.1
	2	295062	5.1	754.2	0.4	760.7	5.34	0	1.5	-0.2
Demand	DD	311996	5.3	636.7	0.5	643.5	5.05	0	1.5	-0.2
	SD	310105	5.9	790.3	0.5	797.7	5.36	0	1.6	-0.1
Setup costs	DF	308045	5.7	686.1	0.5	693.3	5.42	0	1.5	-0.2
	SF	314056	5.5	740.8	0.5	748	4.99	0	1.5	-0.1
Inventory	1	311405	5.6	494.4	0.2	501	5.25	0	0.9	-0.1
	2	310920	5.6	748.8	0.5	755.9	5.11	0	1.7	-0.1
	inf	310828	5.6	897.1	0.9	904.9	5.26	0	2	-0.1
Average		311051	5.6	713.5	0.5	720.6	5.21	0	1.5	-0.1

Table 5.12 – Sensitivity analysis the top-down heuristic, delivery splitting only

Parameter	Value	BUB	CPU_2	CPU_3	CPU_{TSP}	Total CPU	GAP (%)	Del split (%)	Dem split (%)	Gap seq (%)
$\|V\|$	2	312341	5.5	71.2	0.1	77.6	2.43	25.7	0	-0.2
	5	313711	5.7	142.2	0.4	149.1	6.84	40.3	0	-0.5
$\|I\|$	1	180246	1.6	88.6	0.2	90.9	5.8	7.8	0	-0.5
	3	299882	4.3	90.9	0.2	96.1	4.91	43	0	-0.3
	5	458950	11	140.6	0.5	153	3.32	48.3	0	-0.2
$\|R\|$	10	310235	5.4	47.1	0.1	53.2	5.28	33.7	0	-0.2
	20	315817	5.8	166.3	0.5	173.5	4.02	32.4	0	-0.5
$\|W\|$	2	310215	3	68.2	0.2	72	3.09	38.4	0	-0.3
	4	315838	8.3	145.2	0.4	154.7	6.21	27.6	0	-0.3
C^{WR}	0.4	313168	5.7	100.5	0.3	107.1	3.06	35.2	0	-0.3
	0.5	312884	5.6	112.9	0.2	119.5	6.26	30.9	0	-0.3
C^P	1.75	328983	6.1	114.5	0.5	121.6	4.59	32.8	0	-0.3
	2	297070	5.1	98.9	0.1	105	4.73	33.2	0	-0.3
Demand	DD	313989	5.3	97.1	0.3	103.5	4.21	32.7	0	-0.3
	SD	312064	5.9	116.3	0.3	123.2	5.1	33.3	0	-0.3
Setup costs	DF	310022	5.7	107.8	0.2	114.5	4.64	32.6	0	-0.3
	SF	316031	5.6	105.6	0.3	112.1	4.68	33.4	0	-0.3
Inventory	1	313026	5.6	106.7	0.3	113.3	4.66	33	0	-0.3
	2	312701	5.7	237.5	0.3	244.6	5.28	31.8	0	-0.4
	inf	313026	5.6	106.7	0.3	113.3	4.97	33	0	-0.2
Average		313026	5.6	106.7	0.3	113.3	4.66	33	0	-0.3

Table 5.13 – Sensitivity analysis the top-down heuristic, demand and delivery splitting

Parameter	Value	BUB	CPU_2	CPU_3	CPU_{TSP}	Total CPU	GAP (%)	Del split (%)	Dem split (%)	Gap seq (%)
$\|V\|$	2	312007	5.7	137.6	0.4	144.9	3.38	28.9	0.7	-0.2
	5	313201	5.7	235.5	0.4	243.1	6.55	33.4	0	-0.6
$\|I\|$	1	179775	1.6	177	0.3	179.8	5.68	16.9	0.1	-0.7
	3	299415	4.2	174.1	0.2	180	5.55	35.7	0.6	-0.3
	5	458623	11.2	208.4	0.6	222.2	3.66	40.8	0.3	-0.2
$\|R\|$	10	309950	5.4	41.7	0.2	47.7	5.43	31.7	0.4	-0.1
	20	315259	5.9	331.3	0.6	340.3	4.49	30.5	0.3	-0.7
$\|W\|$	2	309875	2.9	63.9	0.4	67.6	3.41	36.7	0.7	-0.5
	4	315333	8.4	309.1	0.4	320.4	6.51	25.6	0	-0.3
C^{WR}	0.4	312797	5.7	174	0.3	181.5	3.72	33.9	0.5	-0.4
	0.5	312411	5.6	199	0.4	206.6	6.22	28.3	0.2	-0.4
C^P	1.75	328574	6.2	198.4	0.4	206.7	4.7	31.1	0.4	-0.4
	2	296634	5.2	174.6	0.4	181.3	5.23	31.2	0.4	-0.4
Demand	DD	313481	5.4	184	0.4	191.2	4.67	30	0.3	-0.4
	SD	311727	6	189	0.4	196.8	5.25	32.3	0.4	-0.4
Setup costs	DF	309599	5.7	187.8	0.4	195.3	5.09	31.1	0.4	-0.4
	SF	315609	5.6	185.2	0.4	192.7	4.83	31.1	0.4	-0.4
Inventory	1	312719	5.7	87.5	0.2	94.2	4.81	29.6	0.3	-0.3
	2	312438	5.6	203.7	0.4	210.7	4.95	32.1	0.8	-0.5
	inf	312655	5.7	268.3	0.6	277.2	5.13	31.7	0	-0.4
Average		312604	5.7	186.5	0.4	194	4.96	31.1	0.4	-0.4

As illustrated in Table 5.10, when there is no demand splitting, the results obtained by the top-down heuristic are somehow homogeneous. The only differences that appear concern the number of retailers and warehouses. Indeed, one can see that with a higher number of retailers or warehouses, the CPU time tends to increase quickly. This is explained by the growing size of the different OWMRs solved in steps 2 and 3. The increase in the number of warehouses seems to be a setting that affects the solution quality more than the other parameters. Indeed, when we go from 2 warehouses to 4 warehouses, the gap with the solution given by our branch-and-cut algorithm goes from 3.41 % to 8.24 %. We can also point out that a lower value of the transportation capacity of the trucks used at the warehouses has a positive impact on the performance of the top-down heuristic. The gap, the CPU time and solution cost all decrease when we have tighter capacity. This is explained by a better value of the lower bound in both OWMRs to be solved in the heuristics. In Table 5.10, the number of trucks and items, and the way the demand and the setup cots are generated do not seem to have an influence on the results of the top-down heuristic. The conclusions drawn here also stand for the case where we allow for delivery splitting only and demand and delivery splitting as illustrated in Tables 5.12 and 5.13, respectively. In those last two cases, we just note that the CPU time taken to solve instances with tighter production capacity increases.

Similar findings are reported in Table 5.11 when we allow demand splitting only. We just note that the CPU time taken to solve instances with only 2 warehouses is higher than when we have 4 warehouses. In such a setting, a tighter capacity, whether at the production plant or for transportation between the warehouses and the retailers, has a less positive impact compared to the situation with no splitting at all. This can be seen by the cost of the solution, the CPU time taken and the gap reported.

Tables 5.14-5.17 report the results of the sensitivity analysis for the bottom-up heuristic. The conclusions drawn on the top-down heuristic for the case where we do not allow for splitting at all, and where we allow for demand splitting only still stand for the bottom-up heuristic, as illustrated in Tables 5.14 and 5.15, respectively. For the case with demand

Table 5.14 – Sensitivity analysis the bottom-up heuristic, no demand nor delivery splitting

Parameter	Value	BUB	CPU_1	CPU_2	CPU_3	Total CPU	GAP (%)	Del split (%)	Dem split (%)	Gap seq (%)
$\|V\|$	2	310063	0	726.6	1.4	728.2	2.2	0	0	-1.2
	5	314630	0	645.9	1.3	647.7	6	0	0	-7.1
$\|I\|$	1	179318	0	361.3	0.1	361.5	5.5	0	0	-7.5
	3	297460	0	858.1	0.8	859.2	4.6	0	0	-1.5
	5	455730	0	829.7	3.1	833.5	2.3	0	0	-2
$\|R\|$	10	309000	0	502.8	1.1	504.1	4.5	0	0	-4.5
	20	315609	0	870	1.5	872	3.7	0	0	-3.6
$\|W\|$	2	310865	0	711.7	0.9	712.8	3.1	0	0	-4
	4	313762	0	662	1.7	664.3	5	0	0	-4.2
C^{WR}	0.4	315363	0	641.3	1.4	643.1	2.8	0	0	-6.4
	0.5	309341	0	731.2	1.2	732.9	5.3	0	0	-1.6
C^P	1.75	326335	0	664.5	1.5	666.4	3.8	0	0	-1.2
	2	297853	0	709.9	1.2	711.4	4.4	0	0	-7
Demand	DD	313447	0	683.4	1.3	685	4.1	0	0	-4.5
	SD	311183	0	690.3	1.4	692.1	4.1	0	0	-3.6
Setup costs	DF	309289	0	687.2	1.2	688.8	4.3	0	0	-4.1
	SF	315336	0	686.5	1.4	688.3	3.9	0	0	-4
Inventory	1	310668	0	641.4	1.3	642.9	4.1	0	0	-1.1
	2	310668	0	569.9	1.3	571.4	4.1	0	0	-1.1
	inf	315797	0	858.9	1.4	860.9	4.1	0	0	-5.7
Average		312378	0	690.1	1.3	691.8	4.1	0	0	-2.6

Table 5.15 – Sensitivity analysis the bottom-up heuristic, demand splitting only

Parameter	Value	BUB	CPU_1	CPU_2	CPU_3	Total CPU	GAP (%)	Del split (%)	Dem split (%)	Gap seq (%)
$\|V\|$	2	305025	0	885.8	1.2	887.1	0.4	76.1	0	-10.9
	5	294894	0	1341.7	0.9	1343	1.2	71.1	0	-6.8
$\|I\|$	1	173768	0	917.6	0.1	917.7	2.3	40.3	0	-5.2
	3	287573	0	1163.6	0.6	1164.5	0.7	83.9	0	-9.1
	5	440903	0	1270.1	2.4	1273	-0.7	96.9	0	-33
$\|R\|$	10	298462	0	957.6	1	958.8	1.9	74.7	0	-8.2
	20	301357	0	1275.3	1	1276.7	-0.3	72.5	0	-9.5
$\|W\|$	2	301535	0	1098.7	0.9	1099.8	0.7	72.4	0	-8.6
	4	298283	0	1133.5	1.2	1135.1	0.9	74.8	0	-9
C^{WR}	0.4	300806	0	910.3	1.1	911.7	-0.2	71.6	0	-13.4
	0.5	299047	0	1312.9	0.9	1314.1	1.7	75.4	0	-2.8
C^P	1.75	315052	0	1203.3	1.1	1204.7	1.4	71.3	0	-5.1
	2	284440	0	1027.2	0.9	1028.4	0.1	75.9	0	-12.5
Demand	DD	302192	0	1107.4	1	1108.7	0.9	73.6	0	-8.1
	SD	297615	0	1124.9	1.1	1126.3	0.7	73.5	0	-9.6
Setup costs	DF	296354	0	1117	1	1118.3	0.7	75.6	0	-10.2
	SF	303447	0	1115.2	1.1	1116.6	0.9	71.6	0	-7.7
Inventory	1	300772	0	1321.5	1	1322.8	0.4	73.7	0	-2.3
	2	297627	0	825.4	1.1	826.7	1	73.3	0	-7.6
	inf	301350	0	1200.4	1	1201.7	1	73.8	0	-16
Average		299916	0	1115.7	1	1117.1	0.8	73.6	0	-8.6

Table 5.16 – Sensitivity analysis the bottom-up heuristic, delivery splitting only

Parameter	Value	BUB	CPU_1	CPU_2	CPU_3	Total CPU	GAP (%)	Del split (%)	Dem split (%)	Gap seq (%)
$\|V\|$	2	310960	0	532.4	1.5	534	4.5	0	2.3	-10.2
	5	309349	0	503.2	1.4	505.1	5.7	0	2.4	-1.2
$\|I\|$	1	180975	0	437.8	0.1	438	6.6	0	2.3	-3.7
	3	296144	0	531.9	0.8	532.9	5.4	0	2.5	-7.8
	5	453414	0	583.2	3.4	587.1	3.4	0	1.7	-13
$\|R\|$	10	307339	0	272.3	1.3	273.8	4.7	0	2.4	-5.2
	20	312954	0	763.5	1.6	765.5	5.6	0	0.8	-6.9
$\|W\|$	2	309491	0	542.1	1	543.2	3.5	0	2.4	-6
	4	310793	0	493.1	1.9	495.4	6.8	0	2.3	-5.9
C^{WR}	0.4	311074	0	420.9	1.5	422.7	4.8	0	2.3	-10.1
	0.5	309239	0	611.3	1.4	613	5.5	0	2.2	-1.1
C^P	1.75	325988	0	533.8	1.6	535.7	5	0	1.9	-1.4
	2	293843	0	500.9	1.3	502.5	5.2	0	2.6	-10.2
Demand	DD	310950	0	514	1.3	515.6	5	0	2.3	-5.7
	SD	309334	0	521.2	1.5	523	5.3	0	2.3	-6.2
Setup costs	DF	307021	0	520.4	1.3	522	5.4	0	2.2	-6
	SF	313268	0	514.7	1.6	516.6	4.9	0	2.4	-5.9
Inventory	1	310024	0	507.7	1.4	509.4	5	0	2.4	-1.4
	2	310024	0	386.2	1.4	387.9	5.1	0	2.4	-1.4
	inf	310390	0	665.1	1.4	666.8	5.3	0	2.2	-14.3
Average		310146	0	519.6	1.4	521.4	5.1	0	2.3	-5.7

Table 5.17 – Sensitivity analysis the bottom-up heuristic, demand and delivery splitting

Parameter	Value	BUB	CPU_1	CPU_2	CPU_3	Total CPU	GAP (%)	Del split (%)	Dem split (%)	Gap seq (%)
$\|V\|$	2	304034	0	924.8	1.1	926.1	0.8	49.6	2.1	-11
	5	300403	0	1842.4	0.9	1843.7	0.4	79.7	0	-20.1
$\|I\|$	1	172171	0	1338.9	0.1	1339.1	1.2	38.2	2.1	-13.3
	3	287053	0	1487.8	0.6	1488.6	1	73.1	0	-17
	5	440770	0	1314.5	2.3	1317.3	-0.4	81.1	0	-22.2
$\|R\|$	10	299349	0	1314.5	1.1	1315.8	1.5	62.9	3.3	-9.4
	20	305200	0	1448.6	1	1449.9	-0.4	66.2	1.3	-21.3
$\|W\|$	2	304782	0	1392.7	0.9	1393.7	0.9	65.4	2.1	-14.9
	4	299698	0	1368.5	1.2	1370.1	0.3	63.7	0	-14.9
C^{WR}	0.4	302406	0	1206.1	1.1	1207.4	-0.3	67.7	2.1	-15
	0.5	302056	0	1554.2	1	1555.5	1.5	61.4	0	-14.8
C^P	1.75	319049	0	1451.8	1.1	1453.1	1.3	61.1	0	-8.1
	2	284826	0	1306.8	1	1308.1	-0.2	68	2.1	-24
Demand	DD	304491	0	1377.9	1	1379.1	0.8	70	3.3	-14.6
	SD	299972	0	1383.2	1.1	1384.5	0.4	59	1.3	-15.3
Setup costs	DF	298582	0	1383.6	1	1384.8	0.5	66.2	2.7	-16.7
	SF	305883	0	1377.5	1.1	1378.9	0.7	62.9	0.8	-13.6
Inventory	1	299643	0	1555.5	1	1556.7	0.3	64.3	1.9	-1.8
	2	307371	0	1163.8	1.1	1165.1	0.6	64.9	1.9	-23.7
	inf	299907	0	1409.5	1	1410.8	0.8	64.5	3.3	-17.6
Average		302307	0	1376.3	1	1377.5	0.6	64.5	2.4	-14.4

splitting (only or with delivery splitting too), there are some more insights that can also explain the results already reported in Section 5.6.1. First, the values of the gap are in general lower than in Tables 5.10-5.15 and the CPU times are larger. Regarding the values of the gap, there is a clear difference when the number of trucks or items increase. When the number of trucks decreases, the gap is lower. On the contrary, when the number of items increases, the gap decreases. This may be explained by a larger use of the available capacities. With few vehicles and multiple items, the gaps are around 1%, illustrating a good performance of the bottom-up heuristic for those values of the parameters. When the number of retailers or warehouses increases, the values of the gap decreases. Higher values of those parameters, making in theory the problem harder to solve, leads to a better performance of the bottom-up approach. Finally, when the transportation capacity gets tighter, the gap decreases, and when the production capacity gets tighter, the gap increases.

Regarding the splitting possibilities, the results obtained by the top-down approach do not lead to clear conclusions. Indeed, the cost of the solution obtained is roughly the same regardless of the splitting possibilities. With the results of the top-down approach, we cannot identify settings that are more beneficial for demand or delivery splitting only, both, or none of them. On the contrary, the results of the bottom-up approach highlight more differences for the cost of the solution depending on the splitting possibilities. In particular, when the number of available trucks is high, one can take easily advantage of any splitting possibilities. This is expected since having more vehicles leads to more flexibility regarding the routes that are constructed. With a high number of items, delivery splitting only and both delivery and demand splitting lead to solutions of a lower cost compared to the case where there is no delivery nor demand splitting. It is expected since the fixed cost for delivery will be shared among more items. Finally, when the production capacity is tighter, it also leads to situations more beneficial for delivery splitting only or both delivery and demand splitting. With a tight production capacity, there is also more gain compared to the cost of a solution obtained by a sequential approach. A similar situation occurs with no limit on the inventory on hand at the retailer level.

Table 5.18 – Results of the top-down heuristic for large instances

Delivery splitting	Demand splitting	Diversification strategy	BUB	CPU_2	CPU_3	CPU_{TSP}	Total CPU	GAP BLB (%)	Feas (%)
x	x	BW	877016	836	2306	2	3331	47.67	95.63
		CC	876970	843	2306	1	3317	47.53	94.62
		CD	875802	820	2153	1	3103	47.42	94.83
x	✓	BW	860381	475	2950	4	3613	55.61	81.18
		CC	862310	471	2978	3	3591	55.81	81.42
		CD	858604	460	2952	2	3530	55.42	81.7
✓	x	BW	873358	469	2901	7	3744	46.34	99.58
		CC	871600	471	2983	5	3723	46.22	99.65
		CD	868457	554	2703	3	3394	46.06	99.06
✓	✓	BW	873213	479	2847	10	3669	49.97	99.48
		CC	872415	477	2927	11	3663	49.37	98.02
		CD	872002	548	2632	9	3364	49.05	99.41

5.6.2 Results on large instances

For the large instances, we consider a number of retailers and warehouses similar to the ones used in Gruson et al. (2019b). The number of retailers is set equal to 50, 100 or 200 and the number of warehouses is set equal to 5, 10 or 20. The number of time periods is set equal to 15.

For these large instances, CPLEX could not find any feasible solution within the time limit of 3 hours, illustrating the difficulty of the problem. We therefore do not report detailed results obtained with CPLEX directly. We however use the lower bound obtained in this set of experiments to measure the performance of our approaches. The gap between this lower bound and the cost of the best solution found by our approaches is given in the column GAP BLB in Tables 5.18 and 5.19.

Tables 5.18 and 5.19 display the results for the large instances for the top-down and bottom-up approaches, respectively. In those tables, we further give the number of feasible solutions obtained, expressed as a proportion of the total number of instances solved, in the Feas column.

In Table 5.18, one can see that the results obtained with the top-down approach are disappointing. First, the total CPU time taken is close to the limit we imposed. A closer look at detailed results indicate that the time limit is actually reached for numerous in-

Table 5.19 – Results of the bottom-up heuristic for large instances

Delivery splitting	Demand splitting	BUB	CPU_1	CPU_2	CPU_3	Total CPU	GAP BLB (%)	Feas (%)
x	x	764532	0	676	840	1729	42.21	99.14
x	✓	647879	0	265	1320	1734	47.68	99.31
✓	x	588384	0	167	1733	2020	30.15	98.38
✓	✓	612391	0	227	1625	2010	35.39	98.75

stances. This is explained by the CPU time taken in Step 3. We note however that our attempt to reduce the CPU time in Step 2, by identifying tailing-off effect, is successful, as shown by the values obtained in the CPU_2 column. Second, the gaps compared to the lower bound given by CPLEX are large, ranging between 46.06 and 55.68%. Note however that this is to be taken cautiously since we have no clue about the strength of the lower bound given by CPLEX. Finally, the number of feasible solutions obtained is quite low in case we allow for demand splitting only. For those instances where we do not find a feasible solution, the heuristic actually never leaves Step 2. Note finally that the diversification strategy has only a very small impact on the results. We just note that the use of the CD strategy leads to slightly lower CPU times and less expensive solutions, as for the small instances.

In Table 5.19, one can see that the bottom-up approach is able to find feasible solutions for almost all instances. This illustrates one strength of the bottom-up approach compared to CPLEX. The gaps compared to the lower bound given by CPLEX are quite large, ranging between 30.15 and 47.68%. Again, given that CPLEX did not find any feasible solution, we cannot be sure that the lower bound we obtained through CPLEX is of good quality. Finally, the CPU time taken is on average still far from the time limit, illustrating the fact that the bottom-up approach is quite efficient. We also note that the increase in CPU time, compared to the small instances, is quite limited.

If we compare the results obtained by the top-down and bottom-up approaches, one can directly see the higher performance of the bottom-up approach. Indeed, the cost of the solutions obtained with the bottom-up approach are much lower than the cost of the solutions obtained with the top-down approach. Regarding the CPU time, again the

bottom-up approach uses less CPU time than the top-down approach. Finally, the number of feasible solutions obtained with the bottom-up approach is slightly higher.

Regarding the splitting possibilities, as for the results on the small instances, one can see that the top-down approach obtains similar costs regardless of the splitting possibilities. On the contrary, the cost of the solutions obtained by the bottom-up approach are different based on the splitting possibilities. On the large instances, as for the small instances, we are still unable to make a fair comparison of the costs of the solutions with demand splitting only compared to the costs of the solutions with delivery splitting only. However, given the relative better performance of the bottom-up approach, we can suspect that delivery splitting only is more beneficial than demand splitting only. This would indicate that the transportation capacity can be better optimized if we allow retailers to be visited by several trucks each time period.

5.7 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 further developed an exact branch-and-cut algorithm. This algorithm allows us to obtain optimal solutions on a set of small instances that are used to measure the performance of the solutions given by our heuris-

tics. On small instances the heuristics are able to find solution that are on average 5.41 and 4.25% from the optimal plan 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. Based on these results we have also tested our heuristics on large instances adapted from Gruson et al. (2019a). On those large instances, our approaches are able to find much more feasible solutions than CPLEX. The bottom-up approach obtains a better performance than the top-down approach, in terms of the CPU time taken and of the quality of the solution.

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 would give us access to a lower bound. In particular, we could develop heuristics that provide a lower bound during the search process. We would also like to further explore the possibility of solving exactly the problem 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., C. Archetti, S. Dauzère-Pérès, D. Feillet and M. Grazia Speranza. 2018, «Comparing sequential and integrated approaches for the production routing problem», *European Journal of Operational Research*, vol. 269, p. 633–646.

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

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

Adulyasak, Y., J.-F. Cordeau and R. Jans. 2015a, «Benders decomposition for production routing under demand uncertainty», *Operations Research*, vol. 63, p. 851–867.

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

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

Anderluh, A., P. C. Nolz, V. C. Hemmelmayr and T. G. Crainic. 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 2019-33*.

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

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

Avci, M. and S. T. Yildiz. 2019, «A matheuristic solution approach for the production routing problem with visit spacing policy», *European Journal of Operational Research*, vol. 279, n° 2, p. 572 – 588.

Baldacci, R., M. Battarra and D. Vigo. 2008, «Routing a heterogeneous fleet of vehicles», in *The Vehicle Routing Problem: Latest Advances and New Challenges*, edited by B. Golden, S. Raghavan and E. Wasil, chap. 1, Springer, Boston, MA, p. 3–27.

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

Brown, G., J. Keega, B. Vigus and K. Wood. 2001, «The Kellog company optimizes production, inventory, and distribution», *Interfaces*, vol. 31, p. 1–15.

Chandra, P. and M. L. Fisher. 1994, «Coordination of production and distribution planning», *European Journal of Operational Research*, vol. 72, n° 3, p. 503–517.

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

Cuda, R., G. Guastaroba and M. G. Speranza. 2015, «A survey on two-echelon routing problems», *Computers & Operations Research*, vol. 55, p. 185–199.

Cunha, J. O. and R. A. Melo. 2016, «On reformulations for the one-warehouse multi-retailer problem», *Annals of Operations Research*, vol. 238, n° 1, p. 99–122.

Darvish, M., C. Archetti, L. C. Coelho and M. G. Speranza. 2019, «Flexible two-echelon location routing problem», *European Journal of Operational Research*, vol. 277, p. 1124–1136.

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

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

Dellaert, N., T. Van Woensel, T. G. Crainic and F. D. Saridarq. 2019, «A multi-commodity two-echelon capacitated vehicle routing problem with time windows: model formulations and solution approach», *CIRRELT Technical Report 2019-43*.

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

Fischetti, M. and A. Lodi. 2003, «Local branching», *Mathematical Programming*, vol. 98, p. 23–47.

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

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

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

Gruson, M., J.-F. Cordeau and R. Jans. 2019b, «Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure», *GERAD Technical Report G-2019-51*.

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

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

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

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

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

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

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

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

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

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

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

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

Wagner, H. M. and T. M. Whitin. 1958, «Dynamic version of the economic lot size model», *Management Science*, vol. 5, p. 89–96.

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

Conclusion générale

L'intégration des décisions opérationnelles au sein d'une entreprise ou à travers la chaîne d'approvisionnement reste un défi tant théorique que pratique. S'il est vrai que la séparation des décisions permet de mieux définir les rôles de chacun et permet de résoudre des problèmes opérationnels a priori moins complexes, il n'en reste pas moins que les bénéfices reliés à l'intégration des décisions opérationnelles sont nombreux. Ces bénéfices sont souvent associés aux bénéfices monétaires, mais ils incluent aussi une meilleure performance opérationnelle, un meilleur niveau de service, une plus grande satisfaction des clients ou encore une plus grande cohérence entre les différentes décisions.

L'objectif de cette thèse est d'utiliser les techniques de la recherche opérationnelle pour développer des modèles et algorithmes de résolution permettant de résoudre des problèmes de planification intégrée. Cette thèse met donc l'emphase sur les gains économiques reliés à l'intégration des décisions opérationnelles, en minimisant les coûts opérationnels dans les fonctions objectifs des modèles de PMNE développés. Plus en détails dans notre cas, nous voulons favoriser l'intégration des décisions de production et distribution dans une chaîne d'approvisionnement à trois niveaux. La principale contribution de cette thèse est l'introduction d'un nouveau problème et le développement de modèles et algorithmes performants pour résoudre le 3LSPD. Avec le développement de ces modèles et algorithmes, nous réalisons une première étape vers l'étude détaillée du 3LSPD.

Modélisation du problème

Le premier article de la thèse se concentre sur la modélisation du 3LSPD. De nombreuses formulations de PMNE ont été proposées, avec des propriétés théoriques démon-

trées. Ces propriétés théoriques ont été confrontées à des performances pratiques. Ainsi, les formulations les plus fortes ne se sont pas toujours révélées être les formulations les plus performantes en pratique, en termes de temps de calcul et de qualité des solutions obtenues.

Le travail sur ces formulations a servi de base pour les autres articles de la thèse. Néanmoins, la contribution de ce premier travail n'est pas uniquement de servir de fondation à l'ensemble de la thèse. En proposant des formulations différentes, nos modèles peuvent être directement utilisables dans des contextes différents, voire pour d'autres problèmes que le 3LSPD. Ainsi, l'utilisation d'indices temporels dans les variables de décision permettrait de modéliser des problèmes de 3LSPD avec des contraintes de périssabilités des produits, ou des contraintes de délai de livraison. Du côté des problèmes existants qui peuvent bénéficier de ce travail, le PRP et l'IRP sont des candidats potentiels. En effet, les formulations utilisées dans ce genre de problème sont souvent des formulations classiques ou de transport. Comme vu dans ce premier article, l'utilisation de formulation échelon ou de la formulation MCE pourrait être bénéfique pour les modèles de PRP et IRP. Dans un même ordre d'idée, ce travail sur les formulations pourrait être bénéfique pour un cas avec des économies d'échelle, que ce soit au niveau de la production, de la distribution, ou du stockage. Cette prise en compte amènerait un indice supplémentaire aux variables utilisées, pour représenter le niveau de rabais. Les formulations de transport et réseau ne semblent pas appropriées dans ce cas, au contraire des autres formulations.

Résolution du problème

Le chapitre 3 ainsi que les second et troisième articles de la thèse s'attardent sur la résolution de plusieurs variantes du 3LSPD. Le chapitre 3 porte sur la résolution du 3LSPD déterministe avec des contraintes de capacité de production. En effet, les résultats du premier projet ont mis en lumière l'incapacité du solveur à trouver des solutions faisables ou optimales en un temps de calcul raisonnable. Il y avait donc le besoin de développer une méthode de résolution efficace. À cet effet, nous avons développé un algorithme de sépa-

ration et génération de colonnes pour résoudre le problème, à partir d'une reformulation de Dantzig-Wolfe. Si le travail sur les modélisations nous a permis d'identifier des sous-structures intéressantes, les performances de notre algorithme n'ont pas dépassé celles du solveur CPLEX, malgré l'ajout de plusieurs améliorations. Il est à noter que nous avions également réalisé des tests sur une version non capacitaire du 3LSPD, toujours sans succès. Si les résultats ne sont pas au rendez-vous, ce travail n'en reste pas moins porteurs de conclusions intéressantes. Pour résoudre exactement une version capacitaire, la séparation et génération de colonnes ne semble pas être une méthode à privilégier, la résolution du problème maître étant trop coûteuse. Toutefois, cette méthode pourrait être utilisable pour une résolution approchée du 3LSPD capacitaire. En effet, les résultats obtenus, en termes d'écart par rapport aux solutions d'un solveur, sont honorables. Ainsi, transformer notre méthode exacte en méthode approchée serait une avenue à explorer, avec une emphase à mettre sur le goulot actuel, soit la résolution du problème maître.

Le second article de la thèse a abordé une version stochastique non capacitaire du problème. L'ajout d'incertitude est cohérent avec la réalité des entreprises qui ne peuvent prédire de manière exacte la demande des clients. Ainsi, cela remet en question la modélisation effectuée dans le premier projet, et appelle au développement de méthode de résolution propre à cette version stochastique non capacitaire. C'est le seul projet de la thèse qui incorpore une dose d'incertitude entourant la demande des détaillants, amenant donc une première contribution importante. Pour ce projet encore, c'est le travail sur les formulations qui a permis d'identifier des sous-structures à exploiter dans le cadre d'une décomposition de Benders. Même avec l'incertitude nous avons pu décomposer notre problème, en modélisant cette incertitude via des scénarios de demande. L'algorithme de séparation et coupes développé, fondé sur la décomposition de Benders, a obtenu des performances bien supérieures à celles obtenues par le solveur CPLEX seul, en termes de temps de calcul et de qualité des solutions obtenues. L'algorithme de résolution de problèmes de flot à coût minimal que nous avons développé explique en grande partie ce succès, et apporte donc une contribution scientifique non négligeable. Grâce à cet algorithme nous avons pu obtenir des coupes de Pareto en résolvant un seul problème, sans

faire appel à un solveur. Ce gain de temps, combiné à l'efficacité des coupes de Pareto, rend notre algorithme de séparation et coupes plus que compétitif. Il est intéressant de noter que cet algorithme fonctionne toujours aussi bien si on inclut la possibilité d'avoir des ventes perdues.

Enfin, le dernier article s'attaque à une extension du 3LSPD, appelée 3LSRP. Pour ce dernier projet, nous relâchons plusieurs hypothèses qui étaient présentes dans les trois premiers projets. En particulier, les détaillants ne sont plus assignés à un unique entrepôt. On a ainsi un problème plus générique à résoudre, qui comporte plus de flexibilité. De plus, de nombreuses contraintes opérationnelles supplémentaires ont été ajoutées, et plusieurs hypothèses ont été relâchées comparativement aux autres projets (contraintes de capacité de production et transport, assignation unique des détaillants aux entrepôts). La difficulté du problème qui en découle nous a naturellement orienté vers le développement de méthodes heuristiques. Deux méthodes ont été proposées. La première part des décisions prises à l'usine pour finalement obtenir les décisions de distribution auprès des détaillants. La seconde procède en ordre inverse et part des détaillants pour aboutir à l'usine de production. Nos heuristiques ont plusieurs avantages. Elles peuvent être facilement adaptées et sont simples. De plus, sur de grandes instances, elles ont été en mesure d'obtenir des solutions faisables au problème quand CPLEX n'a pas été en mesure de fournir de solutions faisables. Si ces heuristiques fonctionnent en décomposant le problème en sous-problèmes, c'est bien les liens entre ces sous-problèmes qui distinguent nos heuristiques d'une méthode séquentielle qui rendrait inexistante l'intégration des décisions opérationnelles. Dans ce dernier projet, nous avons également ajouté de la flexibilité dans les livraisons. Cet ajout est une contribution intéressante, la littérature fixant traditionnellement les visites aux détaillants à un véhicule uniquement. Cette contribution nous permet de proposer des méthodes de résolution plus propices en fonction des libertés de livraison offertes.

Limites de la thèse

Cette thèse comporte plusieurs limites. Premièrement, nous n'avons considéré qu'un seul item dans les trois premier projets. Au niveau de la modélisation, les modèles proposés peuvent facilement être modifiés pour prendre en compte un item supplémentaire. Les algorithmes de séparation et génération de colonnes et de séparation et coupes pourraient également être modifiables pour prendre en compte plusieurs items. En particulier, notre algorithme de séparation et coupes, car prenant appui sur la formulation multi-produits du 3LSPD, ne devrait pas perdre en performance : on garderait la même décomposabilité par sous-problème. Cette limite de l'item unique a été levée dans le dernier article.

Une deuxième limite est la non prise en compte de coûts unitaires de production et de transport. Si cette hypothèse est classique dans les problèmes de lotissement, on pourrait toutefois la relâcher dans nos travaux. Cette hypotèse repose sur des coûts fixes de production et sur le fait que la demande totale est à satisfaire au complet : on obtient ainsi une constante dans la fonction objectif. Encore une fois, avec la prise en compte de coûts unitaires de production, la modélisation ne s'en trouverait que peu affectée. Il n'y a par contre pas d'indice particulier qui nous assurerait des mêmes performances pratiques de nos formulations. Dans les algorithmes développés dans les différents articles, cette prise en compte de coûts unitaire peut se faire également facilement. Concernant la performance de notre algorithme de séparation et coupes pour le deuxième article, il nous faudrait toutefois ajuster notre algorithme de résolution de problème de flôt à coût minimal. Cela devrait augmenter un peu les temps de calcul, mais rien ne peut laisser penser que l'algorithme proposé perdrait en efficacité. Dans un même ordre d'idée, on pourrait relâcher l'hypothèse que les coûts unitaire de stockage sont plus élevés au niveau des détaillants. La modélisation ne serait pas affectée du tout, ni l'algorithme de *branch-and-price* développé dans le second projet, ni les deux heuristiques développées dans le dernier projet. En revanche, il faudrait ajuster l'algorithme de résolution du problème de flot à coût minimal développé dans le cadre du troisième projet. Le changement serait toutefois mineur car il suffit juste d'identifier les chemins les moins coûteux dans un graphe.

Enfin, toujours au niveau des coûts, on pourrait imaginer un cas où les coûts fixes peuvent être transférés d'une période à l'autre. Au niveau de la modélisation, toutes les formulations pourraient être adpatées de la même manière en ajoutant des variables d'ouverture et fermeture. Cela n'aurait pas d'impact sur les méthodes de résolution développées.

Enfin, une dernière limite est le fait que le réseau considéré soit figé dans les différents travaux, sauf le dernier. La raison derrière cette hypothèse rejoint la disposition géographique des différents sites. Ainsi, certains liens entre entrepôts et détaillants ne sont pas utiles. Toutefois, en particulier dans un contexte stochastique, il pourrait être bénéfique d'ajouter de la flexibilité dans le réseau. Une autre situation qui justifierait l'ajout de cette flexibilité serait le fait d'avoir des compétences spécifiques dans les différents entrepôts. Ainsi, chaque entrepôt pourrait être responsable d'un type d'item.

Perspectives

À la lumière des travaux réalisés dans le cadre de cette thèse, plusieurs avenues de recherche restent encore à explorer. D'abord, les limites mentionnées précédemment pourraient être utilisées comme point de départ pour des travaux futurs. En particulier, les différentes hypothèses émises dans les projets pourraient être relâchées, comme nous avons commencé à le faire dans le dernier travail de recherche. Ensuite, on pourrait se tourner vers une chaîne d'approvisionnement ayant plusieurs usines de production. Le fait d'avoir plusieurs usines de production serait cohérent avec la présence de plusieurs items. Dans un souci d'efficacité opérationnelle et de rentabilisation des ressources, chaque usine serait responsable de la production d'un nombre restreint d'items. Dans cette situation, les modèles et algorithmes développés dans le cadre de cette thèse pourraient servir de point de départ ou de base de comparaison. Puis, une perspective intéressante serait la prise en compte de la chaîne logistique amont. L'intégration de nos décisions opérationnelles est en ce moment limitée par l'hypothèse que l'on a les ressources nécessaires pour produire nos produits finis. En incluant des nomenclatures de produit, on ajoute des contraintes liées à la disponibilité des composants et matières premières. Dans le but d'intégrer les dé-

cisions opérationnelles, la prise en compte des décisions de réapprovisionnement auprès des fournisseurs est une avenue intéressante. Cette avenue est d'autant plus intéressante qu'elle a des implications théoriques au niveau des problèmes d'optimisation à résoudre, mais aussi des implications pratiques au niveau des liens et des relations avec les fournisseurs. Enfin, une dernière perspective serait l'intégration des décisions d'ordonnancement de production, dans un contexte multi items. Parmi les trois perspectives mentionnées, la seconde est celle qui me semble être la plus prometteuse. En effet, l'hypothèse d'avoir un approvisionnement qui va se plier à nos besoins, qui émanent des plans intégrés de production et distribution, reste une hypothèse forte. Avoir la prise en compte de l'aspect fournisseur est ainsi intéressant, surtout dans un contexte d'intégration de la chaîne logistique. Cette perspective de recherche comporte non seulement un potentiel scientifique fort avec de nouveaux problèmes classiques de recherche opérationnelle à définir, mais aussi un potentiel pratique avec une preuve des bénéfices de l'intégration. Ce dernier point pourrait servir d'argument pour faire travailler ensemble des entreprises différentes.

La principale contribution de ce travail est le développement d'un cadre général pour l'étude du 3LSPD. D'abord, la variété des formulations de PMNE proposées, notamment en raison de la variété des variables de décision, permet à nos modèles d'être utilisables dans de nombreux cas industriels. Ensuite, l'algorithme de séparation et coupes développé, fondé sur la décomposition de Benders, s'avère être très efficace pour résoudre une version stochastique du problème. Enfin, nous avons proposé de mesurer les différences de coûts qui peuvent exister en fonction des libertés offertes quant aux possibilités de livraison. Ce dernier point amène une contribution nouvelle par rapport aux hypothèse classiquement posées dans la littérature.

Les structures des chaînes d'approvisionnement diffèrent bien entendu d'une entreprise à l'autre et d'une industrie à l'autre. Malgré tout, par la définition d'un nouevau problème en phase avec la réalité des entreprises et par le développement de nouveaux modèles et algorithmes, nous sommes convaincus que les travaux réalisés dans cette thèse participent à la facilitation de l'intégration des décisions opérationnelles pour les entre-

prises manufacturières.

Bibliographie générale

Abdullah, S., A. Shamayleh et M. Ndiaye. 2019, «Three stage dynamic heuristic for multiple plants capacitated lot sizing with sequence-dependent transient costs», *Computers & Industrial Engineering*, vol. 127, p. 1024–1036.

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

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

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

Adulyasak, Y., J.-F. Cordeau et R. Jans. 2015a, «Benders decomposition for production routing under demand uncertainty», *Operations Research*, vol. 63, p. 851–867.

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

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

Aloulou, M. A., A. Dolgui et M. Y. Kovalyov. 2014, «A bibliography of non-deterministic lot-sizing models», *International Journal of Production Research*, vol. 52, p. 2293–2310.

Anderluh, A., P. C. Nolz, V. C. Hemmelmayr et T. G. Crainic. 2019, «Multi-objective optimization of a two-echelon vehicle routing problem with vehicle synchronization and 'grey zone' customers arising in urban logistics», *Cahier du CIRRELT*, vol. CIRRELT-2019-33.

Applegate, D., R. Bixby, V. Chvátal et W. Cook. 2011, «Concorde tsp solver», `http://www.math.uwaterloo.ca/tsp/concorde.html`. Consulté le 2019-10-07.

Arkin, E., D. Joneja et R. Roundy. 1989, «Computational complexity of uncapacitated multi-echelon production planning problems», *Operations Research Letters*, vol. 8, n° 2, p. 61–66.

Avci, M. et S. T. Yildiz. 2019, «A matheuristic solution approach for the production routing problem with visit spacing policy», *European Journal of Operational Research*, vol. 279, n° 2, p. 572 – 588.

Bahl, H. C. et S. Zionts. 1987, «Multi-item scheduling by Benders' decomposition», *The Journal of the Operational Research Society*, vol. 38, n° 12, p. 1141–1148.

Baldacci, R., M. Battarra et D. Vigo. 2008, «Routing a heterogeneous fleet of vehicles», dans *The Vehicle Routing Problem : Latest Advances and New Challenges*, edité par B. Golden, S. Raghavan et E. Wasil, chap. 1, Springer, Boston, MA, p. 3–27.

Barany, I., T. Van Roy et L. A. Wolsey. 1984, «Uncapacitated lot-sizing : The convex hull of solutions», *Mathematical Programming*, vol. 22, p. 32–43.

Barbarasoglu, G. et D. Özgür. 1999, «Hierarchical design of an integrated production and 2-echelon distribution system», *European Journal of Operational Research*, vol. 118, p. 464–484.

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

Bayley, T., H. Süral et J. H. Bookbinder. 2018, «A hybrid Benders approach for coordinated capacitated lot-sizing of multiple product families with set-up times», *International Journal of Production Research*, vol. 56, n° 3, p. 1326–1344.

Bell, W. J., L. M. Dalberto, M. L. Fisher, A. J. Greenfield, R. Jaikumar, P. Kedia, R. G. Mack et P. J. Prutzman. 1983, «Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer», *Interfaces*, vol. 13, p. 4–23.

Ben Mohamed, I., W. Klibi et F. Vanderbeck. 2020, «Designing a two-echelon distribution network under demand uncertainty», *European Journal of Operational Research*, vol. 280, p. 102–123.

Benders, J. F. 1962, «Partitioning procedures for solving mixed-variables programming problems», *Numerische Mathematik*, vol. 4, p. 238–252.

Billington, P. J., J. O. McClain et L. J. Thomas. 1986, «Heuristics for multilevel lot-sizing with a bottleneck», *Management Science*, vol. 32, n° 8, p. 989–1006.

Birge, J. R. et F. Louveaux. 1997, *Introduction to Stochastic Programming*, Springer-Verlag, New-York.

Birge, J. R. et F. V. Louveaux. 1988, «A multicut algorithm for two-stage stochastic linear programs», *European Journal of Operational Research*, vol. 34, n° 3, p. 384–392.

Blumenfeld, D., L. D. Burns, C. F. Daganzo, M. C. Frick et R. W. Hall. 1987, «Reducing logistics costs at General Motors», *Interfaces*, vol. 17, p. 26 – 47.

Bodur, M. et J. R. Luedtke. 2016, «Mixed-integer rounding enhanced Benders decomposition for multiclass service-system staffing and scheduling with arrival rate uncertainty», *Management Science*, vol. 63, n° 7, p. 2073–2091.

Bookbinder, J. H. et J.-Y. Tan. 1988, «Strategies for the probabilistic lot-sizing problem with service-level constraints», *Management Science*, vol. 34, p. 1037–1156.

Bouchard, M., S. D'Amours, M. Rönnqvist, R. Azouzi et E. Gunn. 2017, «Integrated optimization of strategic and tactical planning decisions in forestry», *European Journal of Operational Research*, vol. 259, n° 3, p. 1132 – 1143.

Brahimi, N., N. Absi, S. Dauzère-Pérès et A. Nordli. 2017, «Single-item dynamic lot-sizing problems : An updated survey», *European Journal of Operational Research*, vol. 263, p. 838–863.

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

Brown, G., J. Keega, B. Vigus et K. Wood. 2001, «The Kellog company optimizes production, inventory, and distribution», *Interfaces*, vol. 31, p. 1–15.

de Camargo, R. S., G. de Miranda Jr. et H. P. Luna. 2008, «Benders decomposition for the uncapacitated multiple allocation hub location problem», *Computers & Operations Research*, vol. 35, p. 1047–1064.

Caserta, M. et S. Voß. 2013, «A math-heuristic Dantzig-Wolfe algorithm for capacitated lot sizing», *Annals of Mathematical Artificial Intelligence*, vol. 69, p. 207–224.

Chand, S., V. N. Hsu, S. Sethi et V. Deshpande. 2007, «A dynamic lot sizing problem with multiple customers : customer-specific shipping and backlogging costs», *IIE Transactions*, vol. 39, n° 11, p. 1059–1069.

Chandra, P. 1993, «A dynamic distribution model with warehouse and customer replenishment requirements», *Journal of the Operational Research Society*, vol. 44, p. 681–692.

Chandra, P. et M. L. Fisher. 1994, «Coordination of production and distribution planning», *European Journal of Operational Research*, vol. 72, n° 3, p. 503–517.

Chen, H. 2015, «Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems», *Omega*, vol. 56, p. 25–36.

Chen, Z. et L. Li. 2011, «Coordination mechanism and algorithm for decentralized production-distribution planning», dans *2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce*, p. 1038–1041.

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

Cuda, R., G. Guastaroba et M. G. Speranza. 2015, «A survey on two-echelon routing problems», *Computers & Operations Research*, vol. 55, p. 185–199.

Cunha, J. O. et R. A. Melo. 2016, «On reformulations for the one-warehouse multi-retailer problem», *Annals of Operations Research*, vol. 238, n° 1, p. 99–122.

Dantzig, G. B. et P. Wolfe. 1960, «Decomposition principle for linear programming», *Operations Research*, vol. 8, p. 101–111.

Darvish, M., C. Archetti, L. C. Coelho et M. G. Speranza. 2019, «Flexible two-echelon location routing problem», *European Journal of Operational Research*, vol. 277, p. 1124–1136.

Darvish, M. et L. Coelho. 2018, «Sequential versus integrated optimization : Production, location, inventory control, and distribution», *European Journal of Operational Research*, vol. 268, p. 203–214.

Dayarian, I. et G. Desaulniers. 2019, «A branch-price-and-cut algorithm for a production-routing problem with short-life-span products», *Transportation Science*, vol. 53, p. 829–849.

Dellaert, N., T. Van Woensel, T. G. Crainic et F. D. Saridarq. 2019, «A multi-commodity two-echelon capacitated vehicle routing problem with time windows : model formulations and solution approach», *Cahier du CIRRELT*, vol. CIRRELT-2019-43.

Dhaenens-Flipo, C. et G. Finke. 2001, «An integrated model for an industrial production-distribution problem», *IIE Transactions*, vol. 33, p. 705–715.

Di Summa, M. et L. A. Wolsey. 2008, «Lot-sizing on a tree», *Operations Research Letters*, vol. 36, n° 1, p. 7–13.

Diaby, M. et A. Martel. 1993, «Dynamic lot sizing for multi-echelon distribution systems with purchasing and transportation discounts», *Operations Research*, vol. 41, n° 1, p. 48–59.

Dixon, P. S. et E. A. Silver. 1981, «A heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing problem», *Journal of Operations Management*, vol. 2, p. 23–39.

Duarte, A. J. S. T. et J. M. V. V. de Carvalho. 2015, «A column generation approach to the discrete lot sizing and scheduling problem on parallel machines», dans *Operational Research : IO 2013 - XVI Congress of APDIO, Bragança, Portugal, June 3-5, 2013*, edité par J. P. Almeida, J. F. Oliveira et A. A. Pinto, Springer International Publishing, p. 157–170.

Eppen, G. et R. Martin. 1987, «Solving multi-item capacitated lot-sizing problems with variable definition», *Operations Research*, vol. 35, p. 832–848.

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

Federgruen, A. et M. Tzur. 1999, «Time-partitioning heuristics : Application to one warehouse, multiitem, multiretailer lot-sizing problems», *Naval Research Logistics*, vol. 46, n° 5, p. 463–486.

Fischetti, M. et A. Lodi. 2003, «Local branching», *Mathematical Programming*, vol. 98, p. 23–47.

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

Fisher, M. L. 1985, «An applications oriented guide to lagrangian relaxation», *Interfaces*, vol. 15, n° 2, p. 10–21.

Flynn, B. B., B. Huo et X. Zhao. 2010, «The impact of supply chain integration on performance : A contingency and configuration approach», *Journal of Operations Management*, vol. 28, n° 1, p. 58–71.

Gayon, J.-P., G. Massonnet, C. Rapine et G. Stauffer. 2017, «Fast approximation algorithms for the one-warehouse multi-retailer problem under general cost structures and capacity constraints», *Mathematics of Operations Research*, vol. 42.

Gebennini, E., R. Gamberini et R. Manzini. 2009, «An integrated production – distribution model for the dynamic location and allocation problem with safety stock optimization», *International Journal of Production Economics*, vol. 122, n° 1, p. 286–304.

Gopal, C. et H. Cypress. 1993, *Integrated distribution management : competing on customer service, time and cost*, Business One Irwin, Homewood, Illinois.

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

Gruson, M., M. Bazrafshan, J.-F. Cordeau et R. Jans. 2017, «A comparison of formulations for a three-level lot sizing and replenishment problem with a distribution structure», *Cahier du GERAD, HEC Montréal*, vol. G-2017-59.

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

Gruson, M., J.-F. Cordeau et R. Jans. 2018, «The impact of service level constraints in deterministic lot sizing with backlogging», *Omega*, vol. 79, p. 91–103.

Gruson, M., J.-F. Cordeau et R. Jans. 2019b, «Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure», *Cahier du GERAD, HEC Montréal*, vol. G-2019-51.

Guan, Y., S. Ahmed, G. L. Nemhauser et A. J. Miller. 2006, «A branch-and-cut algorithm for the stochastic uncapacitated lot-sizing problem», *Mathematical Programming*, vol. 105, n° 1, p. 55–84.

Gutiérrez, J., J. Puerto et J. Sicilia. 2004, «The multiscenario lot size problem with concave costs», *European Journal of Operatinal Research*, vol. 156, p. 162–182.

Haq, A. N., P. Vrat et A. Kanda. 1991, «An integrated production-inventory-distribution model for manufacture of urea : a case», *International Journal of Production Economics*, vol. 39, p. 39–49.

Harris, F. W. 1990, «How many parts to make at once», *Operations Research*, vol. 38, n° 6, p. 947–950.

Haugen, K. K., A. Løkketange et D. L. Woodruff. 2001, «Progressive hedging as a metaheuristic applied to stochastic lot-sizing», *European Journal of Operatinal Research*, vol. 132, p. 116–122.

Helber, S., F. Sahling et K. Schimmelpfeng. 2013, «Dynamic capacitated lot sizing with random demand and dynamic safety stocks», *OR Spectrum*, vol. 35, p. 75–105.

van Hoesel, C. P. M. et A. P. M. Wagelmans. 1996, «An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities», *Management Science*, vol. 42, n° 1, p. 142–150.

Huisman, D., R. Jans, M. Peeters et A. P. M. Wagelamns. 2005, «Combining column generation and lagrangian relaxation», dans *Column Generation*, edité par G. Desaulniers, J. Desrosiers et M. M. Solomon, chap. 9, Springer, New York, p. 247–270.

Jans, R. et Z. Degraeve. 2006, «Modeling industrial lot sizing problems : a review», *International Journal of Production Research*, vol. 46, p. 1619–1643.

Jans, R. et Z. Degraeve. 2007, «A new dantzig-wolfe reformulation and branch-and-price algorithm forthe capacitated lot sizing problem with setup times», *Operations Research*, vol. 55, p. 909–920.

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

Karimi, B., G. Fatemi et J. Wilson. 2003, «The capacitated lot sizing problem : a review of models and algorithms», *Omega*, vol. 31, p. 365–378.

Kopanos, G. M., L. Puigjaner et M. C. Georgiadis. 2012, «Simultaneous production and logistics operations planning in semicontinuous food industries», *Omega*, vol. 40, n° 5, p. 634–650.

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

Le, T., A. Diabat, J. P. Richard et Y. Yih. 2013, «A column generation-based heuristic algorithm for an inventory routing problem with perishable goods.», *Optimization Letters*, vol. 7, p. 1481–1502.

Lejeune, M. A. 2006, «A variable neighborhood decomposition search method for supply chain management planning problems», *European Journal of Operational Research*, vol. 175, p. 959–976.

Levi, R., R. Roundy, D. Shmoys et M. Sviridenko. 2008, «A constant approximation algorithm for the one-warehouse multiretailer problem», *Management Science*, vol. 54, n° 4, p. 763–776.

Li, S., B. Ragu-Nathan, T. Ragu-Nathan et S. Subba Rao. 2006, «The impact of supply chain management practices on competitive advantage and organizational performance», *Omega*, vol. 34, p. 107–124.

Li, Z. et J. Hai. 2019, «Inventory management for one warehouse multi-retailer systems withcarbon emission costs», *Computers & Industrial Engineering*, vol. 130, p. 565 – 574.

Lübbecke, M. E. et J. Desrosiers. 2005a, «A primer in column generation», dans *Column Generation*, edité par G. Desaulniers, J. Desrosiers et M. M. Solomon, chap. 1, Springer, New York, p. 1–32.

Lübbecke, M. E. et J. Desrosiers. 2005b, «Selected topics in column generation», *Operations Research*, vol. 53, n° 6, p. 1007–1023.

Maes, J., J. O. McClain et L. N. Van Wassenhove. 1991, «Multilevel capacitated lot sizing complexity and LP-based heuristics», *European Journal of Operational Research*, vol. 53, n° 2, p. 131–148.

Magnanti, T. et R. Wong. 1981, «Accelerating Benders decomposition : algorithmic enhacement and model selection criteria», *Operations Research*, vol. 23, p. 464–484.

Magnanti, T. L., P. Mireault et R. T. Wong. 1986, «Tailoring Benders decomposition for uncapacitated network design», *Mathematical Programming Study*, vol. 26, p. 112–154.

Magnanti, T. L., J. F. Shapiro et M. H. Wagner. 1976, «Generalized linear programming solves the dual», *Management Science*, vol. 22, p. 1195–1203.

Manne, A. S. 1958, «Programming of economic lot sizes», *Management Science*, vol. 4, p. 115–135.

Martel, A. et W. Klibi. 2016, «Supply chain networks optimization», dans *Designing value-creating supply chain networks*, chap. 7, Springer, Cham, p. 243–287.

Melo, R. A. et L. A. Wolsey. 2010, «Uncapacitated two-level lot-sizing», *Operations Research Letters*, vol. 38, n° 4, p. 241–245.

Meng, Q.-C., T. Zhang, M. Li et X.-X. Rong. 2014, «Optimal Order Strategy in Uncertain Demands with Free Shipping Option», *Discrete Dynamics in Nature and Society*, vol. 2014.

Michel, S. et F. Vanderbeck. 2012, «A column-generation based tactical planning method for inventory routing», *Operations Research*, vol. 60, p. 382–397.

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

Monthatipkul, C. et P. Yenradee. 2008, «Inventory/distribution control system in a one-warehouse/multi-retailer supply chain», *International Journal of Production Economics*, vol. 114, n° 1, p. 119–133.

Mourgaya, M. et F. Vanderbeck. 2017, «Column generation based heuristic for tactical planning in multi-period vehicle routing», *Euopean Journal of Operational Research*, vol. 183, p. 1028–1041.

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

Normandin, F. 2016, «La malédiction du silo», https://www.revuegestion.ca/la-malediction-du-silo. Consulté le 2019-10-30.

Özdamar, L. et T. Yazgaç. 1999, «A hierarchical planning approach for a production-distribution system», *International Journal of Production Research*, vol. 37, n° 16, p. 3759–3772.

Papadakos, N. 2008, «Practical enhancements to the Magnanti-Wong method», *Operations Research Letters*, vol. 36, n° 4, p. 444–449.

Perboli, G., R. Tadei et D. Vigo. 2011, «The two-echelon capacitated vehicle routing problem : models and math-based heuristics», *Transportation Science*, vol. 45, p. 364–380.

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

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

Rahmaniani, R., T. G. Crainic, M. Gendreau et W. Rei. 2017, «The Benders decomposition algorithm : A literature review», *European Journal of Operational Research*, vol. 259, n° 3, p. 801–817.

Rockafellar, R. T. et R. J.-B. Wets. 1991, «Scenarios and policy aggregation in optimization under uncertainty», *Mathematics of Operations Research*, p. 119–147.

Sahling, F., L. Buschkühl, H. Tempelmeier et S. Helber. 2009, «Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic», *Computers & Operations Research*, vol. 36, n° 9, p. 2546–2553.

Sanei Bajgiran, O., M. Kazemi Zanjani et M. Nourelfath. 2016, «The value of integrated tactical planning optimization in the lumber supply chain», *International Journal of Production Economics*, vol. 171, p. 22 – 33.

Solyalı, O. et H. Süral. 2012, «The one-warehouse multi-retailer problem : reformulation, classification, and computational results», *Annals of Operations Research*, vol. 196, p. 517–541.

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

Solyalı, O., H. Süral et M. Denizel. 2010, «The one-warehouse multiretailer problem with an order-up-to level inventory policy», *Naval Research Logistics*, vol. 57, n° 7, p. 653–666.

Soysal, M., J. M. Bloemhof-Ruwaard et Bektaş. 2015, «The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations», *International Journal of Production Economics*, vol. 164, p. 366–378.

Taskin, S. et E. J. Lodree Jr. 2010, «Inventory decisions for emergency supplies based on hurricane count predictions», *International Journal of Production Economics*, vol. 126, p. 66–75.

Tempelmeier, H. 2007, «On the stochastic uncapacitated dynamic single-item lot sizing problem with service level constraints», *European Journal of Operational Research*, vol. 181, n° 1, p. 184–194.

Tempelmeier, H. 2011, «A column generation heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint», *Omega*, vol. 39, p. 627–633.

Tempelmeier, H. 2013, «Stochastic lot sizing problems», dans *Handbook of Stochastic Models and Analysis of Manufacturing System Operations*, edité par J. M. Smith et B. Tan, Springer New York, New York, NY, p. 313–344.

Tempelmeier, H. et S. Helber. 1994, «A heuristic for dynamic multi-item multi-level capacitated lot sizing for general product structures», *European Journal of Operational Research*, vol. 75, n° 2, p. 296–311.

Tunc, H., O. A. Kilic, S. A. Tarim et R. Rossi. 2018, «An extended mixed-integer programming formulation and dynamic cut generation approach for the stochastic lot-sizing problem», *INFORMS Journal on Computing*, vol. 30, n° 3, p. 492–506.

Vickery, S. K., J. Jayaram, C. Droge et R. Calantone. 2003, «The effects of an integrative supply chain strategy on customer service and financial performance : an analysis of direct versus indirect relationships», *Journal of Operations Management*, vol. 21, n° 5, p. 523–539.

Wagner, H. M. et T. M. Whitin. 1958, «Dynamic version of the economic lot size model», *Management Science*, vol. 5, p. 89–96.

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

Wei, M., M. Qi, T. Wu et C. Zhang. 2019, «Distance and matching-induced search algorithm for the multi-level lot-sizing problem with substitutable bill of materials», *European Journal of Operational Research*, vol. 277, p. 521–541.

Wentges, P. 1997, «Weighted Dantzig-Wolfe decomposition for linear mixed-integer programming», *International Transactions in Operational Research*, vol. 4, p. 151–162.

Wolsey, L. A. 1998, *Integer Programming*, Wiley, New York.

Wu, T., L. Shi, J. Geunes et K. Akartunali. 2011, «An optimization framework for solving capacitated multi-level lot-sizing problems with backlogging», *European Journal of Operational Research*, vol. 214, p. 428–441.

Yang, W., F. T. S. Chan et V. Kumar. 2012, «Expert systems with applications optimizing replenishment polices using genetic algorithm for single-warehouse multi-retailer system», *Expert Systems With Applications*, vol. 39, n° 3, p. 3081–3086.

Zangwill, W. 1969, «A backlogging model and a multi-echelon model of a dynamic economic lot size production system - a network approach», *Management Science*, vol. 15(9), p. 506–527.

Zhang, S. et H. Song. 2018, «Production and distribution planning in Danone waters China division», *INFORMS Journal on Applied Analytics*, vol. 48, p. 578–590.