



HAL
open science

Algorithmes d'approximation pour la gestion de stock

Guillaume Massonnet

► **To cite this version:**

Guillaume Massonnet. Algorithmes d'approximation pour la gestion de stock. Autre [cs.OH]. Université de Grenoble, 2013. Français. NNT : 2013GRENM034 . tel-00992384

HAL Id: tel-00992384

<https://theses.hal.science/tel-00992384>

Submitted on 17 May 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques-Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Guillaume MASSONNET

Thèse dirigée par **Christophe RAPINE**
et codirigée par **Jean-Philippe GAYON**

préparée au sein de **G-SCOP (Grenoble Science pour la Conception et l'Optimisation de la Production)**
et de **MSTII (Mathématiques, Sciences et Technologies de l'Information, Informatique)**

Algorithmes d'approximation pour la gestion de stocks

Thèse soutenue publiquement le **4 avril 2013**,
devant le jury composé de :

Mr Jean-Philippe GAYON

Maître de conférence, Institut National Polytechnique de Grenoble, Co-Directeur de thèse

Mr Imed KACEM

Professeur, Université de Lorraine, Président et Rapporteur

Mr Tim NONNER

Research Staff Member, IBM Zürich Research Laboratory, Examineur

Mr Christophe RAPINE

Professeur, Université de Lorraine, Directeur de thèse

Mr Gautier STAUFFER

Professeur, Institut National Polytechnique de Grenoble, Examineur

Mr Mathieu VAN VYVE

Professeur, Université Catholique de Louvain, Rapporteur



Table des matières

1	Introduction	9
1.1	Modèles de gestion de stocks	10
1.2	Complexité et approximations	13
1.3	Organisation du document	15
2	État de l’art	17
2.1	Modèles à temps continu	17
2.1.1	Modèles avec demande stationnaire	17
2.1.2	Modèles avec demande variable	25
2.2	Modèles à temps discret	28
2.2.1	Modèles à un seul échelon	29
2.2.2	Modèles multi-échelon	36
2.3	Résumé des contributions	45
2.3.1	Chapitre 3 : Algorithmes d’approximation pour systèmes à un seul échelon à temps continu	45
2.3.2	Chapitres 4 et 5 : Algorithmes d’approximation pour le problème OWMR et extensions aux modèles avec rupture de stock	46
3	Approximation algorithms for deterministic continuous-review inventory lot-sizing problems with time-varying demand	49
3.1	Introduction	49
3.2	Assumptions and notations	52
3.3	An approximation algorithm	54
3.3.1	The balancing policy	54
3.3.2	Performance guarantee of the BL policy	55
3.3.3	Numerical issues and complexity	56
3.3.4	Bad example	57
3.4	Extensions	58
3.4.1	Nonlinear holding costs	58
3.4.2	Perishable products	59
3.4.3	Finite production rate	60

3.4.4	A model with backlogging	62
3.4.5	Time-dependent order costs	64
3.5	Improvement of the performance guarantee	67
3.5.1	A generalized balancing policy	67
3.5.2	Complexity analysis	68
3.6	Conclusion	70
4	A simple and fast 2-approximation for the one-warehouse multi-retailer problem	73
4.1	Introduction	73
4.2	Model and notations	75
4.3	A simple approximation algorithm	77
4.3.1	Phase 1: Decomposition of the problem	77
4.3.2	Phase 2: The uncrossing algorithm	78
4.3.3	Split and conquer	82
4.3.4	Complexity analysis	85
4.3.5	A bad example	86
4.4	Extension to more general holding cost structures	87
4.4.1	Non-linear level-dependent holding costs	87
4.4.2	Shelf age dependent holding costs	90
4.5	Extension to more general procurement cost structures	95
4.5.1	Assumptions	95
4.5.2	A modified decomposition	96
4.6	Computational experiments	99
4.7	Conclusion	101
5	Constant approximation algorithms for the one warehouse multiple retailers problem with backlogging or lost-sales	103
5.1	Introduction	103
5.2	Assumptions and cost structure for the backlogging model	106
5.3	The split and uncross technique	110
5.3.1	Decomposition of the OWMR problem	110
5.3.2	The algorithm	113
5.4	A lost-sales model	116
5.4.1	Cost structure	116
5.4.2	Decomposition in single-echelon systems	117
5.4.3	The algorithm	118
5.5	Conclusion and perspectives	120
6	Conclusion	121

A Proofs of chapter 3	127
A.1 Theorem 3.3.3	127
A.2 Theorem 3.4.2	129
A.3 Theorem 3.4.3	129
A.4 Theorem 3.4.4	131
B Proofs of chapter 4	135
B.1 Proof of the dominance of synchronized policies for J -retailers	135
B.2 Lemma 4.4.1	135
B.3 Lemma 4.4.2	138
B.4 Lemma 4.4.4	140
B.5 Lemma 4.4.5	142
B.6 Proposition 4.5.1	143
C Proofs of chapter 5	145
C.1 Lemma 5.3.1	145
C.2 Lemma 5.3.2	146
C.3 Lemma 5.4.1	149
C.4 Lemma 5.4.2	149

Remerciements

Je tiens à remercier chaleureusement mes directeurs de thèse, Christophe Rapine et Jean-Philippe Gayon, pour leur patience, leur enthousiasme et leur implication sans faille dans ces travaux de recherche. Leur soutien et leurs conseils m'ont plusieurs fois permis de repartir de l'avant après une période difficile. Leurs grandes qualités, aussi bien scientifiques qu'humaines, ont largement contribué à la qualité de nos projets de recherche et m'ont aidé à mûrir professionnellement. Je suis heureux de pouvoir aujourd'hui les considérer non seulement comme des collaborateurs, mais également comme des amis.

Je veux également remercier Imed Kacem et Mathieu Van Vyve pour avoir accepté le rôle difficile de rapporteurs de cette thèse. Leurs remarques et commentaires constructifs m'ont permis d'aboutir à la version finale de ce manuscrit, tandis que leurs questions lors de la soutenance ont ouvert de nouvelles pistes de recherche pertinentes.

Tim Nonner et Gautier Stauffer m'ont fait le plaisir d'accepter de faire partie de mon jury de thèse. Je les en remercie sincèrement, même si je retiendrai certainement plus volontier d'eux les bons moments et les discussions passionnées que nous avons partagé à Grenoble comme à Zürich. Gautier a aussi activement participé à plusieurs projets de recherche dans lesquels j'étais impliqué et nos contributions ne seraient certainement pas aussi originales sans son imagination et ses qualités scientifiques exceptionnelles.

Mon doctorat aurait certainement été bien plus difficile à vivre si je n'avais pas bénéficié d'environnements de travail propices à l'épanouissement, comme cela a pu être le cas à Grenoble, ainsi qu'au IBM Zürich Research Laboratory ou à l'INSEAD Singapore. Je souhaite donc exprimer ici à quel point j'ai apprécié l'excellente ambiance de travail que j'ai pu trouver dans chacun de ces lieux. Je veux donc souhaiter le meilleur pour les années à venir à chacune des personnes qui ont jalonné mon parcours pendant les 4 dernières années.

Enfin, j'adresse un grand merci à ma famille et à mes amis de Grenoble pour les bouffées d'oxygène quotidiennes qu'ils ont représentés durant cette période. Les sorties en montagnes, les bons repas et autres discussions enflammées autour d'un verre resteront d'excellents souvenirs que je ne manquerai pas de me remémorer dans les années à venir !

Chapitre 1

Introduction

Les débuts de la recherche opérationnelle moderne coïncident avec l'apparition des modèles fondateurs de gestion de stocks dans la première moitié du XX^{ème} siècle. Si le champ d'application des techniques de modélisation et d'optimisation propres à cette discipline s'est depuis considérablement diversifié, les problématiques liées aux stocks sont restées une composante majeure et active de ce domaine scientifique. Les raisons de cette popularité proviennent sans doute de l'universalité du concept de stock : depuis les chasseurs-cueilleurs de la préhistoire jusqu'aux réseaux de grande distribution actuels, la gestion de stocks a de tout temps constitué un élément central des organisations humaines. Aujourd'hui, ceux-ci représentent un investissement colossal dans l'économie des entreprises de production et de distribution, pour lesquelles il devient primordial de disposer de politiques de contrôle et d'optimisation efficaces. Dans ce contexte, la théorie de la gestion de stocks vise à proposer des outils d'aide à la décision performants et fiables, à même d'apporter des réponses opérationnelles concrètes aux situations rencontrées en pratique dans l'industrie.

Dans cette thèse, nous nous sommes concentrés sur plusieurs problèmes classiques en gestion des stocks, dont la résolution à l'optimal peut nécessiter un temps de calcul démesuré. Dans ce cas, le principal défi auquel sont confrontés les industriels consiste à trouver un compromis acceptable entre la qualité des solutions fournies et la rapidité de la procédure employée pour les calculer. Afin de satisfaire chacune de ces exigences, nous nous sommes naturellement orientés vers le domaine des algorithmes d'approximation, dont l'utilisation en recherche opérationnelle a déjà apporté de nombreux résultats pour divers modèles complexes. Ce manuscrit regroupe les principales contributions issues de ces travaux de recherche. Nous nous sommes efforcés d'en faire un document cohérent, mettant en relation nos productions scientifiques avec les techniques de la littérature utiles à leur compréhension. De fait, si des connaissances préalables en théorie de gestion de stocks constituent un atout indéniable pour aborder nos travaux, elles ne sont *a priori* pas nécessaires pour appréhender les méthodes et les preuves présentées dans cette thèse. Pour un état de l'art plus général de la discipline, nous renvoyons le lecteur à l'un des nombreux ouvrages de référence, tels que ceux de Zipkin (2000), Axsäter (2006) et Porteus (2002).

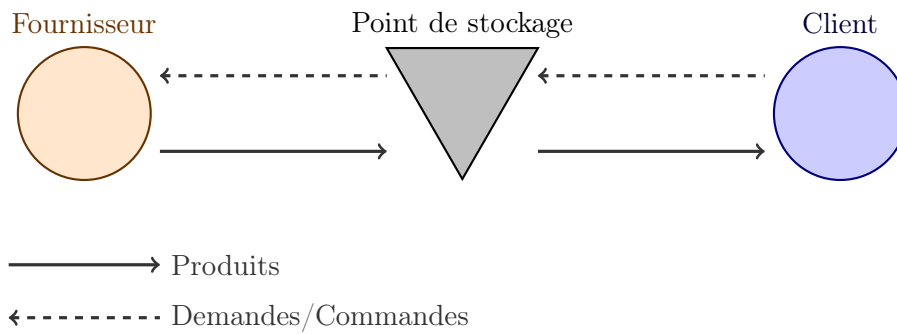


FIGURE 1.1 – Un système à un seul échelon (un seul entrepôt)

1.1 Modèles de gestion de stocks

Généralités

Les modèles de gestion de stocks étudiés dans la littérature sont aussi nombreux que variés et les détailler ici de manière exhaustive serait trop fastidieux. Néanmoins, on retrouve un certain nombre de caractéristiques communes aux problèmes auxquels nous sommes intéressés dans cette thèse. Ainsi, nos travaux se sont concentrés sur des modèles déterministes, pour lesquels il n'existe aucune incertitude sur les paramètres du problème considéré. Les systèmes logistiques que nous étudions sont composés d'un ou plusieurs entrepôts appartenant à la même entité (industriel, commerçant, etc.) dont le but est de satisfaire la demande de ses clients en minimisant ses coûts de fonctionnement. De manière spécifique, on parle de système à un échelon lorsqu'il comporte un seul entrepôt (voir Figure 1.1) et de système multi-échelon lorsqu'on considère un réseau d'entrepôts (voir Figure 1.2). Lorsque le système comporte plusieurs points de stockage, on les distingue en deux catégories. Les points de vente ou de distribution (par exemple des détaillants) sont en contact direct avec les clients et répondent à leurs demandes, tandis que les autres entrepôts servent de points de relais ou d'approvisionnement et sont chargés d'opérations d'assemblage ou de redistribution des biens à d'autres entrepôts composant le réseau.

Quelle que soit sa nature, un entrepôt dispose d'un espace de stockage dans lequel il peut entreposer des produits en attente de servir un client ou d'être transférés vers un autre point de stockage. Le réapprovisionnement de son stock s'opère soit en commandant des unités supplémentaires à son fournisseur (c'est-à-dire d'autres entrepôts du système ou un fournisseur extérieur), soit en produisant lui-même de nouveaux biens (on parle alors de systèmes de production). Les modèles intègrent en général un *délai de livraison* ou de mise en route de production qui modélise le temps de latence entre la décision de réapprovisionner et la réception des unités correspondantes dans le stock. À tout moment, on peut ainsi décrire l'état d'un entrepôt par la quantité de biens déjà stockés ou en train d'être acheminés vers celui-ci. On définit deux variables d'état à partir de ces valeurs : le *niveau de stock* (quantité de produits physiquement stockés dans l'entrepôt) et la *position*

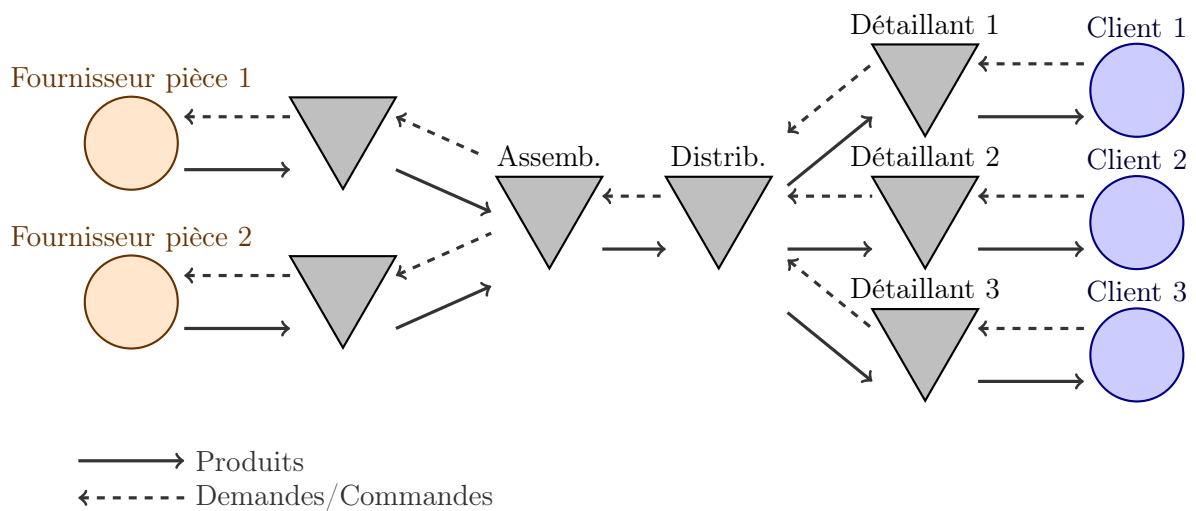


FIGURE 1.2 – Un système multi-échelon (plusieurs entrepôts reliés entre eux) : réception des pièces, assemblage du produit et distribution aux détaillants

de stock (niveau de stock + produits commandés et en cours de transit).

Enfin, si les modèles les plus classiques exigent que la demande d'un client soit satisfaite à la date de livraison convenue, d'autres relâchent cette contrainte et autorisent la rupture de stock. La littérature sur ce type de modèle se divise en deux grandes catégories. La première considère que les clients sont disposés à attendre le prochain réapprovisionnement du stock pour être servis. Cette hypothèse se vérifie en particulier lorsque la concurrence est inexistante ou trop contraignante pour les clients (éloignement géographique, prix rédhibitoires, etc.). La deuxième suppose à l'inverse que les clients sont impatientes et qu'une demande non satisfaite immédiatement entraîne la perte de la vente : on parle alors de modèle avec ventes perdues.

Les différents coûts mis en jeu

Différentes décisions opérationnelles permettent souvent d'améliorer la satisfaction des clients en influençant les commandes et le flux des biens dans les entrepôts. Ces actions entraînent une modification de l'état du système mais induisent également un certain nombre de coûts de fonctionnement ou de rémunération pour certains services tels que le transport d'unités ou la maintenance des entrepôts. Nous détaillons ici les coûts classiques considérés dans les différents travaux de cette thèse :

- *Coûts de commande (ou de production)*. Chaque réapprovisionnement du stock met en jeu différents acteurs et mécanismes. Lorsqu'un entrepôt passe une commande auprès d'un fournisseur extérieur, il doit évidemment payer les produits ou les matières premières qui composent sa demande. De plus, on considère en général qu'une commande entraîne également un coût fixe, indépendant du nombre d'unités commandées. Celui-ci inclut tous les frais supplémentaires dus à l'action même de com-

mander : frais administratifs, frais de transport, main d'oeuvre, inspection, etc. Dans le cas des systèmes de production, ce coût peut par exemple correspondre au démarrage d'une machine utilisée dans la réalisation des produits.

- *Coûts de possession ou de stockage.* En stockant des produits dans un entrepôt, une entreprise immobilise une partie de son capital. Une telle situation réduit donc ses possibilités d'investissements et, par conséquent, ses revenus. Stocker des produits sur une certaine période a également d'autres implications plus directes. Les possibles détériorations, les opérations de maintenance sur la marchandise, la location d'un espace de stockage suffisant, l'assurance des produits stockés, les éventuelles taxes sont autant de facteurs qui augmentent les frais ou diminuent les revenus de l'entreprise lorsqu'elle choisit d'accumuler du stock. Dans cette thèse, nous agrégeons l'ensemble de ces coûts et les désignons sous le terme générique de coût de possession. En pratique, la plupart des modèles suppose que ce coût correspond à un pourcentage de la valeur de la marchandise stockées, mais dépend de la nature de celle-ci. Par exemple, on considère souvent que des produits périssables ou dont la demande suit un phénomène de mode ont un coût de possession important, car leur valeur marchande diminue fortement avec le temps.
- *Coûts de mise en attente des clients ou de vente perdue.* Dans certaines situations, mettre en attente des demandes pour les servir ultérieurement peut être un choix profitable pour une entreprise. Néanmoins en choisissant cette possibilité, celle-ci s'expose au mécontentement de ses clients et à la dégradation de son image de marque. On modélise le plus souvent ces phénomènes par l'application d'une pénalité qui croît en fonction du retard accumulé. En revanche lorsque les clients sont impatientes, une situation de rupture de stock entraîne un manque à gagner lié à la perte d'une vente. Lorsque les demandes doivent absolument être satisfaites, ce coût de vente perdue peut aussi traduire l'achat en urgence des produits manquants ou leur acheminement depuis une autre unité de production, pour un prix plus élevé.

Ces coûts ne sont évidemment pas les seuls à intervenir dans la chaîne logistique. Par exemple, les modèles que nous considérons dans ces travaux supposent que les demandes des clients sont déterministes, c'est-à-dire connues à l'avance. Cela implique un important travail de prospection et de prévision de la demande future qui entraîne des coûts supplémentaires. Cependant, nous supposerons que ceux-ci sont négligeables par rapport aux coûts purement opérationnels introduits plus haut et nous les omettrons donc dans notre étude.

Pour un problème donné, on appelle *politique* un ensemble de règles qui définit les instants de commande ainsi que les quantités commandées. Une politique est dite *réalisable* si elle respecte les contraintes du modèle. Comme mentionné plus haut, l'objectif est de trouver une solution qui minimise l'ensemble des coûts encourus par le système. En particulier, une politique réalisable de coût minimum pour un problème donné est appelée *politique optimale*.

1.2 Complexité et approximations

Complexité d'un problème

La complexité d'un problème peut intuitivement être assimilée à la difficulté de résolution de celui-ci : en gestion de stock, celle-ci traduit l'effort nécessaire pour établir les règles de décision qui définissent la politique optimale du système. Dans cette thèse, nous distinguons deux grandes catégories de problèmes : ceux dans lesquels le temps est un paramètre continu et ceux dans lequel le temps est supposé discret, divisés en pas de temps appelés *périodes*.

À notre connaissance, la notion de complexité pour les problèmes du domaine continu n'est pas définie. Si la résolution analytique de certains modèles représente un défi mathématique de taille, il est difficile d'établir une hiérarchie claire entre les différents systèmes en fonction de leur complexité. En revanche, la théorie de la complexité pour les problèmes de décision ou d'optimisation combinatoires est bien établie et constitue un domaine de recherche à part entière de l'informatique théorique. Celle-ci s'applique en particulier aux problèmes de gestion de stocks discrets. Dans ce cas de figure, les politiques sont le plus souvent déterminées à l'aide d'algorithmes, c'est-à-dire de procédures qui construisent une politique pour le système considéré. L'ordre de grandeur du nombre d'opérations exécutées par l'algorithme détermine son *temps d'exécution* ou sa *complexité*, qui dépend notamment de la taille de l'instance traitée. Ainsi, un algorithme de résolution est dit linéaire si son temps d'exécution est de l'ordre de $O(N)$ opérations pour n'importe quelle instance de taille N , quadratique si il est de $O(N^2)$, etc. Plus généralement, on dit qu'un algorithme est polynomial si sa complexité est de l'ordre de $O(N^k)$ pour résoudre une instance de taille N , avec $k \geq 1$ une constante. On considère en général qu'un temps d'exécution polynomial est "raisonnable", à l'inverse d'un algorithme exponentiel dont le temps d'exécution est de l'ordre de $O(a^N)$ avec $a > 1$. Par exemple si on exécute un algorithme de complexité $O(2^N)$ sur un ordinateur qui effectue un milliard d'opérations à la seconde, le temps d'exécution sera de l'ordre de 1 seconde pour résoudre un problème de taille 30, mais 12 jours pour un problème de taille 50 et 32 mille milliards d'années pour un problème de taille 100. On comprend facilement que pour des problèmes industriels complexes impliquant des milliers de variables et de contraintes, un tel temps de résolution n'est pas envisageable.

La théorie de la complexité est centrale en combinatoire et en informatique théorique : elle définit en particulier des classes de complexité qui permettent de classer les problèmes en fonction de leur difficulté. Cette thèse n'a pas pour objectif de détailler la théorie de la complexité ; cependant celle-ci justifie pour certains problèmes que nous étudions la recherche de solution sous-optimales. Nous renvoyons le lecteur souhaitant approfondir ses connaissances en théorie de la complexité aux ouvrages références dans le domaine (Garey et Johnson (1979) et Papadimitriou (1993)) pour appréhender les principales notions utilisées en combinatoire.

Nom de la classe	Caractérisation des problèmes de la classe
L	Il existe un algorithme déterministe de résolution en $O(\log(N))$
NL	Il existe un algorithme non déterministe de résolution en $O(\log(N))$
P	Il existe un algorithme déterministe de résolution en $O(N^k)$
NP	Il existe un algorithme non déterministe de résolution en $O(N^k)$
EXPTIME	Il existe un algorithme déterministe de résolution en $O(a^N)$

TABLE 1.1 – Un tableau récapitulatif des principales classes de complexité

Le tableau 1.1 regroupe certaines classes de complexité importantes. On a de manière évidente $L \subseteq NL \subseteq P \subseteq NP \subseteq EXPTIME$: en particulier, l'un des problèmes ouverts actuels les plus importants en mathématiques consiste à déterminer si l'inclusion de P dans NP est stricte ou non. En règle générale, on considère en recherche opérationnelle qu'une méthode de résolution à l'optimal est acceptable lorsqu'elle s'exécute en temps polynomial (classe P). Cependant, on peut prouver que certains problèmes auxquels nous nous intéressons sont *NP-difficiles*, c'est-à-dire plus complexes à résoudre que n'importe quel problème appartenant à la classe NP. En particulier, cela implique qu'il est impossible de calculer une solution optimale en temps raisonnable pour n'importe quelle instance du problème sous l'hypothèse $P \neq NP$.

Algorithmes d'approximation

Dans l'idéal, on aimerait disposer d'une méthode fiable et générique qui donne rapidement la meilleure solution au problème que l'on souhaite résoudre. Malheureusement, les systèmes industriels qui motivent l'étude des modèles de gestion de stocks (et plus généralement ceux de la recherche opérationnelle) font fréquemment intervenir un grand nombre de paramètres et bien souvent il n'existe pas de méthode efficace pour les résoudre à l'optimal. Nous désignons ici par "méthode efficace" un algorithme polynomial dans le cas d'un problème discret ou un ensemble d'équations que l'on peut résoudre analytiquement si le problème est à temps continu. En pratique, un compromis doit donc être trouvé entre rapidité, performance et universalité (c'est-à-dire que la méthode peut être appliquée à toutes les instances d'un modèle donné).

Une approche consiste à restreindre le champ d'application visé et à se concentrer sur des cas particuliers du modèle dont les propriétés permettent de simplifier la résolution. Cette démarche est satisfaisante lorsque les caractéristiques des instances pratiques s'y prêtent, mais peut également amener à considérer des hypothèses irréalistes. Une autre approche est de se concentrer sur la recherche d'une solution optimale en s'autorisant éventuellement des temps de résolution exponentiels pour certaines instances problématiques. Du fait de l'explosion combinatoire des problèmes concernés, ce type de stratégie ne peut pas en général s'appliquer sur des instances de grande taille telles que celles rencontrées dans le monde industriel. La dernière approche, la plus commune dans l'industrie, est de se

limiter à la recherche de solutions acceptables au problème sans que celles-ci ne soient nécessairement optimales. D'innombrables méthodes approchées, appelées *heuristiques*, ont ainsi été appliquées avec succès à de multiples problèmes. Certaines s'appuient sur des principes intuitifs pour construire une solution réalisable au problème étudié, comme les algorithmes génétiques, les méthodes de recuit simulé ou la recherche tabou. Ces dernières fournissent une solution réalisable au problème étudié qui présente souvent de bonnes performances en pratique, mais dont l'application à certaines instances particulières peut s'avérer arbitrairement mauvaise.

Nos travaux se sont concentrés sur des heuristiques efficaces (c'est-à-dire qui s'exécutent en temps polynomial) pour lesquelles il est possible de borner l'erreur relative de chaque solution produite. En d'autres termes, si il n'est pas possible de garantir l'optimalité de la solution proposée (du fait de la contrainte sur le temps d'exécution de l'algorithme), on souhaite borner l'erreur maximale sur le coût de celle-ci comparé au coût d'une politique optimale. On appelle de tels algorithmes des *algorithmes d'approximation* ou algorithmes à garantie de performance. La définition suivante formalise la terminologie utilisée, qui se rapporte à des problèmes d'optimisation combinatoire de minimisation.

Définition 1. *Un algorithme d'approximation de garantie α , ou α -approximation, pour un problème de minimisation est un algorithme polynomial qui pour toute instance du problème construit une solution dont la valeur est inférieure à α fois la valeur d'une solution optimale.*

En d'autres termes, nous cherchons des algorithmes pour lesquels nous pouvons prouver que le coût de la solution obtenue n'est jamais pire que α fois le coût optimal, avec $\alpha \geq 1$. Le facteur α est appelé indifféremment *garantie de performance*, *ratio* ou *facteur d'approximation*. On parle d'approximation avec garantie constante lorsque α est indépendant de la taille N de l'instance considérée. En particulier, on peut remarquer qu'un algorithme optimal est une 1-approximation, selon la Définition 1.

Pour une présentation détaillée des techniques d'approximation classiques pour des problèmes combinatoires, le lecteur pourra consulter les ouvrages de Vazirani (2001) et de Williamson et Schmoys (2010).

1.3 Organisation du document

Dans cette thèse, nous proposons de nouveaux algorithmes simples et originaux pour des problèmes classiques de la littérature de gestion de stocks déterministe. Nos contributions, rédigées en anglais, sont regroupées aux chapitres 3, 4 et 5 : elles s'appliquent à des modèles pour lesquels il n'existe pas de méthode efficace pour déterminer la politique optimale. Notre approche a donc privilégié la recherche d'algorithmes d'approximation pour chacun de ces problèmes, afin d'obtenir des techniques de résolution rapides mais présentant néanmoins une garantie de performance intéressante. Si le domaine des algorithmes d'approximation est très actif en recherche opérationnelle et combinatoire, il n'existe que

peu de travaux qui abordent des problèmes de gestion de stock sous cet angle.

Afin d'assurer la cohérence du document dans son ensemble, nous commençons par introduire au chapitre 2 différents problèmes classiques de gestion de stocks en lien avec nos travaux de recherche. En particulier, nous y détaillons les principales méthodes de résolution pour des problèmes simples que nous utilisons ensuite pour aborder les modèles plus complexes considérés dans les chapitres suivants. La partie 2.3 est consacrée à un rapide résumé de nos principales contributions, afin de mieux comprendre le positionnement scientifique de nos travaux.

Dans le chapitre 3, nous nous intéressons à des modèles de gestion de stocks à temps continu. Nous considérons un problème classique dans lequel un entrepôt fait face à des demandes de clients variables dans le temps, ainsi que plusieurs variantes de ce modèle. Nous appliquons une politique basée sur l'équilibrage des différents coûts mis en jeu dans le système et prouvons que celle-ci a une garantie de performance constante pour cette classe de problèmes difficiles à résoudre analytiquement. Les chapitres 4 et 5 se concentrent sur des problèmes dans lesquels le temps est discret, divisé en périodes. Nous étudions deux systèmes de distribution dans lesquels un entrepôt central approvisionne plusieurs détaillants, connus dans la littérature sous le nom de *Joint Replenishment Problem* et *One Warehouse Multi Retailer*. Ces problèmes sont tous deux NP-difficiles et donc trop complexes pour être résolus à l'optimal en temps raisonnable : nous proposons donc des techniques simples qui permettent de construire des politiques réalisables avec garantie de performance constante. Ainsi, le chapitre 4 introduit un nouvel algorithme qui décompose le système en sous-problèmes plus simples à résoudre. Dans le chapitre 5, ce concept est étendu à des modèles avec mise en attente et ventes perdues et nous proposons une première approximation à garantie de performance constante pour ce problème général. Enfin, nous rappelons nos principales contributions et envisageons des perspectives de recherche future en fin de document.

Chapitre 2

Un état de l'art des principaux modèles déterministes en gestion de stocks

La littérature sur la gestion de stocks en environnement déterministe se divise en deux grandes catégories : les modèles à temps continu et les modèles à temps discret. Dans ce chapitre, nous introduisons certains problèmes fondamentaux de chacun de ces domaines, ainsi que certaines méthodes de résolution classiques. L'objectif ici n'est pas de présenter un état de l'art exhaustif, mais plutôt de dégager les modèles en lien avec nos contributions, en s'attardant principalement sur les techniques utiles au lecteur pour la bonne compréhension des chapitres 3, 4 et 5. Une revue de littérature plus conséquente sur les travaux relatifs à chaque modèle sera par la suite proposée en introduction de chacun de ces chapitres.

2.1 Modèles à temps continu

2.1.1 Modèles avec demande stationnaire

Quantité économique de commande : le modèle de Wilson

Le modèle le plus connu dans le domaine de la gestion de stocks est celui proposé par Harris (1913), bien que souvent attribué (à tort) à Wilson (1934). Ce modèle est mieux connu sous le nom de modèle EOQ (pour *Economic Order Quantity*), ou QEC (*Quantité Économique de Commande*) en français. La formule associée est l'un des résultats fondateurs de la gestion de stocks et reste largement utilisée dans l'industrie malgré son apparente simplicité. Ce modèle est constitué d'un seul échelon qui distribue un seul produit à ses clients, avec les hypothèses suivantes :

- La demande est continue et stationnaire.
- Les coûts considérés sont invariants avec le temps.

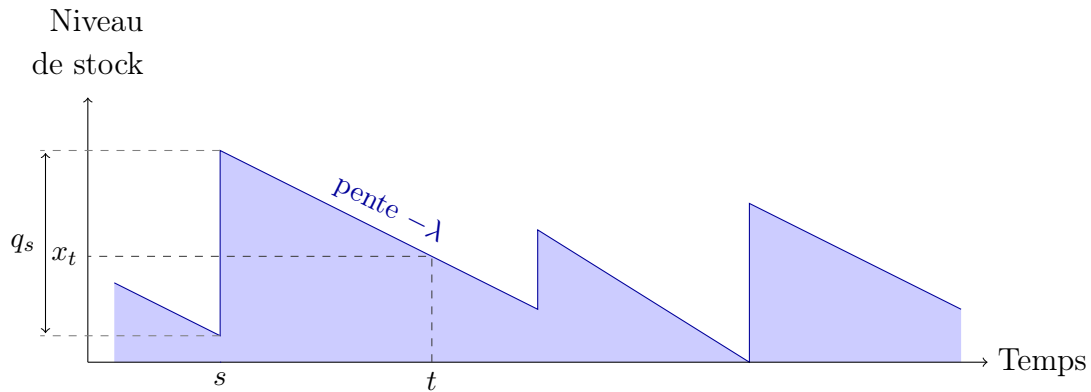


FIGURE 2.1 – Un exemple d'évolution du stock ($\lambda = 0.5$). Le système associé passe une commande à l'instant s pour une quantité q_s .

- Une commande peut être passée à n'importe quel moment : elle arrive alors immédiatement à l'entrepôt dans sa totalité.
- La pénurie est interdite.

Remarque : Beaucoup de problèmes de gestion de stocks tiennent compte d'un *délai de livraison* entre l'instant où la commande est passée et la réception de celle-ci. Néanmoins lorsque le modèle a un comportement déterministe, ce délai peut être considéré comme nul : l'ajout de celui-ci avance simplement chaque commande dans le temps afin de conserver le même niveau de stock que dans la solution sans délai.

On utilise les notations suivantes pour désigner les différents paramètres du modèle :

- λ : demande par unité de temps.
- K : coût fixe de commande en euros (payé à chaque fois que l'entrepôt passe commande, quelle que soit la quantité commandée).
- c : coût d'achat unitaire auprès du fournisseur.
- h : coût de stockage unitaire dans l'entrepôt pendant une unité de temps.

On suppose que l'horizon de temps est infini. L'objectif du problème est de minimiser le coût moyen encouru par chaque unité de produit (commande et possession).

On note x_t le niveau de stock dans l'entrepôt à l'instant $t \geq 0$. On fait l'hypothèse que le stock initial est nul ($x_0 = 0$) et pour tout instant t on note q_t la quantité de biens commandée par l'entrepôt. La quantité q_t est supposée réelle, strictement positive si t est un instant de commande et égale à 0 dans le cas contraire. Le niveau de stock diminue donc à un taux constant de λ par unité de temps, à l'exception des instants de commande où il augmente instantanément de la quantité commandée. Un exemple d'évolution du niveau de stock dans le temps est donné sur la Figure 2.1.

Avant de s'intéresser à la résolution de ce problème, nous commençons par introduire une propriété dominante pour ce modèle. Une propriété \mathcal{P} est dite *dominante* pour un problème donné si pour toute politique réalisable pour ce problème, il existe une politique de coût inférieur ou égal qui respecte la propriété \mathcal{P} . On définit ci-dessous la propriété ZIO

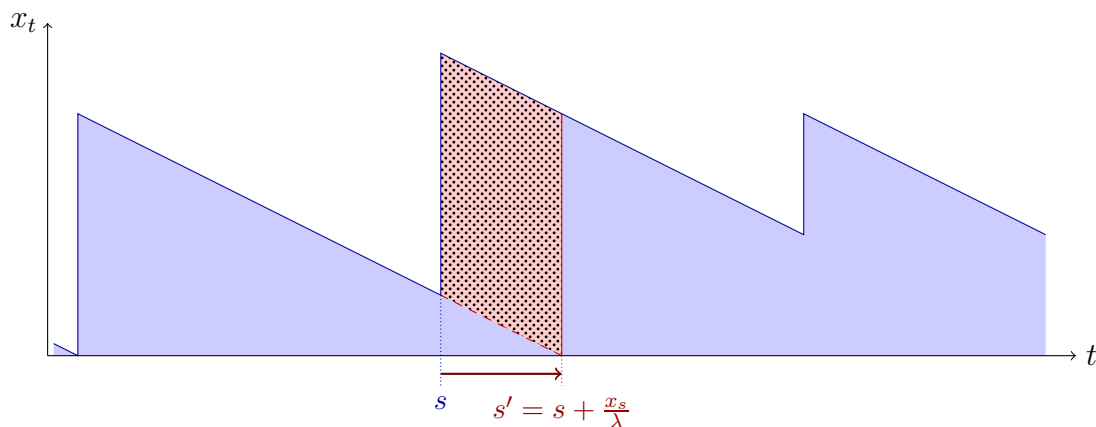


FIGURE 2.2 – Argument d'échange pour la dominance de ZIO

(pour *Zero-Inventory Ordering*) et on prouve sa dominance pour le modèle EOQ.

Définition 2 (Politiques ZIO). *Une politique respecte la propriété ZIO si elle ne commande que quand son niveau de stock est zéro.*

$$q_t > 0 \Rightarrow x_t = 0$$

Propriété 1. *La propriété ZIO est dominante pour le modèle EOQ.*

Preuve

On utilise un argument d'échange pour transformer une politique ne respectant pas ZIO en une politique ZIO de moindre coût. Soit π une politique ne respectant pas la propriété ZIO et soit s le premier instant où π commande alors que $x_s > 0$. On peut alors repousser la commande de ces q_s unités à un instant postérieur $s' = s + \frac{x_s}{\lambda}$ afin que cette commande respecte la propriété ZIO. Le coût de commande pour ces q_s unités est alors le même, tandis que le coût de stockage encouru entre s et s' est diminué de $hq_s(s' - s)$ (cf. Figure 2.2). Le reste des coûts de la politique restant inchangés, on conclut que la propriété ZIO est donc dominante pour le modèle EOQ.

□

La Propriété 1 nous permet de restreindre l'ensemble des politiques considérées et de nous concentrer uniquement sur celles qui ne commandent que lorsque leur stock est épuisé. On appelle *cycle de réapprovisionnement* la période de temps séparant deux commandes successives. Si t est un instant de commande, la propriété ZIO permet alors de calculer la longueur du cycle commençant en t grâce à la quantité q_t d'unités commandées. On note cette longueur $\tau(q_t)$; le prochain instant de commande est alors $t' = t + \tau(q_t)$, où :

$$q_t = \lambda \tau(q_t) \Leftrightarrow \tau(q_t) = \frac{q_t}{\lambda} \quad (2.1)$$

D'après l'équation (2.1), minimiser le coût moyen encouru par unité de produit est équivalent à minimiser le coût moyen par unité de temps. Comme les paramètres du modèle sont tous stationnaires, il suffit de minimiser le coût moyen sur un cycle puis de répéter

ce cycle à l'infini pour obtenir la solution optimale du modèle. Considérons un cycle correspondant à une commande de q unités. Le coût moyen par unité de temps $C(q)$ encouru sur ce cycle peut alors se décomposer comme suit :

$$C(q) = \frac{1}{\tau(q)} \times \left(\underbrace{K + cq}_{\text{coûts de commande}} + h \underbrace{\int_{t=s}^{s+\tau(q)} (q - \lambda(t-s)) dt}_{\text{coût de stockage}} \right)$$

$$= \frac{K\lambda}{q} + c\lambda + h\frac{q}{2} \quad (2.2)$$

La fonction $C(\cdot)$ est deux fois dérivable et sa dérivée seconde $C''(q) = 2K\lambda/q^3$ est strictement positive, donc $C(\cdot)$ est convexe. On peut donc calculer la quantité qui minimise le coût moyen encouru sur un cycle en annulant la dérivée de $C(\cdot)$:

$$C'(q) = 0 \Leftrightarrow q = \sqrt{\frac{2K\lambda}{h}}$$

Définition 3. *La quantité optimale de commande est :*

$$q^* = \sqrt{\frac{2K\lambda}{h}} \quad (2.3)$$

On appelle cette quantité quantité économique de commande (*QEC*), ou quantité EOQ (pour *Economic Order Quantity*).

Le coût moyen encouru par unité de temps s'exprime alors de la façon suivante :

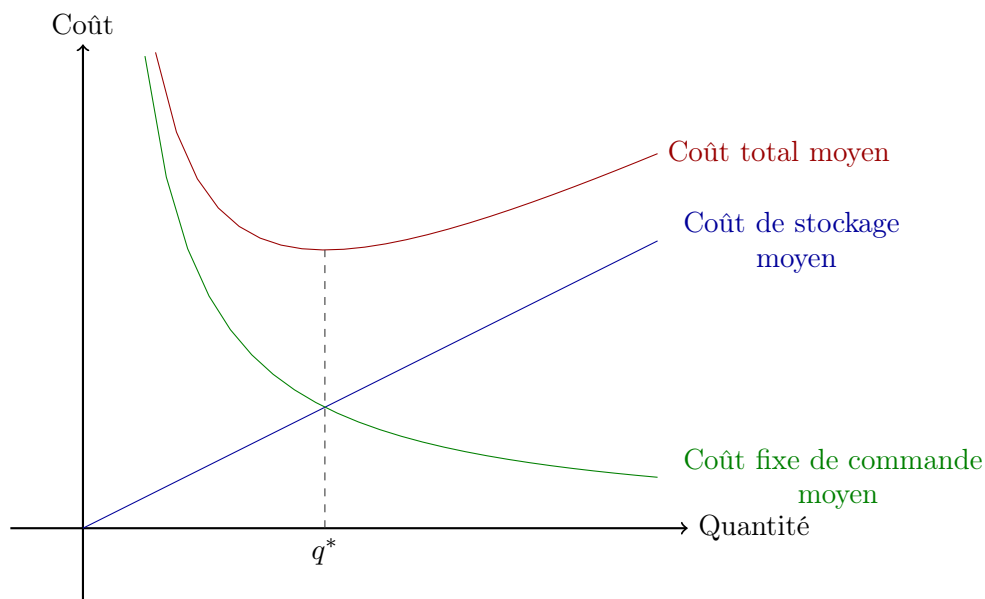
$$C(q^*) = c\lambda + \frac{K\lambda}{\sqrt{\frac{2K\lambda}{h}}} + \frac{1}{2}h\sqrt{\frac{2K\lambda}{h}}$$

$$= c\lambda + \sqrt{\frac{K\lambda h}{2}} + \sqrt{\frac{K\lambda h}{2}}$$

On peut remarquer que le second terme correspond au coût fixe moyen dépensé par unité de temps alors que le troisième provient du coût de stockage moyen par unité de temps. La quantité optimale équilibre donc ces deux termes. Cette propriété est notamment réutilisée dans le chapitre 3 afin d'approximer des politiques optimales pour des systèmes dont les paramètres varient dans le temps.

Extensions du modèle de Wilson

Taux de production. Le modèle de Wilson suppose que lorsque l'entrepôt passe commande, toutes les unités commandées sont livrées en même temps. Cependant si l'entreprise produit les biens qu'elle met en vente, il est probable que les unités soient produites une à une. Une commande correspond alors à un lancement de production et le stock augmente

FIGURE 2.3 – La fonction de coût du modèle EOQ convexe ($K = 40$, $c = 3$, $h = 1$, $\lambda = 0.2$)

graduellement au fur et à mesure que de nouvelles unités deviennent disponibles.

Le modèle de production auquel nous nous intéressons a été introduit par Taft (1918). Les coûts restent inchangés par rapport au modèle EOQ, mais lorsque l'on commande une quantité q , celle-ci arrive progressivement dans le stock suivant un *taux de production* μ , continu et stationnaire, qui traduit la fréquence avec laquelle les produits sortent de l'unité de production. La demande est toujours supposée continue et stationnaire de taux λ . De plus on suppose $\mu > \lambda$: en effet, si $\mu < \lambda$, la cadence de production de la machine n'est pas suffisante pour répondre à la demande et la condition de non-rupture ne peut être respectée. Si $\mu = \lambda$, la solution optimale triviale au problème consiste à lancer une production au temps $t = 0$ pour une quantité infinie, et le stock est alors nul à tout instant puisque chaque unité produite est immédiatement consommée par la demande.

Il est aisé de vérifier que la propriété ZIO est toujours dominante pour ce problème et que la politique optimale est également stationnaire. Une fois de plus, on a équivalence entre la quantité produite q et la longueur d'un cycle $\tau(q)$ par la relation $q = \lambda\tau(q)$. Cependant, chaque cycle comprend maintenant deux phases : une *phase active* (ou phase de production), durant laquelle le stock augmente avec un taux $\mu - \lambda$, et une *phase passive*, pendant laquelle aucune unité n'est produite et le stock diminue suivant le taux λ de la demande. Ces deux phases sont représentées sur la Figure 2.4. La durée de chacune des phases se calcule facilement en remarquant que durant la phase active, une quantité q est produite avec un taux μ . On en déduit le temps actif $\tau_a(q)$ de chaque cycle :

$$\tau_a(q) = \frac{q}{\mu} \quad (2.4)$$

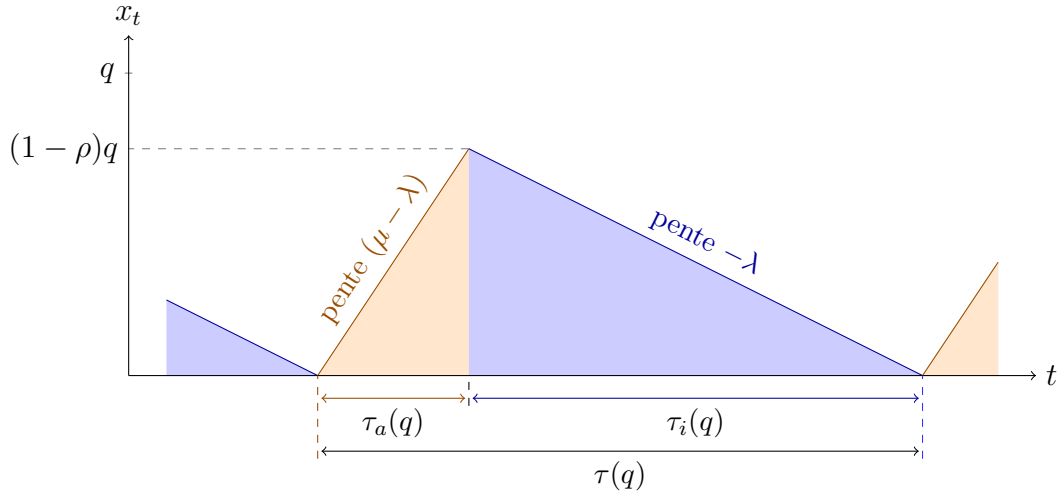


FIGURE 2.4 – Évolution du stock dans un modèle de production ($\lambda = 0.5$, $\mu = 2$)

La phase passive s'étend ensuite sur tout le reste du cycle pour une durée

$$\begin{aligned}
 \tau_i(q) &= \tau(q) - \tau_a(q) \\
 &= \frac{q}{\lambda} - \frac{q}{\mu} \\
 &= \frac{\mu - \lambda}{\mu\lambda}q
 \end{aligned} \tag{2.5}$$

Dans ce qui suit, on note $\rho = \frac{\lambda}{\mu}$. On a donc $\tau_a(q) = \rho\tau(q)$ et $\tau_i(q) = (1 - \rho)\tau(q)$. Au cours d'un cycle, le niveau de stock atteint donc son maximum après un temps $\tau_a(q)$, pour une valeur de $(1 - \rho)q$ (voir Figure 2.4).

On peut alors calculer le coût moyen encouru par unité de temps comme pour le modèle classique :

$$C(q) = \frac{K\lambda}{q} + c\lambda + h(1 - \rho)\frac{q}{2} \tag{2.6}$$

En suivant la même méthode que pour le modèle EOQ, on peut calculer la quantité optimale de production, que l'on définit comme suit :

Définition 4. On appelle quantité économique de production la quantité optimale pour le modèle de production :

$$q^* = \sqrt{\frac{2K\lambda}{(1 - \rho)h}}$$

Comme dans le modèle classique, on remarque que la quantité optimale équilibre le coût fixe de commande moyen et le coût moyen de stockage. Nous proposons au chapitre 3 une extension de ce principe à un modèle avec demande variable.

Modèles avec mise en attente. Les modèles présentés jusqu'à présent interdisent la rupture de stock, mais il peut parfois être intéressant d'intégrer cette possibilité. On consi-

dère ici un système dans lequel la rupture de stock est autorisée et les demandes non satisfaites immédiatement sont différées et servies en priorité lors de la commande suivante (voir Zipkin (2000) pour une discussion plus détaillée du modèle). Cette hypothèse conduit à relâcher la contrainte de positivité sur le niveau de stock dans le modèle de Wilson, amène une plus grande flexibilité du modèle et donc une diminution des coûts encourus (pour un même ensemble de paramètres). Plus précisément, une situation de rupture se traduit par une accumulation de retard que l'on peut assimiler à un stock négatif qui croît avec un taux λ . Les paramètres de ce modèle restent indépendants du temps et on s'intéresse donc une nouvelle fois aux politiques stationnaires pour ce problème. La propriété ZIO introduite plus haut est quant à elle modifiée afin d'offrir la possibilité de rupture :

Propriété 2. *Il existe une politique optimale qui ne commande que lorsque son stock est négatif ou nul.*

$$q_t > 0 \Rightarrow x_t \leq 0$$

Il est aisé de vérifier, en utilisant des arguments similaires à ceux avancés pour la dominance de ZIO, que cette propriété est dominante pour le modèle avec mise en attente.

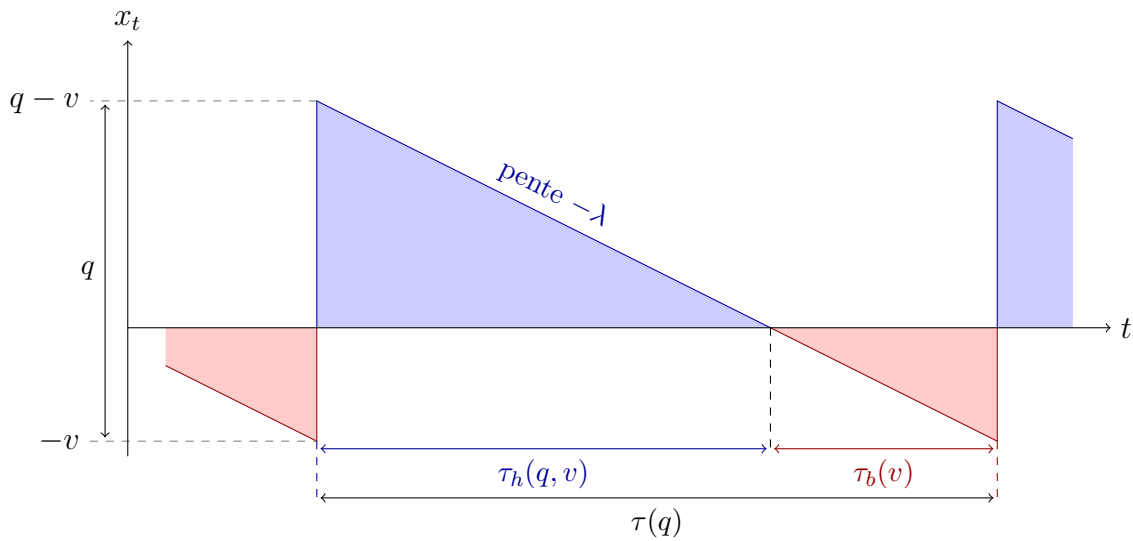
Contrairement au modèle classique, une politique stationnaire respectant la Propriété 2 se définit maintenant par deux paramètres : sa quantité de commande q et son *point de commande* $-v$, qui dénote le seuil du niveau du stock qui déclenche une commande. En d'autres termes, la politique commande dès que son niveau de stock atteint la valeur $-v$, où $v \geq 0$ dénote le nombre maximum de demandes différées sur un cycle. On peut noter que le modèle EOQ classique n'est qu'un cas particulier de ce modèle dont le coût de mise en attente b est infini. Le point de commande est alors 0 et on retrouve la propriété ZIO introduite au début du chapitre.

De la même manière que pour le modèle classique, on note $\tau(q) = \frac{q}{\lambda}$ la longueur d'un cycle. Chaque cycle peut alors se diviser en deux parties : une *période de stockage*, durant laquelle le stock physiquement présent décroît selon le taux de demande, et une *période de rupture*, durant laquelle des demandes sont mises en attente. La première partie du cycle a une longueur $\tau_h(q, v) = \frac{q-v}{\lambda}$, tandis que la seconde a une longueur $\tau_b(v) = \frac{v}{\lambda}$ (voir Figure 2.5).

Le stock moyen sur une période de stockage est donc de $\frac{q-v}{2}$, tandis que le nombre de demandes différées moyen durant la période de rupture est de $\frac{v}{2}$. On peut donc facilement en déduire le coût moyen encouru sur un cycle :

$$\begin{aligned} C(q, v) &= \frac{1}{\tau(q)} \times \left(K + cq + \frac{1}{2}h(q-v)\tau_h(q, v) + \frac{1}{2}bv\tau_b(v) \right) \\ &= \frac{K\lambda}{q} + c\lambda + \frac{1}{2}h\frac{(q-v)^2}{\lambda} + \frac{1}{2}b\frac{v^2}{\lambda} \end{aligned} \quad (2.7)$$

L'expression du coût dépend maintenant de deux variables de décision : la quantité commandée q pour chaque cycle et l'accumulation maximum de demandes différées v . La recherche des valeurs optimales pour q et v s'appuie à nouveau sur la convexité de $C(\cdot, \cdot)$.

FIGURE 2.5 – Le modèle avec demandes différées ($\lambda = 0.5$)

En annulant la dérivée de $C(\cdot, \cdot)$ par rapport à chacune des variables, on obtient les valeurs optimales suivantes :

Définition 5. *La quantité optimale de commande pour le modèle avec demandes différées est*

$$q^* = \sqrt{\frac{2K\lambda}{h}} \sqrt{\frac{1}{\omega}}$$

L'accumulation maximum de demandes différées (ou seuil de commande) est

$$v^* = (1 - \omega)q^*$$

où

$$\omega = \frac{b}{h + b}$$

Si on réutilise ces valeurs dans l'expression du coût moyen défini par l'équation (2.7), on obtient la valeur suivante :

$$C(q^*, v^*) = c\lambda + \underbrace{\sqrt{\frac{K\lambda h\omega}{2}}}_{\text{coût fixe de commande moyen}} + \underbrace{\sqrt{\frac{K\lambda h\omega}{2}}}_{\text{coût de stockage et de mise en attente moyen}}$$

Comme dans le modèle classique, on constate que la politique optimale équilibre le coût fixe avec les coûts de stockage et de mise en attente sur un cycle. Cette observation a permis à Hariga (1996) de proposer une heuristique basée sur ce principe pour des problèmes dont le taux de demande varie dans le temps. L'étude numérique qu'il a mené a déjà démontré que cette approche fournit de bons résultats en pratique. Dans le chapitre 3,

nous avons démontré qu'un tel algorithme d'équilibrage est en fait une 2-approximation pour le problème avec mise en attente et paramètres variables.

Autres extensions. De multiples autres extensions existent pour le modèle EOQ, mais nous ne pouvons les détailler toutes ici par soucis de concision. Néanmoins plusieurs correspondent à des situations fréquemment rencontrées en pratique et il nous semble important de mentionner les principales références en lien avec ces modèles.

Ainsi, Goh (1994) et Giri et Chaudhuri (1997) ont étudié des modèles dans lesquels le coût de possession d'une unité dépend du temps pendant lequel celle-ci est stockée ou du niveau de stock disponible. Il existe également des modèles dans lesquels le prix d'achat des unités varie en fonction de la quantité commandée (voir Monahan (1984)). D'autres font l'hypothèse que les produits stockés se détériorent et proposent des procédures tenant compte de ce phénomène : un état de l'art des modèles sur les produits périssables est dressé par Nahmias (1982), Raafat (1991) et Goyal et Giri (2001).

Enfin, si le modèle classique EOQ admet une solution analytique relativement simple, son extension à des systèmes comportant plusieurs entrepôts rend la résolution beaucoup plus complexe. Aucun des travaux existants dans la littérature n'a, à notre connaissance, proposé de politique optimale pour des modèles de réseaux d'entrepôts. Néanmoins, certaines heuristiques et approximations sont devenues des résultats fondateurs du domaine : le lecteur intéressé est invité à consulter Schwarz (1973) et Roundy (1985, 1986).

2.1.2 Modèles avec demande variable

Les modèles présentés jusqu'à présent supposent que tous les paramètres qui entrent en jeu sont stationnaires. En pratique, il est difficile d'imaginer que cette hypothèse soit vérifiée. Les phénomènes de mode, la période de l'année, la situation économique ou même la météo sont autant de facteurs qui peuvent influencer fortement la demande, tandis que les promotions, l'inflation ou l'augmentation du coût de certaines matières premières peuvent conduire à la fluctuation des coûts de fonctionnement du système. Cependant la stationnarité des paramètres est un élément clé pour la résolution analytique des modèles de gestion de stocks présentés dans les parties précédentes. Ce type d'approche n'est plus applicable si la demande ou les paramètres de coût sont variables.

Cette difficulté est à l'origine de nombreux travaux de recherche qui se focalisent sur des modèles à un échelon dont la demande varie dans le temps. Plus précisément, le problème considéré vise à optimiser la politique de commande d'un entrepôt lorsque la demande est fonction du temps, dénotée $\lambda(t)$. En revanche, les paramètres de coût sont supposés stationnaires.

Suivant ces hypothèses, Resh et al. (1976) se sont concentrés sur un modèle où la demande est proportionnelle au temps ($\lambda(t) = \alpha t$), modèle généralisé quelques temps plus tard par Donaldson (1977) pour une demande affine ($\lambda(t) = \alpha t + \gamma$) puis à une demande de forme "puissance" ($\lambda(t) = \alpha t^\beta$, $\beta > -2$) par Barbosa et Friedman (1978). La méthode de

résolution est similaire pour ces différentes formes de demandes, et consiste à déterminer des fenêtres de commande, c'est-à-dire des intervalles de temps dans lesquels une politique optimale place ses commandes, lorsque le problème est limité à un horizon de temps fini, de longueur T . Lorsque $T \rightarrow \infty$, chacune des fenêtres se réduit à un point dans le temps, qui définit un instant de commande de la politique optimale pour le problème à horizon infini.

Nous ne donnons pas le détail de ces travaux ici, néanmoins nous décrivons leur démarche dans le cas d'une demande de la forme $\lambda(t) = \alpha t^\beta$, avec $\beta > -2$. Considérons le problème sur un horizon de temps fini de longueur T . On note $C(m, \mathbf{t})$ le coût total encouru sur l'horizon de temps par l'unique politique ZIO qui commande m fois dans $[0, T]$, selon le vecteur $\mathbf{t} = t_1, \dots, t_m$. On utilise une approche en deux étapes pour calculer les instants de commande optimaux :

- (a) Pour tout $m \geq 1$, trouver un vecteur $\mathbf{t}^*(m) = (t_1^*(m), \dots, t_m^*(m))$ tel que l'unique politique ZIO qui commande aux instants $t_1^*(m), \dots, t_m^*(m)$ minimise $C(m, \cdot)$. On note $C^*(m) = C(m, \mathbf{t}^*(m))$.
- (b) Trouver la valeur de m qui minimise le coût encouru :

$$m^* = \arg \min_{m \in \mathbb{N}} C^*(m)$$

Si les détails techniques de la preuve sont trop complexes et trop nombreux pour être tous donnés dans ce document, nous essayons de transmettre l'essentiel du raisonnement ici. On définit $\Lambda(t)$, la demande cumulée jusqu'à l'instant t , grâce à l'équation suivante :

$$\Lambda(t) = \int_0^t \lambda(u) du \quad (2.8)$$

On a alors la propriété suivante qui lie deux instants de commande consécutifs dans une politique optimale :

Propriété 3 (Barbosa et Friedman (1978)). *Soit $C(m, \mathbf{t})$ le coût total encouru par la politique qui commande m fois selon le vecteur \mathbf{t} . Pour m fixé, le vecteur $\mathbf{t}^*(m)$ minimise $C(m, \mathbf{t})$ si et seulement si*

$$(t_i^* - t_{i-1}^*)\lambda(t_i^*) = \Lambda(t_{i+1}^*) - \Lambda(t_i^*)$$

Ainsi si on suppose le stock initial nul ($t_0 = 0$), cette propriété implique que le problème de trouver $m-1$ instants de commande se réduit en réalité à trouver un seul d'entre eux, par exemple le premier. Les instants de commande suivants se déduisent ensuite de proche en proche à l'aide de la Propriété 3. On définit maintenant par récurrence une séquence $(\sigma_i)_{i \in \mathbb{N}}$

comme suit :

$$\begin{aligned} \sigma_0 &= 0 \\ \sigma_i &= \begin{cases} \exp(-(1 - \sigma_{i-1})) & \text{si } \beta = -1 \\ [(\beta + 2) - (\beta + 1)\sigma_{i-1}]^{-1/(\beta+1)} & \text{sinon} \end{cases} \quad i = 1, 2, \dots \end{aligned} \quad (2.9)$$

La propriété suivante utilise les éléments σ_i ainsi définis pour déterminer le nombre optimal de commandes m^* .

Propriété 4. *Si la demande est de forme “puissance”, l’unique nombre de commande optimal m^* vérifie l’équation suivante :*

$$\sigma_{m^*} - \sigma_{m^*-1} \leq (\beta + 2) \frac{K}{\alpha h T^{\beta+2}} \leq \sigma_{m^*-1} - \sigma_{m^*-2}$$

Enfin, la *General Root Law* (“loi des racines généralisée”) donne les fenêtres de commandes dans lesquelles se situent les éléments du vecteur $\mathbf{t}^*(m^*)$:

Définition 6 (General Root Law). *Les coordonnées du vecteur $\mathbf{t}^*(m^*)$ satisfont aux inégalités suivantes :*

$$R_i(m^*) \left[(\beta + 2) \frac{K}{\alpha h} \right]^{1/(\beta+2)} \leq t_i^*(m^*) \leq S_i(m^*) \left[(\beta + 2) \frac{K}{\alpha h} \right]^{1/(\beta+2)}$$

avec

$$\begin{aligned} S_i(m^*) &= \frac{\prod_{j=i}^{m^*-1} \sigma_j}{(\sigma_{m^*} - \sigma_{m^*-1})^{1/\beta+2}} \\ R_i(m^*) &= \frac{\prod_{j=i}^{m^*-1} \sigma_j}{(\sigma_{m^*-1} - \sigma_{m^*-2})^{1/\beta+2}} \end{aligned}$$

La définition de ces fenêtres de temps lorsque l’horizon de temps est fini permet ensuite de trouver le premier instant de commande $t_1^*(m^*)$ en horizon infini en utilisant les propriétés limites sur les éléments σ_i . On obtient alors en passant à la limite

$$t_1^* = \lim_{m^* \rightarrow \infty} t_1^*(m^*) = L_1 \left[(\beta + 2) \frac{K}{\alpha h} \right]^{1/(\beta+2)} \quad (2.10)$$

avec

$$L_1 = \lim_{m^* \rightarrow \infty} S_1(m^*) = \lim_{m^* \rightarrow \infty} R_1(m^*)$$

En combinant l’équation (2.10) avec la propriété 3, on peut donc déterminer l’ensemble des instants de commande de la politique optimale. On peut noter que cette politique n’offre pas une solution analytique explicite comme c’est le cas pour le modèle EOQ. Néanmoins, il existe des méthodes numériques efficaces qui permettent d’approximer de façon précise les instants de commande définis grâce à cette méthode.

De nombreuses études ont étendu ce résultat à d'autres fonctions de demandes, en servant notamment de la propriété 3 couplée à des méthodes numériques pour construire une solution optimale. Parmi elles, on peut citer Henery (1979), qui applique ce principe à un modèle dont les demandes sont croissantes log-concaves et Henery (1990) pour un modèle dont les demandes sont périodiques. Hariga (1994) applique une approche semblable pour des systèmes avec mise en attente dont la demande est monotone et log-concave. De nombreux auteurs ont également repris cette méthodologie pour l'appliquer à des modèles à temps continu plus ou moins exotiques, notamment Teng (1994, 1996) puis Teng et al. (1997, 1999, 2002) qui ont employé les mêmes principes pour différentes fonctions de demandes et des modèles incluant la mise en attente, la détérioration du stock, etc.

Si l'approche de Barbosa et Friedman (1978) s'est avérée suffisamment générique pour permettre de résoudre à l'optimal certains modèles (ou prouver des propriétés structurelles de l'optimal), il n'existe pas à ce jour de méthode générale permettant de trouver des solutions proches de l'optimal pour une fonction de demande quelconque. Quelques heuristiques ont été proposées (voir par exemple Silver (1979), Hariga (1996)) pour obtenir des solutions en pratique, mais leur performance ne sont pas garanties et le coût des politiques obtenues par ces méthodes peut se révéler arbitrairement mauvais sur certaines instances. Le chapitre 3 propose une méthode très simple pour aborder de tels problèmes lorsque la demande et les paramètres de coût du modèle sont des fonctions continues par morceaux quelconques du temps. Nous prouvons que le coût de la solution obtenue est toujours inférieure à deux fois le coût d'une solution optimale et adaptons cette méthode à certaines variantes classiques du modèle EOQ avec paramètres variables.

2.2 Modèles à temps discret

La première partie de ce chapitre a introduit des modèles à temps continu, dans lesquels un entrepôt peut passer commande à n'importe quel moment. Néanmoins, il est fréquent que de telles décisions ne puissent être prises qu'à des instants précis, indépendamment de la volonté de l'entreprise. Par exemple, il existe de nombreux secteurs dans lesquels le fournisseur ne peut pas effectuer plus d'une livraison par jour, ce qui oblige l'entrepôt à prévoir ses commandes et son niveau de stock en conséquence. La solution la plus commune pour modéliser de telles situations consiste à considérer un horizon de temps fini de longueur T , discrétisé en *périodes* notées $t = 1, \dots, T$. Les problèmes auxquels nous nous intéressons sont déterministes, c'est-à-dire que l'on suppose la demande connue pour les T périodes à venir. Contrairement au cas continu, propice à l'application de méthodes analytiques, cette approche fait appel à des méthodes combinatoires et permet également de traiter efficacement des systèmes dont la demande varie avec le temps. Dans cette partie, nous présentons plusieurs modèles fondateurs de la gestion de stocks déterministe dans le cas discret, ainsi que divers algorithmes de résolution classiques de la littérature.

2.2.1 Modèles à un seul échelon

Dimensionnement de lot : le modèle ULSP et l'algorithme de Wagner-Whitin

Les problèmes de dimensionnement de lot (ou *lot-sizing problems* en anglais), également désignés par le sigle *ULSP* (*Uncapacitated Lot Size Problem*), considèrent un seul échelon faisant face à la demande de ses clients sur un horizon de temps fini et discret. Les notations suivantes sont utilisées pour les différents paramètres du modèle :

- d_t : demande pour la période $t = 1, \dots, T$.
- K_t : coût fixe de commande à la période t .
- c_t : coût unitaire de commande à la période t .
- h_t : coût unitaire de stockage à la période t .

On suppose qu'à chaque période se produit la séquence d'événements suivante :

1. Une commande est passée et l'entrepôt reçoit les unités commandées. Le délai de livraison est encore considéré comme nul sans perte de généralité.
2. La demande de la période est satisfaite et le niveau de stock décroît en conséquence.
3. Le nouveau niveau de stock est constaté et les coûts de stockage (ou éventuellement de rupture) s'appliquent.

L'objectif des modèles présentés consiste à minimiser la somme des coûts encourus (commande, stockage, etc.) sur l'horizon de temps considéré.

Dans ce qui suit, on note q_t la quantité commandée en période t et x_t le niveau de stock à la fin de la période t , c'est-à-dire après que la demande d_t ait diminué le stock. On peut donc déduire le stock à la fin de la période t à partir du niveau de stock en période $t - 1$ et de la quantité q_t (on suppose $x_0 = 0$) :

$$x_t = x_{t-1} + q_t - d_t \quad (2.11)$$

Comme pour le modèle EOQ présenté en §2.1.1, on suppose que la rupture de stock est interdite : $x_t \geq 0$ pour tout t . On peut facilement prouver en se basant sur des arguments similaires à ceux utilisés dans sa version continue que la propriété ZIO est également dominante pour ce type de modèle discret (voir Wagner et Whitin (1958) pour la preuve de ce résultat).

Propriété 5 (ZIO pour les modèles discrets).

$$\forall t = 1, \dots, T, \quad q_t > 0 \Rightarrow x_{t-1} = 0 \quad (2.12)$$

Ainsi on peut restreindre notre champ de recherche aux politiques qui respectent cette propriété : une commande n'est passée en période t que si le stock restant à la fin de la période précédente est nul. Enfin, on peut noter que l'évolution du stock est différente de celle du modèle continu. La demande diminue le stock instantanément au début de chaque

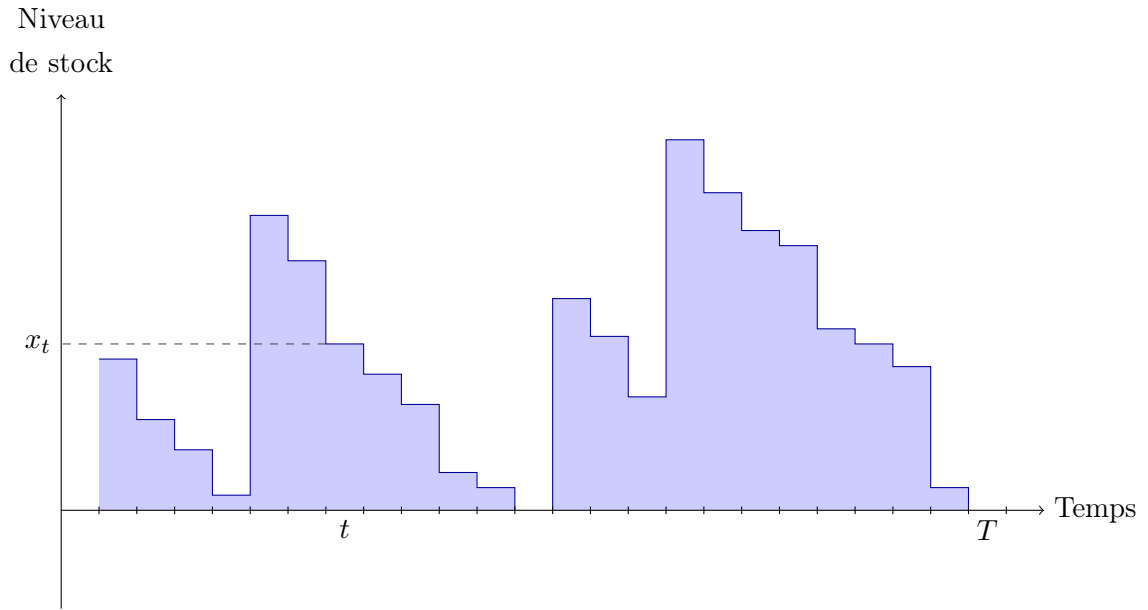


FIGURE 2.6 – Exemple d'évolution du stock pour le problème ULSP (politique non ZIO)

période et la courbe d'évolution du niveau de stock a une forme de fonction en escalier, comme celle représentée sur la Figure 2.6.

Comme la plupart des problèmes combinatoires, le ULSP peut être formulé comme un programme linéaire mixte. On rappelle que l'objectif est de minimiser la somme des coûts encourus sur l'horizon de temps considéré. En plus des variables de décision q_t déjà introduites plus haut, on note y_t la variable binaire égale à 1 si l'entrepôt passe commande en période t et 0 sinon. Enfin, pour deux périodes quelconques s et t , on définit le paramètre suivant, qui représente la demande cumulée de s à t :

$$d_{s,t} = \begin{cases} 0 & \text{si } s > t \\ \sum_{u=s}^t d_u & \text{sinon} \end{cases} \quad (2.13)$$

On peut alors formuler le problème ULSP grâce au programme linéaire mixte suivant :

$$\begin{aligned} \text{Minimiser } w &= \sum_{t=1}^T K_t y_t + c_t q_t + h_t x_t \\ \text{s.c. } x_t &= x_{t-1} + q_t - d_t \quad \forall t = 1, \dots, T \\ q_t &\leq y_t d_{t,T} \quad \forall t = 1, \dots, T \\ x_0 &= 0 \\ y_t &\in \{0, 1\} \quad \forall t = 1, \dots, T \\ q_t &\geq 0 \quad \forall t = 1, \dots, T \end{aligned}$$

Cette formulation en programme linéaire mixte a notamment été exploitée pour trouver des propriétés polyédrales du problème (voir Pochet et Wolsey (1988, 1994)) ou définir l'enveloppe convexe des solutions (voir Küçükyavuz et Pochet (2009)).

De nombreux autres travaux de recherche en lien avec le ULSP se concentrent sur la recherche d'algorithmes efficaces pour résoudre le problème à l'optimal. Les fondations de cette approche ont été posées par Manne (1958) et Wagner et Whitin (1958) qui étudient un modèle dans lequel les coûts de commande unitaires sont nuls ($c_t = 0$ pour tout t). Néanmoins, leur modèle peut en réalité être résolu sous une hypothèse plus faible dite de *non spéculation*, où l'inégalité suivante est satisfaite pour toute période $t < T$:

$$c_{t+1} \leq c_t + h_t \quad (2.14)$$

En d'autres termes, cette hypothèse stipule qu'il est avantageux (hors coût fixe) de commander les unités le plus tard possible pour servir une demande. En particulier, cela permet de se restreindre aux politiques de type *FCFS* (*First Come First Served*) où les demandes sont satisfaites selon leur ordre d'arrivée. On peut prouver que lorsque l'inégalité (2.14) est satisfaite pour tout t , on peut se ramener sans perte de généralité à des paramètres c_t nuls : dans la suite de ce chapitre on adopte donc cette hypothèse pour faciliter la compréhension générale des techniques présentées. On suppose également sans perte de généralité que le stock initial est nul et que $d_1 > 0$. Dans ce contexte, l'algorithme de Wagner-Whitin, introduit par les auteurs du même nom, résout ce problème à l'optimal en temps polynomial. Leur méthode est basée sur la programmation dynamique (voir Bellman (1957)) et utilise la définition récursive de la fonction $f(\cdot)$ suivante :

$$f(0) = 0 \quad (2.15)$$

$$f(1) = K_1 \quad (2.16)$$

$$\forall t \geq 2, f(t) = \begin{cases} \min_{1 \leq s \leq t} \left\{ K_s + \sum_{u=s}^{t-1} h_u d_{u+1,t} + f(s-1) \right\} & \text{si } d_t > 0 \\ f(t-1) & \text{sinon} \end{cases} \quad (2.17)$$

Il est aisé de vérifier que la fonction $f(t)$ renvoie le coût minimum encouru par le système pour servir l'ensemble des demandes jusqu'à la période t incluse. Les équations (2.15) et (2.16) correspondent aux cas de base. En se basant sur le principe de sous-optimalité, l'équation (2.17) décompose le coût pour servir les demandes $1, \dots, t$ en deux parties qui dépendent de la dernière commande passée par le système. Si celle-ci intervient en période $s \leq t$, la première partie correspond alors au coût minimum $f(s-1)$ encouru pour servir les demandes d_1, \dots, d_{s-1} , tandis que la seconde se compose du coût de commande K_s et des éventuels coûts de possession pour servir les demandes d_s, \dots, d_t avec des unités provenant de cette commande.

L'algorithme récursif proposé par Wagner et Whitin se base sur le même principe pour déterminer les périodes de commande. Pour tout $t = 1, \dots, T$, on note π_t^* la politique

optimale pour le problème ULSP limité au t premières périodes de l'horizon. L'algorithme commence par déterminer la meilleure politique π_1^* pour servir la demande d_1 : trivialement, celle-ci consiste à passer une commande en période 1. Soit $t = 2, \dots, T$ et supposons que pour tout $s = 2, \dots, t$ on connaisse la politique optimale π_{s-1}^* pour servir les demandes d_1, \dots, d_{s-1} : par définition son coût est égal à $f(s-1)$. Ces valeurs sont connues grâce aux itérations précédentes de l'algorithme, il est donc facile de déterminer la période s^* qui minimise l'expression (2.17). La politique optimale π_t^* consiste à appliquer $\pi_{s^*-1}^*$ jusqu'à la période $s^* - 1$, puis commander une quantité $d_{s^*,t}$ en période s^* . L'algorithme procède ainsi par itérations successives jusqu'à obtenir la politique optimale π_T^* pour l'horizon entier.

On peut noter que les politiques π_1^*, \dots, π_T^* sont toutes calculées pendant l'exécution. L'itération correspondant à la recherche de π_t^* nécessitant t opérations pour déterminer s^* , l'algorithme effectue au total $\sum_{t=1}^T t = T(T+1)/2$ opérations, pour une complexité finale en $O(T^2)$. Un certain nombre de travaux se sont employés à développer des algorithmes plus efficaces pour résoudre le ULSP. Les plus importants d'entre eux ont été proposés par Federgruen et Tzur (1991), Wagelmans et al. (1992) et Aggarwal et Park (1993), qui utilisent des techniques avancées de programmation dynamique pour résoudre une version générale du problème avec une complexité finale $O(T \log(T))$. De plus dans le cas particulier de modèles qui respectent les hypothèses de Wagner et Whitin (voir inégalité (2.14)), une politique optimale peut être construite en temps linéaire, c'est-à-dire pour une complexité en $O(T)$. En particulier, ces méthodes de résolution peuvent être utilisées pour obtenir des solutions optimales de sous-problèmes de type ULSP issus de la décomposition de systèmes multi-échelon complexes, comme nous le présentons dans les chapitres 4 et 5.

Les méthodes présentées jusqu'à présent permettent toutes d'obtenir une politique optimale pour le problème ULSP, mais il existe également dans la littérature de nombreuses heuristiques pour ce modèle. La plus connue d'entre elles est sans doute celle de Silver et Meal (1973), qui consiste à minimiser le coût unitaire (commande et possession) encouru sur un intervalle de commande. Une autre technique utilise la demande moyenne sur l'horizon comme taux de demande et détermine une quantité de commande grâce à la formule EOQ définie par l'équation (2.3). Bien que très utilisées en pratique, ces deux heuristiques peuvent produire une solution de coût arbitrairement grand comparé à l'optimum pour certaines instances, comme l'ont démontré Axsäter (1982) et Bitran et al. (1984). D'autres auteurs se sont donc attachés à proposer des algorithmes à garantie de performance constante. Axsäter (1982) a ainsi présenté une 2-approximation pour un modèle à coûts constants. De même, De Matteis et Mendoza (1968) et Bitran et al. (1984) ont développé un algorithme d'approximation pour le ULSP quand le coût fixe de commande est constant ($K_t = K$ pour tout t). Leur méthode consiste à équilibrer le coût de possession encouru sur un intervalle de commande avec le coût fixe de commande, comme c'est le cas pour la politique optimale EOQ dans le modèle continu stationnaire. Ainsi, si s est la dernière période de commande

de la politique, la prochaine commande est passée à la période t telle que

$$\sum_{u=s}^{t-1} h_u d_{u+1,t-1} \leq K < \sum_{u=s}^t h_u d_{u+1,t} \quad (2.18)$$

Axsäter (1982) et Bitran et al. (1984) ont prouvé que la garantie de performance de cette méthode est égale à 2. Une idée similaire est appliquée au cas continu dans le chapitre 3 : la politique obtenue présente également une garantie de performance égale à 2.

Pour conclure cette partie, il est intéressant de mentionner que l'on peut également modéliser un problème ULSP sous forme d'un graphe $G = (V, E)$ dans les cas où la propriété ZIO est dominante. L'ensemble V des sommets est constitué des périodes $1, \dots, T$, auxquelles on ajoute une période artificielle $T + 1$ modélisant la fin de l'horizon de temps. Un arc $(u, v) \in E$ correspond à un intervalle de commande, c'est-à-dire à deux périodes de commande u et v telles que pour tout $w \in \{u + 1, \dots, v - 1\}$ on a $q_w = 0$. En résumé, on définit donc :

$$\begin{aligned} V &= \{0, \dots, T + 1\} \\ E &= \{(u, v) : u \in V \text{ et } v \in V \setminus \{u\}\} \end{aligned}$$

La longueur $l_{u,v}$ de l'arc $(u, v) \in E$ correspond au coût de commande en u plus les coûts de possession encourus pour servir les demandes d_u, \dots, d_{v-1} avec des unités commandées en u .

$$l_{u,v} = K_u + \sum_{t=u}^{v-1} h_t d_{t+1,v-1} \quad (2.19)$$

La Figure 2.7 présente un exemple de modélisation d'un problème ULSP par un graphe. Une solution pour le problème est alors équivalente à un chemin dans le graphe G du sommet 1 au sommet $T + 1$. Les sommets utilisés par ce chemin correspondent aux instants de commande tandis que sa longueur totale est égale au coût de la solution ZIO correspondante. La solution optimale au problème ULSP ainsi modélisé est donnée par un plus court chemin de 1 à $T + 1$ dans le graphe G . On peut par exemple utiliser l'algorithme de Dijkstra pour déterminer un tel chemin en temps $O(T^2)$.

Extensions du problème ULSP

La littérature des modèles discrets de gestion de stocks propose un grand nombre de variantes du modèle ULSP dont nous ne pouvons pas donner de liste exhaustive ici. Néanmoins, nous introduisons succinctement certaines d'entre elles dans cette partie afin de fournir au lecteur un contexte suffisant à la compréhension des contributions présentées dans les chapitres 4 et 5.

Si le problème ULSP admet un algorithme de résolution polynomial, l'ajout d'une contrainte de capacité sur les quantités commandées le rend extrêmement complexe. Dans

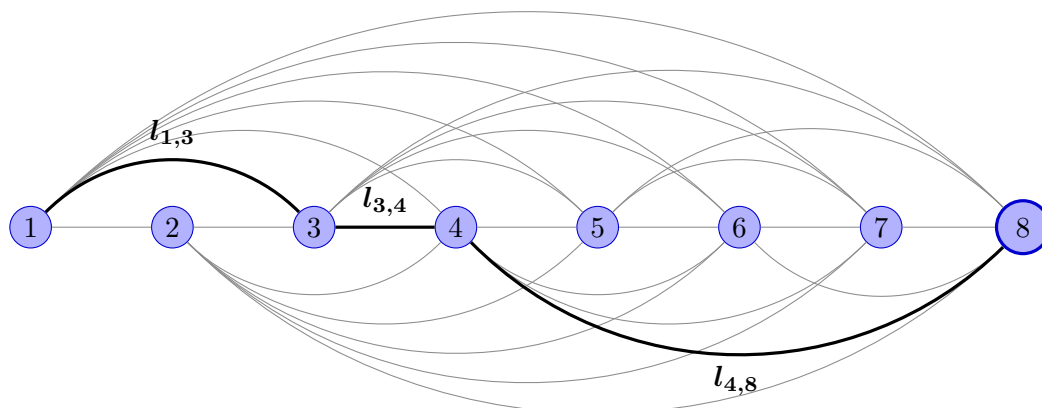


FIGURE 2.7 – Modélisation d'un problème ULSP avec $T = 7$ par un graphe : le chemin indiqué en gras correspond à la solution ZIO qui commande en périodes 1, 3 et 4.

ce modèle appelé CLSP (pour *Capacitated Lot-Sizing Problem*), une quantité q_t commandée par l'entrepôt en période t ne peut dépasser un certain seuil $q_t \leq C_t$. Lorsque les capacités de commande sont stationnaires ($C_t = C$ pour toute période t), Florian et Klein (1971) proposent un algorithme basé sur la programmation dynamique qui résout le problème en temps $O(T^4)$. Ce résultat a par la suite été amélioré par van Hoesel et Wagelmans Hoesel and Wagelmans (1996) qui présentent un algorithme de résolution de complexité $O(T^3)$. En revanche lorsque la capacité de commande varie selon la période, ce même modèle est prouvé NP-difficile par Florian et al. (1980) et Bitran et al. (1982). Enfin, van Hoesel et Wagelmans (2001) ont développé un FPTAS pour ce problème, c'est-à-dire qu'ils proposent une $(1 + \varepsilon)$ -approximation pour n'importe quel $\varepsilon > 0$ dont la complexité est polynomiale en T et en $1/\varepsilon$.

Les modèles dans lesquels la rupture de stock est autorisée occupent également une place importante dans la littérature, qui les distingue principalement en deux catégories : les modèles avec *mise en attente* et les modèles avec *ventes perdues*. Les premiers se rapprochent du modèle du même nom présenté pour le modèle continu. Une demande non satisfaite immédiatement est mise en attente et servie par une commande future : chaque unité de demande en attente à la période t encourt alors un coût b_t . Néanmoins toutes les demandes doivent être satisfaites à la fin de l'horizon de temps. Ce type de modèle a notamment été étudié par Zangwill (1966a), qui propose un algorithme basé sur la programmation dynamique pour résoudre le problème en temps polynomial. Un tel problème peut également être représenté par un graphe biparti $G = (V_1 \cup V_2, E)$ (voir Zipkin (2000)), où $V_1 = \{1, \dots, T + 1\}$ correspond à l'ensemble des périodes de l'horizon et $V_2 = \{\kappa_1, \dots, \kappa_T\}$ représente les instants de commandes potentiels d'une politique. L'ensemble des arcs du graphe se décompose en deux sous ensembles : $E = E^- \cup E^+$, avec $E^- = \{(s, \kappa_t) : t \geq s\}$

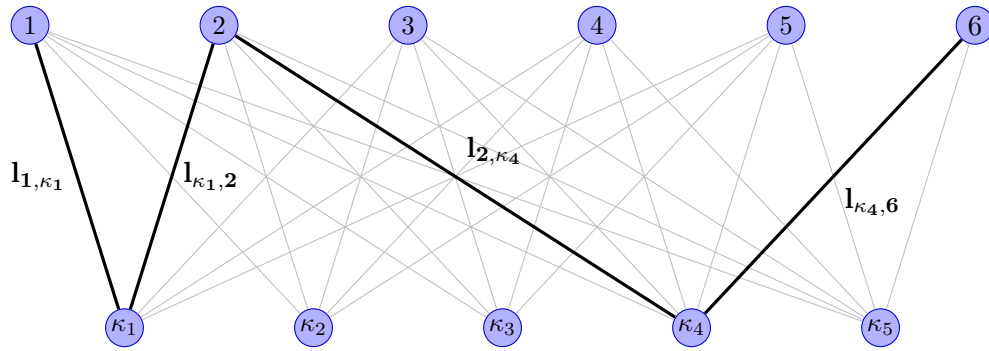


FIGURE 2.8 – Modélisation d'un problème ULSP avec mise en attente par un graphe ($T = 5$) : la politique correspondante au chemin en gras commande en période 1 pour servir les demandes d_1 et d_2 et en période 4 pour servir les demande d_3 , d_4 et d_5 .

et $E^+ = \{(\kappa_t, u) : u > t\}$. De la même manière que pour le ULSP sans rupture, un chemin dans ce graphe décrit une politique pour le modèle. Si deux arcs consécutifs $(s, \kappa_t) \in E^-$ et $(\kappa_t, u) \in E^+$ sont utilisés par le chemin, les demandes $d_s, d_{s+1}, \dots, d_{u-1}$ sont servie par une commande en période t : plus précisément les demandes d_s, \dots, d_{t-1} sont mises en attente jusqu'à t tandis qu'une quantité égale à $d_{t,u-1}$ approvisionne le stock en période t pour servir les demandes d_t, \dots, d_{u-1} . On peut alors associer les coûts suivants aux différents arcs du graphe :

$$(s, \kappa_t) \in E^- : l_{s, \kappa_t} = \sum_{v=s}^{t-1} b_v d_{s,v} \quad (2.20)$$

$$(\kappa_t, u) \in E^+ : l_{\kappa_t, u} = K_t + \sum_{v=t}^{u-1} h_v d_{v+1, u-1} \quad (2.21)$$

La Figure 2.8 présente un exemple d'un tel graphe. Une politique optimale pour le problème ULSP avec mise en attente considéré se ramène alors à un plus court chemin dans le graphe correspondant.

Contrairement au modèle ci-dessus, les problèmes avec ventes perdues considèrent que toute demande qui n'est pas immédiatement satisfaite quitte le système et entraîne un coût correspondant à la perte de la vente. Ces problèmes peuvent également être résolus à l'optimal par programmation dynamique. Un premier modèle a été introduit par Sandbothe et Thompson (1990), qui considèrent une version du problème avec limite de capacité. Lorsque cette contrainte est relâchée, Aksen et al. (2003) ont proposé un algorithme de résolution quadratique pour le problème, basé encore une fois sur la programmation dynamique.

Dans les chapitres 4 et 5, nous nous intéressons à des modèles multi-échelon composé d'un entrepôt central qui approvisionne plusieurs détaillants. Notre méthode de résolution s'appuie sur la décomposition d'un système en plusieurs sous-problèmes à un échelon, que nous recombinaisons ensuite pour obtenir une politique réalisable pour le problème d'origine. Dans ce contexte, nous nous appuyons également sur la résolution de modèles

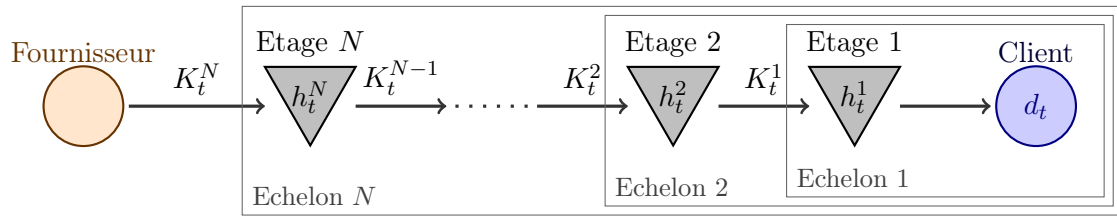


FIGURE 2.9 – Un système série

multi-item, qui consistent en un entrepôt unique distribuant N types de produits différents. Une commande passée en période t détaille alors la quantité q_t^i désirée pour chaque produit $i = 1, \dots, N$, mais un seul coût fixe de commande K_t est encouru (on parle en anglais de *joint-ordering cost*). En revanche, le coût de possession unitaire dépend du type de produit stocké : on note h_t^i ce coût pour le produit i en période t . La résolution de ce type de modèle peut se faire de la même façon que dans le cas à un seul produit (voir chapitre 4 pour une explication détaillée). Cette classe de problèmes a surtout fait l'objet de nombreux travaux dans sa version NP-difficile avec contrainte de capacité. Levi et al. (2008b) ont notamment démontré que ce problème est NP-difficile et introduit des inégalités valides, dites de couverture de flot, à sa formulation en programme linéaire mixte. En utilisant la relaxation du programme linéaire correspondant, les auteurs ont également proposé une 2-approximation pour ce modèle. Certains cas particuliers de ce problème peuvent cependant être résolu à l'optimal en temps polynomial (voir Anily et al. (2009)).

2.2.2 Modèles multi-échelon

Systèmes série

Un système série est composé de N échelons qui se succèdent et par lesquels transitent des produits finis ou semi-finis, depuis un fournisseur extérieur jusqu'à la demande des clients. Seul l'échelon 1 est en contact direct avec la demande finale : les autres échelons servent simplement de relais ou correspondent à une étape de la transformation des matières premières ou produits semi-finis vers le produit final distribué aux clients. Pour $i = 1, \dots, N - 1$, l'échelon i approvisionne son stock en commandant des unités disponibles dans le stock de l'échelon $i + 1$, tandis que l'échelon N passe ses commandes auprès d'un fournisseur extérieur de capacité infinie. On considère ici encore un horizon de temps fini, de longueur T , durant lequel l'entrepôt 1 doit satisfaire sans délai la demande d_t des clients pour toute période $t = 1, \dots, T$. De façon similaire au cas de un seul échelon, l'entrepôt i encourt un coût fixe de commande K_t^i si il s'approvisionne auprès de son prédécesseur en période t . De plus, chaque unité stockée dans l'entrepôt i en période t encourt un coût de possession (unitaire) h_t^i . L'objectif est de trouver un plan de commande pour chaque entrepôt de manière à satisfaire toutes les demandes à temps en minimisant les coûts encourus sur l'horizon. La Figure 2.9 représente un système série à N échelons.

Parmi les modèles multi-échelon de la littérature, le système série est l'un des seuls pour

lequel on connaît des algorithmes de résolution efficaces (voir Zangwill (1966b) et Love (1972)). Ceux-ci sont basés sur la programmation dynamique et permettent de résoudre un problème série avec une complexité de l'ordre de $O(NT^4)$. Cependant, il existe des systèmes légèrement plus restrictifs, appelés systèmes série *imbriqués*, pour lesquels la procédure de résolution s'exécute en temps $O(NT^3)$: nous présentons succinctement cette dernière dans cette partie. Un système série imbriqué présente la caractéristique que toute politique réalisable est dominée par une politique imbriquée, selon la définition 7 énoncée ci-dessous.

Définition 7. *Une politique est dite imbriquée si pour tout $i = 1, \dots, N - 1$ la propriété suivante est vérifiée pour toute période $t = 1, \dots, T$:*

Si l'échelon $i+1$ commande en période t alors son successeur i dans la chaîne commande également en période t .

Les politiques imbriquées sont notamment dominantes si pour toute période t et tout échelon $i = 1, \dots, N - 1$, on a $h_t^i \geq h_t^{i+1}$. On peut alors définir alors le *coût de possession échelon*, noté \hat{h}_t^i , pour tout $i = 1, \dots, N - 1$ et toute période t :

$$\begin{aligned}\hat{h}_t^N &= h_t^N \\ \hat{h}_t^i &= h_t^i - h_t^{i+1}\end{aligned}\tag{2.22}$$

Dans ce qui suit, on se restreint à des systèmes série dont le coût échelon est positif ou nul ; en d'autres termes, on suppose que le coût de possession unitaire du produit augmente lorsque celui-ci remonte la chaîne du fournisseur vers le client. Pour un système vérifiant cette propriété, L'idée sous-jacente de l'algorithme est simple : pour tout étage $i = 1, \dots, N$ et toutes périodes $u \leq v$, on calcule les valeurs $\hat{l}_{u,v}^i$ comme dans l'équation (2.19) en utilisant le coût de possession échelon \hat{h}_t^i et la demande finale. On définit également $f_i(u, v)$ comme le coût optimal du système série restreint aux échelons $1, \dots, i$ sur l'horizon de temps $u, u + 1, \dots, v - 1$. Les valeurs $f_1(\cdot, \cdot)$ correspondent alors simplement à la résolution à l'optimal d'un système à un échelon obtenu à partir de l'entrepôt 1 pris séparément. On peut ensuite calculer $f_{i+1}(\cdot, \cdot)$ récursivement à partir de $f_i(\cdot, \cdot)$ et des longueurs $\hat{l}_{u,v}^{i+1}$: on définit un graphe $G_{i+1} = (V, E_{i+1})$ similaire à celui d'un problème à un échelon mais dont les longueurs des arcs sont modifiées selon la formule suivante :

$$l_{u,v}^{i+1} = \hat{l}_{u,v}^{i+1} + f_i(u, v)\tag{2.23}$$

La longueur d'un arc $(u, v) \in E_i$ du graphe G_i détermine donc le coût minimum encouru par le système restreint aux échelons $1, \dots, i$ entre les périodes u et v , si l'échelon i commande en période u . Un plus court chemin entre les sommets 1 et $T + 1$ dans le graphe G_i donne alors une solution optimale sur l'horizon de temps pour le système restreint. En particulier, un plus court chemin de 1 à T dans le graphe G_N correspond à une solution optimale pour le système série considéré.

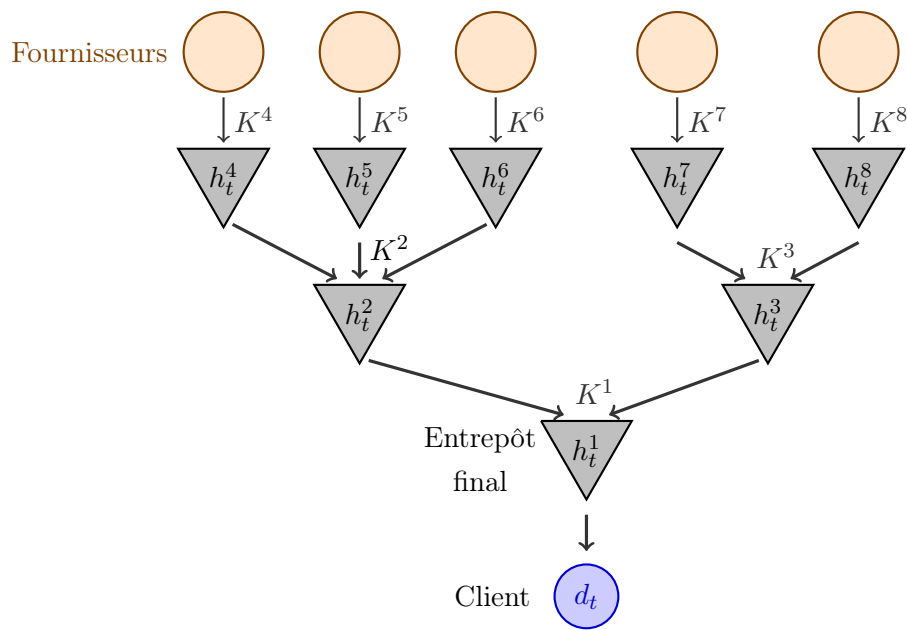


FIGURE 2.10 – Un exemple de système d'assemblage.

Systèmes convergents (assemblage)

On appelle systèmes convergents les réseaux d'entrepôts dans lesquels chaque point de stockage a au plus un successeur dans la chaîne logistique (voir Figure 2.10). Une telle structure est en général utilisée pour modéliser une chaîne d'assemblage, dans laquelle un entrepôt commande plusieurs produits semi-finis afin de les combiner et se rapprocher du bien final (indexé par 1 comme dans le système série), distribué au client par un unique entrepôt en bout de chaîne. Les hypothèses sont semblables à celles qui régissent un système série imbriqué : chaque entrepôt encourt un coût fixe de commande lorsqu'il réapprovisionne son stock et les coûts de possession unitaires augmentent au fur et à mesure que le produit se rapproche de la fin de son assemblage.

Bien que le système d'assemblage soit un modèle multi-échelon classique de la littérature, nos travaux ne se sont pas concentrés sur ce type de système convergent. Par souci de concision, nous ne le présentons donc pas en détail ici. On peut néanmoins noter que plusieurs résultats d'approximation existent déjà dans la littérature pour ce problème. En particulier, Roundy (1993) a introduit un modèle dont les paramètres sont constants et proposé une 2-approximation empruntant des idées à des politiques de gestion de stocks existantes pour des systèmes à temps continu. Plus tard, Levi et al. (2006) se sont intéressés à une version NP-difficile du problème avec paramètres variables et possibilité de mise en attente. Les auteurs adoptent une approche primal-dual qui leur permet notamment de construire une solution réalisable au problème avec une garantie de performance égale à 2.

Systèmes divergents (distribution)

Par le terme “systèmes divergents”, nous désignons des réseaux logistiques dans lesquels les entrepôts non terminaux (c’est-à-dire qui ne sont pas en contact direct avec la demande des clients) ont un seul fournisseur, mais approvisionnent un ou plusieurs autres entrepôts de la chaîne logistique. L’étude de tels systèmes est pertinente dans le cas de larges réseaux de distribution à l’échelle d’un pays ou d’un continent : les produits achetés auprès de différents fournisseurs alimentent le stock d’un entrepôt central et celui-ci répartit ces unités entre différents entrepôts régionaux. Chaque entrepôt régional les distribue ensuite à des détaillants qui les utilisent pour servir la demande de leurs clients. Ce type de système soulèvent un grand nombre de problématiques, tant au niveau de la coordination que dans la répartitions des biens entre les différents acteurs du réseau. Les modèles qui dérivent de ces systèmes de distribution sont généralement complexes et certains d’entre eux sont maintenant considérés comme des problèmes classiques en théorie de la gestion de stocks.

Nos travaux se sont concentrés sur deux de ces problèmes, désignés dans la littérature sous le nom de problème OWMR (pour *One-Warehouse Multi-Retailer*) et JRP (pour *Joint Replenishment Problem*). Plus précisément, le problème OWMR vise à optimiser la distribution de produits via un réseau logistique composé d’un entrepôt central qui approvisionne N détaillants, sur un horizon de temps discret et fini, de longueur T . Le JRP est un cas particulier du problème OWMR, où l’entrepôt central sert uniquement de plateforme de routage pour répartir les produits entre les détaillants et ne garde aucune unité dans son stock.

Chaque détaillant $i = 1, \dots, N$ connaît la demande de ses clients d_1^i, \dots, d_T^i pour les T prochaines périodes et doit satisfaire chacune de ces demandes à temps. Il peut pour cela stocker des unités commandées au préalable à l’entrepôt central, qui lui-même commande des unités à un fournisseur extérieur au système. On désigne l’entrepôt central par l’index 0 et on définit les paramètres de coût du système comme suit :

- K_t^i : coût fixe de commande en période t pour l’entrepôt i
- h_t^i : coût de possession unitaire en période t pour l’entrepôt i

On suppose de plus que la rupture de stock est interdite. La Figure 2.11 représente un problème OWMR à N détaillants. L’objectif du problème est de trouver un plan de commande pour chaque entrepôt qui minimise la somme des coûts de fonctionnement encourus par le système sur l’horizon de temps.

Ces deux modèles (OWMR et JRP) ont été prouvé NP-difficile par Arkin et al. (1989), qui proposent une réduction polynomiale du JRP au problème 3-SAT, lui-même NP-difficile (voir Garey et Johnson (1979)). Par extension, le problème OWMR est donc également NP-difficile. De plus, Chan et al. (2000) ont montré qu’il n’existe pas d’algorithme d’approximation à garantie de performance constante pour ce problème lorsque les coûts fixes de commande des détaillants sont variables dans le temps. Leur preuve s’appuie sur une réduction du JRP au problème de couverture d’ensemble. Un résultat classique de la lit-

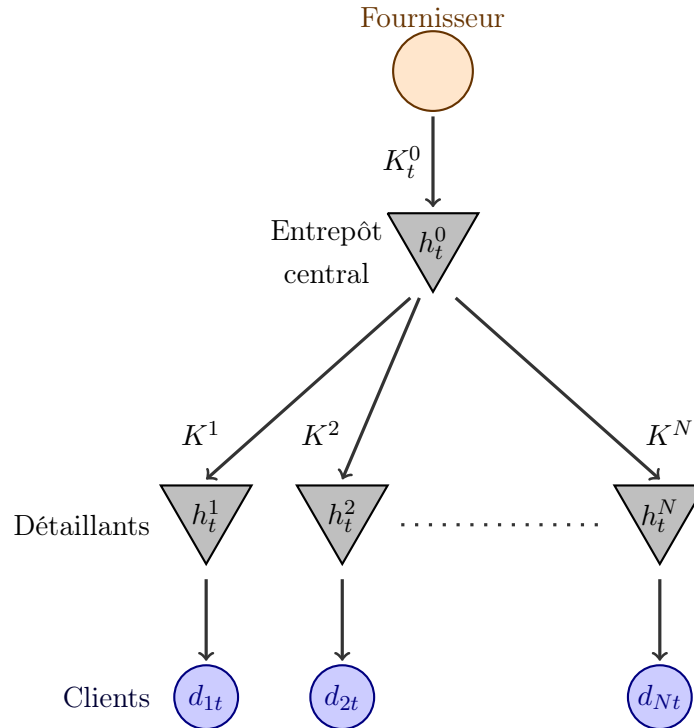


FIGURE 2.11 – Un système divergent de type OWMR

térature (voir Feige (1998)) stipule que si $P \neq NP$, ce dernier n'admet pas d'algorithme polynomial dont la garantie de performance soit meilleure que $\gamma \log(N)$, où $\gamma > 0$ et N est la taille de l'instance considérée. Ainsi, dans l'optique de proposer des algorithmes performants pour la résolution de ces problèmes, nous nous restreindrons dans ce qui suit à des modèles pour lesquels $K_t^i = K^i$ pour tout $t = 1, \dots, T$ et $i = 1, \dots, N$. Notons que cette hypothèse concerne uniquement les détaillants : les coûts de commande de l'entrepôt central dépendent *a priori* de la période considérée. D'importantes propriétés structurelles du modèle ont été introduites par Schwarz (1973), notamment la dominance des politiques ZIO à chaque entrepôt. On rappelle que pour servir une demande d_t^i , une unité doit d'abord être commandée par l'entrepôt central à une certaine période r , puis par le détaillant i en période s , avec $r \leq s \leq t$. Dans ce qui suit, on notera $[r, s]$ ($r \leq s$) une telle paire de périodes de commande. Comme les coûts de possession sont positifs et que les quantités commandées ne sont pas bornées par une contrainte de capacité, on peut se restreindre aux politiques qui servent l'intégralité de chaque demande d_t^i en utilisant une unique paire $[r, s]$.

De nombreux autres travaux se sont focalisés sur la recherche de solutions réalisables satisfaisantes pour le problème OWMR et le JRP : nous en introduisons quelques uns pour conclure ce chapitre. Les contributions que nous présentons aux chapitres 4 et 5 se concentrent sur ces deux modèles et proposent une nouvelle technique de résolution simple et originale.

Heuristiques pour le JRP. La première heuristique connue pour ce problème, introduite par Kao (1979), considère un modèle de JRP dans lequel tous les coûts sont stationnaires (c'est-à-dire $K_t^0 = K^0$ et $h_t^i = h^i$ pour tout i, t). La méthode proposée procède en plusieurs passes qui comportent chacune au plus N itérations. À chaque itération, l'algorithme résout un problème à un échelon avec des coûts de commande modifiés. Plus précisément, on introduit pour tout $i = 1, \dots, N$ l'ensemble $\tau_i \subset \{1, \dots, T\}$ qui contient les périodes de commande de la politique pour le ULSP correspondant au détaillant i . Initialement, on pose $\tau_i = \emptyset$ pour tout i . À chaque passe, l'algorithme considère successivement chaque détaillant i et lui associe le problème ULSP avec les paramètres suivant :

- La demande en période t est égale à d_t^i (c'est-à-dire que seule la demande du détaillant i est prise en compte).
- Le coût de possession unitaire est stationnaire, égal à h^i .
- La valeur du coût fixe de commande pour une période t est définie par l'équation suivante :

$$K_t = \begin{cases} K^i & \text{si } t \in \bigcup_{j \neq i} \tau_j \\ K^i + K^0 & \text{sinon} \end{cases} \quad (2.24)$$

L'algorithme résout alors ce problème à l'optimal et redéfinit τ_i comme l'ensemble des périodes de commande du détaillant i ainsi obtenues. Ainsi à chaque itération, un des ensembles τ_i est modifié et affecte la définition des problèmes ULSP suivants. L'algorithme procède ainsi jusqu'à résoudre le problème ULSP associé au détaillant N , puis démarre une nouvelle passe en commençant à nouveau par le détaillant 1. La procédure se poursuit jusqu'à ce que la solution converge. On peut prouver que cela se produit en un nombre fini de passes et que chaque itération améliore la solution courante; en revanche Joneja (1990) trouve un exemple pour lequel la solution finale obtenue peut être arbitrairement mauvaise.

Joneja (1990) présente également une nouvelle heuristique pour le même problème. L'idée sous-jacente s'inspire de celle introduite dans De Matteis et Mendoza (1968) et consiste à équilibrer coûts de possession et coûts de commande sur un cycle d'approvisionnement. L'algorithme parcourt l'horizon de la période 1 à la période T et maintient en permanence à jour une liste de détaillants "candidats" à la commande. Supposons que la procédure traite la période t et pour $i = 1, \dots, N$, soit t_0 la dernière période de commande de l'entrepôt central précédant t et t_i la dernière période de commande du détaillant i précédant t . On définit alors les deux quantités suivantes :

$$H_{it}(t_i) = \sum_{u=t_i}^t h^i d_{[u+1, t-1]}^i \quad (2.25)$$

$$S_{it}(t_0, t_i) = h^i (t_0 - t_i) \sum_{u=t_0}^t d_u^i \quad (2.26)$$

En d'autres termes, $H_{it}(t_i)$ représente le coût de stockage total encouru par le détaillant i depuis sa dernière commande tandis que $S_{it}(t_0, t_i)$ correspond à l'économie potentielle que réaliserait i si il commandait en période t_0 . On dit que le détaillant i est *candidat à la commande* en période t si $H_{it}(t_i) \geq K^i$. L'algorithme se décrit alors comme suit :

Étape 1 On fixe $t_i = 1$ pour tout $i = 0, \dots, N$.

Étape 2 On incrémente $t : t \leftarrow t + 1$. Si $t > T$, la procédure se termine. Sinon, pour tout i tel que $S_{it}(t_0, t_i) > K^i$ le détaillant i se rajoute à la commande i et on fixe $t_i = t_0$.

Étape 3 Si $\sum_{i=1}^N (H_{it}(t_i) - K^i)^+ \geq K^0$, l'entrepôt central passe une commande qui inclut tous les candidats à la commande en période t . On fixe alors $t_0 = t$ et $t_i = t$ pour tout i candidat à la commande en période t . On continue en appliquant l'Étape 2 de l'algorithme.

Joneja propose une preuve que l'heuristique simple décrite ci-dessus est une 3-approximation pour le JRP. Malheureusement, les arguments qu'il avance pour démontrer ce résultat se sont depuis avérés erronés (Simchi-Levi 2002, communication privée).

Plus récemment, Chan et al. (2000) ont étudié une version du JRP dont les coûts de commande sont concaves par morceaux. Dans ce cadre, les politiques ZIO ne sont plus dominantes, ce qui rend l'étude du modèle plus complexe. Néanmoins, leurs travaux montrent qu'appliquer la meilleure politique ZIO à ce type de système ne détériore la solution que d'un facteur $\frac{4}{3}$ dans le pire cas. En s'appuyant sur ce résultat, ils proposent alors une heuristique qui utilise la relaxation linéaire du problème d'origine pour construire une politique ZIO pour ce JRP général. La procédure obtenue se révèle efficace en pratique, comme le montre leur étude numérique.

Enfin, Levi et al. (2006) ont étudié un modèle de JRP avec des coûts de possession généraux et ont développé une 2-approximation pour ce problème basée sur une méthode primal-dual. Décrire leur algorithme dans le détail serait trop long ici, mais les idées principales utilisées peuvent être résumées simplement. Leur technique se décompose en deux phases. Durant la première ils construisent une solution fractionnaire au problème dual du JRP considéré et lui associent une solution entière du problème primal. Leur algorithme construit cette solution pas à pas, en incrémentant les variables duales jusqu'à des valeurs-seuils qui déterminent la solution primale associée. Ils affinent ensuite la solution obtenue afin de diminuer les coûts encourus lors de la première phase, tout en conservant une solution réalisable au JRP considéré. Cette approche leur permet de démontrer que la politique produite par leur algorithme a un coût égal dans le pire cas à deux fois le coût d'une politique optimale. De plus, ils démontrent que leur approche est suffisamment générique pour pouvoir être adaptée facilement à d'autres problèmes de gestion de stocks, comme le JRP avec mise en attente et le système d'assemblage.

Algorithme d'approximation pour le problème OWMR. Si il existe déjà un certain nombre de résultats d'approximation pour le JRP, on trouve beaucoup moins d'heuristiques pour le problème plus général OWMR dans la littérature. Le seul algorithme d'approximation à garantie de performance constante a été proposé par Levi et al. (2008c) et nous donnons ici les principales idées de leur technique.

On commence par présenter la formulation étendue du problème sous forme de programme linéaire en nombres entiers sur laquelle ils s'appuient pour développer leur algorithme. Pour tout $i = 0, \dots, N$ et $t = 1, \dots, T$, on introduit y_t^i , la variable binaire égale à 1 si l'entrepôt i commande en période t et 0 sinon. On rappelle qu'il existe une solution optimale au problème dans laquelle toute demande est satisfaite au moyen d'unités commandées en une unique période r à l'entrepôt central et une unique période $s \geq r$ au détaillant i . On définit en conséquence pour tout détaillant i la variable binaire x_{rs}^{it} , égale à 1 si la demande d_t^i est commandée en périodes r et s respectivement à l'entrepôt central et au détaillant i . Enfin, on définit pour tout $r \leq s \leq t$ le paramètre H_{rs}^{it} égal au coût de possession encouru par le système pour stocker la demande d_t^i à l'entrepôt central de la période r à la période $s - 1$, puis au détaillant i de la période s à la période $t - 1$:

$$H_{rs}^{it} = \left(\sum_{u=r}^{s-1} h_u^0 + \sum_{v=s}^{t-1} h_v^i \right) d_t^i \quad (2.27)$$

En réalité, les hypothèses faites par Levi et al. (2008c) sur les paramètres H_{rs}^{it} sont plus générales que la définition donnée ici par l'équation (2.27), mais par souci de clarté nous nous restreindrons à ce cas particulier. En utilisant les notations introduites plus haut, ils proposent une formulation du problème grâce au programme linéaire en nombres entiers (IP) suivant :

$$(IP) \left\{ \begin{array}{l} \text{Minimiser} \quad \sum_{r=1}^T y_r^0 K_r^0 + \sum_{i=1}^N \sum_{s=1}^T y_s^i K^i + \sum_{i=1}^N \sum_{t=1}^T \sum_{r,s:r \leq s \leq t} H_{rs}^{it} x_{rs}^{it} \\ \text{s.c.} \quad \sum_{r,s:r \leq s \leq t} x_{rs}^{it} = 1, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T \text{ tels que } d_t^i > 0 \quad (1) \\ \sum_{r:r \leq s} x_{rs}^{it} \leq y_s^i, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall s = 1, \dots, t \quad (2) \\ \sum_{s:r \leq s \leq t} x_{rs}^{it} \leq y_r^0, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall r = 1, \dots, t \quad (3) \\ x_{rs}^{it}, y_r^0, y_s^i \in \{0, 1\}, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \\ \forall s = 1, \dots, t, \forall r = 1, \dots, s \quad (4) \end{array} \right.$$

La contrainte (1) permet de s'assurer que chaque demande est satisfaite à temps. Les contraintes (2) et (3) garantissent qu'une demande ne peut être servie que par une commande existante du détaillant i et de l'entrepôt central, respectivement. Il est aisé de vérifier que la fonction objectif donne bien le coût total encouru selon l'affectation des

variables et par conséquent le programme (IP) est une formulation correcte du problème OWMR considéré.

On considère maintenant le programme relaxé (LP) dérivé de (IP), dans lequel on remplace la contrainte (4) par $x_{rs}^{it}, y_r^0, y_s^i \geq 0$. On peut résoudre (LP) à l'optimal et le coût de la solution obtenue (\hat{x}, \hat{y}) est une borne inférieure au coût optimal du problème original, noté \mathcal{C}^* . L'algorithme développé par Levi et al. (2008c) utilise cette solution relaxée pour déterminer une politique réalisable pour le problème OWMR initialement considéré. On commence par définir deux paramètres : un pas $c \in (0, 1]$ arbitraire et un coefficient de bascule α_0 tiré aléatoirement et de manière uniforme dans l'intervalle $(0, c]$. Soit $\hat{Y} = \sum_{r=1}^T \hat{y}_r^0$ le poids total cumulé des commandes fractionnaires passées par la solution optimale de (LP) et soit W le plus petit multiple de c plus grand que \hat{Y} : $W = \lceil \hat{Y}/c \rceil$. Le coefficient de bascule α_0 permet alors de définir un ensemble de points de bascule pour l'entrepôt central $\{\alpha_0 + cw : w = 0, \dots, W - 1\}$, qui servent ensuite à déterminer en quelles périodes celui-ci commande dans la solution finale proposée par l'algorithme. Plus précisément, l'entrepôt central commande en période $m = 1, \dots, T$ si l'intervalle $(\sum_{r=1}^{m-1} \hat{y}_r^0, \sum_{r=1}^m \hat{y}_r^0]$ contient au moins un des points de bascule ainsi défini.

La politique de commande de l'entrepôt central est donc obtenue grâce au paramètre aléatoire α_0 et définit par conséquent un ensemble aléatoire de périodes de commande. Les auteurs prouvent que l'espérance du coût de commande encouru par l'entrepôt central dans la solution produite par leur algorithme est au plus égale à $1/c$ fois le coût de commande de l'entrepôt central dans la solution optimale de (LP). En faisant varier le pas c choisi, ils proposent également deux procédures similaires pour calculer les périodes de commande aux différents détaillants. Ils prouvent de plus qu'en sélectionnant la meilleure solution parmi les deux obtenues, l'espérance du coût total encouru par la politique choisie est dans le pire cas égale à 1.8 fois le coût total encouru par la solution optimale de (LP). Enfin, ils présentent une méthode pour transformer leur procédure aléatoire en procédure déterministe sans détériorer le ratio d'approximation. Ils obtiennent ainsi un algorithme d'approximation polynomial déterministe dont la garantie de performance est égale à 1.8.

Comme on peut le constater, la procédure proposée par Levi et al. (2008c) est relativement fastidieuse et implique la manipulation d'outils mathématiques complexes pour obtenir une solution réalisable au problème OWMR. Dans le chapitre 4, nous proposons une méthode simple et efficace pour déterminer une politique réalisable à ce problème à partir des politiques optimales de systèmes ULSP. Notre méthode s'avère également générique et s'applique à de nombreuses extensions du modèle OWMR classique. En particulier, le chapitre 5 généralise le principe de notre algorithme à des systèmes où la rupture de stock est tolérée.

2.3 Résumé des contributions

Dans cette partie, nous présentons succinctement les principales contributions de cette thèse. Notre objectif est de donner une vision globale des méthodes que nous proposons et de leur positionnement dans la littérature, sans rentrer dans les détails techniques des analyses et des hypothèses pour chaque modèle. Celles-ci seront par la suite formalisées dans les chapitres 3, 4 et 5.

2.3.1 Chapitre 3 : Algorithmes d'approximation pour systèmes à un seul échelon à temps continu

La partie 2.1 se concentre principalement sur des modèles dont la demande est stationnaire. Néanmoins, cette hypothèse n'est souvent pas vérifiée en pratique, où fluctuations saisonnières et phénomènes de mode influencent la demande et les coûts du système. La littérature comporte déjà plusieurs résultats tenant compte de ces variations, tels que les travaux fondateurs de Barbosa et Friedman (1978) présentés dans la partie 2.1.2. Néanmoins, l'ensemble des approches existantes se limite à traiter des cas particuliers de fonction de demande, dont les propriétés analytiques permettent de développer des solutions optimales. En pratique, cette stratégie peut s'avérer relativement fastidieuse et à ce jour aucune méthode générique efficace ne permet d'appréhender ces problèmes lorsque les demandes et les coûts sont des fonctions quelconques du temps.

Dans le chapitre 3, nous présentons une politique de commande pour des modèles à un seul échelon, qui ne requiert que peu d'hypothèses sur les paramètres du système. Plus précisément, nous supposons simplement que le taux de demande $\lambda(\cdot)$ et les coûts de possession $h(\cdot)$ sont des fonctions positives et continues par morceaux du temps. Notre approche repose sur un mécanisme de décision simple qui s'inspire de la propriété d'équilibrage des coûts de commande et de possession sur un cycle pour la politique optimale dans le cas d'une demande stationnaire. L'idée de notre algorithme est alors de calculer les différents instants de commande de façon itérative, en utilisant la dominance de la propriété ZIO pour déterminer les quantités de commande. Pour un instant de commande donné, nous calculons ainsi l'instant de commande suivant tel que les coûts de possession sur le cycle ainsi défini soit égaux au coût fixe de commande.

Nous prouvons que l'algorithme ainsi obtenu a une garantie de performance égale à 2 pour ce modèle dont la demande et les coûts de possession sont des fonctions quelconques. On peut noter que le concept d'équilibrage des coûts a déjà été utilisé dans la littérature des modèles discrets par Bitran et al. (1984), avec une garantie de performance similaire. Nous démontrons également que cette approche peut être étendue à différentes variantes du modèle de base et nous sommes ainsi en mesure de proposer des algorithmes d'approximation pour des modèles avec produits périssables, avec taux de production, avec mise en attente ou dont le coût fixe de commande est une fonction monotone du temps. Enfin, nous présentons une amélioration de la politique, avec une garantie de performance égale à $3/2$,

pour des cas particuliers de fonctions de demande et de coûts de possession. Notamment, les hypothèses des travaux de Barbosa et Friedman (1978) sont suffisantes pour appliquer notre procédure modifiée.

2.3.2 Chapitres 4 et 5 : Algorithmes d'approximation pour le problème OWMR et extensions aux modèles avec rupture de stock

Dans la partie 2.2, nous avons introduit plusieurs modèles de gestion de stock à temps discret composés d'un ou plusieurs échelons. Nous avons vu que la littérature comporte déjà des méthodes de résolution pour des problèmes comportant un seul entrepôt, dont le célèbre algorithme de Wagner et Whitin pour le problème ULSP. En revanche, à l'exception du cas des systèmes série, il existe peu de techniques efficaces pour traiter des modèles intégrant plusieurs entrepôts reliés entre eux. En particulier, les systèmes de distribution divergents évoqués dans ce chapitre allient un défi théorique considérable de par leur complexité avec une application pratique évidente dans de nombreux réseaux logistiques existants.

Dans le chapitre 4, nous considérons un problème OWMR à temps discret similaire à celui étudié par Levi et al. (2008c), dans lequel un entrepôt central approvisionne plusieurs détaillants qui font face à la demande finale des clients. Le problème étant NP-difficile, le calcul d'une solution optimale en temps raisonnable pour toute instance est irréaliste. Nous avons donc cherché à proposer un algorithme d'approximation simple, qui limite le temps d'exécution tout en garantissant la qualité de la solution obtenue. Globalement, notre procédure se divise en deux étapes majeures. Dans la première, nous brisons la corrélation entre les différents entrepôts du système et proposons une décomposition de celui-ci en plusieurs sous-problèmes à un seul échelon que l'on peut résoudre grâce à l'algorithme de Wagner et Whitin. Dans un deuxième temps, nous recombinaisons ces politiques indépendantes en synchronisant certaines commandes des détaillants avec l'entrepôt central. L'algorithme obtenu est donc purement combinatoire, c'est-à-dire qu'il ne fait pas appel à la résolution d'un programme linéaire.

Notre démarche nous permet de présenter plusieurs résultats nouveaux sur le problème OWMR considéré. D'abord, nous introduisons une nouvelle structure de coûts de possession, appelée structure métrique, qui généralise celle introduite par Levi et al. (2008c). Ces hypothèses moins contraignantes permettent entre autres de modéliser certains cas particuliers du problème. Nous démontrons également que le coût des politiques à un échelon mises en jeu dans les sous-problèmes de la décomposition constitue une borne inférieure au coût d'une solution optimale. Ce résultat propose donc une alternative à la borne inférieure classique issue de la relaxation linéaire du problème. Enfin, notre analyse montre que sous ces hypothèses relativement générales, notre algorithme a une garantie de performance égale à 2 pour une complexité quadratique en la taille de l'instance considérée. La méthode que nous proposons se révèle donc rapide et construit une solution réalisable de coût limité pour ce problème NP-difficile. De plus nous estimons sa mise en pratique assez aisée : la résolution des sous-problèmes peut facilement s'effectuer à l'aide de l'algorithme

de Wagner-Whitin ou de ses variantes, tandis que la recombinaison des politiques ainsi obtenues s'exécute en temps linéaire en la taille de l'instance.

Notre procédure se révèle suffisamment générique pour être adaptée à des structures de coût plus complexes, telles que des coûts de possession non linéaires ou des coûts de commande qui varient en fonction de la quantité commandée. À notre connaissance, il s'agit des premiers algorithmes de résolution à garantie de performance constante pour ces modèles généraux de OWMR. Enfin, nous menons en fin de chapitre une étude numérique comparant les performances de notre algorithme avec celui de Levi et al. sur un ensemble d'instances générées aléatoirement. Les résultats de ces expérimentations confirment la faible complexité de notre méthode, malgré une performance moyenne légèrement moins bonne que celle obtenue par Levi et al. (2008c).

Dans le chapitre 5, nous étendons le modèle précédent pour intégrer la possibilité de mettre en attente des clients ou de perdre certaines ventes. Nous reprenons une structure de coûts de type métrique, mais celle-ci est étendue afin d'offrir la possibilité de ne pas satisfaire les demandes à temps. Nous démontrons que la décomposition introduite pour le système précédent peut être adaptée à ces nouveaux modèles et nous proposons de façon analogue de nouvelles bornes inférieures pour le coût optimal du problème avec mise en attente des clients, ainsi que pour celui du problème avec ventes perdues. Nous adaptons ensuite notre procédure pour chacun de ces deux modèles, obtenant dans les deux cas des garanties de performance constantes. Pour le modèle avec mise en attente, la politique obtenue est une 3-approximation et préserve la faible complexité algorithmique démontrée pour le modèle classique. Lorsque le problème est réduit à son cas particulier du JRP, nous améliorons la garantie de performance, avec un ratio d'approximation égal à 2. Dans le cas du modèle avec ventes perdues, une adaptation simple de l'algorithme nous permet d'obtenir une 2-approximation, pour un temps de résolution toujours quadratique. Ces résultats, en plus de constituer les premières approximations avec facteur constant pour ces problèmes, confirment donc la généralité de notre approche pour ce type de modèle multi-échelon.

Chapitre 3

Approximation algorithms for deterministic continuous-review inventory lot-sizing problems with time-varying demand

3.1 Introduction

The Economic Order Quantity (EOQ) problem deals with a single location that faces a demand of constant rate λ . In this model, costs are incurred when an order is placed as well as when units are physically held in the stock. The goal is to determine a continuous-review policy of minimal cost. More precisely, placing an order incurs a fixed order cost K and a linear order cost c while holding an inventory unit incurs a cost h per unit of time. This problem has been solved for a long time by Harris (1913) in the early twentieth century and popularized by Wilson (1934). Since then, many extensions and variations have been studied including production capacity, backorders, perishability, multi-echelon systems (see e.g. Zipkin (2000) for a state of the art).

One of the important limitations of the EOQ model is the assumption of time-independent parameters. In particular, it assumes a constant demand rate λ , which is not realistic in numerous practical situations. Many authors relax this assumption and consider a more general model with time-varying demand rate $\lambda(t)$. However they only solve the problem under very restrictive assumptions. Resh et al. (1976) consider a time proportional demand rate ($\lambda(t) = \alpha t$) with time-independent cost parameters. Later, Donaldson (1977) proposes an optimal policy for a linear demand rate ($\lambda(t) = \alpha t + \beta$) and Barbosa and Friedman (1978) generalize to power-form demand rates ($\lambda(t) = \alpha t^\beta$, $\beta > -2$). This result is then extended by Henery (1979) to increasing log-concave demand patterns and Hariga (1994) who studies a more general model for any monotonic log-concave demand rate when shortages are allowed. Finally, Henery (1990) focuses on non monotonic demand patterns

in the special case of cyclic demands.

These papers use a similar general approach that consists in finding the optimal policy for a fixed number of orders over the planning horizon, then to determine the optimal number of orders minimizing the total cost. This concept has been studied and generalized by Benkherouf and Gilding (2009), who show that the optimal cost over a finite planning horizon is a convex function of the number of orders. However, this method requires specific properties on the cost function and therefore the papers mentioned above restrict their attention to specific demand patterns and time-independent cost parameters. There is no general technique to determine an optimal continuous-review policy for time-varying parameters without the restrictive assumptions discussed above.

As the problem is difficult for more general demand patterns, the literature proposes several heuristics based on empirical methods such as the greedy (or myopic) approach. Silver (1979) introduces a heuristic where the length of a replenishment cycle is chosen such that the average cost is minimized (locally) on the replenishment cycle. This heuristic is inspired from the well known Silver and Meal (1973) heuristic designed for a periodic-review setting. Another heuristic, widely used in practice, consists of averaging the demand rate and apply the EOQ formula with the average demand to compute the order times. Goyal and Giri (2003) consider an extension of Silver (1979) with backorders and time-varying demand rate $\lambda(t)$, production rate $\mu(t)$ and deterioration rate $\theta(t)$. They also use a greedy approach, optimizing the average cost on each cycle rather than globally. Many other heuristics and extensions have been considered in the literature. We refer the reader to Goyal and Giri (2001) and Bakker et al. (2012) for a review of deteriorating-inventory models with time-varying demand and to Teng et al. (2007) for some references on the Economic Production Quantity (EPQ) model with time-varying demand.

An important heuristic discretizes the time horizon and use dynamic programming techniques to solve the corresponding periodic-review problem. Indeed, there exists several polynomial time algorithms that are optimal for the periodic-review lot-sizing problem. In the discrete time version, the planning horizon is divided into n periods. In each period a demand occurs and an order can be placed to replenish the stock. Wagner and Whitin (1958) present an algorithm of complexity $O(n^2)$, while Aggarwal and Park (1993) propose a $O(n)$ algorithm to solve this problem using Monge arrays. However, it is very unlikely that the optimal solution for the discretized problem matches an optimal policy for the original continuous-review version.

More generally, the previous heuristics have an important drawback: They can perform arbitrarily bad (i.e. no guarantee of performance has been proven). Even in the periodic-review setting, Axsäter (1982) shows that the heuristic of Silver and Meal (1973) can perform arbitrarily bad. That is, on some instances, the cost of the heuristic is arbitrary large compared to the cost of the optimal policy. Similarly, the EOQ heuristic (that averages the demand and uses the EOQ formula) is also shown to perform arbitrarily bad (see Bitran et al. (1984)), even if the average demand is recalculated whenever an order is

placed.

In this paper, we aim to derive tractable policies for the continuous lot-sizing problem with time-varying parameters that have provable performance guarantees. A policy is said to have a performance guarantee (or a worst-case guarantee) of α if its cost is at most α times the cost of an optimal policy. In the periodic-review setting with time-varying demand, Axsäter (1982) shows that a policy that balances holding and set-up costs in each replenishment cycle has a performance guarantee of two. Bitran et al. (1984) extend the results of Axsäter to the case with time-varying holding costs. Recently, Van den Heuvel and Wagelmans (2010) have proven that in fact this worst-case guarantee cannot be improved for online heuristics, i.e. procedures that use a forward induction mechanism and cannot modify their past ordering periods.

In this paper, we apply the ideas of cost balancing introduced by Bitran et al. (1984) to the continuous-review problem and prove that the performance guarantee of two remains valid for this problem. We then show that the ideas developed are quite generic and can be extended to several important models from the inventory literature (production rate, deterioration rate, non-linear holding costs, models with shortages, time-varying order costs) while preserving the performance guarantee. Similarly to Bitran, our method uses a myopic mechanism to decide whether to place a new order or increase the quantity ordered in the previous one and hence is unlikely to achieve a worst-case guarantee lower than two. However, we introduce at the end of this paper an alternative procedure that enables us to improve the performance guarantee to $3/2$, with some additional assumptions.

Note that Hariga (1996) applies a similar cost balancing technique to the time-varying demand model with shortages and compares its performances to several other heuristics widely used in practice. Although his study presents numerical experiments and compare the performances of the different techniques, it focuses on time-independent cost parameters. Moreover, the theoretical performance and complexity are not discussed in his paper. Thus, to the best of our knowledge, this paper is the first to derive algorithms and prove their guarantee of performance for lot-sizing problems with time-varying parameters in a continuous-time setting.

The rest of the paper is organized as follows: in §3.2, we formally introduce the basic lot-sizing problem and useful notations to describe the solutions and prove their guarantee. §3.3 explains how to adapt the balancing policy of Bitran et al. (1984) for a continuous-time setting and prove that the resulting policy preserves the performance guarantee of 2. In §3.4, we introduce several extensions to the lot-sizing model and show how the balancing algorithm can be modified to solve them. Finally, we generalize the concept of cost balancing to present a new policy for the lot-sizing model and prove its worst-case guarantee is 1.5 in §3.5.

3.2 Assumptions and notations

We consider a single location that faces a continuous demand with rate $\lambda(t)$ at time t . We denote by $\Lambda(s, t) = \int_s^t \lambda(u)du$ the cumulated demand over $[s, t]$. Holding an inventory unit at time t incurs a cost $h(t)$ per unit of time. We assume that $\lambda(\cdot)$ and $h(\cdot)$ are piecewise continuous functions. Placing an order at time t incurs a fixed order cost (or set-up cost) $K(t)$ and a linear order cost $c(t)$.

The inventory level at time 0, before placing the first order, is denoted by x_0 . For the majority of the models presented in this paper, it is dominant to place the first replenishment at time t_0 such that $x(t_0) = 0$ and $\lambda(t_0) > 0$: Hence we will assume w.l.o.g. that the following initial conditions (IC) holds, unless explicitly said otherwise.

$$(IC) \quad \begin{cases} x_0 = 0 \\ \lambda(0) > 0 \end{cases}$$

In what follows, a *policy* is defined as a set of rules to determine the ordering times and quantities for each instance of the problem. The objective is to find a policy minimizing costs over a finite horizon $[0, T]$ while satisfying all the demands: In particular, a policy that achieves the lowest possible cost for any instance of the problem is said to be optimal.

When the demand rate is constant, the cost parameters are time-independent and the horizon is infinite, this problem reduces to the EOQ problem and can be solved analytically (see e.g. Zipkin (2000)). On the contrary, it is generally hard to solve the problem to optimality when parameters are time-dependent. The aim of this paper is to develop approximation algorithms for the time-varying version of the problem.

Without loss of generality, we assume that the procurement leadtime is null: All demands being deterministic, a deterministic positive leadtime simply shifts the decision earlier in time. We now introduce some useful notations and concepts for the rest of the paper. Let $x^P(u)$ be the inventory level at time u , under some policy P and let $\mathcal{C}^P(s, t)$ be the cost incurred by a policy P over $(s, t]$. More precisely, $\mathcal{C}^P(s, t)$ includes holding cost $\int_s^t h(u)x^P(u)du$ and set-up costs over $(s, t]$, excluding the set-up cost at time s , if any. Then, for any sequence $a_0 = 0 < a_1 < \dots < a_{n+1} = T$, the total cost \mathcal{C}^P of policy P over the whole time horizon can be decomposed as

$$\mathcal{C}^P = \delta_0^P K(0) + \mathcal{C}^P(0, T) = \delta_0^P K(0) + \sum_{i=0}^n \mathcal{C}^P(a_i, a_{i+1}) \quad (3.1)$$

where δ_0^P is equal to 1 if P orders at time 0 and 0 otherwise. Notice that when condition (IC) is satisfied and policy P is feasible, we have necessarily $\delta_0^P = 1$. Furthermore since intervals $(a_i, a_{i+1}]$ partition the time horizon, equation (3.1) partitions the cost incurred by any policy according to the sequence $(a_i)_{i=0, \dots, n+1}$. In particular, the following proposition holds:

Proposition 3.2.1. *Let π be a feasible policy for the continuous-review problem and let $0 = a_0 < a_1 < \dots < a_n < a_{n+1} = T$ be a sequence of points in time. Then if $\delta_0^\pi \leq \delta_0^P$ and $\mathcal{C}^\pi(a_i, a_{i+1}) \leq \alpha \mathcal{C}^P(a_i, a_{i+1})$ for all $i = 0, \dots, n$ and all feasible policy P , π has a performance guarantee of α .*

Proof. Summing all the partial costs as in equation (3.1) leads to $\mathcal{C}^\pi \leq \alpha \mathcal{C}^P$ for any feasible policy P . In particular, this inequality holds when P is an optimal policy and the proof follows. \square

Given a policy P , a (replenishment) cycle $(s, t]$ is defined as a time interval such that an order is placed at time s , an order is placed at time $t > s$ and no order is placed inbetween. We say that a policy satisfies the *Zero Inventory Ordering* property, or is ZIO, if it orders only when its inventory level is zero. Note that under a ZIO policy P , the quantity ordered at the beginning of a cycle $(s, t]$ is exactly $\Lambda(s, t)$, the cumulated demand over the cycle. Moreover, the inventory level can be easily expressed in cycle $(s, t]$ as $x^P(u) = \Lambda(u, t)$. Hence the cumulated holding cost $H(s, t)$ of a ZIO policy over a cycle $(s, t]$ can be simply expressed as an integral of holding cost and demand functions.

$$H(s, t) = \int_s^t h(u) \Lambda(u, t) du$$

Note that since $h(\cdot)$ and $\Lambda(\cdot, \cdot)$ are both nonnegative piecewise continuous functions, $H(s, \cdot)$ is a nondecreasing continuous function for all s . We give below a summary of the notations used through the paper:

$\lambda(t)$	demand rate at time t
$\Lambda(s, t)$	cumulated demand over $[s, t]$
$\Lambda(t)$	cumulated demand from time 0 to t : $\Lambda[0, t]$
$h(t)$	per-unit holding cost at time t
$K(t)$	fixed order cost (or set-up cost) at time t
$c(t)$	linear order cost at time t
T	time horizon
$x^P(u)$	inventory level at time u , under policy P
$\mathcal{C}^P(s, t)$	cost of policy P over $(s, t]$ (time s excluded)
\mathcal{C}^P	cost of policy P over $[0, T]$
$H^P(s, t)$	holding cost of policy P over cycle $(s, t]$
$H(s, t)$	holding cost of a ZIO policy over cycle $(s, t]$

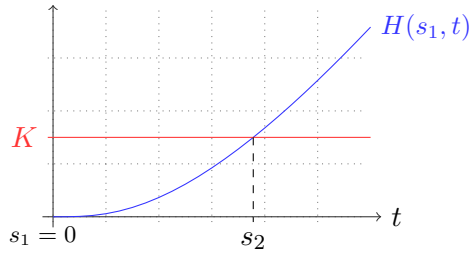


Figure 3.1: Computing the next ordering time s_2 when the current order is placed at time $s_1 = 0$ ($K = 1.5$, $h(t) = \frac{\ln(t+1)}{t+1}$, $\lambda(t) = \frac{2\ln(t+1)}{(t+1)}$).

3.3 An approximation algorithm

In this section, we present the central idea of our algorithm. Roughly, the main concept is to balance in each replenishment cycle the holding costs with the fixed order cost. For ease of understanding, we first make similar assumptions as Bitran et al. (1984) who have investigated a periodic-review version of this problem. More precisely, we assume that the fixed order cost and the linear order cost are time-independent, that is $K(t) = K$ and $c(t) = c$ for all $t \in [0, T]$. These assumptions will be relaxed in §3.4. Notice that without loss of generality, we can assume $c = 0$ as any policy incurs a linear order cost of exactly $c\Lambda(0, T)$ over the planning horizon to satisfy all the demands. Throughout the remainder of this section, we restrict our attention to policies that respect the Zero Inventory Ordering (ZIO) property, as it is dominant for the model we are interested in.

3.3.1 The balancing policy

The concept of cost balancing is well known in the inventory literature and can even be found in the classical EOQ problem, where all parameters are time-independent. Indeed, it is well known that for this special case the optimal policy balances exactly the holding costs with the order costs. More precisely, the optimal average cost \mathcal{C}^* over an infinite horizon is

$$\mathcal{C}^* = \underbrace{\sqrt{\lambda K h / 2}}_{\text{Holding costs}} + \underbrace{\sqrt{\lambda K h / 2}}_{\text{Fixed order costs}}$$

In the time-varying setting, we introduce the *balancing policy*, denoted BL in what follows, which balances in each cycle the two parts of the cost discussed above. The BL policy is inspired by the periodic-review policy proposed by Axsäter (1982) and Bitran et al. (1984). In fact, it is a ZIO policy whose order times are determined by a forward induction as follows: A first order is placed at the last moment when the inventory is nonnegative. If the BL policy places an order at time s , the next order time is then defined as the last time t such that

$$H(s, t) \leq K \tag{3.2}$$

Recall that $H(s, t)$ represents the holding cost incurred by a ZIO policy over cycle $(s, t]$.

As $H(s, \cdot)$ is a nondecreasing continuous function and $H(s, s) = 0$, inequality (3.2) is tight for at least one point in $[s, T]$ if and only if $H(s, T) \geq K$ (see Figure 3.1). When the latter condition is not satisfied then s is the last order time of the BL policy and the quantity ordered at s is $\Lambda(s, T)$, such that all the remaining demand is ordered at s . More precisely, we define the BL policy as follows:

Definition 3.3.1. The BL policy is the ZIO replenishment policy whose order times $s_1 < s_2 < \dots < s_n$ are given by Algorithm 1.

Algorithm 1 BL policy

```

set  $s_1 = 0$ 
set  $n \leftarrow 1$ 
while  $H(s_n, T) \geq K$  do
     $s_{n+1} \leftarrow \max \{t \leq T : H(s_n, t) = K\}$ 
     $n \leftarrow n + 1$ 
end while
return  $(s_1, \dots, s_n)$ 

```

Since the BL policy is ZIO, it is completely defined by the vector of order times (s_1, \dots, s_n) . By convention we set $s_{n+1} = T$. Then for $i \geq 1$, the quantity ordered by the BL policy at time s_i is exactly the cumulated demand over $[s_i, s_{i+1}]$, i.e. $\Lambda(s_i, s_{i+1})$.

Remark. Since Algorithm 1 uses a forward mechanism that does not modify past decisions, the procedure can easily be applied in a similar fashion to problems with an infinite or a rolling time horizon.

3.3.2 Performance guarantee of the BL policy

The BL policy is interesting for its simplicity and for the quality of the solution obtained. In particular, we show in this section that the cost of the BL policy is at most twice the cost of an optimal policy. We first present a lower bound on the cost incurred by any policy between two instants $s < t$:

Lemme 3.3.1. *For any instants $s < t$ and any feasible policy P , we have*

$$\mathcal{C}^P(s, t) \geq \min\{H(s, t), K\}$$

Proof. If policy P places an order on $(s, t]$, clearly $\mathcal{C}^P(s, t) \geq K$. Otherwise, the inventory level $x^P(s)$ of policy P at instant s is at least $\Lambda(s, t)$ to prevent stockout on $[s, t]$. Since no order is placed on $(s, t]$, we have that for all $u \in (s, t]$, $x^P(u) \geq \Lambda(u, t)$ for $u \in (s, t]$ and thus $H^P(s, t) = \int_s^t h(u)x^P(u)du \geq \int_s^t h(u)\Lambda(u, t)du = H(s, t)$. The results follows. \square

We now use this result to state and prove the following theorem on the performance guarantee of the BL policy:

Theorem 3.3.2. *For piece-wise continuous functions $\lambda(\cdot)$ and $h(\cdot)$ and constant functions $K(\cdot)$ and $c(\cdot)$, the BL policy has a worst-case guarantee of two. That is, the cost incurred by the BL policy is at most twice the cost of an optimal policy.*

Proof. Let P be a feasible policy for the continuous-review lot-sizing problem considered and let s_1, \dots, s_n be the sequence of order times of the BL policy. We prove that on any time interval $(s_i, s_{i+1}]$, the cost incurred by the BL policy is at most twice the cost incurred by P .

Since P is feasible, we clearly have from the initial condition (IC) $\delta_0^{\text{BL}} = \delta_0^{\text{P}} = 1$. On the last interval $(s_n, T]$, the cost incurred by BL is $H(s_n, T)$, which is lower than K by construction. According to Lemma 3.3.1, we have

$$\begin{aligned} \mathcal{C}^{\text{P}}(s_n, T) &\geq \min\{H(s_n, T), K\} \\ &= H(s_n, T) \\ &= \mathcal{C}^{\text{BL}}(s_n, T) \end{aligned}$$

Now consider an interval $(s_i, s_{i+1}]$, with $1 \leq i \leq n-1$. By construction, the BL policy incurs a holding cost $H(s_i, s_{i+1}) = K$ and thus $\mathcal{C}^{\text{BL}}(s_i, s_{i+1}) = H(s_i, s_{i+1}) + K = 2K$. Due to Lemma 3.3.1, any feasible policy pays at least $\min\{H(s_i, s_{i+1}), K\} = K$ on $(s_i, s_{i+1}]$. In particular, we have that $\mathcal{C}^{\text{P}}(s_i, s_{i+1}) \geq K$ and therefore

$$\mathcal{C}^{\text{BL}}(s_i, s_{i+1}) \leq 2\mathcal{C}^{\text{P}}(s_i, s_{i+1})$$

for all $i = 0, \dots, n-1$. The theorem then follows from Proposition 3.2.1. □

Remark. The BL policy is optimal in the case of the EOQ problem.

Assume that $\lambda(t) = \lambda$ and $h(t) = h$ for all t . In this case, we have $\Lambda(s, t) = (t-s)\lambda$ and $H(s, t) = \lambda h(t-s)^2/2$. Solving Equation (3.2) yields

$$t - s = \sqrt{\frac{2K}{\lambda h}}$$

The time between orders is precisely the optimal order interval for the EOQ model (see e.g. Zipkin (2000)), which proves the optimality of the BL policy.

3.3.3 Numerical issues and complexity

In this section, we discuss how the order times can be computed by the balancing algorithm in practice. We first turn our attention to the issue of computing the next order time t given the current order time s . From the definition of $H(s, \cdot)$, we have that $H(s, s) = 0$ and $H(s, \cdot)$ is a nondecreasing continuous function on $[s, T]$. Hence there exists at least one solution $t \in [s, T]$ to the equation $H(s, t) = K$ (unless $H(s, T) < K$). However, in most cases there is no analytical expression of t . One can then use classical

root-finding algorithms, such as the bisection method, to compute an approximate value of t . The next ordering moment t' found by such techniques is such that $t' \in [t - \theta, t + \theta]$ for a given precision $\theta > 0$ and the final complexity depends on θ . For instance when using the bisection method, each computation of t' requires $\mathcal{O}(\log(T/\theta))$ evaluations of function $H(s, \cdot)$.

For some classes of functions $h(\cdot)$ and $\lambda(\cdot)$, like polynomial, exponential, sinusoidal, the integral $H(s, t)$ can be expressed analytically. Otherwise, one can use classical numerical methods to evaluate this integral with some precision error. In what follows, we assume that $H(s, t)$ can be computed (exactly) in time $\mathcal{O}(1)$ and we only focus on the complexity of determining the order times s_1, \dots, s_n of the BL policy. Notice that the number of order times of the BL policy, as for an optimal policy, may be arbitrary large (for instance considering an exponential holding cost $h(t) = e^t$). Thus the time complexity can not be bounded in general with respect to the instance size. For this reason we consider in what follows a natural restriction of the problem.

Assume that the demand function and the holding cost function are both upper bounded on $[0, T]$. That is, there exists $\hat{\lambda}, \hat{h}$ such that for all $t \in [0, T]$, $\lambda(t) \leq \hat{\lambda}$ and $h(t) \leq \hat{h}$. Then the following theorem holds:

Theorem 3.3.3. *For any $\varepsilon > 0$, the BL policy can be implemented to provide a solution with a performance guarantee of $(2 + \varepsilon)$ using $\mathcal{O}(\Theta \log \Theta)$ evaluations of $H(\cdot, \cdot)$, where $\Theta = (1 + \frac{1}{\varepsilon}) \frac{\hat{\lambda} \hat{h}}{K} T^2$.*

Proof. See A.1. □

Remark. In practice, $H(s, t)$ is often too complex to be computed exactly in constant time. However, if one uses a numerical method to compute $H(s, t)$ with maximum error $\gamma \geq 0$ in time $\mathcal{O}(f(\gamma, T))$, this error can be somehow included in the general error ε of BL. Specifically, the maximum error on $H(\cdot, \cdot)$ must satisfy $\gamma < \varepsilon K$. The total complexity of the procedure is then $\mathcal{O}(\Theta' \log \Theta' f(\gamma, T))$, where $\Theta' = \frac{1 + \varepsilon}{\varepsilon K - \gamma} \hat{\lambda} \hat{h} T^2$.

3.3.4 Bad example

We now build an instance of the problem for which the BL policy reaches its worst-case bound of 2. We consider a finite time horizon T and stationary holding and fixed ordering costs $K = h = 1$. Given the time interval $I = \bigcup_{i=0}^{\lceil T \rceil} [0, T] \cap [i - 1/T, i + 1/T]$, the demand rate $\lambda(t)$ is defined as follows:

$$\lambda(t) = \begin{cases} 1 & \text{if } t \in I \\ 0 & \text{otherwise} \end{cases}$$

For clarity, we assume from now on that T is integer. One can easily verify that the

BL policy applied to this instance orders at times $0, \dots, T-1$ and incurs a total cost of

$$\mathcal{C}^{\text{BL}} = \sum_{i=0}^{T-1} 2K = 2T$$

Let P denote the ZIO policy that orders at times $0, i-1/T$ for $i = 1, \dots, T$. The total cost incurred by P is then

$$\begin{aligned} \mathcal{C}^{\text{P}} &= K + \frac{h}{2T} + \sum_{i=1}^{T-1} \left(K + \frac{h}{T} \right) + K + \frac{h}{2T} \\ &= T + 1 + \sum_{i=1}^{T-1} \frac{1}{T} + \frac{1}{T} = T + 2 \end{aligned}$$

and we have as $T \rightarrow +\infty$

$$\lim_{T \rightarrow +\infty} \frac{\mathcal{C}^{\text{BL}}}{\mathcal{C}^{\text{P}}} = 2$$

3.4 Extensions

The previous section presents a cost balancing technique for a basic continuous lot-sizing model, but the underlying idea appears to be rather generic. In this section, we apply this concept to several important extensions of the lot-sizing problem. Although in most cases Algorithm 1 cannot be used directly, we adapt the policy to the specific settings of each extension and prove that the performance guarantee of two found in §3.3 remains valid for these more general models.

3.4.1 Nonlinear holding costs

The analysis used in §3.3 remains valid for nonlinear holding costs. In this model, the holding cost incurred depends on both the time moment and the current inventory on hand. Specifically, we denote $h(x, t)$ the holding cost value at t for holding x units in stock. The only (natural) assumption we make is that $h(x, t)$ is nondecreasing in x .

Essentially, one only has to check that Lemma 3.3.1 remains valid in this context to ensure the approximation result holds. As before, let $H(s, t)$ be the holding cost incurred on a cycle $(s, t]$ by a ZIO policy. We have $H(s, t) \equiv \int_s^t h(\Lambda(u, t), u) du$. Now consider a feasible policy P that does not order in the time interval $(s, t]$. The holding cost $H^{\text{P}}(s, t)$

incurred by P on $(s, t]$ satisfies:

$$\begin{aligned} H^P(s, t) &\equiv \int_s^t h(x^P(u), u) du \\ &= \int_s^t h(x^P(s) - \Lambda(s, u), u) du \\ &\geq \int_s^t h(\Lambda(s, t) - \Lambda(s, u), u) du \\ &= \int_s^t h(\Lambda(u, t), u) du = H(s, t) \end{aligned}$$

where the second equality is due to the fact that P does not order in (s, t) and subsequent inequalities come from the nondecreasing property of $h(\cdot, u)$. Therefore Lemma 3.3.1 is still valid.

The policy we propose in this case follows exactly Algorithm 1. This policy satisfies the ZIO property, thus the same arguments as the ones used in Theorem 3.3.2 hold and the performance guarantee of 2 remains valid in the general case of nonlinear holding costs.

3.4.2 Perishable products

Classical inventory models generally assume that one can store units indefinitely to meet future demands. However, many products do not satisfy this assumption in practice, e.g. because they deteriorate or become obsolete (see Goyal and Giri (2001) for a recent review on these models). In this section, we consider a model in which the inventory on hand at time t deteriorates at rate $\theta(t)$, incurring a per unit deterioration cost $a(t)$. In a cycle $(s, s']$, the inventory level $x(t)$ satisfies the following first-order differential equation:

$$\frac{dx(t)}{dt} + \theta(t)x(t) = -\lambda(t), \quad x(s') = 0$$

which can be solved easily, see e.g. Goyal and Giri (2003). Note that even though we only present our results for the simple inventory model derived from §3.3, this equation can also be solved with production rate $\mu(t)$ and backorders (see Goyal and Giri (2003)).

The sum of holding and deterioration costs over a time interval $[s, s']$ for a policy P can be expressed as:

$$H^P(s, s') = \int_s^{s'} [h(t) + a(t)\theta(t)]x^P(t)dt = \int_s^{s'} \tilde{h}(t)x^P(t)dt$$

where $\tilde{h}(t) = h(t) + a(t)\theta(t)$. Using this modified holding cost parameter, we can apply Algorithm 1 as in §3.3. The arguments used to prove the performance guarantee of BL remain valid in this more general context and thus the resulting algorithm is a 2-approximation for the model with perishable products.

3.4.3 Finite production rate

The model studied in §3.3 assumes that replenishments are instantaneous. We now relax this assumption and focus on a model where units are produced according to a time-dependent, piecewise continuous production rate $\mu(\cdot)$. In addition, let

$$\mu(s, s') = \begin{cases} \int_s^{s'} \mu(z) dz & \text{if } s \leq s' \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

denote the cumulative production capacity on between instants s and s' . If a production starts at time s for q units, all units are available on hand at time $s+t$ such that $\mu(s, t) = q$. Each production start incurs a time-independent fixed cost K (the nonspeculative linear production cost is omitted w.l.o.g. as before). The objective is again to minimize total costs over $[0, T]$ without stockout.

Although classical production models assume that the production rate $\mu(t)$ is larger than the demand rate $\lambda(t)$, for all t , we relax this constraint and consider a more general situation where the only assumption is that the production capacity is sufficient to satisfy the entire demand without stockout, that is $\mu(0, t) \geq \Lambda(0, t)$ for all t .

Observe that a policy P cannot produce more than $\mu(s, t)$ units during a time interval $[s, t]$. Thus a necessary condition for P to avoid stockouts is that for any time $t \geq s$, $x^P(s) \geq \Lambda(s, t) - \mu(s, t)$.

Property 3.4.1. *Let $x_{min}(s) = \max_{t \geq s} \{\Lambda(s, t) - \mu(s, t)\}$.*

Any feasible policy P satisfies

$$\forall s \in [0, T], x^P(s) \geq x_{min}(s)$$

When $t = s$, this condition simply implies that $x^P(s) \geq 0$. According to this new definition, we modify the initial condition (IC) and assume w.l.o.g. in this model that $x_0 = x_{min}(0)$ and $\lambda(0) > 0$.

Note that in the special case where $\mu(t) \geq \lambda(t)$ for all t , we have $x_{min}(s) = 0$ for all s and the ZIO property remains dominant. However in this more general production model, it is not necessarily the case. Consequently we focus on policies that order at time s only if their inventory level is equal to $x_{min}(s)$. In the remainder of this section, we call such policies MIO (Minimum Inventory Ordering) policies. In addition to Property 3.4.1, the final stock of a MIO policy must be null ($x(T) = 0$). MIO policies are clearly dominant for our problem since for any non MIO policy, one can easily build a MIO policy with lower cost by delaying orders. Figure 3.2 illustrates a MIO policy.

For a MIO policy P , one can easily compute the stock evolution during a replenishment cycle $(s, t]$. First, note that by definition we have $x^P(s) = x_{min}(s)$ and $x^P(t) = x_{min}(t)$. Due to the continuous production process, cycle $(s, t]$ consists now of an active period $(s, u]$ (when production is on-going) and an idle period $(u, t]$ (when production is over). The idle

Algorithm 2 Balancing policy for production models (PB)

```

set  $s_1 \leftarrow 0$ 
set  $n \leftarrow 1$ 
while  $H_{\text{PB}}(s_n, T) \geq K$  do
     $s_{n+1} \leftarrow \max \{t \leq T : H_{\text{PB}}(s_n, t) = K\}$ 
     $n \leftarrow n + 1$ 
end while
return  $(s_1, \dots, s_n)$ 

```

For $i = 1, \dots, n$, the quantity ordered by policy PB at time s_i is then $q_i = \Lambda(s_i, s_{i+1}) + x_{\min}(s_{i+1}) - x_{\min}(s_i)$ and the following theorem holds:

Theorem 3.4.2. *The PB policy has a worst-case guarantee of 2 for the general production model.*

Proof. See A.2. □

3.4.4 A model with backlogging

One major constraint in the previous models is that demand has to be satisfied on time. However, there exists more flexible models in which the demand does not necessarily have to be fulfilled on time. We now focus on a model in which backlogging is allowed and present an approximation algorithm based on the same balancing idea as the one discussed in §3.3.

We still assume that a policy incurs a fixed order cost K for each order placed while holding inventory induces a per-unit holding cost of $h(t)$ at time t . On the other hand, any unmet demand is *backlogged*: Customers are willing to wait until the stock on hand is sufficient to fulfill their requirement. Thus backlogged demand is served with a subsequent order placed later in time and incurs a per-unit backlogging cost of $b(t)$ at time t .

Without loss of generality, we assume the units are consumed on a first-ordered first consumed basis. Moreover, we modify the initial condition (IC) and simply consider that $x_0 \leq 0$. The balancing policy for models with backlogging uses an idea rather similar to the one used in the basic model: It balances the costs incurred between two consecutive order times with the fixed order cost K . The main difference is that instead of using only the holding costs, both holding and backlogging costs are considered. However, the policy decides to serve or backlog demand in a way that minimizes the sum of the costs incurred over the cycle. Each cycle $(s, t]$ is consequently split into two parts: $(s, u]$, during which inventory is held, and $(u, t]$, when backorders accumulate.

In the remainder of the section, the balancing policy for models with backlogging is denoted BB. For a given policy P , let $B^P(s, t)$ be the backlogging cost incurred by P over $(s, t]$. In addition, we denote $B(s, t)$ the backlogging cost incurred over $(s, t]$ by a policy

that does not order in (s, t) and whose inventory at time s is 0. Formally, the algorithm works as follows:

Algorithm 3 Balancing policy for models with backlogging (BB)

```

set  $s_0 \leftarrow 0$ ,  $s_1 \leftarrow \max \{t \leq T : B(0, t) = K\}$ 
set  $n \leftarrow 1$ 
while  $H(s_n, T) > K$  do
     $s_{n+1} \leftarrow \max \left\{ t \leq T : \min_{u \in [s_n, t]} \{H(s_n, u) + B(u, t) = K\} \right\}$ 
     $n \leftarrow n + 1$ 
end while
return  $(s_1, \dots, s_n)$ 

```

Given $s < t$, the algorithm computes the quantity that minimizes the holding and backlogging costs incurred over $(s, t]$. This quantity has to satisfy the backorder accumulated during the previous cycle until time s , plus the cumulative demand faced during the first part of the cycle (when inventory is physically held).

Since satisfying the backlogged units at s does not induce any cost, the algorithm aims to find the instant $u^* \in (s, t)$ that minimizes the function $f_{s,t}(u) = \int_s^u h(z)\Lambda(z, u)dz + \int_u^t b(z)\Lambda(u, z)dz$. If $s < t$ are two consecutive ordering times of the BB policy, the quantity ordered at time s increases the inventory level up to $\Lambda(s, u^*)$. Note that consequently the remaining $\Lambda(u^*, t)$ units are backlogged and served at time t . Finally if the last order computed by the algorithm is $s_n < T$, the BB policy orders up to $\Lambda(s_n, T)$ at time s_n . Otherwise ($s_n = T$), it orders a quantity at time T that satisfies all the demand backlogged in $(s_{n-1}, s_n]$. We now state and prove the following theorem on the performance guarantee of the balancing algorithm for models with backlogging:

Theorem 3.4.3. *The BB policy for continuous lot-sizing models with backlogging has a worst-case guarantee of two.*

Proof. See A.3. □

To conclude this section, we discuss the computation of u^* for a given ordering cycle $(s, t]$. Consider the first derivative of $f_{s,t}(\cdot)$:

$$\begin{aligned} \frac{df_{s,t}(u)}{du} &= \int_s^u h(z)\lambda(u)dz - \int_u^t b(z)\lambda(u)dz \\ &= \left(\int_s^u h(z)dz - \int_u^t b(z)dz \right) \lambda(u) \end{aligned}$$

Since $\lambda(\cdot)$ is a nonnegative function, it is easy to see that for $s < t$, there exists $u^* \in (s, t)$

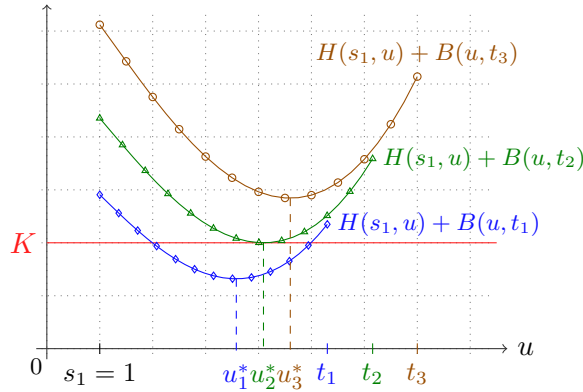


Figure 3.3: Computing u^* for different values of t ($t_1 = 5.3$, $t_2 = 6.156$, $t_3 = 7$) when the previous order is placed at time $s_1 = 1$ ($K(t) = 2$, $h(t) = 1$, $p(t) = 1.5$, $\lambda(t) = \frac{e^{\sqrt{t+1}}}{4(t+1)} \left(1 - \frac{1}{\sqrt{t+1}}\right)$). The next order chosen by the BB policy is $s_2 = t_2$ since the minimum cost incurred over $(s_1, t_2]$ is exactly equal to K . Furthermore the quantity ordered at s_1 increases the inventory level up to $\Lambda(s_1, u_2^*)$.

such that:

$$\begin{cases} \frac{df_{s,t}(u)}{du} \leq 0 & \text{for all } u \in (s, u^*] \\ \frac{df_{s,t}(u)}{du} \geq 0 & \text{for all } u \in (u^*, t] \end{cases}$$

Therefore $f_{s,t}(\cdot)$ is unimodal and reaches its minimum at $u^* \in (s, t)$ such that:

$$\int_s^{u^*} h(z)dz = \int_{u^*}^t b(z)dz$$

This property of functions $f_{s,t}(\cdot)$ is illustrated on Figure 3.3.

In practice, one can use classical search algorithms such as the Fibonacci search technique (see Kiefer (1953)) to compute the minimum $u^* \in (s, t)$ in $\mathcal{O}(\log(T/\theta))$, where θ is the chosen precision. Thus assuming that $\lambda(\cdot)$ (resp. $h(\cdot)$, $p(\cdot)$) is bounded by a constant $\hat{\lambda}$ (resp. \hat{h} , \hat{p}), one can choose Θ as in §3.3.3 and compute the order times in $\mathcal{O}(\Theta(\log \Theta)^2)$ (using the precision $\theta = T/\Theta$) while bounding the error in the cost by a chosen gap ε .

3.4.5 Time-dependent order costs

In §3.3 we restricted our attention to time-independent fixed order costs. We now relax this assumption and allow for monotonic time-varying fixed order costs and nonspeculative linear order costs. Specifically, we assume that function $K(t)$ is continuous in time and either nonincreasing or nondecreasing with t , while $c(\cdot)$ satisfies for all $ds > 0$:

$$c(s + ds) \leq c(s) + \int_s^{s+ds} h(u)du \tag{3.6}$$

In other words, given two potential order times $s_1 < s_2$ to serve a specific demand at $t \geq s_2$, it is always cheaper to order at s_2 , rather than order earlier at s_1 and hold the corresponding units for a longer period. One can notice that inequality (3.6) resemble the well-known assumption of nonspeculative motivations often encountered in discrete lot-sizing models.

Intuitively, the replenishment rule used by the policy consists in balancing in each cycle the holding costs and the linear order costs incurred with the minimum fixed order cost over this cycle. Due to our assumption, this minimum fixed order cost is realized at one of the extremities of the cycle. That is, if $K(\cdot)$ is nonincreasing (resp. nondecreasing) the holding cost $H(s, t) + c(s)\Lambda(s, t)$ incurred over a cycle $(s, t]$ is balanced with $K(s)$ (resp. $K(t)$). The procedure is then applied in a forward or in a backward manner depending on the variation of $K(\cdot)$. Notice that Bitran et al. (1984) also define a forward and a backward method for the discrete time problem when ordering costs are time-independent. In our case, these two algorithms are defined as follows:

Algorithm 4 Forward balancing policy for nonincreasing order costs

```

set  $s_1 \leftarrow 0$ 
set  $n \leftarrow 1$ 
while  $c(s_n)\Lambda(s_n, T) + H(s_n, T) \geq K(T)$  do
     $s_{n+1} \leftarrow \max \{t \leq T : c(s_n)\Lambda(s_n, t) + H(s_n, t) = K(t)\}$ 
     $n \leftarrow n + 1$ 
end while
return  $(s_1, \dots, s_n)$ 

```

Remark. When $K(\cdot)$ is nondecreasing and $x_0 > 0$, placing an order before the initial stock is depleted may reduce the total cost incurred by the policy. Hence in this case, we relax the initial condition discussed in §3.2 and only assume $x_0 \geq 0$.

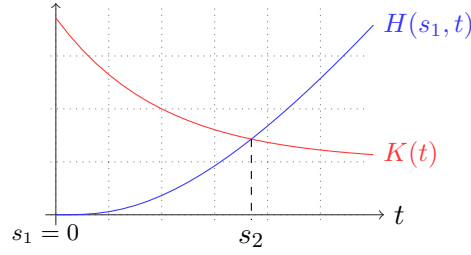


Figure 3.4: $K(t) = e^{1-\frac{t}{2}}$, $h(t) = \frac{\ln(t+1)}{t+1}$,
 $\lambda(t) = \frac{2\ln(t+1)}{(t+1)}$.

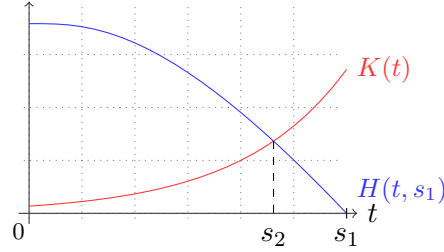


Figure 3.5: $K(t) = e^{1-\frac{6-t}{2}}$, $h(t) = \frac{\ln(t+1)}{t+1}$,
 $\lambda(t) = \frac{2\ln(t+1)}{(t+1)}$.

Algorithm 5 Backward balancing policy for nondecreasing order costs

```

set  $s_0 = T$ 
set  $t_0 \leftarrow \max \{t \leq T : \Lambda(0, t) \leq x_0\}$ 
set  $n \leftarrow 0$ 
while  $c(t_0)\Lambda(t_0, s_n) + H(t_0, s_n) \geq K(t_0)$  do
     $s_{n+1} \leftarrow \min \{t \geq t_0 : c(t)\Lambda(t, s_n) + H(t, s_n) = K(t)\}$ 
     $n \leftarrow n + 1$ 
end while
if  $s_n > t_0$  then
     $s_{n+1} \leftarrow \min \{0 \leq t \leq t_0 : c(t)\Lambda(t_0, s_n)$ 
         $+ H(t, s_n) - H(t, t_0) \leq K(t)\}$ 
     $n \leftarrow n + 1$ 
end if
return  $(s_n, \dots, s_1)$ 

```

Figures 3.4 and 3.5 show how the forward and backward algorithms compute the next and the previous order, respectively.

Note that in the case of the backward algorithm, the order times are computed in a reverse order: $0 \leq s_n \leq \dots \leq s_1 \leq T$. Moreover, the ZIO rule is not necessarily respected for the first order. Let x_0 be the initial inventory on hand and let $t_0 = \max \{t \leq T : \Lambda(0, t) \leq x_0\}$: If $s_n < t_0$, the quantity ordered at time s_n is $q_n = \Lambda(t_0, s_{n-1})$.

All the other orders are of size $q_i = \Lambda(s_i, s_{i-1})$.

Theorem 3.4.4. *For time-dependent order costs, the forward (resp. backward) balancing policy has a worst-case guarantee of two when $K(\cdot)$ is a continuous nonincreasing (resp. nondecreasing) function.*

Proof. See A.4 □

3.5 Improvement of the performance guarantee

In this section, we present a generalization of the balancing technique used in §3.3 and use it to improve the performance guarantee from 2 to $3/2$. The idea remains to balance the costs incurred over a time interval $(s, s']$ with order costs but we now consider the possibility to place additional orders in the interval. We present this modified algorithm in the simple situation of Section 3.3, in which order costs are time-independent and holding costs are linear.

3.5.1 A generalized balancing policy

Consider a time interval $(s, t]$ and let k be a positive integer. We define $G_k(s, t)$ as the minimum cost over (s, t) incurred by a feasible policy ordering at most k times in (s, t) .

Note that a policy that minimizes the cost $G_k(s, t)$ is ZIO. Thus for $k = 0$ we have $G_0(s, t) = H(s, t)$ and clearly the BL policy incurs on each of its ordering interval $(s, s']$ a cost $G_0(s, s')$. We can generalize Lemma 3.3.1 to the following result:

Lemma 3.5.1. *Let $k \in \mathbb{N}$, $s < t$ and P a feasible policy. We have*

$$\mathcal{C}^P(s, t) \geq \min\{G_k(s, t), (k + 1)K\}$$

Proof. The proof is immediate due to the definition of G_k . If policy P places at most k orders inside the time interval, then $\mathcal{C}^P(s, t) \geq G_k(s, t)$. Otherwise $\mathcal{C}^P(s, t) \geq (k + 1)K$. □

For a given positive integer k , we propose the following balancing policy, denoted BL_k in the remainder of this section. Its principle is to balance $G_k(s, t)$ with $(k + 1)K$ in each of its replenishment cycle $(s, t]$. That is, given an instant s , the policy computes the largest instant $t \leq T$ such that $G_k(s, t) \leq (k + 1)K$. Notice that in this case policy BL_k orders at instants s and t and possibly up to k times in the interval (s, t) . We call *main* order times the instants s and t , while the other order times in (s, t) are called *secondary* order times. Although main and secondary orders play the same role in the final solution, we differentiate these two types of orders to simplify the definition and the proof of performance of the algorithm. The following result generalizes Theorem 3.3.2:

Theorem 3.5.2. *For $K(\cdot)$ and $c(\cdot)$ constant functions and $h(\cdot)$ and $\lambda(\cdot)$ a piece-wise continuous function, the BL_k policy has a worst-case guarantee of $\frac{k+2}{k+1}$.*

Proof. The proof is very similar to the one of Theorem 3.3.2 and only the main arguments are given here. Let P be a feasible policy and let s_1, \dots, s_n be the sequence of the main order times of the BL_k policy. First notice that any feasible policy P places an order at time 0 and therefore $\delta_0^{BL_k} = \delta_0^P = 1$. Now consider an interval $(s_i, s_{i+1}]$, with $0 \leq i \leq n-1$. By construction, the BL_k policy incurs a cost $\mathcal{C}^{BL_k}(s_i, s_{i+1}) \leq G_k(s_i, s_{i+1}) + K = (k+2)K$ in this time interval. Due to Lemma 3.5.1, policy P pays at least $\min\{G_k(s_i, s_{i+1}), (k+1)K\} = (k+1)K$. It results that $\mathcal{C}^{BL_k}(s_i, s_{i+1}) \leq \frac{k+2}{k+1}\mathcal{C}^P(s_i, s_{i+1})$. Then consider the last interval. On $[s_n, T]$, we have $\mathcal{C}^{BL_k}(s_n, T) = G_k(s_n, T)$, which is lower than $\mathcal{C}^P(s_n, T)$ due to Lemma 3.5.1. As a consequence the inequality $\mathcal{C}^{BL_k}(s_i, s_{i+1}) \leq \frac{k+2}{k+1}\mathcal{C}^P(s_i, s_{i+1})$ holds for every interval $(s_i, s_{i+1}]$ and the proof follows. \square

3.5.2 Complexity analysis

Contrarily to the balancing policy presented in §3.3, the family of algorithms presented in the previous section does not fall within the class of online heuristics defined by Van den Heuvel and Wagelmans (2010) and enables us to design efficient procedures to solve the problem introduced in §3.2. In particular, the resulting performance guarantee can be as close as aimed to the cost of an optimal policy. However the problem of determining (analytically or numerically) the minimal cost $G_k(s, t)$ on an interval (s, t) is likely to require a large computing effort, especially for large values of k . In this section, we first discuss the computation of $G_k(s, t)$ for an arbitrary k . Then, for $k = 1$, we show that the BL_1 policy can be computed in $\mathcal{O}(\Theta(\log \Theta)^2)$ with some restrictions on the demand and holding cost functions.

Arbitrary k

When k secondary orders are placed in interval (s, t) at times v_1, \dots, v_k , the inventory holding cost of a ZIO policy on interval (s, t) is

$$f(v_1, \dots, v_k) = \sum_{i=0}^k \int_{v_i}^{v_{i+1}} h(x)\Lambda(x, v_{i+1})dx$$

where $v_0 = s$ and $v_{k+1} = t$.

The problem is then to minimize f with the constraint that $v_0 = s \leq v_1 \leq \dots \leq v_{k+1} = t$. We have for $1 \leq i \leq k$

$$\frac{\partial f}{\partial v_i}(v_1, \dots, v_k) = \lambda(v_i)h(v_{i-1}, v_i) - h(v_i)\Lambda(v_i, v_{i+1})$$

where $h(x, y) = \int_x^y h(t)dt$.

Hence the optimal secondary order points satisfy the following equality for $1 \leq i \leq k$:

$$\lambda(v_i)h(v_{i-1}, v_i) = h(v_i)\Lambda(v_i, v_{i+1}) \quad (3.7)$$

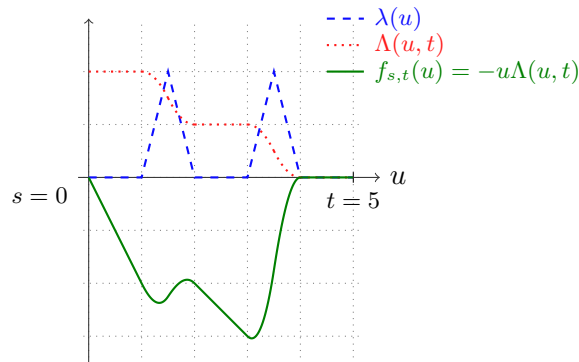


Figure 3.6: A bad example for function $f_{s,t}(\cdot)$ with triangular demand, $s = 0$, $t = 5$ and $h = 1$.

Note that this condition is necessary but not sufficient since there may exist several solutions to Equation (3.7). For time-independent holding cost $h(t) = h$, condition (3.7) reduces to a simpler condition that was established by Barbosa and Friedman (1978): $(v_i - v_{i-1})\lambda(v_i) = \Lambda(0, v_{i+1}) - \Lambda(0, v_i)$. However even in this restrictive setting, the problem of solving equation (3.7) is difficult, except for very specific demand functions (linear, power-form).

$k = 1$

We now turn our attention to the case $k = 1$, where the problem is to find a single optimal intermediary order. In this case, there is a single variable v_1 that we rename v for ease of understanding. If we assume that the demand function $\lambda(\cdot)$ and the holding cost function $h(\cdot)$ are once derivative, we can compute the second derivative of $f(\cdot)$:

$$\frac{d^2 f}{dv^2}(v) = \lambda'(v)h(s, v) + 2\lambda(v)h'(v) - h'(v)\Lambda(v, t)$$

As a consequence, $f(\cdot)$ is convex if and only if the following condition is satisfied:

$$\lambda'(v)h(s, v) + 2\lambda(v)h'(v) - h'(v)\Lambda(v, t) \geq 0 \quad (3.8)$$

Unfortunately, there exists numerous demand functions that do not satisfy inequality (3.8), even when the holding cost function is constant (see for example Figure 3.6). However, we now exhibit two cases satisfying inequality (3.8). Both cases relax some assumptions made in the existing literature.

Case 1: $h(\cdot)$ and $\lambda(\cdot)$ once derivative, $h(\cdot)$ non-increasing and $\lambda(\cdot)$ non-decreasing.

In this case $-h'(\cdot)$ and $\lambda'(\cdot)$ are positive and the inequality (3.8) is trivially satisfied. These assumptions relax the framework used in Resh et al. (1976), Donaldson (1977) and Henery (1979) to any increasing demand function.

Case 2: $h(\cdot)$ once derivative, non-increasing and $\lambda(t) = \alpha t^\beta$, with $\alpha \geq 0$ and $\beta \geq -2$.

In this case $-h'(\cdot)$ is nonnegative and $h(s, v) \geq (v - s)h(v)$. The second derivative of $f(\cdot)$ then becomes

$$\begin{aligned} \frac{d^2 f(v)}{dv^2} &= \alpha\beta v^{\beta-1}h(s, v) + 2\alpha v^\beta h(v) - h'(v)\Lambda(v, t) \\ &\geq \alpha h(v) \left(\beta(v - s)v^{\beta-1} + 2v^\beta \right) \\ &\geq \alpha h(v) \left((\beta + 2)v^\beta - \beta s v^{\beta-1} \right) \end{aligned}$$

Hence for $\beta \geq 0$ we have

$$\frac{d^2 f(v)}{dv^2} \geq 2\alpha h(v)v^\beta$$

and for $-2 \leq \beta < 0$, we have

$$\frac{d^2 f(v)}{dv^2} \geq \alpha h(v)(\beta + 2)v^\beta$$

In both cases, $\frac{d^2 f(v)}{dv^2} \geq 0$ and $f(\cdot)$ is convex.

Note that this setting extends the problem studied in Barbosa and Friedman (1978) to a time-varying holding cost function.

As a consequence, BL_1 provides solutions that are both computationally tractable and provably close to the optimal, while relaxing assumptions made in the existing literature. The following theorem summarizes the performance guarantee and the complexity of the technique in practice.

Theorem 3.5.3. *Assume that $\lambda(\cdot)$ and $h(\cdot)$ are bounded, once derivative and satisfy inequality (3.8) for all $s < v < t$. Then for any $\varepsilon > 0$, the BL_1 policy can be implemented to provide a solution with a performance guarantee of $(3/2 + \varepsilon)$ using $\mathcal{O}(\Theta(\log \Theta)^2)$, where Θ is defined as in 3.3.3.*

Proof. The proof is similar to the one of Theorem 3.3.3, except that in this case for every t candidate for the next order time, the algorithm has to compute the optimal intermediary order v . For $s < t$, one can use the convexity of $f(\cdot)$ and apply the bisection method to compute the optimal secondary order v^* in $\mathcal{O}(T/\theta)$ for a chosen precision θ . Hence for an arbitrary $\varepsilon > 0$, one can define Θ as in §3.3.3 and choose a precision $\theta = T/\Theta$. The final complexity of BL_1 for a maximum error of ε is then $\mathcal{O}(\Theta(\log \Theta)^2)$. \square

3.6 Conclusion

The EOQ formula for time-independent lot-size models has been extensively studied and is now considered as a classical result in the literature. On the contrary, the optimal

policy for its time-varying extension is still an open question and the existing research is limited to restricted cases where demand has a specific pattern (i.e. linear, power form, monotonic log-concave, etc.).

In this paper, we introduce a cost balancing technique and apply it to continuous-review inventory models when demands and cost parameters are time-varying. We prove that the resulting algorithm has a worst-case guarantee of two for the lot-size problem with time-varying parameters. In addition, we show that the underlying idea is rather generic and adapt it to many important extensions, such as perishable products, production systems or models with backlog (among others). For specific demand patterns, we also improve upon the performance guarantee of two and propose a 1.5 approximation algorithm for increasing or power form demand. Finally, we show that the complexity of our algorithm depends mainly on the chosen precision for the approximation ratio. The resulting running time makes it an interesting practical tool for decision makers.

One natural direction for further research is to experiment the cost balancing technique on numerical examples and compare its performances with existing optimal policies or heuristics. Furthermore, we believe that the concept is very generic and can also be applied to multi-echelon systems under mild restrictions. In particular, a first step in this direction would be to study how the algorithm can be extended to series systems.

Chapitre 4

A simple and fast 2-approximation for the one-warehouse multi-retailer problem

4.1 Introduction

We study an important two-level distribution system in which a central warehouse supplies N retailers facing the final customers demands. More precisely, we want to optimize over a discrete and finite planning horizon the distribution of a single item in this logistic network, known in the inventory literature as the *One-Warehouse Multi-Retailer* (OWMR) problem. We assume each retailer has a perfect knowledge of its customers demands and thus we focus on a deterministic version of this model. Two types of costs are incurred while the goods move through the system. Namely, a location pays a procurement cost whenever it replenishes its inventory and incurs holding costs when it stores physical units in its stock. The objective of the OWMR problem is to find a planning for the orders at each location (i.e. period and quantity) such that all the demands are fulfilled the demands on time while minimizing the sum of the procurement and holding costs incurred by the system. The *Joint Replenishment Problem* (JRP) is a special case of the OWMR problem where the warehouse operates as a cross-docking station (i.e. holds no inventory). This can be captured in the model via a prohibitive holding cost at the warehouse.

The JRP and the OWMR problem are known to be NP-hard, see Arkin et al. (1989). This paper aims at designing efficient approximation algorithms for these problems. Recall that a polynomial time algorithm for a minimization problem is said to be a α -approximation, or to have a performance guarantee of α , if for any instance it delivers a solution of cost at most α times the optimal cost.

Literature Review. Both OWMR and JRP problems have been studied extensively in the literature for continuous-review models. Schwarz (1973) is among the first to study the continuous-time version of the OWMR problem under constant-rate demands. While the

complexity status of the problem under this setting is still open, Roundy (1985) introduced in its seminal work a 98%-effective algorithm which was later revisited and extended by Muckstadt and Roundy (1993) to other multi-echelon problems. More recently, Stauffer (2012) presented a simple 1.27-approximation based on the recombination of the optimal single-echelon strategies, which is also one of the basic ingredients of the approach presented in this paper. In the special case of the JRP, Nonner and Sviridenko (2013) proposed a EPTAS in a stationary continuous-time setting. We refer the reader to Aksoy and Erenguc (1988) and Khouja and Goyal (2008) for a detailed survey of the JRP.

In this article we focus on the discrete-time version of the problem, i.e. we consider periodic-review policies. Only few papers propose algorithms or heuristics to solve the OWMR problem in its discrete-time version. Federgruen and Tzur (1999) have applied a time-partitionning heuristic, but considered bounded demands and parameters for the analysis of their worst-case bound. Chan et al. (2000) and later Shen et al. (2009) studied the class of Zero-Inventory-Ordering (ZIO) policies for the OWMR problem, i.e. in which the locations place orders only when their current inventory level is zero. More recently, Levi et al. (2006) proposed a 2-approximation algorithm for the JRP and Levi et al. (2008c) improved this result to a 1.8-approximation algorithm for the OWMR problem. Their results hold for shelf age dependent holding costs that are more general than linear holding costs (see Federgruen and Wang (2013) for a discussion on shelf age and level-dependent holding costs). Furthermore in the special case of JRP with deadlines, Nonner and Souza (2009) have improved this result to a $5/3$ -approximation using a similar LP-rounding technique.

Contributions of this work. In contrast with the approximation algorithms presented in Levi et al. (2008c) and Nonner and Souza (2009) which exploit a linear integer formulation of the problem via LP-rounding techniques, we develop a combinatorial algorithm, based on a natural decomposition of the problem into single-echelon problems. This algorithm can be implemented to run in polynomial time and yields a performance guarantee of two for both problems. In addition, we introduce a new lower bound for this problem as a byproduct of our analysis. The complexity of our algorithm can even be made linear (i.e. in $O(NT)$) for linear holding costs. While we do not match the best approximation guarantee of Levi et al. (2008c), we believe that the simplicity of the method and its computational complexity make it both a valuable theoretical approach and a practical tool for managers. Besides, we demonstrate that our method can be applied to a significantly broader class of holding and procurement cost structures than the ones introduced in these papers. To the best of our knowledge, we obtain the first approximation algorithms for the OWMR problem either with non-linear holding costs or with FTL/LTL procurement costs.

The rest of the paper is organized as follows. In §4.2, we present the model and important notations used throughout the rest of the paper. In §4.3, we introduce a natural decomposition of the OWMR problem into single-echelon subproblems on a simple model

with a basic cost structure. We then propose a new method, called *uncrossing algorithm*, that recombines the solutions to these subproblems into a feasible solution for the original problem and prove that the overcost incurred by this solution is bounded by a constant factor.

The remaining sections emphasize how the method can easily be adapted to numerous cost structures that capture more intricate cases. §4.4 shows how to modify the decomposition in order to tackle more complex holding cost structures, namely level-dependent and shelf age holding costs. Specifically, we develop in the former section a 2-approximation when holding costs are nonlinear functions of the inventory on hand, while the latter generalizes the so-called Monge holding cost structure introduced in Levi et al. (2008c). We also show in §4.5 that the algorithm can be adapted to general procurement costs that include the well-known FTL (*Full Truck Load*) and LTL (*Less than Truck Load*) transportation cost structures. Note that the FTL procurement cost structure is sometimes referred to as ordering with *soft capacities*. A preliminary version of this paper was presented in SODA 2011, see Stauffer et al. (2011).

4.2 Model and notations

We consider a two-level distribution network composed of one central warehouse (indexed by 0) and N final retailers (indexed from 1 to N) facing the customers demand. The planning horizon is finite, discretized into T time periods. Goods enter the system from an outside supplier of infinite capacity that replenishes the inventory of the warehouse, which in turn supplies the retailers. Assuming deterministic lead-time, we consider without loss of generality that the orders are delivered instantaneously from one location to another. Each retailer $i = 1, \dots, N$ faces in period $t = 1, \dots, T$ a deterministic demand d_t^i that must be satisfied on time, i.e. neither backlogging nor lost-sales are allowed. Hence any unit that satisfies a demand (i, t) is ordered in a period s at retailer i and a period r at the warehouse such that $r \leq s \leq t$. In the remainder of this paper, we denote $[r, s]$ such a pair of orders, where the demand served with this pair will be clear from the context.

We describe below the different costs incurred while products move across the network. The precise definitions and assumptions relative to the different cost structures will be given in the corresponding section of the paper.

Holding costs. Each location can store goods in its inventory to serve demands – or, in the case of the warehouse, retailers orders – in future periods. Most models in the literature consider that a per-unit holding cost h_t^i is incurred to keep one unit of inventory in location i from period t to $t + 1$. In what follows, this simple cost structure is referred to as the *linear* holding costs. In particular, we shall use this setting to explain the principles of our algorithms in §4.3. We also study two more general holding cost structures: Inventory *level-dependent* holding costs in §4.4.1 and *shelf age* dependent holding costs in §4.4.2. In the level-dependent case, the

holding cost incurred in location i from period t to $t + 1$ is a nondecreasing function $h_t^i(\cdot)$ of its inventory level x . Linear holding costs are clearly a special case of level-dependent holding costs for which $h_t^i(x) = h_t^i \cdot x$. In the shelf age case, we denote h_{rs}^{it} the total holding cost incurred by a unit that serves demand d_t^i through the pair of orders $[r, s]$. Such a cost structure has been studied by Levi et al. (2008c). Linear holding costs are again a special case of shelf age dependent holding costs, where $h_{rs}^{it} = \sum_{u=r}^{s-1} h_u^0 + \sum_{u=s}^{t-1} h_u^i$.

In the remainder of the paper, we assume that we can partition the set of retailers into two subsets I_J (J -retailers) and I_W (W -retailers) based on their holding cost. Specifically, it is cheaper to hold inventory in a J -retailer rather than in the warehouse, while storing units at W -retailers is more expensive. This assumption is classical in the literature: In particular in the case of linear holding costs we have for all t , $h_t^i \leq h_t^0$ if $i \in I_J$ and $h_t^i \geq h_t^0$ if $i \in I_W$. It is well-known that there exists an optimal solution that does not hold any stock at the warehouse for the J -retailers for the linear and shelf age holding cost structure. This dominance also holds under the level-dependent holding costs we consider in §4.4.1, see Appendix B.1 for a formal proof of this result.

Procurement costs. In general, we denote $p_t^i(q)$ the procurement cost incurred by location i to place an order of size q in period t . Many models in the literature focus on the special case of fixed ordering costs, in which $p_t^i(q) = K_t^i$ for all q , i.e. the procurement cost does not depend on the size q of the order. We shall use this simpler setting to present our algorithm in §4.3. In §4.5, we consider a more general ordering cost structure in which procurement costs $p_t^i(\cdot)$ model the well-known *Full Truck Load* (FTL) and *Less than Truck Load* (LTL) settings.

Note that Chan et al. (2000) have shown that if the procurement costs at the retailers vary over time, the OWMR problem is as hard to approximate as the set cover problem, even in the simple case of fixed ordering costs. Thus it is unlikely that there exists an approximation algorithm with constant guarantee unless $\mathcal{P} = \mathcal{NP}$, see Feige (1998). We hence assume in this paper that the procurement costs at each retailer $i > 0$ are stationary, i.e. $p_t^i(\cdot) = p^i(\cdot)$ for all periods $t = 1, \dots, T$.

The objective of the OWMR problem is to find a planning for the orders at each location (i.e. period and quantity) that minimizes the sum of the procurement costs and holding costs in the system to fulfill the demands at the retailers. With respect to the vocabulary of inventory control, a solution to the problem is called a *policy* and is represented by a $N + 1$ -uplet $\pi = (\pi_0, \pi_1, \dots, \pi_N)$ where each π_i is the set of pairs (order period, quantity) for location i . We say that a policy π is *feasible* for the OWMR problem if any unit of demand d_t^i is ordered via a valid pair of orders $[r, s]$. That is, π_0 orders the unit in period r and π_i in period s , with $r \leq s \leq t$. A policy is said to be FCFS (*First Come First Served*) if units of product are consumed to satisfy demands at each location in the same order

they are supplied. In addition, we say that a policy π_i for a specific location $i = 1, \dots, N$ is ZIO if it places orders only when its current inventory level is zero. It is easy to check that for the holding cost structures considered in this paper, there exists an optimal policy $\pi^{OPT} = (\pi_0^{OPT}, \pi_1^{OPT}, \dots, \pi_N^{OPT})$ such that π_i^{OPT} is FCFS and ZIO for all i . However, the dominance of ZIO policies does not hold under the more general procurement costs presented in §4.5.

Finally, given a policy π for the entire system, we denote $\mathcal{C}(\pi)$ the total cost incurred by π over the planning horizon. This cost can be split into two parts: The total procurement cost $\mathcal{K}(\pi)$ and the total holding cost $\mathcal{H}(\pi)$. Thus we have:

$$\mathcal{C}(\pi) = \mathcal{K}(\pi) + \mathcal{H}(\pi)$$

4.3 A simple approximation algorithm

In this section, we introduce the main ideas of our technique, called *uncrossing algorithm*, to build a feasible solution to an OWMR problem. For ease of understanding, we start by focusing only on the mechanisms involved and present the procedure on a simple model with stationary costs. Specifically, we assume that each location i incurs a fixed ordering cost K^i and a per-unit holding cost $h_t^i = h^i$. Recall that in this case, the definition of the sets I_W and I_J is straightforward since we have $h^i \geq h^0$ for all $i \in I_W$ and $h^i < h^0$ for all $i \in I_J$.

Our procedure works in two main phases, where the output of the first phase is used as an input for the second one. In what follows, we detail each phase of the algorithm and prove that the error on the cost of the final policy obtained is bounded by a constant factor.

4.3.1 Phase 1: Decomposition of the problem

The algorithm we present in this section recombines single-echelon policies to build a feasible solution for the original problem. We now propose a simple decomposition of an OWMR problem into $N + 1$ single-echelon systems:

- (S_i) Retailer i is considered as a single-echelon location facing demand d_t^i , with holding cost h^i and ordering cost K^i .
- (S_0) The warehouse is regarded as a single-echelon, multi-item system facing for each period t a demand d_t^i for item i , with a fixed ordering cost K^0 . A different holding cost is incurred depending on which item (retailer) a unit serves: An item $i \in I_J$ incurs a holding cost h^i while an item $i \in I_W$ incurs a holding cost h^0 .

The single-echelon problems (S_i) are commonly referred to as the *Uncapacitated Lot Sizing Problem*, which has been extensively studied in the literature since the seminal paper of Wagner and Whitin (1958), see for instance Brahimi et al. (2006) for a recent

survey on single item lot-sizing problems or Pochet and Wolsey (2006) for polyhedral approaches. Note that problem (S_0) at the warehouse is multi-item while problems at retailers are single-item. It is well-known that there exists an optimal policy which is ZIO if the holding costs are positive and demands have to be delivered on time.

In the remainder of the paper, we denote $\mathcal{C}_i(\pi_i)$ the cost incurred by policy π_i in system (S_i) . Similarly to a policy π for the OWMR problem, one can split $\mathcal{C}_i(\pi_i)$ into two parts for each $i = 0, \dots, N$. Therefore we denote $\mathcal{K}_i(\pi_i)$ (resp. $\mathcal{H}(\pi_i)$) the total ordering cost (resp. holding cost) of policy π_i at location i and we have for all $i = 0, \dots, N$:

$$\mathcal{C}_i(\pi_i) = \mathcal{K}_i(\pi_i) + \mathcal{H}_i(\pi_i)$$

Finally, notice that the decomposition presented above induces two natural lower bounds for the problem. On one hand, (S_0) is simply a OWMR problem with no fixed ordering cost for the retailers. On the other hand, the set of all independent single-echelon systems (S_i) for $i = 1, \dots, N$ is a OWMR problem with no fixed ordering cost at the warehouse. Therefore, any optimal solution for one of these two systems is a lower bound for the original problem, and the following lemma holds:

Lemma 4.3.1. *Let \mathcal{C}^* be the cost of an optimal policy for the OWMR problem with linear holding costs, and let $\pi_0^*, \pi_1^*, \dots, \pi_N^*$ be optimal policies for systems $(S_0), (S_1), \dots, (S_N)$, respectively. Then we have:*

$$\mathcal{C}^* \geq \max \left\{ \mathcal{C}_0(\pi_0^*), \sum_{i=1}^N \mathcal{C}_i(\pi_i^*) \right\}$$

4.3.2 Phase 2: The uncrossing algorithm

In this section, we turn our attention to the problem of building a feasible policy for an OWMR instance from single-echelon solutions to problems (S_i) . From now on, we assume that we are given a feasible ZIO policy π_i to problem (S_i) for all $i = 0, \dots, N$. Note that since policies π_i are computed independently, they define a policy $\pi = (\pi_0, \pi_1, \dots, \pi_N)$ that is not necessarily feasible for the OWMR problem. However, there exists some situations that we call *uncrossing policies*, in which a small adjustment suffices to transform π into a feasible policy, at no overcost. Taking advantage of this observation, the *uncrossing algorithm* we present in this section works as follows: First transform policy π into an uncrossing policy π' , then convert π' into a feasible policy π^u for the original problem.

Before giving the details of the algorithm, we start by introducing some additional notations. We denote $R = \{r_1, \dots, r_w\}$ the set of periods where the warehouse orders according to policy π_0 . For convenience we add to R an artificial period $r_{w+1} = T + 1$ corresponding to the end of the planning horizon, with no ordering cost. In the final policy π^u the warehouse will order exactly in the same periods R , but possibly different quantities. Given a set R – or a policy π_0 – and a period $s = 1, \dots, T$, let s^+ and s^- be the first period

of R after s and the last period of R prior to s , respectively:

$$s^+ = \min \{r \in R : r \geq s\} \quad (4.1)$$

$$s^- = \max \{r \in R : r \leq s\} \quad (4.2)$$

In what follows, the set R used to define s^+ and s^- will be clear from the context.

For a policy at a given location, we call *ordering interval* a pair s, s' of successive ordering periods (for convenience we allow $s' = T + 1$ when s is the last order) i.e. an order is placed in s and no order is placed in $\{s + 1, \dots, s' - 1\}$. We say that a period t is in the ordering interval s, s' if $t \in \{s, \dots, s' - 1\}$.

Definition 4.3.1. An ordering interval s, s' at retailer i *crosses* the orders at the warehouse if there exists an ordering interval r, r' at the warehouse such that $r < s < r' < s'$. A policy is said to be *uncrossing* if none of its ordering intervals crosses the orders at the warehouse.

For simplicity if the ordering interval s, s' crosses, we say that order s is crossing. Due to the definition, an order in period s at a retailer is not crossing either if $s \in R$ (s coincides with an order at the warehouse, i.e. is synchronized), or if the ordering interval s, s' is contained in an ordering interval of the warehouse. We now propose a simple algorithm that uncrosses the crossing orders of policies π_1, \dots, π_N with respect to π_0 and builds a feasible policy for the original OWMR problem.

Uncrossing algorithm

Step 1. For each $i = 1, \dots, N$, uncross every crossing order of π_i : Consider an ordering interval s, s' of π_i crossing an ordering interval r, r' of π_0 ($r < s < r' < s'$). Uncross s, s' by placing an additional order of retailer i in period r' . That is, for each crossing order s of π_i , retailer i places an additional order in the first warehouse ordering period after s . Repeat this step until no ordering interval of π_i crosses: We denote π'_i the resulting (uncrossing) policy.

Step 2. For all $i \in I_J$, modify the ordering periods of π'_i to keep only synchronized orders: For each ordering interval r, r' of policy π_0 at the warehouse, we cancel all the orders of retailer i in $[r, r')$ and replace them (if any) by a unique order in period r . In other words we shift to the left and collapse together all the orders inside an ordering interval of the warehouse.

Step 3. Let r and s be the ordering period of a unit serving demand d_t^i in policy π_0 and π_i , respectively. Then the final solution serves this unit with the pair of orders $[r^{it}, s^{it}]$, where:

$$[r^{it}, s^{it}] = \begin{cases} [s^+, s^+] & \text{if } t \geq s^+ \\ [s^-, s^-] & \text{if } i \in I_J \text{ and } t < s^+ \\ [r, s] & \text{if } i \in I_W \text{ and } t < s^+ \end{cases} \quad (4.3)$$

We denote $\pi^u = (\pi_0^u, \pi_1^u, \dots, \pi_N^u)$ the final policy obtained.

Figure 4.1 illustrates on an example with two retailers the evolution of the inventory level at the different locations after each step of the uncrossing algorithm. Notice that since the ZIO property is dominant in the linear holding cost setting, another alternative is to compute the quantities ordered in each location to obtain a ZIO feasible policy:

- *Location i :* For each ordering interval s, s' of policy π'_i (including the orders added in Step 1), order in period s a quantity

$$q_s^i = \sum_{t=s}^{s'-1} d_t^i$$

- *Central warehouse:* For each ordering interval r, r' of π_0 , the warehouse orders in period r a quantity equal to the sum of the quantities ordered by all the retailers in periods $r, \dots, r' - 1$, that is:

$$q_r^0 = \sum_{t=r}^{r'-1} \sum_{i=1}^N q_t^i$$

One can prove that Step 3 assigns each demand d_t^i to the cheapest pair of orders available in π_0 and π'_i and hence the resulting policy is ZIO. However, this may not be true for all

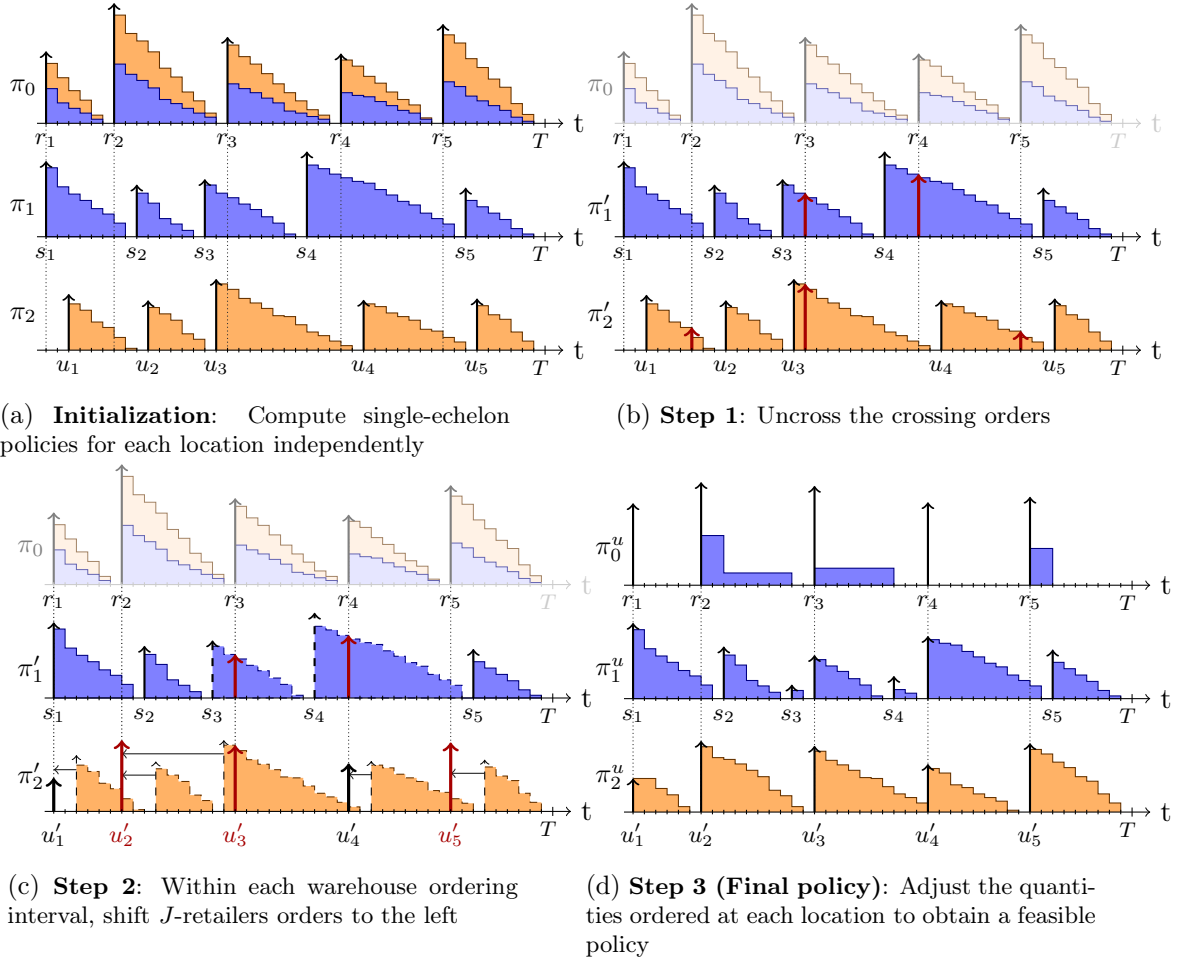


Figure 4.1: The three steps of the uncrossing algorithms with $T = 75$, $N = 2$, $I_W = \{1\}$ and $I_J = \{2\}$

cost structures: In fact, §4.5 discusses the case of a procurement cost structure in which the ZIO property is not dominant. In this case, the uncrossing algorithm produces a solution that is not necessarily ZIO, but we shall prove that its performance guarantee remains the same.

We now turn our attention to the overcost incurred by the uncrossing algorithm. The next lemma states that the uncrossing solution π^u is feasible for the original problem and bound this overcost with the cost incurred by the single-echelon policies $\pi_0, \pi_1, \dots, \pi_N$.

Lemma 4.3.2. *Given single-echelon feasible policies π_i to problems (S_i) , the uncrossing algorithm builds in time $O(NT)$ a feasible and uncrossing policy π^u for the OWMR problem with linear holding costs such that*

$$\mathcal{C}(\pi^u) \leq \mathcal{C}_0(\pi_0) + \sum_{i=1}^N \mathcal{C}_i(\pi_i) + \sum_{i=1}^N \mathcal{K}_i(\pi_i)$$

Proof. We do not detail the proof of this lemma since we prove the same result for the

more general cases of level-dependent and shelf age holding cost structures in the following sections (see the proofs of Lemmas 4.4.1 and 4.4.4). Basically, the upper bound on the final costs comes from two key observations: Firstly, Step 1 at most doubles the ordering cost of policies π_i for $i = 1, \dots, N$. This is straightforward since it adds at most one order in π'_i for each existing order in π_i . Secondly, Step 2 and 3 of the algorithm can only decrease the costs incurred by π' . It is clearly the case for Step 2, which only cancels some orders of π' . For Step 3, we prove that if a demand d_t^i is ordered in period r in π_0 and s in π_i , the final holding cost incurred by π^u using the pair of orders $[r^{it}, s^{it}]$ is lower than the sum of the holding costs incurred for the same demand in policies π_0 and π_i . \square

Notice that the bound given by Lemma 4.3.2 is valid for any set of feasible single-echelon policies $\pi_0, \pi_1, \dots, \pi_N$. In particular, the inequality holds when the uncrossing algorithm builds a solution upon optimal single-echelon policies $\pi_0^*, \pi_1^*, \dots, \pi_N^*$ for subproblems $(S_0), (S_1), \dots, (S_N)$. Hence we have from Lemmas 4.3.1 and 4.3.2:

$$\begin{aligned} \mathcal{C}(\pi^u) &\leq \mathcal{C}_0(\pi_0^*) + 2 \sum_{i=1}^N \mathcal{C}_i(\pi_i^*) \\ &\leq 3 \max \left\{ \mathcal{C}_0(\pi_0), \sum_{i=1}^N \mathcal{C}_i(\pi_i) \right\} \\ &\leq 3\mathcal{C}^* \end{aligned}$$

Therefore the cost of the final solution obtained is at most three times the optimal cost.

4.3.3 Split and conquer

In this section, we decompose in a more subtle way the OWMR problem into $N + 1$ instances of single-echelon lot-sizing problems. The goal is to improve upon the 3-approximation algorithm we proposed in the previous section. Indeed in § 5.3.1, we brutally relax the original problem to obtain two uncorrelated lower bounds on the optimal cost (see Lemma 4.3.1). We then use these lower bounds to prove a performance guarantee of 3. Here we slightly modify the subproblems by simply considering single-echelon systems (\hat{S}_i) with holding costs halved $\hat{h}^i \equiv \frac{1}{2}h^i$ for all $i = 0, \dots, N$. This allows us to refine the lower bound and to exploit the fact that the uncrossing algorithm at most doubles the ordering costs while leaving the holding costs untouched. Building upon this, we obtain a 2-approximation for the OWMR problem.

Using the notation \hat{h}^i introduced above, we define the split and uncross algorithm, which decomposes into two phases:

Split and uncross algorithm

Phase 1 (Splitting phase). Split the original problem into $N + 1$ single-echelon problems:

(\widehat{S}_i) Retailer i is considered as a single-echelon location facing demand d_t^i , with ordering cost K^i and holding cost $\hat{h}^i \equiv \frac{1}{2}h^i$.

(\widehat{S}_0) The warehouse is regarded as a single-echelon, multi-item system facing for each period t a demand d_t^i for item i , with a fixed ordering cost K^0 . The holding cost incurred for item i is equal to \hat{h}^i if $i \in I_J$ and $\hat{h}^0 \equiv \frac{1}{2}h^0$ if $I \in I_W$.

The algorithm then computes optimal solutions $\hat{\pi}_i^*$ for each single-echelon problem, $i = 0, \dots, N$.

Phase 2 (Uncrossing phase). Apply the uncrossing algorithm to policies $\hat{\pi}^* = (\hat{\pi}_0^*, \dots, \hat{\pi}_N^*)$

Since holding costs are proportional to the quantity held, one can also halve the demands instead of the holding costs. The algorithm then adds another step to multiply all the quantities ordered by two. This clearly holds in the shelf age holding cost structure in which holding costs are proportional to the demand, but does not apply to the (non-linear) level-dependent holding cost structure introduced in §4.4.1.

Note that any feasible policy π_i for system (\widehat{S}_i) is also feasible for system (S_i), which considers the whole holding cost h^i instead of \hat{h}^i . In what follows, we denote $\widehat{\mathcal{E}}_i(\pi_i)$ (resp. $\widehat{\mathcal{H}}_i(\pi_i)$) the total cost (resp. holding cost) in system (\widehat{S}_i) of a feasible policy π_i (as opposed to its costs in system (S_i) defined in §5.3.1). It is straightforward to see that

$$\widehat{\mathcal{E}}_i(\pi_i) = \widehat{\mathcal{H}}_i(\pi_i) + \mathcal{K}_i(\pi_i)$$

and

$$\mathcal{H}_i(\pi_i) = 2\widehat{\mathcal{H}}_i(\pi_i)$$

We shall prove that the cost of the policy built by this new algorithm is at most twice the optimal cost. Recall that in the previous section, we have only derived uncorrelated lower bounds from the decomposition into $N + 1$ subproblems. In fact while the ordering costs are somewhat split between (S_0), (S_1), \dots , (S_N), the holding cost incurred by the single-echelon solutions can overlap. On the other hand, problems (\widehat{S}_i) use only a fractional part of the holding costs and thus also split the holding costs between the warehouse and the retailers. The following lemma uses this property to introduce a more intertwined lower bound, which states that the *summation* of the optimal cost at the warehouse and at the retailers in subproblems (\widehat{S}_i) is a lower bound of any feasible policy for the OWMR problem. One can note the difference with Lemma 4.3.1, where the *maximum* between the cost of the warehouse system (S_0) and the sum of the costs of the retailers (S_i) is a lower

bound:

Lemma 4.3.3. *Let \mathcal{C}^* be the cost of an optimal policy for the OWMR problem with linear holding cost structure, and $\hat{\pi}_i^*$ be an optimal policy for the single-echelon problem (\hat{S}_i) , $i = 0, \dots, N$. We have:*

$$\mathcal{C}^* \geq \widehat{\mathcal{C}}_0(\hat{\pi}_0^*) + \sum_{i=1}^N \widehat{\mathcal{C}}_i(\hat{\pi}_i^*)$$

Proof. Here again, the result is proven in the more general case of level-dependent and shelf age holding cost structures. We refer the reader to the proofs of Lemmas 4.4.2 and 4.4.5 for each case. \square

To conclude this section, we show that the split and uncross algorithm is a 2-approximation for the OWMR problem with a linear holding cost structure. Notice that for all $i = 0, \dots, N$, the cost of policy $\hat{\pi}_i^*$ when applied to problem (S_i) is:

$$\begin{aligned} \mathcal{E}_i(\hat{\pi}_i^*) &= \mathcal{H}_i(\hat{\pi}_i^*) + \mathcal{K}_i(\hat{\pi}_i^*) \\ &= 2\widehat{\mathcal{H}}_i(\hat{\pi}_i^*) + \mathcal{K}_i(\hat{\pi}_i^*) \end{aligned}$$

Since Phase 2 of the new algorithm applies exactly the uncrossing algorithm introduced in the previous section, we have from Lemma 4.3.2:

$$\begin{aligned} \mathcal{C}(\hat{\pi}^u) &\leq \mathcal{C}_0(\hat{\pi}_0^*) + \sum_{i=1}^N \mathcal{E}_i(\hat{\pi}_i^*) + \sum_{i=1}^N \mathcal{K}_i(\hat{\pi}_i^*) \\ &\leq \mathcal{H}_0(\hat{\pi}_0^*) + \mathcal{K}_0(\hat{\pi}_0^*) + \sum_{i=1}^N \mathcal{H}_i(\hat{\pi}_i^*) + 2 \sum_{i=1}^N \mathcal{K}_i(\hat{\pi}_i^*) \\ &\leq 2\widehat{\mathcal{H}}_0(\hat{\pi}_0^*) + \mathcal{K}_0(\hat{\pi}_0^*) + \sum_{i=1}^N 2\widehat{\mathcal{H}}_i(\hat{\pi}_i^*) + 2 \sum_{i=1}^N \mathcal{K}_i(\hat{\pi}_i^*) \\ &\leq 2\widehat{\mathcal{C}}_0(\hat{\pi}_0^*) + \sum_{i=1}^N 2\widehat{\mathcal{C}}_i(\hat{\pi}_i^*) \end{aligned}$$

Therefore from Lemma 4.3.3, we conclude that the worst-case guarantee of the split and uncross algorithm is two:

Theorem 4.3.4 (Split and uncross algorithm). *The uncrossing algorithm based on optimal policies to the decomposition (\hat{S}_i) has a performance guarantee of 2 for the one-warehouse multiretailer problem with linear holding costs.*

Notice that the ordering periods at the warehouse are not changed in $\hat{\pi}^u$. Thus cost $\mathcal{C}(\hat{\pi}^u)$ is in fact bounded by $2 \sum_{i=0}^N \widehat{\mathcal{C}}_i(\hat{\pi}_i^*) - \mathcal{K}_0(\hat{\pi}_0^*)$. It follows that we can get *a posteriori* a better guarantee than 2 for our algorithm: For instance if in the relaxation the total ordering costs at the warehouse represents 1/4 of the total cost of $\hat{\pi}^*$ (i.e. the four parts of

the costs are balanced), we get a 1.75-approximate solution. An open question is whether we can use this property to improve *a priori* our approximation guarantee, possibly using a different splitting of the demands among the retailers and the warehouse.

4.3.4 Complexity analysis

The strength of such a technique lies mainly in its simplicity as well as in its low computational complexity. First, it is easy to see that given single-echelon policies for all locations the uncrossing algorithm can be performed in $O(NT)$. It remains to determine the computational complexity of solving to optimality each single-echelon problem (S_i) for $i = 0, \dots, N$. If the ZIO property is dominant, one can classically (see for instance Zipkin (2000)) represent the single-echelon problem (S_i) with a graph $G_i = (V_i, E_i)$ where the vertices are the different periods (i.e. $V_i = \{1, \dots, T + 1\}$) and an edge $(s, t) \in E_i$ represents two consecutive ordering periods (or the last ordering period and $T + 1$ if necessary), i.e. $E_i = \{(s, t) : 1 \leq s < t \leq T + 1\}$. In addition, for $i \geq 1$ we add an artificial vertex labeled 0 to V_i and artificial edges $(0, t)$ to E_i for $t = 1, \dots, T + 1$ to consider the possible zero demands at the beginning of the planning horizon. For the single-echelon problems corresponding to the retailers (i.e. $i \in \{1, \dots, N\}$), the length $l_{s,t}^i$ of each edge $(s, t) \in E_i$ is set to:

$$l_{s,t}^i = \begin{cases} K^i + \sum_{u=s}^{t-1} (u-s)h^i \cdot d_u^i & \text{if } s \geq 1 \\ 0 & \text{if } s = 0 \text{ and } \sum_{u=1}^{t-1} d_u^i = 0 \\ \infty & \text{otherwise} \end{cases}$$

This length corresponds to the cost incurred for placing an order in period s and holding units to satisfy the demands of period s till period $t - 1$. On the other hand, these lengths for the graph G_0 corresponding to the warehouse (multi-item uncapacitated lot-sizing problem) are set to:

$$l_{s,t}^0 = K^0 + \sum_{u=s}^{t-1} \left(\sum_{i \in I_W} (u-s)h^0 \cdot d_u^i + \sum_{i \in I_J} (u-s)h^i \cdot d_u^i \right)$$

Since h^i are nonnegative, these lengths are also nonnegative. One can then easily derive a production planning for the inventory problem (S_i) from a path $0 = t_1^i < t_2^i < \dots < t_m^i = T + 1$ by producing in every period corresponding to a vertice $1 \leq t_k^i \leq T$ in the path. The optimal solution for the corresponding single-echelon problem is then simply the shortest path from 0 to $T + 1$ in G_i . Bellman's algorithm solves such a problem in $O(T^2)$ time, leading to an overall complexity of $O(NT^2)$.

It is interesting to note that when the system has a linear holding cost structure, one can use advanced dynamic programming techniques to solve the single-echelon problems more efficiently. In the Wagner-Whitin case, there even exists improved algorithms that

find an optimal solution to the single-echelon problem in linear time (See Federgruen and Tzur (1991), Wagelmans et al. (1992) and Aggarwal and Park (1993)). Using one of these techniques, it is possible to solve the $N + 1$ single-echelon problems with linear holding cost structure in time $O(NT)$ and the overall complexity of our algorithm then decreases to $O(NT)$.

To the best of our knowledge, the complexity of our algorithm outperforms any existing approximation algorithm for the OWMR problem.

4.3.5 A bad example

In this section, we exhibit an instance of the OWMR problem for which the worst-case bound presented in Theorem 4.3.4 is tight. We consider a single warehouse that supplies two retailers over three periods, with a stationary linear holding cost structure. The parameters have the following values:

- $K^0 = 1, h^0 = 1$
- $K^1 = 0, h^1 = 0, d_1^1 = 1, d_2^1 = 0, d_3^1 = 0$
- $K^2 = z, h^2 = 2, d_1^2 = 0, d_2^2 = 1, d_3^2 = 1 + \varepsilon$, where $z \geq 2$ and $0 < \varepsilon < 1$

The optimal ZIO policies for the single-echelon systems with holding costs halved are unique and can be computed easily:

- The *warehouse* orders in periods 1 and 3.
- *Retailer 1* orders in period 1.
- *Retailer 2* orders in period 2.

Using these optimal single-echelon policies, the uncrossing policy produces a solution for the entire system by adding an order for retailer 2 in period 3, holding only one unit in the warehouse in period 1 (to serve d_2^2). The final cost it incurs is then:

$$\begin{aligned} \mathcal{C}(\pi^u) &= 2K^0 + K^1 + 2K^2 + h^0 \times 1 \\ &= 2z + 3 \end{aligned}$$

Another feasible policy for this instance is such that the warehouse and retailer 1 order only in period 1, while retailer 2 orders only in period 2. The warehouse then holds demands d_2^2 and d_3^2 in period 1, before it supplies retailer 2 in period 2. Thus, it incurs a total cost of

$$\begin{aligned} \mathcal{C}(\pi^*) &= K^0 + K^1 + K^2 + h^0 \times (2 + \varepsilon) + h^2 \times (1 + \varepsilon) \\ &= z + 5 + 3\varepsilon \end{aligned}$$

The worst case bound is then reached asymptotically as z goes to infinity:

$$\lim_{z \rightarrow \infty} \frac{\mathcal{C}(\pi^u)}{\mathcal{C}(\pi^*)} = \lim_{z \rightarrow \infty} \frac{2z + 3}{z + 5 + 3\varepsilon} = 2$$

4.4 Extension to more general holding cost structures

4.4.1 Non-linear level-dependent holding costs

Assumptions

We consider a level-dependent holding cost structure: Recall that $h_t^i(x)$ represents the price to hold an amount x of products in stock at location i from period t to $t + 1$. Notice that unlike the shelf age holding cost structures, the level-dependent holding cost is memoryless, i.e. the price paid in period t only depends on the current stock level, no matter how long the products have been in stock. We show in what follows that it is possible to extend the main results of the previous section and obtain a 2-approximation for non-linear holding costs under quite weak assumptions, namely:

- (LD1)** *Non-decreasing property.* We assume that holding cost $h_t^i(\cdot)$ is non-decreasing with respect to the stock level for each location i and for each period t , that is $x \leq y \Rightarrow h_t^i(x) \leq h_t^i(y)$ for all $t = 1, \dots, T$. In addition, we assume that $h_t^i(0)$ is nonnegative.
- (LD2)** *Sub-additivity property.* We assume that holding cost $h_t^0(\cdot)$ at the warehouse is sub-additive with respect to the stock level in each period, that is $h_t^0(x+y) \leq h_t^0(x) + h_t^0(y)$ for all $t = 1, \dots, T$.
- (LD3)** *Partition of the retailers.* For each retailer i , either $h_t^i(x) \geq h_t^0(x)$ holds for any period t and any stock level x (i is a W -retailer), or $h_t^i(y+q) - h_t^i(y) \leq h_t^0(x+q) - h_t^0(x)$ holds for any period t and any quantities x, y and q (i is a J -retailer).

As long as shortages and backlogging are not allowed (i.e. inventory levels are constrained to be non-negative), the non-decreasing property appears to be fulfilled by any reasonable holding cost function. Sub-additivity is a common property in economical function that captures economies of scale. Note that the sum of two sub-additive functions is a sub-additive function, as well as the product by a positive constant. Functions like \sqrt{x} or $\lceil x/B \rceil$ are examples of sub-additive functions. The latter function may model the practical case where racks of capacity B are used to store the items and a cost is paid for each additional rack.

These assumptions on holding costs appear to be sufficiently weak to include a large variety of real situations. However a more stringent condition on the marginal holding costs has to be satisfied in order to partition the set of retailers into two subsets I_W and I_J . Specifically property (LD3) states that a retailer i is in I_W if, for all periods, it is more expensive to hold inventory at the retailer than at the warehouse, i.e. $h_t^i(x) \geq h_t^0(x)$ for any period t and any stock level x . On the other hand, any retailer in I_J is such that there exists an optimal policy keeping no stock at the warehouse to supply its orders, i.e. it orders only in periods when the warehouse orders. Notice that this property may be imposed by practical considerations that force the warehouse to behave as a cross-dock

facility for a specific subset of retailers. More generally a retailer i belongs to I_J if its marginal holding cost is always less than or equal to the marginal holding cost at the warehouse. The condition on the marginal holding costs for a J -retailer ensures precisely that it is dominant to synchronize its orders with the warehouse (see Appendix B.1 for the proof). For example, consider the case where the holding cost of the warehouse is of the form $\alpha[x/B] + \beta x$, which models a unitary holding cost plus a rack overcost as already discussed. A retailer i belongs to I_W if its holding cost is greater than $h^0(x)$ and to I_J if its marginal holding cost is at most β . For instance if retailer i has linear holding cost γx , it belongs to set I_W if $\gamma \geq \alpha + \beta$ and to set I_J if $\gamma \leq \beta$. In the remainder of this section we assume that any retailer is either a W -retailer or a J -retailer as defined by property (LD3).

A modified decomposition (Splitting phase)

Similarly to the linear holding costs case, we decompose the OWMR problem into $N+1$ independent single-echelon systems. Recall that in the previous models, the performance guarantee of 2 can be obtained by halving either the holding costs or the demands of each single-echelon systems. However, when dealing with non-linear holding cost functions, the two approaches are not equivalent anymore. Since we are interested in bounding the costs incurred by our algorithm, we can only halve the holding cost functions when dealing with this specific cost structure. Thus we consider the following subsystems:

- (\widehat{S}_i) Retailer i is considered as a single-echelon location facing demand d_t^i , with fixed ordering cost K^i and holding cost $\hat{h}_t^i(\cdot) \equiv \frac{1}{2}h_t^i(\cdot)$ for all $t = 1, \dots, T$.
- (\widehat{S}_0) The warehouse is regarded as a single-echelon, multi-item system with a fixed ordering cost K_r^0 in period r . Each retailer $i \in I_J$ plays the role of a product i , while product 0 represents the aggregated demand of all the retailers in I_W . We charge the units x on hand in period t at cost $\hat{h}_t^0(\cdot) \equiv \frac{1}{2}h_t^0(\cdot)$ for product 0, and at cost $\hat{h}_t^i(\cdot)$ for each product $i \in I_J$.

Notice that if $I_J = \emptyset$, system (\widehat{S}_0) reduces to a single-item system of holding cost $\hat{h}_t^0(\cdot)$ facing the echelon demand. In the case $I_W = \emptyset$, system (\widehat{S}_0) is a JRP problem with no minor ordering cost. In a similar way as in §4.3.3 we solve independently to optimality the N single-echelon problems at the retailers and the multi-item problem at the warehouse to obtain a vector $\hat{\pi}^* = (\hat{\pi}_0^*, \dots, \hat{\pi}_N^*)$ of optimal ZIO policies. We then rebuild a feasible policy π^u for the original problem using the uncrossing algorithm.

Analysis

We now show that the performance guarantee of the Split and Uncross algorithm remains the same under non-linear holding costs. The analysis follows the two steps defined by Lemmas 4.3.2 and 4.3.3. We prove that these results remain valid under the more general

case of level-dependent holding cost by combining similar arguments with properties (LD1), (LD2) and (LD3).

Lemma 4.4.1. *Given independent single-echelon feasible ZIO policies π_i to problems (\widehat{S}_i) , the uncrossing algorithm builds a feasible policy π^u to the OWMR problem with level-dependent holding costs such that:*

$$\mathcal{C}(\pi^u) \leq 2 \left(\widehat{\mathcal{C}}_0(\pi_0) + \sum_{i=1}^N \widehat{\mathcal{C}}_i(\pi_i) \right)$$

Proof. A formal proof of this result can be found in Appendix B.2, but we give the reader some insights here. First, it is easy to see that when recomposing independent uncrossing policies into a feasible policy π^u for OWMR, the stock level at the warehouse and at the W -retailers, in policy π^u , is less than or equal to the one in the independent policies in each period. Due to the non-decreasing property of the holding costs h_i^t with respect to the stock level for each location and period, the resulting policy pays less to hold inventory. The same holds for J -retailers except for periods where the algorithm shifts orders to synchronize with the warehouse. However we can show that the holding cost paid in π_0 can be used to pay for the extra holding costs incurred in π_i^u . Then, transforming policies π_i at the retailers into uncrossing policies at most doubles the total ordering cost, while changing from holding costs $\widehat{h}(\cdot)$ to $h(\cdot)$ simply doubles the total holding cost paid by the policy. This leads to the inequality expressed in Lemma 4.4.1. \square

The next step to complete our analysis is to extend the lower bound of Lemma 4.3.3 to non-linear level-dependent holding costs. Sub-additivity of holding costs at the warehouse is required to prove the following result:

Lemma 4.4.2. *Let \mathcal{C}^* be the cost of an optimal policy for the OWMR problem with level-dependent holding cost structure and let $\widehat{\pi}_i^*$ be an optimal policy for single-echelon problem (\widehat{S}_i) , for $i = 0, \dots, N$. If the holding costs follow properties (LD1), (LD2) and (LD3) then*

$$\mathcal{C}^* \geq \widehat{\mathcal{C}}_0(\widehat{\pi}_0^*) + \sum_{i=1}^N \widehat{\mathcal{C}}_i(\widehat{\pi}_i^*)$$

Proof. The proof uses quite similar arguments as the ones of Lemma 4.3.3. We combine these ideas with properties (LD1), (LD2) and (LD3) to obtain the lower bound. A detailed proof is given in Appendix B.3. \square

As a corollary of the Lemmas 4.4.1 and 4.4.2 we have the following theorem, which extends Theorem 4.3.4 to non-linear holding cost functions:

Theorem 4.4.3. *The uncrossing algorithm applied to optimal single-echelon policies for systems (\widehat{S}_i) is a 2-approximation for the OWMR problem assuming non-decreasing level-dependent holding costs at each location and sub-additive holding costs at the warehouse,*

see properties (LD1), (LD2) and (LD3). Its time complexity is $O(NT^3)$, provided that any cost $h_t^i(x)$ can be evaluated in constant time.

Proof. The time complexity corresponds to the resolution to optimality of the different (single-echelon) uncapacitated lot sizing problems. With general holding costs, the computation of the value of the arcs as described in §4.3.4 requires $O(T^3)$ operations for a retailer problem (\widehat{S}_i) and $O(NT^3)$ operations for the warehouse problem (\widehat{S}_0). The other steps are linear in the size of the OWMR problem, i.e. in $O(NT)$. To the best of our knowledge this is the first approximation algorithm for such general holding costs. \square

4.4.2 Shelf age dependent holding costs

The metric holding cost structure

Although level-dependent holding cost structure is a classical assumption in the inventory literature, it remains restrictive and fails to modelize many practical situations. In Levi et al. (2008c), the authors introduce a shelf age dependent holding cost structure that captures additional phenomena such as perishable goods. In this case the cost of holding one unit of product depends on how long this specific unit has been physically held in the stock. Specifically, the cost incurred by a unit ordered with the pair $[r, s]$ to serve demand d_t^i incurs a total holding cost of h_{rs}^{it} . This cost includes in particular the cost of holding this unit from period r to s at the warehouse and from period s to t at retailer i . Notice that although we refer to them as holding costs, these parameters can also include per-unit ordering costs at the warehouse and at the retailer. The authors assume that these holding cost parameters obey the following so-called Monge properties (see Levi et al. (2008c) for a detailed discussion of these properties):

- (SA1) *Non-negativity.* Parameters h_{rs}^{it} are nonnegative.
- (SA2) *Monotonicity with respect to s .* Each retailer i is in exactly one of the two following situations: Either h_{rs}^{it} is non-increasing in $s \in [r, t]$ for each demand point (i, t) and warehouse order r , or h_{rs}^{it} is non-decreasing in $s \in [r, t]$ for each demand point (i, t) and warehouse order r . This property defines a partition of the set of retailers into two subsets: I_W and I_J , respectively. A retailer in I_W will be called for short a W -retailer, similarly a J -retailer if it belongs to I_J .
- (SA3) *Monotonicity with respect to r .* For each demand point (i, t) and retailer order in period s , h_{rs}^{it} is non-increasing in $r \in [1, s]$. Moreover for each retailer i in I_J and demand point (i, t) , for each $r' < r \leq t$, the inequality $h_{r'r}^{it} \geq h_{rr}^{it}$ holds.
- (SA4) *Monge property.* For each demand point (i, t) with $i \in I_W$, and $r_2 < r_1 \leq s_2 < s_1 \leq t$, the inequality $h_{r_2, s_1}^{it} + h_{r_1, s_2}^{it} \geq h_{r_2, s_2}^{it} + h_{r_1, s_1}^{it}$ holds.

These properties are satisfied by linear holding costs. We call this holding cost structure *Monge holding costs*. Observe that in the linear holding cost setting, I_W corresponds to

the set of retailers for which it is cheaper to hold inventory at the warehouse while I_J refers to retailers for which it is cheaper to hold inventory at the retailer. In this paper we generalize the Monge holding costs by relaxing property (SA4) to property (SA4'):

(SA4') *Triangle inequality.* For each demand point (i, t) with $i \in I_W$, and $r < s < t$, the inequality $h_{rt}^{it} + h_{ss}^{it} \geq h_{rs}^{it}$ holds.

We call this new structure *metric holding costs* for property (SA4') resembles a triangle inequality. Indeed this inequality implies that the cost of holding a unit at the warehouse from period r to t plus the cost of holding this unit at the retailer from period s to t encapsulates the cost of keeping the unit at the warehouse from period r to s and then at the retailer from period s to t . It is clearly true for linear holding costs. Moreover one can see that the metric holding costs is a generalization of the Monge holding costs, in the sense that Property (SA4) implies (SA4'): The triangle inequality is obtained by instantiating the Monge property with $r_2 = r$, $r_1 = s_2 = s$ and $s_1 = t$. That is a system obeying the Monge holding costs also obeys the metric holding costs. Finally, note that the metric holding costs (and its special case, the Monge holding costs) ensure that each demand d_t^i can be served from a unique pair of orders $[r, s]$ in an optimal solution. Since demands are nonnegative by definition, this means one can include d_t^i into a parameter $H_{rs}^{it} = h_{rs}^{it} \cdot d_t^i$ such that if h_{rs}^{it} satisfies properties (SA1), (SA2), (SA3) and (SA4'), so does H_{rs}^{it} . Hence we shall assume w.l.o.g. that $d_t^i = 1$ for all i, t such that $d_t^i \geq 0$ in the remainder of this section.

To motivate further the interest of metric holding costs over Monge holding costs, we now exhibit an example of a metric holding cost structure that does not satisfy (SA4). Consider that cost h_{rs}^{it} can be split between the warehouse and retailer i : More precisely this cost can be written as $h_{rs}^{it} = \omega(s - r) + \rho_{st}^i$, where $\omega(\tau) \geq 0$ is the cost of carrying a unit at the warehouse during τ periods and $\rho_{st}^i \geq 0$ is the cost of carrying a unit at retailer i from period s to t . On one hand, the triangle inequality of property (SA4') is satisfied if $\omega(\cdot)$ is a non-decreasing function, which is a natural assumption: It means that holding a unit longer at the warehouse incurs a larger warehouse holding cost. On the other hand, $\omega(\cdot)$ must be convex in the set of integers (i.e. $\omega(\tau + 1) - \omega(\tau) \geq \omega(\tau) - \omega(\tau - 1), \forall \tau \geq 1$) to satisfy the Monge property. The convexity condition is obtained by instantiating the Monge inequality with $r_1 = r_2 + 1$, $s_1 = s_2 + 1$ and $\tau = s_2 - r_2$. For instance, holding costs $\omega(\tau) = \sqrt{\tau}$ at the warehouse and $\rho_{st}^i = t - s$ at a retailer $i \in I_W$ satisfy (SA1), (SA2), (SA3), (SA4') but not (SA4). That is, the latter example obeys the metric holding costs but not the Monge holding costs. Note that Levi et al. (2008c) explicitly use the Monge property (SA4) in their analysis to prove the guarantee of their algorithm.

The authors also introduce an integer linear formulation for the OWMR problem with Monge holding costs. This formulation remains valid for the metric holding costs structure. The purpose of introducing formulation (IP) is only to prove the validity of the lower bounds given by Lemma 4.4.5: In particular no LP-relaxation of this large formulation

need to be solved by our algorithm. Recall that a pair of orders $[r, s]$ corresponds to a unit ordered in period r at the warehouse and in period $s \geq r$ by retailer i . The OWMR problem can be modeled by the following integer linear program (IP): For each demand point (i, t) the variable $x_{rs}^{it} \in \{0, 1\}$ indicates if demand d_t^i is ordered with respect to the pair $[r, s]$. The variable y_s^i indicates if retailer i orders in period s and similarly y_r^0 indicates if the warehouse orders in period r . Constraint (1) in formulation (IP) ensures that each demand is served by a certain pair of orders while constraints (2) and (3) ensure that a pair of orders $[r, s]$ is used to serve a demand d_t^i only if retailer i and the warehouse order in period s and r , respectively.

$$(IP) \left\{ \begin{array}{l} \min \sum_{r=1}^T y_r^0 K^0 + \sum_{i=1}^N \sum_{s=1}^T y_s^i K^i + \sum_{i=1}^N \sum_{t=1}^T \sum_{r,s:r \leq s \leq t} H_{rs}^{it} x_{rs}^{it} \\ \text{s.t.} \quad \sum_{r,s:r \leq s \leq t} x_{rs}^{it} = 1, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T \text{ such that } d_t^i > 0 \quad (1) \\ \sum_{r:r \leq s} x_{rs}^{it} \leq y_s^i, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall s = 1, \dots, t \quad (2) \\ \sum_{s:r \leq s \leq t} x_{rs}^{it} \leq y_r^0, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall r = 1, \dots, t \quad (3) \\ x_{rs}^{it}, y_r^0, y_s^i \in \{0, 1\}, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall s = 1, \dots, t, \forall r = 1, \dots, s \end{array} \right.$$

Decomposition

We again introduce for all i, t and all pair of orders $[r, s]$ an artificial holding cost parameter $\hat{h}_{rs}^{it} = \frac{1}{2} h_{rs}^{it}$ that we use to define the subproblems of the decomposition in the case of (shelf age) metric holding costs:

- (\hat{S}_i) Retailer i is considered as a single-echelon location facing demand d_t^i , with ordering cost K^i and holding cost \hat{h}_{ss}^{it} for all $t = 1, \dots, T$ and ordering period $s \leq t$.
- (\hat{S}_0) The warehouse is regarded as a single-echelon, multi-item system facing for each period t a demand d_t^i for item i , with a fixed ordering cost K_r^0 in period r . A different holding cost is incurred depending on which item (retailer) the units are intended to serve: if the units are ordered in period r , the corresponding holding cost is then \hat{h}_{rt}^{it} for $i \in I_W$ and \hat{h}_{rr}^{it} for $i \in I_J$.

Here again, the algorithm starts by solving independently to optimality the N single-echelon problems at the retailers and the multi-item problem at the warehouse to obtain a vector $\hat{\pi}^* = (\hat{\pi}_0^*, \dots, \hat{\pi}_N^*)$ of optimal ZIO policies and uncross these solutions to build a feasible policy π^u for the OWMR problem with metric holding costs.

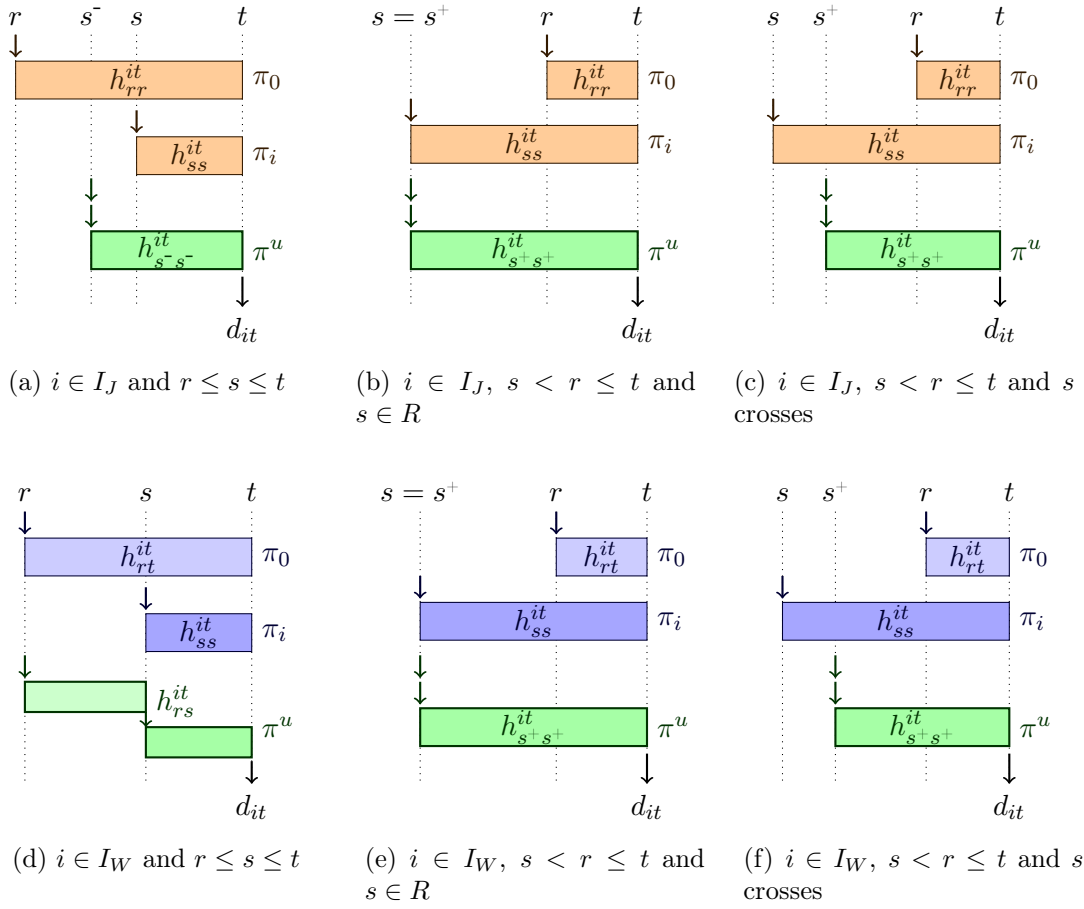


Figure 4.2: Analysis of the holding cost incurred by the uncrossed policy to serve one unit of demand (i, t) , assuming this unit is ordered in period $r \leq t$ (resp. period $s \leq t$) in the single-echelon policy π_0 (resp. π_i).

Analysis

The next lemma extends the results of Lemma 4.3.2 to an OWMR problem with metric holding cost structure. A formal proof is detailed in Appendix B.4. Figure 4.2 gives an insight of this result by comparing the different costs involved in the analysis.

Lemma 4.4.4. *Given single-echelon feasible policies π_i to problems (S_i) , the uncrossing algorithm builds in time $O(NT)$ a feasible and uncrossing policy π^u for the OWMR problem with metric holding costs such that*

$$\mathcal{C}(\pi^u) \leq \mathcal{C}_0(\pi_0) + \sum_{i=1}^N \mathcal{C}_i(\pi_i) + \sum_{i=1}^N \mathcal{K}_i(\pi_i)$$

It remains to show that the sum of the costs of the optimal single echelon solutions to subproblems $(S_0), (S_1), \dots, (S_N)$ is a lower bound on the optimal cost. One can prove this result by simply breaking apart an optimal policy to the original problem into feasible

policies for these subsystems, in a similar fashion as the proof of Lemma 4.4.2. We present an alternative proof for this bound that uses the formulation (IP) and decomposes the integer program into $N + 1$ simpler integer programs corresponding to problems (\widehat{S}_i) . Appendix B.5 further details the proof of the following lemma:

Lemma 4.4.5. *Let \mathcal{C}^* be the cost of an optimal policy for the OWMR problem with metric holding cost structure, and $\widehat{\pi}_i^*$ be an optimal policy for the single-echelon problem (\widehat{S}_i) , $i = 0, \dots, N$. We have:*

$$\mathcal{C}^* \geq \widehat{\mathcal{C}}_0(\widehat{\pi}_0^*) + \sum_{i=1}^N \widehat{\mathcal{C}}_i(\widehat{\pi}_i^*)$$

To conclude this section, we show that the split and uncross algorithm is a 2-approximation for the OWMR problem with a metric holding cost structure. Since Phase 2 of the procedure applies exactly the uncrossing algorithm introduced in the previous section, one can use similar arguments, combined with Lemmas 4.4.4 and 4.4.5, to prove that the following theorem holds:

Theorem 4.4.6. *The uncrossing algorithm based on optimal policies to the decomposition (\widehat{S}_i) has a performance guarantee of 2 for the one-warehouse multiretailer problem with metric holding costs.*

It is interesting to note that even under a metric holding cost structure, one can represent the single-echelon subproblems with a graph as in the linear case. The length of each edge $l_{s,t}^i$ in the graph G_i representing retailer i is defined as

$$l_{s,t}^i = \begin{cases} K^i + \sum_{u=s}^{t-1} H_{ss}^{iu} & \text{if } s \geq 1 \\ 0 & \text{if } s = 0 \text{ and } \sum_{u=1}^{t-1} d_u^i = 0 \\ \infty & \text{otherwise} \end{cases}$$

Similarly, the value of each edge in the graph G_0 is defined as follows:

$$l_{s,t}^0 = K^0 + \sum_{u=s}^{t-1} \left(\sum_{i \in I_W} H_{su}^{iu} + \sum_{i \in I_J} H_{ss}^{iu} \right)$$

Hence these problems can be solved in $O(T^2)$ time using classical shortest path dynamic programming algorithms, leading to an overall complexity of $O(NT^2)$.

4.5 Extension to more general procurement cost structures

4.5.1 Assumptions

In the previous sections we have assumed a fixed cost per order at each location. We now aim to generalize our approach to more general procurement costs $p_t^i(x)$ charged at location i to order an amount x of units in period t . In the literature, two procurement costs have been extensively studied, due to their practical relevance: the FTL (Full Truck Load) and the LTL (Less than Truck Load) procurement costs. Indeed, supplies are often delivered by batch, corresponding to truck capacity, and it is common to pay a fixed cost per truck plus a variable cost according to the actual load of the truck. They take the following form :

FTL. The cost to order $x > 0$ units in period t at location i is $p_t^i(x) = K_t^i + \lceil x/B^i \rceil k^i$.

That is a fixed cost k^i is charged for each batch of size B^i used to supply the order, in addition to a fixed ordering cost K_t^i .

LTL. The cost to order $x > 0$ units in period t at location i is $p_t^i(x) = K_t^i + \lfloor x/B^i \rfloor k^i + f^i(x - B^i \lfloor x/B^i \rfloor)$, where f is a non-decreasing function such that $f^i(0) = 0$ and $f^i(x) \leq k^i$ for $x \leq B^i$. That is, in addition to a fixed ordering cost K_t^i , a fixed cost k^i is charged for each full batch of size B^i used to supply the order and the (last) partially loaded batch is charged according to f . We call f the *LTL freight cost function*.

Notice that we restrict ourself to the case of a stationary batch size B^i and a stationary fixed cost per batch k^i at each location. In the remainder of this section we also assume, according to the result of Chan et al. (2000), that the procurement cost $p_t^i(\cdot)$ at each retailer is stationary over time, that is $p_t^i(\cdot) = p^i(\cdot)$ (or $K_t^i = K^i$ for all $i = 1, \dots, N$). Observe that a fixed ordering cost is a special case of FTL/LTL procurement costs with fixed cost per batch equal to zero. FTL/LTL cost structures have been widely investigated in the lot sizing literature. They are sometimes called multiple setup costs, stepwise costs or truckload discount. In Li et al. (2004) the authors propose a $O(T^3 \log T)$ algorithm for the single-echelon single-item problem with nondecreasing concave holding cost and non-decreasing LTL freight cost function f . The complexity can be reduced to $O(T^3)$ for traditional linear holding costs using Monge arrays.

Although we are mainly interested in FTL/LTL costs, most of our developments hold for slightly more general procurement costs $p_t^i(x)$. Indeed, we will only assume in the following that the functions $p_t^i(\cdot)$ are non decreasing, sub-additive and stationary at each retailer. In addition we assume the the procurement cost $p_t^0(\cdot)$ at the warehouse is *2-linearly sandwiched*:

Definition 4.5.1. Given a positive value λ , functions $p_t(\cdot), t = 1, \dots, T$ are said to be *λ -linearly sandwiched* if there exists some constants $(A_t)_{t=1, T}$ and a constant b such that

for every period t we have:

$$A_t + bx \leq p_t(x) \leq \lambda(A_t + bx) \quad \forall x > 0$$

These assumptions at the retailers are clearly satisfied for any stationary FTL/LTL procurement costs. At the warehouse, the following proposition gives conditions for FTL/LTL procurement costs to be 2-linearly sandwiched. The proof is developed in Appendix B.6.

Proposition 4.5.1.

- Any FTL procurement cost is 2-linearly sandwiched.
- LTL procurement costs are 2-linearly sandwiched if $k^i \leq K_t^i$ for all $i = 0, \dots, N$ and all periods

Our motivation for requiring the procurement costs at the warehouse to be 2-linearly sandwiched is algorithmic. Indeed, to the best of our knowledge, no polynomial time algorithm has been proposed for the multi-item uncapacitated lot sizing problem with multiple setup, while an efficient solution to this latter problem is central for our decomposition approach. Using this assumption, we are able to derive very easily a 2-approximation for this latter problem, which is sufficient to build a 2-approximation for the OWMR problem, as we detail in the next section.

4.5.2 A modified decomposition

We again decompose the OWMR problem into $N + 1$ independent single-echelon systems. In what follows, we assume that holding costs are non-linear as in the previous section, but one can apply the same method with a metric holding cost structure, using the decomposition introduced in §4.3.3. We assume from now that the procurement cost $p_t^0(\cdot)$ at the warehouse is 2-linearly sandwiched between some affine procurement costs $(A_t^0 + bx)$ and $2(A_t^0 + bx)$. We consider the following subproblems:

- (\widehat{S}_i) Retailer i is considered as a single-echelon location facing demand d_t^i , with procurement cost $p^i(\cdot)$, and holding cost $\widehat{h}_t^i(x) \equiv \frac{1}{2}h_t^i(x)$ for all $t = 1, \dots, T$.
- (\widehat{S}_0) The warehouse is regarded as a single-echelon, multi-item system with a fixed ordering cost A_r^0 in period r . Each retailer $i \in I_J$ is considered as a product i , while product 0 represents the aggregated demand of all retailers in I_W (for convenience, we sometimes distinguish product 0 according to its retailer $i \in I_W$ and call it product i again). We charge the units x on hand in period t at cost $\widehat{h}_t^0(x) \equiv \frac{1}{2}h_t^0(x)$ for product 0 and at cost $\widehat{h}_t^i(x)$ for each product $i \in I_J$.

Observe that system (\widehat{S}_0) is identical to the single-echelon problem we have considered for the warehouse in the previous section of the paper and thus an optimal policy can be found efficiently. Another consequence is that ZIO policies remain dominant for system (\widehat{S}_0). This is not necessarily true for a system (\widehat{S}_i) associated with a retailer, since an optimal policy may order a fraction of a subsequent demand to complete a batch.

We show in what follows how the uncrossing algorithm, presented for ZIO policies, can be easily adapted to deal with the so called *Positive Consumption Ordering* (PCO) policies. Remind that we assume without loss of generality that demands are satisfied in a FIFO manner in each policy. Hence for a given period t we can define without ambiguity the *consumption in t from (an order in period) $s \leq t$* as the number of units ordered in s that are used to serve demands till period t included. We then say that a policy for a retailer i is a *Positive Consumption Ordering* policy if for each order s , the consumption in period s from order s is strictly positive. That is, a PCO policy orders in a period s only if at least a part of the quantity ordered is used to serve the demand of period s . Clearly ZIO policies are a special case of PCO policies that require in addition that all units serving a specific demand are ordered in the same period. Also note that due to the FIFO discipline, the units on hand at the beginning of an ordering period s have all been ordered in the previous ordering period. This simple observation on PCO policies is in fact the main property used in our analysis. It is easy to see that PCO policies are dominant for each system (\widehat{S}_i) : For any $i > 0$, there exists an optimal policy that orders in a period s only if the consumption of the order in period s is not null. Otherwise the order in period s can simply be postponed, maintaining a feasible policy without increasing the total cost (the procurement costs are stationary and the holding costs can only decrease when an order is delayed).

The split and uncross algorithm can be applied directly to the optimal PCO policies $\widehat{\pi}_i^*$ for systems (\widehat{S}_i) . As in the previous sections, it builds a feasible uncrossing policy π_i^u for each retailer i . Since for each retailer i the procurement cost is non decreasing and stationary, the orders added in Step 1 of the algorithm at most double the ordering costs incurred by policy $\widehat{\pi}_i^*$. On the other hand Step 2 only merges existing orders and therefore can only decrease the total ordering cost of retailer i as the procurement costs are sub-additive and stationary. On the other hand, the quantities ordered at the warehouse in policy π_0^u may be totally different than the quantities ordered in policy $\widehat{\pi}_0^*$. However by construction policy π^u orders in the same periods, which allows us to bound the overcost incurred by this modification. In fact, Lemma 4.5.2 below shows that the procurement cost of policy π_0^u for the warehouse is at most twice the procurement cost of policy $\widehat{\pi}_0^*$.

Lemme 4.5.2 (Sandwiched procurement costs). *Consider a single-echelon system with λ -linearly sandwiched procurement costs $p_t(\cdot)$. Let ξ and ζ be two feasible policies that order respectively quantities x_t and z_t in period t , such that $x_t = 0 \Rightarrow z_t = 0$. That is, policy ζ orders in period t only if policy ξ orders. Then the overall procurement cost of ζ is at most λ times the overall procurement cost of ξ :*

$$\sum_{t=1}^T p_t(z_t) \leq \lambda \sum_{t=1}^T p_t(x_t)$$

Proof. The proof is immediate from the definition of linearly sandwiched procurement

costs. Let X (resp. Z) be the set of periods when policy ξ (resp. ζ) orders. By definition we have $\sum_{t=1}^T p_t(x_t) \geq \sum_{t \in X} A_t + b \sum_{t=1}^T x_t$ and $\sum_{t=1}^T p_t(z_t) \leq \lambda(\sum_{t \in Z} A_t + b \sum_{t=1}^T z_t)$. Since policy ζ can order only if policy ξ orders, we have $Z \subseteq X$. In addition any feasible policy must order exactly the total demands on the horizon, that is $\sum_{t=1}^T x_t = \sum_{t=1}^T z_t$. The result follows. \square

As a direct consequence of Lemma 4.5.2 we have $\mathcal{K}(\pi^u) \leq 2 \sum_{i=0}^N \mathcal{K}_i(\hat{\pi}_i^*)$. Turning our attention to the holding costs, we can use similar arguments as the ones considered in the proof of Lemma 4.4.1 (or Lemma 4.3.2 in the case of metric holding costs) to show that $\mathcal{H}(\pi^u) \leq 2 \sum_{i=0}^N \mathcal{H}_i(\hat{\pi}_i^*)$. As a result, we have

$$\begin{aligned} \mathcal{C}(\pi^u) &= \mathcal{K}(\pi^u) + \mathcal{H}(\pi^u) \\ &\leq 2 \sum_{i=0}^N \mathcal{C}_i(\hat{\pi}_i^*) \end{aligned} \quad (4.4)$$

Following the proof of Lemma 4.4.2 (or in the case of metric holding costs, Lemma 4.3.3), it is easy to prove that the optimal cost is bounded as follows:

$$\sum_{i=0}^N \mathcal{C}_i(\hat{\pi}_i^*) \leq \mathcal{C}^* \quad (4.5)$$

Finally we have to check that each problem (\hat{S}_i) of the decomposition can be solved to optimality in polynomial time. Recall that the problem (\hat{S}_0) has an affine procurement costs and thus can be solved as in §4.4.1 (or §4.4.2). On the other hand, each single-echelon problem (\hat{S}_i) is a lot-sizing with FTL/LTL cost structures and one can use the algorithm of Li et al. (2004) to compute an optimal policy with a complexity of at most $O(T^3 \log T)$. More precisely, the following theorem holds:

Theorem 4.5.3. *The OWMR problem with 2-linearly sandwiched FTL/LTL procurement costs can be approximated within a factor 2 in time $O(NT^3)$ for a metric holding cost structure and in time $O(NT^3 \log T)$ for sub-additive and concave traditional holding costs.*

In particular, the result holds if every location $i = 0, \dots, N$ of the system has either a FTL procurement cost or *any* LTL procurement cost such that the fixed ordering part in any period is greater than or equal to the fixed cost per batch (i.e. $k^i \leq K_t^i$ for all t).

Note that the subproblem (\hat{S}_0) at the warehouse have a classical ordering cost structure and therefore one can directly apply the results of the previous sections if the procurement costs of each retailer is only composed of a fixed ordering cost. In this case, the overall complexity is not altered and we have as a corollary of Theorem 4.5.3:

Corollary. *The OWMR problem with 2-linearly sandwiched FTL/LTL procurement costs at the warehouse and fixed ordering costs at the retailer can be approximated within a*

factor 2, in linear time $O(NT)$ for linear traditional holding cost, in time $O(NT^2)$ for metric holding cost and in time $O(NT^3)$ for sub-additive traditional holding costs.

Note that one could use the same idea, i.e. substitute the FTL/LTL procurement costs at the warehouse by affine ordering costs, and apply for instance the algorithm of Levi et al. (2008c) to solve the problem. However our analysis takes advantage of the property that the warehouse orders in π^u are placed in the same periods as $\hat{\pi}_0^*$. In particular in the case of fixed ordering costs, the total warehouse ordering cost incurred by π^u is equal to the total ordering cost incurred by $\hat{\pi}_0^*$ in (\hat{S}_0) . Recall that in the FTL/LTL case, we transform the general procurement cost structure into virtual fixed ordering cost parameters. The performance guarantee of two obtained with our algorithm then uses the fact that this operation at most doubles the real ordering costs incurred at the warehouse, therefore filling the gap observed in the case of fixed ordering costs. This approach may not work with other methods: For instance it is not clear whether the rounding technique proposed by Levi et al. (2008c) can improve the performance guarantee of 3.6 they prove in their paper.

Finally, note that one could derive a model that considers level dependent and shelf age holding costs with a FTL/LTL procurement cost structure and apply the split and uncross algorithm in the same fashion with the decomposition of §4.5.2 to build a feasible solution to this general OWMR problem. Since the same algorithm is used for both holding cost structures, the same analysis can be conducted on each part of the holding costs and the performance guarantee of two holds under these all-inclusive assumptions.

4.6 Computational experiments

In this section we present the first numerical results of our algorithm. For this purpose, we apply both the split and uncross algorithm and the rounding algorithm introduced by Levi et al. (2008c) on the same set of test problems and compare the results obtained with each procedure. Recall that these two algorithms start by deciding the warehouse ordering periods before computing a feasible policy for the entire system. In these numerical experiments, we use the improved version of these two algorithms that computes directly the optimal policy for each retailer given the warehouse orders. All computations were run on a DELL OptiPlex 960. Both algorithms were coded in C++ and applied on numerous test instances, described below. The solutions to the linear programs used to define lower bounds for the different problems and to build the solutions of Levi et al. (2008c) were computed with CPLEX.

Instances. We generate each location of the system separately and combine them to obtain instances of OWMR problems. More precisely, we restrict ourselves to cases where all the system costs are time-independent. That is, we consider constant ordering costs and traditional linear holding costs for each location, which take their values in $\{0.1, 1, 5, 9, 100\}$.

Number of retailers	Split and uncross		Levi et al. (2008c)	
	Z_{SU}/Z_{LP}	CPU time (s)	Z_L/Z_{LP}	CPU time (s)
N = 2	1.063	10^{-6}	1.021	0.35
N = 5	1.056	10^{-4}	1.031	0.80
N = 10	1.053	$2 \cdot 10^{-4}$	1.033	1.79
N = 25	1.048	$6 \cdot 10^{-4}$	1.035	6.16
N = 50	1.047	$1.5 \cdot 10^{-3}$	1.036	18.24

Table 4.1: Results obtained by the split and uncross algorithm and the rounding algorithm of Levi et al. (2008c) on OWMR instances ($T = 50$).

For each $i = 1, \dots, N$ and each possible pair of costs (K^i, h^i) , we simply generate retailers subproblems in which demands d_{it} are i.i.d. random variables with uniform distribution $\mathcal{U}(a, b)$ with $(a, b) \in \{(0, 1), (0, 5), (1, 1), (1, 5), (5, 5)\}$. To obtain an instance of a OWMR of size NT , we randomly select a warehouse and N retailers and simulate a sequence of T demands for each of them.

Numerical results. Both algorithms have been applied to 25000 instances of the problem. We define the linear relaxation of (IP) (see §4.4.2) as in Levi et al. (2008c), where constraint (4) in (IP) is replaced by the following inequality:

$$x_{rs}^{it}, y_r^0, y_s^i \geq 0$$

In what follows, we denote Z^* the cost of an optimal solution to (IP) and Z_{SU} (resp. Z_L) the total cost incurred by the split and uncross algorithm (resp. the rounding procedure of Levi et al. (2008c)): We clearly have $Z^* \leq Z_{SU}$ and $Z^* \leq Z_L$ from the optimality of Z^* . Let Z_{LP} be the cost of an optimal solution to the resulting linear program (LP). In particular, Z_{LP} is a lower bound on Z^* and therefore on Z_L and Z_{SU} .

We consider a various number of retailers $N \in \{2, 5, 10, 25, 50\}$: For all N , we generate 2500 different instances and run both algorithms on each of them. We then compare their running time and compute the ratio of each of their costs with Z_{LP} . Table 4.1 summarizes the results obtained when the planning horizon is $T = 50$.

While their rounding technique has a better performance on average, the running time of the split and uncross procedure is remarkably low compared to the one of Levi et al.. Moreover as the number of retailers grows, the average performances of both algorithms tend to converge asymptotically to a similar average gap of 4% (see Figure 4.3), giving our algorithm a clear edge on large systems from its better computational time. Hence the quality of the solution obtained, combined with the low complexity of our method, confirm that it is a legitimate practical tool. One can also notice that the performance of both algorithms in practice are much lower than their theoretical worst-case guarantee. In particular, the total cost incurred by *any* solution built with the split and uncross algorithm stands within 20% of Z_{LP} .

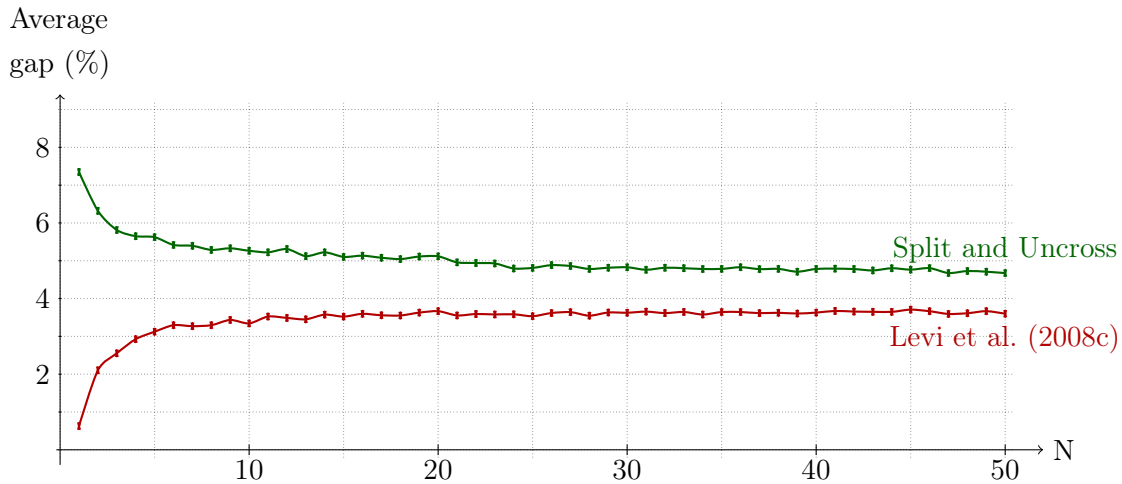


Figure 4.3: Evolution of the average gap with the LP solution as the number of retailers increases ($T = 50$).

An interesting question is then to find out if there exists instances in which the split and uncross algorithm outperforms the procedure of Levi et al. (2008c). For instance when the number of retailers is equal to $N = 40$ and for various horizon lengths ($T \in \{5, 10, 50, 100, 200\}$), Figure 4.4 shows that in practice this is the case for about 40% of the instances tested. This result implies that in some situations, our technique builds a better solution than the rounding algorithm of Levi et al. while being more efficient in terms of running time. The next step of this study should focus on isolating the type of instance that favors one method over the other to emphasize the strengths and weaknesses of each procedure. Furthermore, this could lead to develop improved versions of both algorithms by making some local adjustments in order to decrease the overall cost.

4.7 Conclusion

In this paper, we introduce a new simple and fast technique to approximate the OWMR problem from optimal single-echelon solutions. We then combine this algorithm with a decomposition approach for a relaxed problem by splitting the demands over the different locations and show that the resulting algorithm provides a 2-approximation for the original OWMR problem, which runs in linear time in the case of linear holding costs and fixed ordering costs. This is to the best of our knowledge the most efficient algorithm proposed in the existing literature. Moreover, we show that our technique extends to the framework of non-linear level dependent holding costs and a new shelf age holding cost structure. Finally the procedure also applies with a similar performance guarantee to the case of FTL/LTL procurement costs, with fixed cost per batch and a truckload discount. This is the first combinatorial algorithm we are aware of that finds an approximate solution for the OWMR problem with such general holding costs and procurement costs. Moreover in each

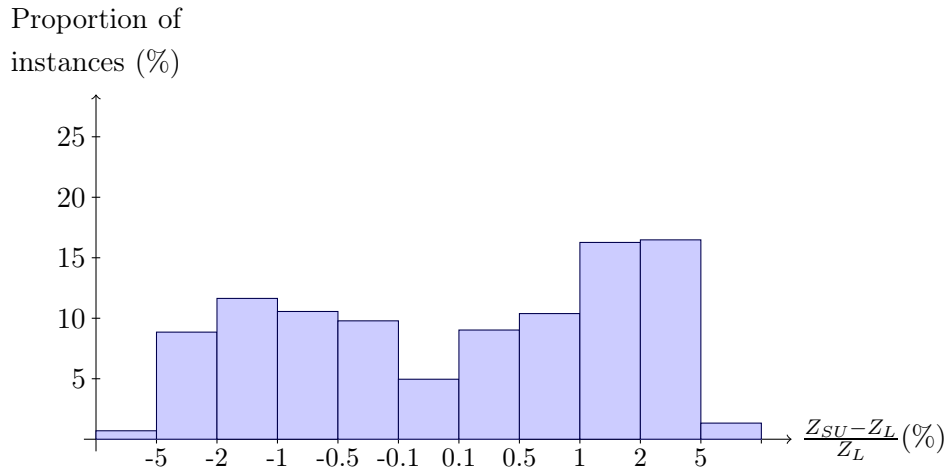


Figure 4.4: Comparison of the results obtained with the split and uncross algorithm vs. the rounding technique of Levi et al. ($N = 40$). Negative values on the x -axis correspond to instances in which the split and uncross algorithm builds a better solution than the rounding procedure.

case the algorithm runs in polynomial time in the size of the instance. We are convinced that we can mix our results to consider even more general holding cost structures, with both a non-linear traditional holding cost and a metric holding cost feature. Finally, we implement our technique and show that its performances in practice are much better than the theoretical worst-case guarantee derived in our analysis.

The main principles of our algorithm are quite general and thus are very likely that they can be applied in other fields than inventory management. This will be the topic of further research, as well as the extension of this technique to more complex inventory problems. In particular it would be of great interest to consider a capacity limitation on the size of the orders at the warehouse as well as the possibility of backloging the demands. However these important extensions certainly require additional conceptual developments.

Chapitre 5

Constant approximation algorithms for the one warehouse multiple retailers problem with backlogging or lost-sales

5.1 Introduction

The *One Warehouse Multi-Retailer (OWMR)* problem and its special case the *Joint Replenishment Problem (JRP)* have drawn a lot of attention in the recent deterministic inventory theory literature. Both models consider a divergent system and aim at optimizing the flow of goods through different locations, from an external supplier to the final customers. More precisely, we study a system in which a single warehouse receives inventory from an exogenous supplier of infinite capacity and in turn replenishes N final retailers facing the customers demands. Each retailer has a forecast of the demands to come for the next T periods of time: Therefore in each period it faces a deterministic demand that has to be filled by the end of the planning horizon. Any unit is therefore first ordered by the warehouse, then by the corresponding retailer before it finally serves the demand. Hence each retailer can only order units that are available (i.e. physically held) at the warehouse. Costs are incurred when a location places an order to replenish its stock, when units are physically held in the system or when demand is not immediately satisfied. In the latter case, we consider two classical assumptions and study a backlogging model on one hand and a lost-sales one on the other. The goal is then to find a distribution strategy that minimizes the total cost incurred to fill – or not – the demands by moving the products through the divergent network described above.

Most – if not all – of the existing work on the deterministic OWMR problem focuses on the case where demands must be satisfied on time. In this paper, we relax this assumption and consider OWMR problems in which retailers can backlog a demand to fill it later (or

simply decide to loose the sale). Under these assumptions, we use a decomposition of the original system into simple single-echelon subproblems to build a constant approximation algorithm for the OWMR problem with backlogging or lost-sales. That is, we bound the ratio between the cost incurred by our solution and the optimal cost by a constant α (we say that α is the performance guarantee of the algorithm or that the algorithm is an α -approximation). This means that for any instance of a problem, our algorithm builds a solution whose cost is at most α times the cost of an optimal solution. To the best of our knowledge, this is the first algorithm with a constant performance guarantee for these general versions of the OWMR problem. Moreover when restricted to the JRP, our technique matches the best known approximation proposed by Levi et al. (2006) for the problem with backlogging.

In its continuous-time version, there already exists some papers that approximate the OWMR problem when neither backorders nor lost-sales are allowed. Roundy (1985) proposed a 98%-effective algorithm that has later been extended in Muckstadt and Roundy (1993) to other multi-echelon problems. The same problem was recently studied by Stauffer (2012), who proposed a simple 1.27-approximation based on the recombination of optimal single-echelon strategies.

When the planning horizon is discrete, the OWMR problem and the JRP are both known to be NP-hard since the work of Arkin et al. (1989), therefore computing an optimal solution for these problems in polynomial time is unrealistic unless $P=NP$. However in the special case of the JRP, there exists some polynomial time algorithms when the number of periods is fixed (Veinott (1969)) or for a fixed number of items (Zangwill (1966b) and Kao (1979)). When the dimensions of the problem are not fixed, most of the research has focused on finding structural properties, heuristics or approximation algorithms for these two inventory problems. When demands and costs are bounded by constants, Federgruen and Tzur (1994) use a branch-and-bound method and a partitioning heuristic to yield epsilon-optimal solutions. The only approximation algorithm we are aware of when backlogging is allowed is the primal-dual method introduced by Levi et al. (2006), which yield a performance guarantee of two for this problem. For a more detailed review of the JRP literature, we refer the reader to Aksoy and Erenguc (1988) and Khouja and Goyal (2008).

In the more general case of the OWMR problem, the existing work is more scarce. When backlogging and stockouts are prohibited, there exists papers that propose methods to solve the OWMR problem. A model was introduced early by Schwarz (1973), who derived some structural properties on the problem, such as the dominance of the so-called *Zero Inventory Ordering* (ZIO) property, i.e. there exists an optimal policy in which the locations place orders only when their current inventory level is zero. Chan et al. (2000) studied the class of ZIO policies under a general cost structure and derived efficient heuristics based on a LP relaxation of the problem. Later Shen et al. (2009) extended their work and proposed a generic LP approach for a large class of inventory problems. In particular,

they relate their general OWMR model to a multi-commodity network flow problem and prove that its solution solves the original model to within ε -optimality. In the classical version of the model, Levi et al. (2008c) used a LP-rounding approach and provided 1.8-approximation algorithm for the OWMR problem without backlogging. This result was later improved by Nonner and Souza (2009) to a 5/3-approximation in the special case of JRP with deadlines. Finally, the most recent approach has been developed by Stauffer et al. (2011), who introduce a split and uncross technique that builds a 2-approximation for the OWMR problem. The procedure is based on a simple decomposition of the system into single-echelon subproblems and apply to a broad class of models with different cost structures.

When the model does not require the demands to be filled on time, the two most common assumptions in the inventory literature consider that unmet demands are either backlogged (*backorder* models) or that the corresponding sale is lost (*lost-sales* models). The former assumes that customers are willing to wait and unmet demands are satisfied later in time, once there is sufficient physical inventory in stock. In the latter, a demand that is not satisfied is simply lost, i.e. customers are impatient and leave the system if their demand is not filled immediately. Such models have been extensively studied in the lot-sizing literature, where a single location faces customers demands over a discrete, finite planning horizon.

Zangwill (1966a) was among the first to extend the pioneering work of Wagner and Whitin (1958) to incorporate the possibility to backlog units. Later, Federgruen and Tzur (1993) and Aggarwal and Park (1993) used more advanced dynamic programming techniques to solve the problem and improved the time complexity. An interesting extension of the lot-sizing problem is the multi-item version, which considers that several items are distributed, each incurring its own holding and backlogging costs. Several papers (see Pochet and Wolsey (1988, 1994) and Küçükyavuz and Pochet (2009)) have focused on the integer programming formulation of this problem in the case of single or multiple items. In particular, they propose valid inequalities for the problem, develop efficient separation algorithms and study the convex hull of the solutions. The special case of the JRP is a generalization of the multi-item problem in which additional ordering costs are incurred whenever a specific item replenishes its own stock. Levi et al. (2006) developed an algorithm based on primal-dual approach for this problem with backlogging and proved its cost is at most twice the optimal cost in the worst case.

The literature on lost sales models is more recent: Sandbothe and Thompson (1990) were the first to propose a forward algorithm to solve a lot-sizing problem with production capacity constraints and stockouts. Later, Aksen et al. (2003) introduced a dynamic programming approach to solve efficiently the uncapacitated problem with time-varying costs. Liu et al. (2007) considered a model where inventory is bounded, present an optimal dynamic program for this model and test it on industrial problem successfully. Other papers deal with the multi-item versions of the basic lot-sizing problem. In particular, Absi and

Kedad-Sidhoum (2008) extends the original work of Sandbothe and Thompson (1990) to a multi-item version and use a MIP approach to develop an effective method to find near optimal solutions.

Our contributions

Our work is strongly connected to the split and uncross technique presented in Stauffer et al. (2011), in which we introduced a simple 2-approximation algorithm for the OWMR problem without backorders. In this work, we extend this method to more general models allowing backorders or lost-sales and derive a constant approximation algorithm in each case. In addition, we present a new lower bound for each problem, based on a decomposition into simple single-echelon systems. We then recombine the optimal solutions to the subproblems into a feasible solution to the original problem. In the case of the OWMR problem with backlogging, we prove that the cost of the solution obtained is guaranteed to be at most three times the optimal cost under mild assumptions on the cost structure. Moreover, we show that in the special case of the JRP, the performance guarantee can be improved to two. Finally, we adapt our technique to the OWMR problem with lost-sales and build a 2-approximation for this problem. To the best of our knowledge, this paper presents the first constant approximation algorithms for the backlogging and the lost-sales version of the OWMR problem. Furthermore in the JRP case, our algorithm matches the best known performance guarantee obtained by Levi et al. (2006) when backorders are allowed.

The remainder of this paper is organized as follows. In §5.2, we formally introduce the assumptions and notations used throughout the following sections. In §5.3, we present the split and uncross technique and provide a 3-approximation algorithm for the backorder version of the OWMR problem. In §5.4, we show that in our cost structure, the lost-sales model is simply a special case of the backlogging problem. We are then able to improve the performance guarantee of our algorithm to two by simply modifying one step of our algorithm.

5.2 Assumptions and cost structure for the backlogging model

In this section, we focus on models where unmet demand is backordered. In this version of the OWMR problem, demands faced by the retailers are not necessarily satisfied on time but can instead be backlogged and served by an order placed later in time. We now discuss the assumptions and notations used throughout the remainder of the paper. In particular, we present a general cost structure that extends the cost parameters introduced in Levi et al. (2008c) and Stauffer et al. (2011).

We consider N retailers that face customers demands over a finite planning horizon, discretized into T periods. For each retailer $i = 1, \dots, N$ and each period $t = 1, \dots, T$, let d_t^i be the deterministic demand for retailer i in period t , to which we also refer as the

demand point (i, t) . Each retailer orders units from a central warehouse, which in turn orders from an external supplier. Recall that we assume that the decisions are centralized in order to minimize the total cost incurred by the system. As a consequence, orders placed by the retailers are always filled on time by units coming from the on-hand inventory of the warehouse. If the warehouse orders from its supplier in period t , it incurs a fixed ordering cost K_t^0 , regardless of how many units are ordered. Similarly, retailer i pays a fixed ordering cost K_t^i for ordering from the warehouse in period t . Chan et al. (2000) have shown that if the ordering costs at the retailers vary over time, the OWMR problem is as hard to approximate as the set cover problem and thus admits no constant guarantee unless $P = NP$ (Feige (1998)). Therefore we assume in this paper that the ordering costs at each retailer $i > 0$ are stationary, i.e. $K_t^i = K^i$ for all periods $t = 1, \dots, T$. The leadtimes are deterministic, thus we assume without loss of generality that the orders are delivered instantaneously from one location to another.

The most common assumption in the inventory literature is to consider that holding and backlogging units induce linear costs that are proportional to the inventory level in each location. However, it is worth noticing that the algorithms developed in the following sections only need to satisfy weaker properties to yield a constant performance guarantee for the OWMR problem with backorders. For clarity reasons, we start by introducing the assumptions on the cost structure in the linear case first. We then relate this basic parameters to the so-called metric carrying cost structure and generalize the assumptions to this more complex setting.

The linear cost structure

In the traditional linear cost structure, each unit physically held in location i in period t incurs a holding cost $h_t^i \geq 0$, while each backlogged unit for a specific retailer $i \geq 1$ induces a penalty cost of $b_t^i \geq 0$. Note that the warehouse cannot backlog the orders of its retailers and therefore it does not incur any backorder penalty cost. Another common assumption for the OWMR problem is that the set of retailers can be partitioned into two subsets I_J and I_W such that:

$$I_J = \{i = 1, \dots, N : h_t^i \leq h_t^0 \text{ for all } t\} \quad (5.1)$$

$$I_W = \{i = 1, \dots, N : h_t^i \geq h_t^0 \text{ and } b_t^i \geq h_t^0 \text{ for all } t\} \quad (5.2)$$

In the remainder of this paper, retailers in set I_J are also called J -retailers, while retailers from set I_W are called W -retailers. To say it simply, it is cheaper to store units at J -retailers than at the warehouse, while it is cheaper to hold units at the warehouse than at W -retailers. Note that in addition we assume that for all $i \in I_W$ and $t = 1, \dots, T$, it is also cheaper to hold unit at the warehouse rather than backlogging it at retailer i in period t . In other words, the backlogging cost of any W -retailer is greater than the holding cost at the warehouse. This assumption matches many practical situations, since

backlogging units is often more expensive than holding them in the stock, which is in turn more expensive than holding them in the central warehouse in the case of W -retailers. Note that the latter property is therefore weaker than these practical situations.

The metric carrying cost structure

We now introduce more general cost parameters, called *carrying cost parameters*, that encapsulate both holding and backlogging costs in a single notation. Namely, a unit ordered in period r at the warehouse and in period $s \geq r$ at retailer i (we denote by $[r, s]$ such a pair of orders) to satisfy a unit of demand d_t^i incurs a per-unit carrying cost of ϕ_{rs}^{it} . That is, ϕ_{rs}^{it} encapsulates the cost for holding one unit at the warehouse from r to s , then either holding it at retailer i from period s to t (if $s \leq t$) to serve demand in period t , or backlogging one unit of demand (i, t) from period t to s (when $s > t$). Note that this cost structure can capture additional phenomena than the traditional linear holding/backlogging costs. For instance, it allows us to modelize situations in which the holding cost incurred for a specific item at the central warehouse depends on which retailers it replenishes. It can also capture additional phenomena such as perishability, with a prohibitive holding cost if the period of storage in a location is longer than a certain threshold.

Linear cost parameters are clearly a special case of carrying cost parameters: Let (i, t) be a demand point, r an ordering period of the warehouse and $s \geq r$ an ordering period of retailer i . Then the total carrying cost incurred to serve demand (i, t) with the pair of orders $[r, s]$ under a linear cost structure is:

$$\phi_{rs}^{it} = \begin{cases} \left(\sum_{u=r}^{s-1} h_u^0 + \sum_{v=s}^{t-1} h_v^i \right) d_t^i & \text{if } s \leq t \\ \left(\sum_{u=r}^{s-1} h_u^0 + \sum_{v=t}^{s-1} b_v^i \right) d_t^i & \text{otherwise} \end{cases} \quad (5.3)$$

As already mentioned, the carrying cost parameters do not need to satisfy equation (5.3) for us to derive constant approximations for the backlogging model we consider. In fact, we simply assume that parameters ϕ_{rs}^{it} satisfy the following five properties:

- (P1) *Non-negativity.* The parameters ϕ_{rs}^{it} are nonnegative.
- (P2) *Piecewise monotonicity with respect to s .* Every retailer i is in exactly one of the two following situations: Either ϕ_{rs}^{it} is non-increasing in $s \in [r, t]$ and non-decreasing in $s \in [t, T]$ for each demand point (i, t) and warehouse order r , or ϕ_{rs}^{it} is non-decreasing in $s \in [r, T]$ for each demand point (i, t) and warehouse order r . This property defines a partition of the set of retailers into two subsets: I_W and I_J , respectively.
- (P3) *Monotonicity with respect to r .* For each retailer $i = 1, \dots, N$, each demand point (i, t) and retailer order in period $1 \leq s \leq T$, ϕ_{rs}^{it} is non-increasing in $r \in [1, s]$. Moreover if $i \in I_J$, we have: $\phi_{rr}^{it} \geq \phi_{r'r'}^{it}$ for $r \leq r' \leq t$ and $\phi_{rr}^{it} \leq \phi_{r'r'}^{it}$ for $t \leq r \leq r'$.

(P4) *Triangle inequality.* For each demand point (i, t) with $i \in I_W$, we have

$$\phi_{rs}^{it} \leq \begin{cases} \phi_{rt}^{it} + \phi_{ss}^{it} & \text{if } r \leq s \leq t \\ \phi_{rt}^{it} + \phi_{ts}^{it} & \text{if } r \leq t < s \\ \phi_{rr}^{it} + \phi_{ts}^{it} & \text{otherwise} \end{cases}$$

(P5) *Backlogging cost bounding.* If $i \in I_W$, for each demand point (i, t) and retailer i order in period $s > t$, we have $\phi_{ts}^{it} \leq 2\phi_{ss}^{it}$

In the remainder of the paper, we assume the carrying cost parameters satisfy the five properties introduced above. It is straightforward to check that properties (P1), (P3) and (P4) capture the traditional per-unit holding/penalty cost structure. Moreover if the set of retailers is partitioned into two subsets I_W and I_J as defined by (5.1) and (5.2), properties (P2) and (P5) are also satisfied and therefore the traditional holding/penalty cost structure is a particular case of the metric carrying cost structure. As already discussed, these assumptions cover many other practical cases.

In the remainder of this paper, we call a solution π to our problem a *policy*. Any policy π for the problem is represented by a $(N + 1)$ -uplet $\pi = (\pi_0, \pi_1, \dots, \pi_N)$ where each π_i is the set of pairs (ordering period, quantity) for location i . A policy π is *feasible* for the OWMR problem if the two following conditions are satisfied:

Condition 1. For all $i = 1, \dots, N$ policy π_i orders at least the sum of the demands d_t^i over the entire planning horizon.

Condition 2. Let r and r' be two consecutive ordering periods of the warehouse. Then the quantities ordered by π_0 in period r are sufficient to serve the orders placed by all the retailers in periods $r, \dots, r' - 1$.

Given a policy π for the entire system, we denote $\mathcal{C}(\pi)$ the total cost incurred by π over the planning horizon. This cost can be split into two parts: The total ordering cost denoted $\mathcal{K}(\pi)$ and the total carrying (i.e. holding/backlogging) cost denoted $\Psi(\pi)$ and thus we have:

$$\mathcal{C}(\pi) = \mathcal{K}(\pi) + \Psi(\pi)$$

We conclude this section by pointing out three dominant properties on any feasible policy for the problem with the metric carrying cost structure. First, it is easy to prove that there exists an optimal policy $\pi^{OPT} = (\pi_0^{OPT}, \pi_1^{OPT}, \dots, \pi_N^{OPT})$ such that for all $i = 1, \dots, N$, π_i^{OPT} orders only when its inventory level is nonpositive. In addition, the metric carrying cost structure ensures that there exists an optimal policy in which each demand d_t^i is served from a unique pair of orders $[r, s]$, which is a classical dominant property in most of the inventory models. This allows us to incorporate the amount d_t^i directly into the cost parameters by setting $\psi_{it}^{rs} \equiv \phi_{rs}^{it} d_t^i$, where ϕ_{rs}^{it} is a per-unit cost satisfying properties (P1)-(P5). Since $d_t^i \geq 0$ for all i, t , parameters ψ_{rs}^{it} also satisfy properties (P1)-(P5) and

correspond to the total carrying cost incurred to serve the entire demand d_t^i from the pair of orders $[r, s]$. Therefore in the remainder of the paper we use parameters ψ_{rs}^{it} to prove our approximation results. Finally, it is clear from property (P2) that there exists an optimal policy in which all J -retailers orders are synchronized with a warehouse order. In particular, we shall assume w.l.o.g. that the optimal policies considered in the following sections satisfy this property.

5.3 The split and uncross technique

We now present how to adapt the split and uncross technique introduced in Stauffer et al. (2011) to obtain an approximation algorithm for the OWMR with backlog. Basically, our algorithm works in two steps: First decompose the original problem into several single-echelon subproblems, then solve each of them and recombine the resulting policies into a feasible policy for the OWMR problem. The policy we propose for this extension of the problem works in the same fashion. However, a direct application of their algorithm does not yield a constant guarantee for this general problem, even when the subproblems consider the possibility to backlog demands. As a consequence, we propose a different decomposition based on new virtual single-echelon subproblems and a slightly more complex recombination procedure in the second step. This section details how these two steps are modified and prove that the resulting algorithm has a worst-case performance of three.

5.3.1 Decomposition of the OWMR problem

The first step of our algorithm is to decompose a general OWMR problem with backlog into several single-echelon subproblems. We show that by considering modified cost parameters we can derive a new lower bound for the original problem.

- (\widehat{S}_0) The warehouse is regarded as a single-echelon, multi-item system with backlogging facing for each period t a demand d_t^i for item $i = 1, \dots, N$. A fixed ordering cost K_r^0 is incurred for placing an order in period r . If a demand (i, t) is ordered in period $r \leq t$, it incurs a holding cost of $\frac{1}{3}\psi_{rt}^{it}$ if $i \in I_W$ and $\frac{1}{2}\psi_{rr}^{it}$ if $i \in I_J$. On the other hand if $r > t$, it incurs a backlogging cost of $\frac{1}{3}\psi_{rr}^{it}$ if $i \in I_W$ and $\frac{1}{2}\psi_{rr}^{it}$ if $i \in I_J$.
- (\widehat{S}_i) Retailer i is considered as a single-echelon system with backlog facing demand d_t^i with ordering cost K^i . The carrying cost incurred to order in period s to serve the demand in period t is equal to $\frac{2}{3}\psi_{ss}^{it}$ if $i \in I_W$ and $\frac{1}{2}\psi_{ss}^{it}$ if $i \in I_J$.

Problem (\widehat{S}_0) and problems (\widehat{S}_i) are all equivalent to single-echelon lot-sizing problems with backlog. However, note that the first one corresponds to the multi-item version of this problem. These problems have been widely studied in the literature and it is well-known that finding their optimal solution can be reduced to finding a shortest path in a graph $G_j = (V_j, E_j)$ representing problem (\widehat{S}_j) (see Figure 5.1). For all $j = 0, \dots, N$, $V_j = \{0, \dots, T + 1\}$ is the set of periods, where 0 and $T + 1$ are two artificial periods representing

the beginning and the end of the planning horizon. Given two consecutive orders u and v and $t \in [u, v]$, properties (P2) and (P3) ensure that it is suboptimal to serve demand d_t^i with an order placed earlier than u or later than v . As a consequence for $i = 1, \dots, N$ an edge $(u, v) \in E_i$ represents two consecutive orders in a solution and its length is equal to the minimal possible cost incurred for placing an order in period u and carrying the units necessary to serve demands d_u^i, \dots, d_{v-1}^i by ordering them either in period u or in period v . Therefore its length $l_{u,v}^i$ is equal to:

$$l_{u,v}^i = \begin{cases} \sum_{t=1}^{v-1} \frac{2}{3} \psi_{vv}^{it} & \text{if } u = 0 \text{ and } i \in I_W \\ \sum_{t=1}^{v-1} \frac{1}{2} \psi_{vv}^{it} & \text{if } u = 0 \text{ and } i \in I_J \\ K^i + \sum_{t=u}^{v-1} \frac{2}{3} \min \{ \psi_{uu}^{it}, \psi_{vv}^{it} \} & \text{if } u > 0 \text{ and } i \in I_W \\ K^i + \sum_{t=u}^{v-1} \frac{1}{2} \min \{ \psi_{uu}^{it}, \psi_{vv}^{it} \} & \text{otherwise} \end{cases}$$

The definition of an edge is similar for the warehouse, except that in this case the system has to serve demands of all the items between two consecutive orders in periods u and v . Its length $l_{u,v}^0$ is then set to:

$$l_{u,v}^0 = \begin{cases} \sum_{t=1}^{v-1} \left(\sum_{i \in I_W} \frac{1}{3} \psi_{vt}^{it} + \sum_{i \in I_J} \frac{1}{2} \psi_{vv}^{it} \right) & \text{if } u = 0 \\ K_u^0 + \sum_{t=u}^{v-1} \left(\sum_{i \in I_W} \frac{1}{3} \min \{ \psi_{ut}^{it}, \psi_{vv}^{it} \} + \sum_{i \in I_J} \frac{1}{2} \min \{ \psi_{uu}^{it}, \psi_{vv}^{it} \} \right) & \text{otherwise} \end{cases}$$

Note that these lengths are computed in time $O(T^2)$ for the single-item problem and $O(NT^2)$ for the multi-item case. Moreover, from the nonnegativity of parameters ρ_s^{it} and ω_r^{it} , they are nonnegative and therefore one can easily find a shortest path from node 0 to $T + 1$ in graph G_j using classical shortest path dynamic algorithms. For instance Dijkstra's algorithm solves such problems to optimality with a time complexity of $O(NT^2)$ for (\widehat{S}_0) and $O(T^2)$ for each (\widehat{S}_i) , $i = 1, \dots, N$.

We denote $\hat{\pi}_j$ a feasible policy for system (\widehat{S}_j) . Note that if $j = 0$, $\hat{\pi}_0$ is a set of $(N + 1)$ -uplet (r, \mathbf{q}_r) , where $r \in \{1, \dots, T\}$ and \mathbf{q}_r is the N -uplet of quantities ordered for each item in period r . For all j , we denote $\widehat{\mathcal{E}}_j(\hat{\pi}_j)$ the total cost incurred by policy $\hat{\pi}_j$ in system (\widehat{S}_j) . In the same fashion as the one used for the global cost, it can be split between two components:

$$\widehat{\mathcal{E}}_j(\hat{\pi}_j) = \mathcal{K}_j(\hat{\pi}_j) + \widehat{\Psi}_j(\hat{\pi}_j)$$

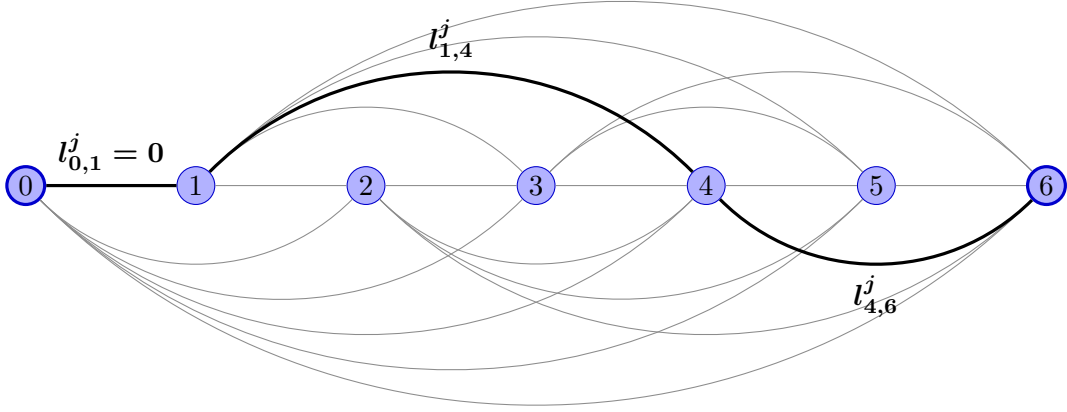


Figure 5.1: An example of graph modeling problem (\widehat{S}_j) with $T = 5$. The bold path corresponds to the solution in which the policy orders in periods 1 and 4, for a total cost of $l_{0,1}^j + l_{1,4}^j + l_{4,6}^j$.

where $\mathcal{K}_j(\hat{\pi}_j)$ (resp. $\widehat{\Psi}_j(\hat{\pi}_j)$) denotes the total ordering (resp. carrying) cost incurred by policy $\hat{\pi}_j$ in system (\widehat{S}_j) .

To conclude this section, we introduce a new lower bound for the optimal solution of the OWMR problem with backlog using the decomposition discussed above. Assume that we compute an optimal policy $\hat{\pi}_j^*$ to problem (\widehat{S}_j) for all j . The next lemma uses the costs incurred by these policies in the single-echelon systems (\widehat{S}_j) to define a lower bound on the cost of any policy for the original problem. In particular, the sum of these costs is a lower bound on its optimal cost.

Lemma 5.3.1. *Consider a OWMR problem with backlog and a metric carrying cost structure. Let \mathcal{C}^* be the cost of an optimal policy for this problem and let $\hat{\pi}_0^*, \hat{\pi}_1^*, \dots, \hat{\pi}_N^*$ be optimal single-echelon policies for problems $(\widehat{S}_0), (\widehat{S}_1), \dots, (\widehat{S}_N)$ as defined in the decomposition. Then the following inequality holds:*

$$\mathcal{C}^* \geq \sum_{j=0}^N \widehat{\mathcal{C}}_j(\hat{\pi}_j^*)$$

Proof. The proof of this lemma is detailed in Appendix C.1. Basically, we start from an optimal policy $\pi^{\text{OPT}} = (\pi_0^{\text{OPT}}, \pi_1^{\text{OPT}}, \dots, \pi_N^{\text{OPT}})$ and decompose it in $N + 1$ single-echelon policies $\bar{\pi}_0, \bar{\pi}_1, \dots, \bar{\pi}_N$, one for each system (\widehat{S}_j) . We then show that the sum of the costs incurred by policies $\bar{\pi}_j$ in systems (\widehat{S}_j) is lower than the optimal cost \mathcal{C}^* for the original problem. The proof then follows from the optimality of policies $\hat{\pi}_j^*$. \square

5.3.2 The algorithm

In this section, we modify the uncrossing method and use the decomposition $(\widehat{S}_0), (\widehat{S}_1), \dots, (\widehat{S}_N)$ to build a feasible solution for the original problem. In addition we prove that when the solution is built upon the optimal single-echelon policies for these independent systems, the policy obtained has a cost of at most three times the optimal cost.

Let $\hat{\pi}_0, \dots, \hat{\pi}_N$ be feasible policies for problems $(\widehat{S}_0), \dots, (\widehat{S}_N)$. We denote $R = \{r_1, \dots, r_w\}$ the set of periods when the warehouse orders according to policy $\hat{\pi}_0$. For convenience we add to R an artificial period $r_{w+1} = T + 1$ corresponding to the end of the planning horizon, with no ordering cost. Given a set R – induced by a policy π_0 – and a period $s = 1, \dots, T$, let s^+ and s^- be the first period of R after s and the last period of R prior to s , respectively:

$$s^+ = \min \{r \in R : r \geq s\} \quad (5.4)$$

$$s^- = \max \{r \in R : r \leq s\} \quad (5.5)$$

In what follows, the set R used to define s^+ and s^- will be clear from the context.

For a policy at a given location, we call *ordering interval* a pair s, s' of successive ordering periods (for convenience we assume $s' = T + 1$ when s is the last order) i.e. two orders are placed in periods s and s' and no order is placed in $\{s + 1, \dots, s' - 1\}$. We say that a period t is in the ordering interval s, s' if $t \in \{s, \dots, s' - 1\}$. A straightforward way to obtain a feasible policy from the independent policies $\hat{\pi}_j$ would be to keep the ordering periods of each policy and simply adjust the quantities ordered such that it minimizes the holding and backlogging costs incurred. However by doing so, it is impossible to control the overcost induced by this operation: In particular, the main problematic situation we want to avoid arises when demand d_t^i is ordered in $\hat{\pi}_i$ *before* it is ordered in $\hat{\pi}_0$. We call such a situation a *crossing order*, as defined below:

Definition 5.3.1. An ordering interval s, s' at retailer i *crosses* the orders at the warehouse if there exists an ordering interval r, r' at the warehouse such that $r < s < r' < s'$. A policy is said to be *uncrossing* if none of its ordering intervals crosses the orders at the warehouse.

For simplicity if ordering interval s, s' crosses, we say that order s is crossing. Due to the definition, an order in period s at a retailer is not crossing either if $s \in R$ (s coincides with an order at the warehouse, i.e. is synchronized), or if the ordering interval s, s' is contained in an ordering interval of the warehouse. Figure 5.2 illustrates the definition of crossing orders.

Notice that if an ordering interval s, s' of policy $\hat{\pi}_i$ crosses an ordering interval r, r' of $\hat{\pi}_0$, one can easily *uncross* this order by simply adding an order of retailer i in period r' . As its name suggests, the *uncrossing algorithm* presented below is based on this idea combined with an adjustment of the quantities ordered in each location. This algorithm is similar to

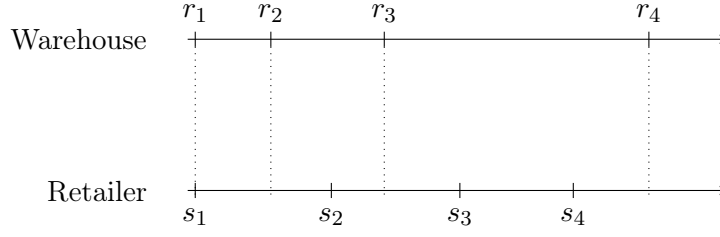


Figure 5.2: An example of a crossing order: Orders s_1 and s_3 do not cross any ordering interval of the warehouse – because $s_1 \in R$ and $s_3 \in (r_3, r_4)$ – while order s_2 crosses in r_3 .

the uncrossing algorithm introduced in Stauffer et al. (2011) in the first two steps. However the quantities adjustments made in this version are more involved than the ones presented in their work.

Uncrossing Algorithm

Step 1. For each $i = 1, \dots, N$, uncross every crossing order of $\hat{\pi}_i$: Consider the first ordering interval s, s' of $\hat{\pi}_i$ crossing an ordering interval r, r' of $\hat{\pi}_0$ ($r < s < r' < s'$). Uncross s, s' by placing an additional order of retailer i in period r' . Repeat this step until no ordering interval of $\hat{\pi}_i$ crosses.

Step 2. For all $i \in I_J$, we modify the ordering periods of $\hat{\pi}_i$ to keep only synchronized orders: For each ordering interval r, r' of policy $\hat{\pi}_0$ at the warehouse, we cancel all the orders of retailer i in $[r, r')$ and replace them (if any) by a unique order in period r . In other words we shift to the left and collapse together all the orders of the retailer inside an ordering interval of the warehouse.

Step 3. For all $i = 1, \dots, N$ and $t = 1, \dots, T$, let r and s be the periods when $\hat{\pi}_0$ and $\hat{\pi}_i$ order to serve demand d_t^i , respectively. The final policy then uses the pair of orders $[r^{it}, s^{it}]$ to serve demand d_t^i , where

$$[r^{it}, s^{it}] = \begin{cases} [s^-, s] & \text{if } i \in I_W \text{ and } r \leq s \\ [s^-, s^-] & \text{if } i \in I_J \text{ and } r \leq s \\ [s^+, s^+] & \text{otherwise} \end{cases} \quad (5.6)$$

Note that Step 1 and 2 are executed in time $O(NT)$ while Step 3 has a complexity of $O(NT^2)$. Therefore, the final complexity of the uncrossing algorithm (including the resolution of the single-echelon subproblems) is $O(NT^2)$, which is remarkably low in comparison to other existing techniques.

We now focus on the total cost incurred by π^u , the final policy built by the uncrossing algorithm. For a policy π and two periods $r \leq s$, we say that π *opens* the pair of orders $[r, s]$ for retailer i if π places a warehouse order in period r and a retailer i order in period s . The following lemma bounds the overcost incurred by the uncrossing algorithm compared to the costs of the independent single-echelon policies $\hat{\pi}_j$.

Lemme 5.3.2. *The uncrossing algorithm applied to single-echelon policies $\hat{\pi}_0, \dots, \hat{\pi}_N$ produces an uncrossed feasible solution π^u for the OWMR problem. The total cost incurred by the resulting policy satisfies:*

$$\mathcal{C}(\pi^u) \leq 3 \sum_{j=0}^N \widehat{\Psi}_j(\hat{\pi}_j) + \mathcal{K}_0(\hat{\pi}_0) + 2 \sum_{i=1}^N \mathcal{K}_i(\hat{\pi}_i)$$

Moreover in the special case of the JRP (i.e. if $I_W = \emptyset$), we have:

$$\mathcal{C}(\pi^u) \leq 2 \sum_{j=0}^N \widehat{\Psi}_j(\hat{\pi}_j) + \mathcal{K}_0(\hat{\pi}_0) + 2 \sum_{i=1}^N \mathcal{K}_i(\hat{\pi}_i)$$

Proof. Proof See Appendix C.2. □

We now show that when the split and uncross algorithm builds a policy π^u upon optimal policies $\hat{\pi}_0^*, \hat{\pi}_1^*, \dots, \hat{\pi}_N^*$ for single-echelon systems $(\widehat{S}_0), (\widehat{S}_1), \dots, (\widehat{S}_N)$, $\mathcal{C}(\pi^u)$ is at most three times the optimal cost \mathcal{C}^* . More precisely, the 3-approximation algorithm works as follows:

Step 1 Compute optimal policies $\hat{\pi}_0^*, \hat{\pi}_1^*, \dots, \hat{\pi}_N^*$ for problems $(\widehat{S}_0), (\widehat{S}_1), \dots, (\widehat{S}_N)$ independently.

Step 2 Apply the uncrossing algorithm to policies $\hat{\pi}_0^*, \dots, \hat{\pi}_N^*$.

Let π^{u*} be the resulting policy: From Lemma 5.3.1 and 5.3.2, we have:

$$\begin{aligned} \mathcal{C}(\pi^{u*}) &\leq 3 \sum_{j=0}^N \widehat{\Psi}_j(\hat{\pi}_j^*) + \mathcal{K}_0(\hat{\pi}_0^*) + 2 \sum_{i=1}^N \mathcal{K}_i(\hat{\pi}_i^*) \\ &\leq 3 \sum_{j=0}^N \mathcal{C}_j^*(\hat{\pi}_j^*) \\ &\leq 3\mathcal{C}^* \end{aligned}$$

We conclude that the following theorem holds:

Theorem 5.3.3. *The policy π^{u*} obtained after applying the split and uncross algorithm to policies $\hat{\pi}_0^*, \hat{\pi}_1^*, \dots, \hat{\pi}_N^*$ has a performance guarantee of three for the OWMR problem with backlog.*

Notice that in the special case of the JRP, this inequality can be refined to reach a

performance guarantee of two:

$$\begin{aligned} \mathcal{E}(\pi^{u*}) &\leq 2 \sum_{j=0}^N \widehat{\Psi}_j(\hat{\pi}_j^*) + \mathcal{K}_0(\hat{\pi}_0^*) + 2 \sum_{i=1}^N \mathcal{K}_i(\hat{\pi}_i^*) \\ &\leq 2 \sum_{j=0}^N \widehat{\mathcal{E}}_j(\hat{\pi}_j^*) \\ &\leq 2\mathcal{E}^* \end{aligned}$$

Theorem 5.3.4. *The policy π^{u*} obtained after applying the split and uncross algorithm to policies $\hat{\pi}_0^*, \hat{\pi}_1^*, \dots, \hat{\pi}_N^*$ has a performance guarantee of two for the JRP with backlog.*

5.4 A lost-sales model

While in the backorder model we assume that customers are willing to wait for their demands to be fulfilled, another alternative is to consider a model in which customers are impatient and every unmet demand is simply lost. Most of the lost-sales literature focuses on cost structures in which it is dominant to serve demands on a first-come first-served basis. Although this is clearly the case for the linear cost setting, our model uses a general metric cost structure derived from the one introduced in §5.2, in which the latter property is not necessarily dominant.

In this section, we introduce a modified decomposition for the problem and exhibits a lower bound for the OWMR problem with lost-sales. We then show how to adapt the split and uncross technique to this problem and build a 2-approximation from the subproblems of the decomposition. This last result generalizes the one obtained in Stauffer et al. (2011) to a lost-sales model and presents an alternative proof of the main theorem that does not rely on the IP formulation of the problem.

5.4.1 Cost structure

We use the general cost structure introduced in §5.2 to modelize the case of lost-sales problem and simply consider it as a special case of the backorder version, where parameters ϕ_{rs}^{it} are constant when demand d_{it} is not satisfied immediately. We have $\phi_{rs}^{it} = \phi^{it}$ for all $s > t$ and the following inequality holds for parameters ψ_{rs}^{it} :

$$\psi_{rs}^{it} = \psi^{it} = \phi^{it} d_{it} \quad \text{for all } s > t \quad (5.7)$$

Thus the parameter ψ^{it} corresponds to the total lost-sales cost associated to demand point (i, t) . In other words, if the system serves demand d_{it} with the pair of orders $[r, s]$ where $s \leq t$, it incurs a cost ψ_{rs}^{it} while if it does not serve this demand, the cost for losing the sale is equal to ψ^{it} . An other way to see the lost-sales penalty cost is to consider that there exists an alternative stock from which we can order units to satisfy an unmet

demand, for a total ordering cost of ψ^{it} . Note that in our model one can speculate on the penalty cost and deliberately choose to order from this alternative stock if the penalty cost is attractive compared to the cost of serving the demand using the inventory on hand.

Constraint (5.7) ensures that properties (P1)-(P4) are satisfied when $s > t$. On the other hand, property (P5) is not necessary in the lost-sales case and therefore we only assume that parameters ψ_{rs}^{it} satisfy properties (P1)-(P4) and equation (5.7) in the remainder of this section.

5.4.2 Decomposition in single-echelon systems

In the same fashion as the backorder model, we decompose the system into $N + 1$ independent single-echelon problems with lost-sales. However in this version, the holding costs and lost-sales penalty costs are halved instead of divided by three. Similarly to the JRP case, increasing the fraction of the real costs used in the subproblems is crucial to obtain a performance guarantee of two.

The decomposition for the OWMR with lost-sales is defined as follows:

- (\tilde{S}_0) The warehouse is regarded as a single-echelon, multi-item system with lost-sales, facing for each period t a demand d_{it} for item i . A fixed ordering cost K_r^0 is incurred for placing an order in period r . The holding cost incurred if demand (i, t) is ordered in period r is equal to $\frac{1}{2}\psi_{rr}^{it}$ if $i \in I_J$ and $\frac{1}{2}\psi_{rt}^{it}$ if $i \in I_W$. If the demand is lost, the system incurs a lost-sales cost equal to $\frac{1}{2}\psi^{it}$.
- (\tilde{S}_i) Retailer i is considered as a single-echelon system with lost-sales, facing demand d_{it} with ordering cost K^i . The holding cost incurred to order to serve demand (i, t) with an order in period s is equal to $\frac{1}{2}\psi_{ss}^{it}$, while the per-unit cost if the demand is lost is equal to $\frac{1}{2}\psi^{it}$.

Similarly to the previous sections, we define $\tilde{\mathcal{E}}_i(\tilde{\pi}_i)$ (resp. $\tilde{\Psi}_i(\tilde{\pi}_i)$) as the total (resp. carrying) cost incurred by $\tilde{\pi}_i$ in system (\tilde{S}_i) and we have for all $i = 0, \dots, N$:

$$\tilde{\mathcal{E}}_i(\tilde{\pi}_i) = \mathcal{K}_i(\tilde{\pi}_i) + \tilde{\Psi}_i(\tilde{\pi}_i)$$

We first discuss how to solve these single-echelon problems. Similarly to the systems with backlog, this can be done using classical shortest path algorithms in the same type of graph (see Figure 5.1). We define a graph G_j corresponding to each single-echelon problem (\tilde{S}_j). The main difference comes from the computation of the length of each edge (u, v) , that we detail below.

We first introduce the following virtual cost parameters:

$$\tilde{\omega}_r^{it} = \begin{cases} \min \{ \psi_{rr}^{it}, \psi^{it} \} & \text{if } i \in I_J \\ \min \{ \psi_{rt}^{it}, \psi^{it} \} & \text{otherwise} \end{cases} \quad (5.8)$$

$$\tilde{\rho}_s^{it} = \min \{ \psi_{ss}^{it}, \psi^{it} \} \quad (5.9)$$

These are the virtual minimum cost between satisfying or loosing demand d_{it} when the last order is placed in period r or s . One can precompute these parameters in time $O(T^2)$ for one retailer and $O(NT^2)$ for the warehouse. We then define the length of each edge (u, v) in graphs G_i and G_0 as follows:

$$\forall i = 1, \dots, N, \quad \tilde{l}_{u,v}^i = \begin{cases} \sum_{t=1}^{v-1} \frac{1}{2} \psi^{it} & \text{if } u = 0 \\ K^i + \sum_{t=u}^{v-1} \frac{1}{2} \tilde{\rho}_u^{it} & \text{otherwise} \end{cases}$$

$$\tilde{l}_{u,v}^0 = \begin{cases} \sum_{i=1}^N \sum_{t=1}^{v-1} \frac{1}{2} \psi^{it} & \text{if } u = 0 \\ K_u^0 + \sum_{i=1}^N \sum_{t=u}^{v-1} \frac{1}{2} \tilde{\omega}_u^{it} & \text{otherwise} \end{cases}$$

These lengths are computed in time $O(NT^2)$. As for the backorder version, the time complexity to compute the $N + 1$ optimal policies to problems (\tilde{S}_j) is equal to $O(NT^2)$.

We now focus on the lower bound resulting from this decomposition. The following lemma derives from Lemma 5.3.1:

Lemme 5.4.1. *Consider a OWMR problem with lost-sales. Let \mathcal{E}^* be the cost of an optimal policy for this problem and let $\tilde{\pi}_0^*, \tilde{\pi}_1^*, \dots, \tilde{\pi}_N^*$ be optimal single-echelon policies for problems $(\tilde{S}_0), (\tilde{S}_1), \dots, (\tilde{S}_N)$ as defined in the decomposition. Then the following inequality holds:*

$$\mathcal{E}^* \geq \sum_{j=0}^N \mathcal{E}_j(\tilde{\pi}_j^*)$$

Proof. Proof The proof is similar to the one of Lemma 5.3.1 (see Appendix C.3). \square

5.4.3 The algorithm

To conclude this section, we show how to adapt the split and uncross technique to the lost-sales case and build a 2-approximation algorithm. Let $\tilde{\pi}_0, \dots, \tilde{\pi}_N$ be feasible policies for problems $(\tilde{S}_0), \dots, (\tilde{S}_N)$ and let $R = \{r_1, \dots, r_w, r_{w+1}\}$ the set of periods when policy $\tilde{\pi}_0$ places order, where again $r_{w+1} = T + 1$ is an artificial period corresponding to the end of the planning horizon. The algorithm is very similar to the one presented in §5.3.2: The two first steps are untouched, while the last one is modified as follows:

Step 3. For all $i = 1, \dots, N$ and $t = 1, \dots, T$, if either $\tilde{\pi}_0$ or $\tilde{\pi}_i$ does not satisfy d_{it} , then the demand is lost. Otherwise, let r and s be the periods when $\tilde{\pi}_0$ and $\tilde{\pi}_i$ order demand d_{it} , respectively. The final policy then uses the pair of orders $[r^{it}, s^{it}]$ to

serve demand d_{it} , where

$$\lceil r^{it}, s^{it} \rceil = \begin{cases} [s^-, s] & \text{if } i \in I_W \text{ and } r \leq s \\ [s^-, s^-] & \text{if } i \in I_J \text{ and } r \leq s \\ [s^+, s^+] & \text{otherwise} \end{cases} \quad (5.10)$$

In other words, every demand that is lost in *either* π_0 or π_i is also lost in the final solution. Otherwise, if *both* π_0 and π_i serve the demand, it is satisfied by pi^u in the final policy using the cheapest possible pair of orders.

It is straightforward that the lower bound from Lemma 5.3.2 holds in this special case of the problem. However, since we only halve the holding and penalty costs in the subproblems instead of dividing them by three, we can improve upon this result to get the following lower bound:

Lemma 5.4.2. *The uncrossing algorithm applied to single-echelon policies $\tilde{\pi}_0, \dots, \tilde{\pi}_N$ produces an uncrossed feasible solution $\tilde{\pi}^u$ for the OWMR problem with lost-sales. The total cost incurred by the resulting policy satisfies:*

$$\mathcal{C}(\tilde{\pi}^u) \leq \mathcal{K}_0(\tilde{\pi}_0) + 2\tilde{\Psi}_0(\tilde{\pi}_0) + 2 \sum_{i=1}^N \tilde{\mathcal{C}}_i(\tilde{\pi}_i)$$

Proof. Proof See Appendix C.4. □

Finally, assume the solution built by the split and uncross algorithm when applied to the optimal single-echelon policies $\tilde{\pi}_0^*, \tilde{\pi}_1^*, \dots, \tilde{\pi}_N^*$ for problems $(\tilde{S}_0), (\tilde{S}_1), \dots, (\tilde{S}_N)$. Let $\tilde{\pi}^{u*}$ be the resulting policy: From Lemma 5.4.1 and 5.4.2, we have:

$$\begin{aligned} \mathcal{C}(\tilde{\pi}^{u*}) &\leq \mathcal{K}_0(\tilde{\pi}_0^*) + 2\tilde{\Psi}_0(\tilde{\pi}_0^*) + 2 \sum_{i=1}^N \tilde{\mathcal{C}}_i(\tilde{\pi}_i^*) \\ &\leq 2 \sum_{j=0}^N \tilde{\mathcal{C}}_j(\tilde{\pi}_j^*) \\ &\leq 2\mathcal{C}^* \end{aligned}$$

and the following theorem states the approximation ratio of the technique:

Theorem 5.4.3. *The policy $\tilde{\pi}^{u*}$ obtained after applying the split and uncross algorithm to policies $\tilde{\pi}_0^*, \tilde{\pi}_1^*, \dots, \tilde{\pi}_N^*$ has a performance guarantee of two for the OWMR problem with lost-sales.*

5.5 Conclusion and perspectives

Although the literature on discrete deterministic divergent inventory systems is already substantial, almost all the existing works assume that demands must be satisfied on time. Our work propose a first constant approximation when backlog and/or lost-sales are allowed by expanding the scope of the split and uncross technique proposed by Stauffer et al. (2011). We derive new, intuitive lower bounds for the problems considered and prove that our algorithms have a constant performance guarantee under a general cost structure that can capture many practical situations. In addition we show that the performance guarantee can be improved in special cases of our model, namely the JRP and a lost-sales model.

Although these results fill a gap in the literature on deterministic inventory theory, there remains some room for improvement. One obvious way for future research is to try to improve the performance guarantee of three in the backlogging case. One could also want to generalize the assumptions on the cost structure by eliminating the backlogging cost bounding constraint or considering nonlinear carrying costs. Finally, it would be of great interest to see how these algorithms perform in practice using some industrial data.

Chapitre 6

Conclusion

Contributions de cette thèse

Au cours de ces travaux, nous avons abordé plusieurs modèles de gestion de stock déterministes pour lesquels il n'existe pas de méthode de résolution exacte efficace dans la littérature. Pour chacun des problèmes étudiés, nous nous sommes efforcés de proposer des techniques simples et originales, qui garantissent la qualité de la solution obtenue. Cette approche, axée sur la recherche d'algorithmes d'approximation, nous a permis de dégager plusieurs résultats théoriques intéressants pour des problèmes de la littérature réputés difficiles. En outre, nous avons également pu extraire de chacun des modèles considérés un certain nombre de propriétés structurelles fondamentales qui ont par la suite permis d'étendre nos méthodes de résolution à plusieurs variantes plus générales, sans pour autant modifier profondément les mécanismes employés.

Dans le domaine continu, nous nous sommes concentrés sur un modèle à un seul échelon dont la demande et les paramètres de coût varient dans le temps. Ces hypothèses empêchent généralement de calculer une politique de commande optimale, sauf pour certains cas particuliers de fonctions de demandes (voir état de l'art du chapitre 3). Pour traiter le cas de demandes quelconques, nous avons adapté des algorithmes déjà utilisés sur des modèles discrets et proposé une méthode d'équilibrage de coûts pour construire une solution avec une garantie de performance égale à 2. Cette technique s'est avérée suffisamment générique pour pouvoir par la suite être étendue à plusieurs autres modèles classiques tels que les systèmes de production ou des systèmes avec mise en attente. Enfin, nous avons démontré que lorsque la demande répond à certains critères généraux, une modification simple de notre algorithme permet d'améliorer le facteur d'approximation à 1.5.

Une telle approche algorithmique pour des problèmes à temps continu est peu commune dans la littérature, qui se concentre majoritairement sur des études analytiques permettant de dégager les propriétés dominantes des politiques optimales. Nos travaux s'inscrivent donc dans une démarche originale qui vise avant tout à proposer des politiques simples et intuitives permettant d'aborder une large classe de problèmes difficiles.

Dans un second temps, nous nous sommes intéressés à des problèmes à temps discret complexes, en se consacrant principalement à l'étude de modèles multi-échelle divergents. En particulier, nos travaux se sont concentrés sur un problème combinatoire NP-difficile classique de la littérature, appelé OWMR, pour lequel il existe peu d'algorithmes de résolution avec garantie de performance. Nous avons abordé le problème sous un angle original en proposant une décomposition naturelle de celui-ci en sous-problèmes à un seul échelon. Cette approche nous a notamment permis d'établir une nouvelle borne inférieure sur le coût d'une solution optimale. Nous avons également développé un algorithme de recombinaison pour reconstruire facilement une solution au problème initial à partir de sa décomposition. On peut remarquer que la procédure finale obtenue est purement combinatoire, tandis que la majorité des méthodes de la littérature (Levi et al. (2008c), Chan et al. (2000)) nécessitent la résolution à l'optimal d'un programme linéaire. De plus, celle-ci s'exécute en temps linéaire ou quadratique (selon la structure du problème) tout en garantissant un ratio d'approximation de 2, ce qui rend la perspective de son utilisation pratique attrayante. Les premiers résultats numériques obtenus confirment l'efficacité théorique de la méthode, en particulier dans le cas de systèmes comprenant de nombreux détaillants.

Au-delà du résultat d'approximation pour ce problème spécifique, notre approche s'est avérée suffisamment générique pour pouvoir s'adapter facilement à des structures de coût plus complexes, pour lesquelles il n'existe aucune technique d'approximation connue. Ainsi, nous avons prouvé que la garantie de performance de notre algorithme est conservée lorsque l'on applique celui-ci à des problèmes dont les coûts de possession sont non linéaires, ou dont les coûts de commande sont de type FTL/LTL. En particulier, l'utilisation de ces derniers est pertinente dans de nombreux cas pratiques, par exemple lorsque les marchandises sont acheminées jusqu'à un entrepôt au moyen de véhicules dont la capacité est limitée. Enfin, nous avons proposé une modification de notre méthode pour traiter des systèmes qui incluent la possibilité de mettre en attente des clients ou qui autorisent la perte de certaines ventes. Ces derniers résultats constituent à notre connaissance les premières approximations proposées dans la littérature pour ces deux modèles. La grande flexibilité de notre méthode nous encourage à penser qu'elle pourrait être adaptée à d'autres problèmes multi-échelle de gestion de stock, pour lesquels on ne connaît pas d'approximation.

De manière générale, les nouvelles techniques que nous avons introduites se démarquent des algorithmes existants par leur simplicité, tant dans les mécanismes mis en jeu que dans l'analyse de leurs performances. En particulier, nos résultats ne nécessitent pas une connaissance pointue d'outils d'optimisation complexes, ce qui rend leur compréhension accessible à un large panel de professionnels de l'industrie et en fait des outils de choix pour une application pratique.

Perspectives

En abordant des problèmes classiques sous un angle nouveau, les travaux présentés dans cette thèse ouvrent la voie à de nombreuses autres pistes de recherche. Si les contributions théoriques de ces travaux ont déjà été analysées, nous pensons également que le champ d'application de ces nouvelles méthodes d'approximation peut encore s'étendre à d'autres modèles de gestion des stocks, voire à d'autres problèmes d'optimisation issus de domaines différents. Parmi les nombreuses perspectives de recherche, trois axes nous semblent se dégager par leur potentiel à apporter des résultats significatifs.

Extensions et variantes des modèles étudiés. Parmi toutes les pistes envisagées, celles que nous explorons aujourd'hui se positionnent majoritairement sur des problèmes discrets. La plus concrète d'entre elles vise à étendre la procédure présentée au chapitre 4 à des systèmes divergents de taille plus importante. Ainsi, nous nous sommes intéressés à un modèle de réseau de distribution arborescent dont la racine se compose d'un entrepôt central qui passe commande auprès d'un fournisseur extérieur et approvisionne plusieurs autres points de stockage. À la différence du problème OWMR, ces derniers peuvent à leur tour servir de plateformes de distribution à la racine de sous-systèmes arborescents. Un tel réseau peut alors être divisé en $m + 1$ étages, où l'étage $l = 0, \dots, m$ est défini comme l'ensemble des entrepôts séparés du fournisseur extérieur par l points de stockages intermédiaires. L'étage 0 ne contient que l'entrepôt central, l'étage 1 comprend l'ensemble de ses successeurs directs dans la chaîne, et ainsi de suite jusqu'à l'étage m , constitué des détaillants en contact avec la demande finale. On peut constater que le problème OWMR étudié aux chapitres 4 et 5 est un cas particulier de celui-ci, composé de deux étages, où l'étage 1 est constitué de l'ensemble des détaillants. Notre objectif sera de généraliser notre approche et notre analyse pour ce problème à des systèmes comportant un nombre arbitraire d'étages. Les premiers résultats que nous avons obtenus pour un système à $m + 1$ étages ont abouti à un algorithme dont la garantie de performance, égale à m , dépend de la taille de l'instance considérée. Une prochaine étape sera de réduire ce facteur d'approximation afin d'obtenir une garantie constante. Ce travail, en cours de rédaction, pourrait constituer un premier résultat pour l'étude de réseaux logistiques à grande échelle.

Il existe également un certain nombre de variantes du problème OWMR classique pour lesquelles un résultat d'approximation constituerait une avancée significative. Par exemple, nous étudions la possibilité d'incorporer à ce modèle une capacité de commande limitée pour les détaillants. Une telle contrainte est courante dans l'industrie mais son intégration au problème complique nettement la résolution de celui-ci. La perspective d'une méthode de résolution simple et performante est donc particulièrement attrayante pour traiter ces situations complexes.

En plus des réseaux divergents déjà étudiés, nous considérons également la possibilité d'adapter notre procédure à d'autres familles de systèmes multi-échelon, tels que les mo-

dèles convergents de type assemblage. Il existe déjà dans la littérature un certain nombre de méthodes qui s'appliquent à ces deux classes de problèmes (voir par exemple l'algorithme primal-dual proposé par Levi et al. (2006)). Nous pensons que notre technique d'approximation appartient à cette catégorie d'algorithmes "modulables", dont le principe peut s'étendre à plusieurs types de systèmes. Un premier pas pour valider l'aspect générique de la méthode serait donc de l'adapter au problème d'assemblage présenté dans la partie 2.2.2, avant d'envisager son extension à d'autres problèmes NP-difficiles de la littérature.

Dans le cas continu, nous souhaitons développer des politiques utilisant le même concept pour d'autres variantes classiques du modèle à un échelon, comme les modèles avec capacité de commande ou de stockage limitée. Une autre direction de recherche intéressante consisterait à étendre l'idée d'équilibrage des coûts à des modèles multi-échelon pour lesquels on ne connaît pas de politique optimale. En particulier, l'application d'une méthode de ce type à des systèmes série imbriqués pourrait amener au développement d'un premier algorithme d'approximation lorsque la demande varie avec le temps.

Étude numérique. L'efficacité et la performance des algorithmes présentés dans cette thèse ont été prouvées d'un point de vue théorique, mais une expérimentation numérique poussée paraît néanmoins nécessaire afin de les valider en pratique, en particulier dans le cas du problème OWMR. Les essais numériques proposés dans la partie 4.6 posent les bases de ce travail et les premiers résultats obtenus semblent confirmer le bien-fondé de notre approche. Nous souhaitons par conséquent pousser l'expérimentation plus en avant, avec pour objectif immédiat de déterminer si nos algorithmes offrent une alternative crédible aux techniques déjà existantes dans la littérature. Lors de cette mise en application, nous nous concentrerons en priorité sur deux caractéristiques principales : la qualité des solutions fournies, d'une part, et la rapidité de leur exécution, d'autre part. Un bon compromis entre ces deux critères achèverait de nous convaincre de leur intérêt pratique.

Dans le cadre de cette étude, il serait également intéressant de développer un jeu d'instances de test varié qui puisse servir de référence pour l'expérimentation des méthodes de résolution, comme il en existe déjà pour le problème du voyageur de commerce (TSP). Une telle bibliothèque de test pourrait mettre en avant les forces et les faiblesses de chacun des algorithmes testés, permettant d'isoler certains cas critiques et de dégager des pistes d'amélioration possibles. Enfin, à plus long terme, une application industrielle de nos méthodes pourra être envisagée si l'étude numérique révèle de bons résultats.

Études de modèles stochastiques. L'un des aspects majeurs de la gestion de stock n'a pas été abordé dans cette thèse. En effet, les modèles déterministes que nous avons présenté supposent que tous les paramètres du système sont parfaitement connus, ce qui n'est en général pas le cas en pratique. De nombreux travaux de recherche s'orientent ainsi vers des problèmes stochastiques afin de prendre en compte certains phénomènes aléatoires

dans leur résolution, tels que les fluctuations des demandes ou les pannes du système.

Dans ce cadre, la littérature propose déjà des algorithmes d'approximation stochastiques dont les résultats laissent entrevoir un large champ d'application potentiel. Une piste de recherche naturelle sera donc d'explorer ce nouveau domaine, en transposant notamment les problèmes que nous avons déjà étudiés dans un cadre stochastique. Nous pourrions par exemple nous appuyer sur une récente série d'articles qui a introduit des politiques d'approximation stochastiques simples basée sur l'équilibrage des coûts encourus par un système (voir Levi et al. (2007), Levi et al. (2008a), Levi et al. (2008d)). L'approche développée dans ces travaux a déjà permis de construire des approximations pour des modèles à un échelon dont les demandes aléatoires peuvent être corrélées. Ces résultats nous semblent prometteurs et nous explorons actuellement la possibilité d'adapter des techniques semblables au JRP stochastique.

Les systèmes industriels qui inspirent les modèles de gestion de stocks et les problématiques qu'ils soulèvent n'ont jamais cessé de se complexifier. Ce phénomène se répercute sur les méthodes utilisées pour les traiter, qui se spécialisent et font souvent intervenir des outils d'optimisation sophistiqués afin de maîtriser le flux des produits à travers la chaîne logistique. La taille considérable des réseaux actuels a également fait du temps de calcul des solutions l'une des préoccupations majeures des industriels, dont les décisions doivent répondre rapidement et efficacement aux situations rencontrées. Dans ce contexte, une approche par les algorithmes d'approximation nous a permis de proposer de nouvelles solutions simples et performantes pour des problèmes complexes de gestion de stocks. Plus généralement, cette méthodologie présente le double avantage de dégager des propriétés théoriques sur les problèmes étudiés tout en proposant des outils pratiques de résolution, dont le temps de calcul reste raisonnable. Au vu des résultats obtenus dans cette thèse et de ceux existants dans la littérature, il semble donc pertinent d'explorer plus en avant cet axe de recherche à l'avenir.

Annexe A

Proofs of chapter 3

A.1 Proof of Theorem 3.3.3

For ease of understanding, we assume the initial inventory is zero ($x_0 = 0$) and that $\lambda(0) > 0$. Therefore the first order is placed at time 0 by any feasible policy. Now assume the current order is placed at time $s \geq 0$ and that $H(s, T) \geq K$. The BL algorithm defines the next order time as $t = \max \{u \leq T : H(s, u) \leq K\}$. Numerically, using the bisection method with a precision δ , the next order time t' is found in $[t - \delta, t + \delta]$. We choose the precision δ of the bisection method to be :

$$\delta = \frac{K}{\hat{\lambda} \hat{h} T} \frac{\varepsilon}{1 + \varepsilon} = \frac{T}{\Omega}$$

In addition we require that two consecutive order times are separated at least by δ , that is $t' \geq s + \delta$. Therefore the BL policy orders at most T/δ times on the planning horizon. If the computational time to evaluate $H(s, t)$ is $\mathcal{O}(1)$ for any $s \leq t$, the final complexity of the BL policy is then $\mathcal{O}((T/\delta) \log (T/\delta)) = \mathcal{O}(\Omega \log \Omega)$.

It remains to prove that applying the bisection method with the precision δ leads to the expected approximation factor of $2 + \varepsilon$. Recall that the next order time t' found by the bisection method belongs to $[t - \delta, t + \delta]$, even if we impose $t' \geq s + \delta$. For conciseness we denote by $\Delta H(s, r)$ the difference $H(s, r + \delta) - H(s, r)$ for any instant $r \geq s$. According to Lemma 3.3.1 and using the fact that $H(s, t) = K$, we can bound the cost ratio between BL and any feasible policy P on $(s, t']$ as follows :

Case 1. $t' \in [t - \delta, t]$:

$$\begin{aligned}
\frac{\mathcal{C}^{\text{BL}}(s, t')}{\mathcal{C}^{\text{P}}(s, t')} &\leq \frac{H(s, t') + K}{H(s, t')} \\
&\leq \frac{H(s, t - \delta) + K}{H(s, t - \delta)} \\
&= \frac{2H(s, t) - (H(s, t) - H(s, t - \delta))}{H(s, t) - (H(s, t) - H(s, t - \delta))} \\
&= \frac{2H(s, t) - \Delta H(s, t - \delta)}{H(s, t) - \Delta H(s, t - \delta)} \\
&= 2 + \frac{\Delta H(s, t - \delta)}{K - \Delta H(s, t - \delta)}
\end{aligned}$$

Case 2. $t' \in (t, t + \delta]$:

$$\begin{aligned}
\frac{\mathcal{C}^{\text{BL}}(s, t')}{\mathcal{C}^{\text{P}}(s, t')} &\leq \frac{H(s, t + \delta) + K}{K} \\
&= \frac{2K + (H(s, t + \delta) - H(s, t))}{K} \\
&= 2 + \frac{\Delta H(s, t)}{K}
\end{aligned}$$

We next bound the holding cost $\Delta H(s, r)$ for any instants $s \leq r \leq T - \delta$:

$$\begin{aligned}
H(s, r + \delta) - H(s, r) &= \int_s^{r+\delta} h(u) (\Lambda(r + \delta) - \Lambda(u)) du \\
&\quad - \int_s^r h(u) (\Lambda(r) - \Lambda(u)) du \\
&= \int_s^r h(u) (\Lambda(r + \delta) - \Lambda(r)) du \\
&\quad + \int_r^{r+\delta} h(u) (\Lambda(r + \delta) - \Lambda(u)) du \\
&\leq (\Lambda(r + \delta) - \Lambda(r)) \int_s^{r+\delta} h(u) du \\
&\leq (\Lambda(r + \delta) - \Lambda(r)) \int_0^T h(u) du \\
&\leq \hat{\lambda} \hat{h} T
\end{aligned}$$

Let $\alpha = \frac{\hat{\lambda} \hat{h} T}{K} \delta = \frac{\varepsilon}{1 + \varepsilon}$. The previous inequality shows that $\Delta H(s, r) \leq \alpha K$ for any instants s, r with $s \leq r \leq T - \delta$. Using this upper bound, we obtain in the two previous cases :

Case 1. Since $x \mapsto x/(K - x)$ is clearly a nondecreasing function of x on $[0, K)$ we have :

$$\begin{aligned} \frac{\mathcal{C}^{\text{BL}}(s, t')}{\mathcal{C}^{\text{P}}(s, t')} &\leq 2 + \frac{\alpha K}{K - \alpha K} = 2 + \frac{\alpha}{1 - \alpha} \\ &= 2 + \varepsilon \end{aligned}$$

Case 2. We have directly, using that $x/(1 + x) \leq x$ for any $x > 0$:

$$\begin{aligned} \frac{\mathcal{C}^{\text{BL}}(s, t')}{\mathcal{C}^{\text{P}}(s, t')} &\leq 2 + \frac{\alpha K}{K} = 2 + \alpha \\ &\leq 2 + \varepsilon \end{aligned}$$

Therefore in both cases the performance ratio is bounded by $2 + \varepsilon$. In particular, the previous inequalities are true for P an optimal policy and the result follows.

A.2 Proof of Theorem 3.4.2

Consider a feasible policy P for the production problem and two consecutive order time instants s and t of PB. Similarly to Lemma 3.3.1, we establish that the cost $\mathcal{C}^{\text{P}}(s, t)$ incurred by P on time interval $(s, t]$ is at least $K + H_{\min}(s, t)$.

As already noticed, the holding cost of any feasible policy on $(s, T]$ is at least $H_{\min}(s, t)$. Thus the inequality clearly holds if P orders (i.e. starts a production) in the time interval $(s, t]$. Conversely assume that P does not order on $(s, t]$. Let $u \in (s, t]$ be the end of the active period of PB in the time interval. We establish that the stock level in policy P is greater or equal to the stock level in policy PB at any point in time of $(s, t]$. That is, the balancing policy PB carries the minimum possible inventory for a policy that does not order inside the time interval $(s, t]$. First note that for any $z \in [u, t]$, we have $x^P(z) \geq x^{\text{PB}}(z)$ as P satisfies demand in $[u, t]$ without stockouts. In addition, since $x^P(s) \geq x_{\min}(s) = x^{\text{PB}}(s)$, $x^P(u) \geq x^{\text{PB}}(u)$ and PB continuously produces in $[s, u]$, we also have $x^P(z) \geq x^{\text{PB}}(z)$ for all $z \in [s, u]$. As a result, the holding cost incurred by policy P on $(s, T]$ is at least $H(s, t) = K + H_{\min}(s, t)$.

Thus for any policy P we have $\mathcal{C}^{\text{P}}(s, t) \geq K + H_{\min}(s, t)$ and a proof similar to the one of Theorem 3.3.2 leads to the same approximation guarantee of two.

A.3 Proof of Theorem 3.4.3

Consider a feasible policy P for the problem with shortages and let $s_1 < \dots < s_n$ be the sequence of orders found by the BB policy. We prove that the total cost \mathcal{C}^{BB} incurred by BB on the time horizon is at most twice \mathcal{C}^{P} , the total cost incurred by policy P .

We start by considering the interval $[0, s_1]$. Recall that $u_0 = \max \{u \geq 0 : \Lambda(0, u) \leq x(0)\}$ is the latest moment such that demand can be satisfied with the inventory on hand at time 0. On $[0, s_1]$, the BB policy incurs a total cost of $H(0, u_0) + B(u_0, s_1) + K = 2K$. On the other hand, either P places an order in $[0, s_1]$ and incurs a total cost of at least K , or it does not and thus it incurs at least $H(0, u_0) + B(u_0, s_1) = K$ on $[0, s_1]$. In both cases, we have $\mathcal{C}^{\text{BB}}(0, s_1) \leq 2\mathcal{C}^{\text{P}}(0, s_1)$.

For the last cycle, we distinguish between two cases, depending on whether $s_n < T$ or $s_n = T$. If $s_n < T$, the BB policy incurs a total cost of $H(s_n, T) \leq K$ on $(s_n, T]$. As policy P is feasible, either it orders on $(s_n, T]$ or it does not and then its inventory level at s_n is at least $\Lambda(s_n, T)$. Therefore it incurs on $(s_n, T]$ a cost of at least $\mathcal{C}^{\text{P}}(s_n, T) \geq H(s_n, T) = \mathcal{C}^{\text{BB}}(s_n, T)$. If $s_n = T$, the BB policy incurs a cost

$$\begin{aligned} \mathcal{C}^{\text{BB}}(s_{n-1}, s_n) &= K + \min_{u \in [s_{n-1}, s_n]} \{H(s_{n-1}, u) + B(u, T)\} \\ &\leq 2K \end{aligned}$$

Note that since BB orders at T , the algorithm ensures that $H(s_{n-1}, T) > K$ by construction. Therefore, if P orders in $(s_{n-1}, T]$ it incurs an order cost of K , while if it does not it incurs a holding cost greater than K . Consequently we have $\mathcal{C}^{\text{BB}}(s_{n-1}, T) \leq 2\mathcal{C}^{\text{P}}(s_{n-1}, T)$.

Now let s_i and s_{i+1} be two consecutive order times of the BB policy, with $s_{i+1} < T$. The total cost incurred by BB over $(s_i, s_{i+1}]$ is

$$\begin{aligned} \mathcal{C}^{\text{BB}}(s_i, s_{i+1}) &= \min_{u \in [s_i, s_{i+1}]} \{H(s_i, u) + B(u, s_{i+1})\} + K \\ &= 2K \end{aligned}$$

On the other hand, either P orders in $(s_i, s_{i+1}]$ and incurs a cost of at least K or it does not and then there exists $u^{\text{P}} \in [s_i, s_{i+1}]$ such that $x^{\text{P}}(z) \geq 0$ for all $z \in [s_i, u^{\text{P}}]$ and $x^{\text{P}}(z) \leq 0$ for all $z \in (u^{\text{P}}, s_{i+1}]$. Hence P incurs a holding and backlogging cost of at least $\min_{u \in [s_i, s_{i+1}]} \{H(s_i, u) + B(u, s_{i+1})\} = K$. Therefore we have $\mathcal{C}^{\text{BB}}(s_i, s_{i+1}) = 2K \leq 2\mathcal{C}^{\text{P}}(s_i, s_{i+1})$.

If we sum up the costs incurred on each of these intervals, we obtain :

$$\begin{aligned} \mathcal{C}^{\text{BB}}(0, T) &= \mathcal{C}^{\text{BB}}(0, s_1) + \sum_{i=1}^{n-1} \mathcal{C}^{\text{BB}}(s_i, s_{i+1}) + \mathcal{C}^{\text{BB}}(s_n, T) \\ &\leq 2\mathcal{C}^{\text{P}}(0, s_1) + \sum_{i=1}^{n-1} 2\mathcal{C}^{\text{P}}(s_i, s_{i+1}) + 2\mathcal{C}^{\text{P}}(s_n, T) \\ &\leq 2\mathcal{C}^{\text{P}}(0, T) \end{aligned}$$

and the proof follows.

A.4 Proof of Theorem 3.4.4

In this proof, we modify the notation introduced in §3.2 as follows : For $s < t$ and P a policy for the problem, let $\mathcal{C}^{\text{P}}(s, t)$ be the sum of the fixed order cost incurred by P in $(s, t]$ plus the linear order cost and holding cost incurred by the units used to serve demands in $(s, t]$. For instance, assume that P orders at time $v \in (s, t]$ to satisfy demands in $[u, t]$ and at time $r < s$ to serve demands in (s, v) . Then we have :

$$\begin{aligned} \mathcal{C}^{\text{P}}(s, t) &= c(r)\Lambda(s, v) + c(v)\Lambda(v, t) + \int_r^s h(x)\Lambda(s, v)dx \\ &\quad + \int_s^v h(x)\Lambda(x, v)dx + \int_v^t h(x)\Lambda(x, t)dx \end{aligned}$$

Note that this is an extension of the original definition of $\mathcal{C}^{\text{P}}(\cdot, \cdot)$ and thus all the previous proofs remain valid with this new definition.

We start by proving the result for the forward algorithm. Let P be a feasible policy for the problem with nonincreasing order costs. The first order of the balancing policy is placed at time $t_0 = \max\{t \leq T : \Lambda(0, t) \leq x_0\}$. Thus P necessarily places an order at time $u \in (0, t_0]$ for an order cost $K(u) \geq K(t_0)$ and we have $\mathcal{C}^{\text{P}}(0, t_0) \geq \mathcal{C}^{\text{BL}}(0, t_0)$. Let $i < n$ and focus on the ordering cycle $(s_i, s_{i+1}]$. We bound the cost incurred by policy P over $(s_i, s_{i+1}]$ as follows :

Case 1. P places an order at time $u \in (s_i, s_{i+1}]$: It incurs an order cost $K(u) \geq K(s_{i+1})$.

Case 2. P does not order in $(s_i, s_{i+1}]$: It incurs a holding cost of at least $H(s_i, s_{i+1})$. Moreover, P orders the units used to serve the demands in $(s_i, s_{i+1}]$ at a previous point in time $s' < s_i$, incurring an additional linear order cost and holding cost of $(c(s') + \int_{s'}^s h(u)du) \Lambda(s_i, s_{i+1})$. According to equation 3.6, we thus have :

$$\begin{aligned}
\mathcal{C}^P(s_i, s_{i+1}) &= \left(c(s') + \int_{s'}^s h(u) du \right) \Lambda(s_i, s_{i+1}) + H(s_i, s_{i+1}) \\
&\geq c(s_i) \Lambda(s_i, s_{i+1}) + H(s_i, s_{i+1}) \\
&\geq K(s_{i+1})
\end{aligned}$$

Thus for any $i < n$ and ordering cycle $(s_i, s_{i+1}]$, we have :

$$\mathcal{C}^{\text{BL}}(s_i, s_{i+1}) = 2K(s_{i+1}) \leq 2\mathcal{C}^P(s_i, s_{i+1})$$

Finally, one can use similar arguments to prove that we have

$$\mathcal{C}^{\text{BL}}(s_n, T) = c(s_n) \Lambda(s_n, T) + H(s_n, T) \leq \mathcal{C}^P$$

and the cost incurred by the balancing policy over the entire planning horizon can be bounded as follows :

$$\begin{aligned}
\mathcal{C}^{\text{BL}}(0, T) &= \mathcal{C}^{\text{BL}}(0, t_0) + \sum_{i=1}^{n-1} \mathcal{C}^{\text{BL}}(s_i, s_{i+1}) + \mathcal{C}^{\text{BL}}(s_n, T) \\
&= \mathcal{C}^{\text{BL}}(0, t_0) + \sum_{i=1}^{n-1} 2K(s_{i+1}) + \mathcal{C}^{\text{BL}}(s_n, T) \\
&\leq \mathcal{C}^P(0, t_0) + \sum_{i=1}^{n-1} 2\mathcal{C}^P(s_i, s_{i+1}) + \mathcal{C}^P(s_n, T) \\
&\leq 2\mathcal{C}^P(0, T)
\end{aligned}$$

For the backward algorithm, the cost accounting for a cycle $(s, t]$ of the balancing policy is slightly modified since we now account for the order cost in period s and the holding cost over $[s, t)$. As a consequence, a cycle is denoted $[s, t)$ in the remaining of the proof.

Let P be a feasible policy for the problem with nondecreasing order costs. First, consider the case where the first order is place earlier than t_0 : $s_n < t_0$. By definition we have :

$$\begin{aligned}
\mathcal{C}^{\text{BL}}(t_0, s_{n-1}) &= K(s_n) + c(s_n) \Lambda(t_0, s_{n-1}) \\
&\quad + H(s_n, s_{n-1}) - H(s_n, t_0) \\
&\leq 2K(s_n)
\end{aligned}$$

On the other hand, either P orders at some time instant $u \in [s_n, s_{n-1})$ and incurs a fixed order cost $K(u) \geq K(s_n)$, or it does not and the units used to serve demands in

$(t_0, s_{n-1}]$ are ordered in some previous period $s' < s_n$. Therefore we have :

$$\begin{aligned}
\mathcal{C}^P(t_0, s_{n-1}) &= \left(c(s') + \int_{s'}^{s_n} h(u) du \right) \Lambda(t_0, s_{n-1}) \\
&\quad + H(s_n, s_{n-1}) - H(s_n, t_0) \\
&\leq c(s_n) \Lambda(t_0, s_{n-1}) \\
&\quad + H(s_n, s_{n-1}) - H(s_n, t_0) \\
&\leq K(s_n)
\end{aligned}$$

For $i \leq n$, we bound the cost incurred by policy P over the cycle $[s_i, s_{i-1})$ using similar arguments as the forward case : Either P places an order in $[s_i, s_{i-1})$ and incurs an order cost of at least $K(s_i)$ or it does not and thus incurs a linear order cost and holding cost of at least $K(s_i)$ to serve demands in $[s_i, s_{i-1})$. Therefore we have :

$$\begin{aligned}
\mathcal{C}^{\text{BL}}(0, T) &= \mathcal{C}^{\text{BL}}(0, s_n) + \sum_{i=1}^n \mathcal{C}^{\text{BL}}(s_i, s_{i-1}) \\
&\leq \mathcal{C}^P(0, s_n) + \sum_{i=1}^n 2K(s_i) \\
&\leq \mathcal{C}^P(0, s_n) + 2 \sum_{i=1}^n \mathcal{C}^P(s_i, s_{i-1}) \\
&\leq 2\mathcal{C}^P(0, T)
\end{aligned}$$

and the proof follows.

Annexe B

Proofs of chapter 4

B.1 Proof of the dominance of synchronized policies for J -retailers

Consider a J -retailer i as defined in § 4.4.1 : For any period t and any quantities x , y and q , the inequality $h_t^i(y+q) - h_t^i(y) \leq h_t^0(x+q) - h_t^0(x)$ holds. We prove that it is dominant for i to synchronize its orders with the warehouse. Assume a policy π does not satisfy this property and let s be the first period when retailer i orders while the warehouse does not. Let r be the previous ordering period of the warehouse and q be the quantity ordered in period s by retailer i . We denote by x_t and y_t the stock on hand at the warehouse and at the retailer, respectively. We now consider the alternative policy π' that shifts the order of period s to period r , ordering the quantity q . The stock levels of policy π' on interval $[r, s)$ are $x'_t = x_t - q$ and $y'_t = y_t + q$. If we compute the holding costs paid by π and π' on $[r, s)$, we get $\sum_{t=r}^{s-1} (h_t^0(x_t) + h_t^i(y_t))$ and $\sum_{t=r}^{s-1} (h_t^0(x_t - q) + h_t^i(y_t + q))$, respectively. The inequality on marginal costs ensures that the cost incurred by π' is less than or equal to the cost of π . Before period r and after period s both policies are identical by construction. Using an interchange argument, we conclude that it is dominant to synchronize all the orders of retailer i with the warehouse.

B.2 Proof of Lemma 4.4.1, §4.4.1

Consider that we are given independent single-echelon feasible ZIO policies π_i for problems (\widehat{S}_i) or (S_i) . We prove here that the uncrossing algorithm builds a feasible policy π^u for the OWMR problem that costs at most twice the sum of the costs of all local policies π_i in (\widehat{S}_i) . We start by proving a result similar to Lemma 4.3.2, that is :

$$\mathcal{C}(\pi^u) \leq \sum_{i=0}^N \mathcal{C}_i(\pi_i) + \sum_{i=1}^N \mathcal{K}_i(\pi_i) \quad (\text{B.1})$$

We first introduce some additional notations. Recall that product 0 stands for the aggregated demands of all the W -retailers. For $i = 0, \dots, N$, we denote x_{it}^u the inventory level at location i in the final policy π_i^u in period t . Similarly for $i = 1, \dots, N$ let x_{it} be the inventory level of policy π_i in period t . We use a different notation for the warehouse, in order to distinguish between product 0 and products $i \in I_J$. Specifically, for $i \in \{0\} \cup I_J$, x_{it}^0 denotes the inventory level of policy π_0 for product i in period t .

Let r, r' be an ordering interval of policy π_0^u . By construction π^u is an uncrossing policy and thus policy π_i^u is exactly in one of the following situations :

- (a) Either retailer i does not order within the interval $[r, r' - 1]$
- (b) Or retailer i orders only in period r within the interval $[r, r' - 1]$, its next order occuring in period r' or later,
- (c) Or retailer i orders in period $u > r$ within the interval $[r, r' - 1]$. Notice that it includes the case where retailer i orders more than once within the interval. The fact that π^u is uncrossing implies that necessarily retailer i orders in period r' .

It is easy to see that Step 1 at most doubles the ordering costs of each policy π_i for $i = 1, \dots, N$. Step 2 can then only decrease the number of orders of J -retailers and thus does not increase the costs. Therefore we only need to prove that Step 3 does not increase the total holding cost incurred by the uncrossed policies $\pi' = (\pi'_0, \pi'_1, \dots, \pi'_N)$ (where $\pi'_0 = \pi_0$). We prove the latter result for the W -retailers and the warehouse first and then for the J -retailers, separately.

Recall that the ZIO property is dominant for policies π_i and π_i^u : This ensures that for all $i \in I_W$ and $t = 1, \dots, T$, $x_{it}^u \leq x_{it}$ as Step 1 adds orders to π_i and Step 2 does not cancel any orders of W -retailers. Therefore from (LD1) we have for all $t = 1, \dots, T$

$$h_t^i(x_{it}^u) \leq h_t^i(x_{it}) \quad (\text{B.2})$$

Moreover, the holding cost incurred by policy π_0^u comes only from units supplying W -retailers, as synchronized policies are dominant for J -retailers. Thus consider an ordering interval r, r' of π_0^u and a period $r \leq t < r'$. Notice that any W -retailer i in situation (a) or (b) on r, r' reduces x_{0t} by $\sum_{v=t+1}^{r'-1} d_v^i$, while it reduces x_{0t} by x_{it}^u otherwise. Hence the inventory level of policy π_0^u in period t is

$$x_{0t}^u = x_{0t} - \sum_{i \in I_W} \min \left\{ x_{it}^u, \sum_{v=t+1}^{r'-1} d_v^i \right\} \leq x_{0t}$$

Since $h_t^0(\cdot)$ is nondecreasing, the total holding cost incurred by π_0^u for product 0 is lower than the holding cost of product 0 in π_0 : For all $t = 1, \dots, T$ we have

$$h_t^0(x_{0t}^u) \leq h_t^0(x_{0t}) \quad (\text{B.3})$$

Now consider a retailer $i \in I_J$. We prove that the holding cost incurred by π_i^u is lower

or equal than the sum of the holding cost incurred by π_i plus the holding cost incurred by item i in policy π_0 . It suffices to note that for any warehouse ordering interval r, r' , π_i^u is either in situation (a) or (b) since Step 2 of the algorithm ensures that all orders of J -retailers are synchronized with the warehouse. Furthermore, any period t such that $x_{it}^u > x_{it}$ corresponds to the case where orders have been shifted to the left in Step 2 and induces that $x_{it}^u = x_{it}^0$. Partitioning the periods according to this inequality yields for all $i \in I_J$ and $t = 1, \dots, T$:

$$\begin{aligned} h_t^i(x_{it}^u) &= \begin{cases} h_t^i(x_{it}) & \text{if } x_{it}^u \leq x_{it} \\ h_t^i(x_{it}^0) & \text{otherwise.} \end{cases} \\ &\leq h_t^i(x_{it}) + h_t^i(x_{it}^0) \end{aligned} \quad (\text{B.4})$$

Summing inequalities (B.2), (B.3) and (B.4) over all the retailers, we obtain for all $t = 1, \dots, N$:

$$\sum_{i=0}^N h_t^i(x_{it}^u) \leq \sum_{i=1}^N h_t^i(x_{it}) + h_t^0(x_{0t}^0) + \sum_{i \in I_J} h_t^i(x_{it}^0) \quad (\text{B.5})$$

and summing inequalities (B.5) over all t yields :

$$\begin{aligned} \mathcal{H}(\pi^u) &= \sum_{t=1}^T \sum_{i=0}^N h_t^i(x_{it}^u) \\ &\leq \sum_{t=1}^T \left(\sum_{i=1}^N h_t^i(x_{it}) + h_t^0(x_{0t}^0) + \sum_{i \in I_J} h_t^i(x_{it}^0) \right) \\ &= \sum_{i=1}^N \mathcal{H}_i(\pi_i) + \mathcal{H}_0(\pi_0) \end{aligned} \quad (\text{B.6})$$

Inequality (B.1) follows.

The proof of the Lemma is then straightforward :

$$\begin{aligned}
\mathcal{C}(\pi^u) &\leq \mathcal{C}_0(\pi_0) + \sum_{i=1}^N \mathcal{C}_i(\pi_i) + \mathcal{K}_i(\pi_i) \\
&= \mathcal{H}_0(\pi_0) + \mathcal{K}_0(\pi_0) + \sum_{i=1}^N \mathcal{H}_i(\pi_i) + 2\mathcal{K}_i(\pi_i) \\
&\leq 2 \sum_{i=0}^N \mathcal{K}_i(\pi_i) + \sum_{i=0}^N \mathcal{H}_i(\pi_i) \\
&= 2 \sum_{i=0}^N \mathcal{K}_i(\pi_i) + \sum_{i=0}^N \widehat{\mathcal{H}}_i(\pi_i) \\
&= 2 \sum_{i=0}^N \widehat{\mathcal{C}}_i(\pi_i)
\end{aligned}$$

B.3 Proof of Lemma 4.4.2, §4.4.1

Although Lemma 4.4.2 states the same lower bound as Lemma 4.3.3, the non-linearity of the holding costs leads to a quite different proof. However, the main ideas remain similar : Based on an optimal policy for the OWMR problem, we exhibit feasible policies $\tilde{\pi}_i$ for each system (\widehat{S}_i) such that $\widehat{\mathcal{C}}_0(\tilde{\pi}_0) + \sum_{i=1}^N \widehat{\mathcal{C}}_i(\tilde{\pi}_i)$ is less than or equal to \mathcal{C}^* . Lemma 4.4.2 then follows from the optimality of each policy $\widehat{\pi}_i^*$ for system (\widehat{S}_i) .

Consider an optimal policy $\pi^{\text{OPT}} = (\pi_0^{\text{OPT}}, \pi_1^{\text{OPT}}, \dots, \pi_N^{\text{OPT}})$ of cost \mathcal{C}^* for the OWMR instance. According to (LD3), one can choose policy π^{OPT} such that no inventory is held at the warehouse for J -retailers. We denote by x_{it}^{OPT} the inventory level of π^{OPT} in period t in location i and we let $x_t^e \equiv x_{0t}^{\text{OPT}} + \sum_{i \in I_W} x_{it}^{\text{OPT}}$ be the aggregated stock level of the warehouse and W -retailers altogether, which represents the echelon stock of the system when set I_J is empty. We now specify a feasible policy $\tilde{\pi}_i$ for each system (\widehat{S}_i) :

- For each $i > 0$, policy $\tilde{\pi}_i$ is identical to policy π^{OPT} restricted to retailer i : It orders the same quantities in the same periods.
- For the multi-item problem (\widehat{S}_0) , policy $\tilde{\pi}_0$ orders when policy π^{OPT} places its warehouse orders. In addition, the quantity ordered in a period r corresponds to the cumulative demand of retailers in I_W until the next ordering period (excluded), plus all the quantities ordered in period r by retailers in I_J .

We first prove that the sum of the costs of policies $\tilde{\pi}_i$ applied to systems (\widehat{S}_i) is lower than the optimal cost \mathcal{C}^* for the original problem.

It is straightforward to see that the total ordering cost paid by policies $\tilde{\pi}_i$ is exactly the total ordering cost paid by policy π^{OPT} , since they order at the same periods. Therefore we focus on the holding costs incurred by policies $\tilde{\pi}_i$. We denote by \tilde{x}_{it} the inventory level at retailer i in period t for policy $\tilde{\pi}_i$, $i > 0$, and by \tilde{y}_{it} the inventory level of product i for policy $\tilde{\pi}_0$ at the warehouse. Recall that product 0 represents the aggregation of the

W -retailers. For this product, we have for each period t :

$$\begin{aligned} \hat{h}_t^0(\tilde{y}_{0t}) &= \frac{1}{2}h_t^0(\tilde{y}_{0t}) \leq \frac{1}{2}h_t^0(x_t^e) \\ &\leq \frac{1}{2}h_t^0(x_{0t}^{\text{OPT}}) + \frac{1}{2} \sum_{i \in I_W} h_t^0(x_{it}^{\text{OPT}}) \\ &\leq \frac{1}{2}h_t^0(x_{0t}^{\text{OPT}}) + \frac{1}{2} \sum_{i \in I_W} h_t^i(x_{it}^{\text{OPT}}) \end{aligned}$$

The first inequality comes from (LD1), the non-decreasing property of $h_t^0(\cdot)$. Indeed in any feasible policy the stock level at the warehouse plus the stock level at a subset of retailers has to be sufficient to satisfy the demands of these retailers until the next ordering period of the warehouse (which is precisely \tilde{y}_{0t} if we focus on product 0), which implies that $\tilde{y}_{0t} \leq x_t^e$. The second inequality directly comes from (LD2), the sub-additivity of h_t^0 . The definition of set I_W implies the last inequality.

Now if we restrict our attention to items $i \in I_J$ of system (\hat{S}_0) , we have by construction $\tilde{y}_{it} = x_{it}^{\text{OPT}}$ in each period. Hence the holding cost $\hat{h}_t^i(\tilde{y}_{it})$ paid for such a product $i \in I_J$ in period t is exactly $\frac{1}{2}h_t^i(x_{it}^{\text{OPT}})$. Adding the holding costs paid for all products altogether in system (\hat{S}_0) , we obtain for each period t that :

$$\hat{h}_t^0(\tilde{y}_{0t}) + \sum_{i \in I_J} \hat{h}_t^i(\tilde{y}_{it}) \leq \frac{1}{2}h_t^0(x_{0t}^{\text{OPT}}) + \frac{1}{2} \sum_{i=1}^N h_t^i(x_{it}^{\text{OPT}})$$

Finally the stock level in each system (\hat{S}_i) , is identical to the stock level at retailer i in policy π^{OPT} . Hence we also have for each period t :

$$\hat{h}_t^i(\tilde{x}_{it}) = \frac{1}{2}h_t^i(x_{it}^{\text{OPT}}) \quad \forall i = 1, \dots, N$$

As a consequence for each period t , the total holding cost paid by all the policies $\tilde{\pi}_i$ in (\hat{S}_i) is at most $\frac{1}{2}h_t^0(x_{0t}^{\text{OPT}}) + \sum_{i=1}^N h_t^i(x_{it}^{\text{OPT}})$, which is a lower bound on the holding cost paid by policy π^{OPT} , since h_t^0 takes only positive values. Summing the holding costs paid by all the

policies $\tilde{\pi}_i$, we obtain for each period t that :

$$\begin{aligned}
\sum_{i=0}^N \widehat{\mathcal{H}}(\hat{\pi}_i^*) &= \hat{h}_t^0(\tilde{y}_{0t}) + \sum_{i \in I_J} \hat{h}_t^i(\tilde{y}_{it}) + \sum_{i=1}^N \hat{h}_t^i(\tilde{x}_{it}) \\
&\leq \frac{1}{2} h_t^0(x_{0t}^{\text{OPT}}) + \sum_{i=1}^N h_t^i(x_{it}^{\text{OPT}}) \\
&\leq \sum_{i=0}^N h_t^i(x_{it}^{\text{OPT}}) \\
&= \mathcal{H}(\pi^{\text{OPT}})
\end{aligned}$$

The lemma follows.

B.4 Proof of Lemma 4.4.4, §4.4.2

First, we prove that the final policy built by the uncrossing algorithm is uncrossing and feasible. By construction π' is clearly uncrossing from Step 1 of the uncrossing, while Step 2 does not alter this property. Hence since Step 3 does not add any order to π' , the final policy π^u is uncrossing. To prove that π^u is feasible, we simply have to prove that for each demand (i, t) , the final pair of orders $[r^{it}, s^{it}]$ used in π^u is valid, i.e. r^{it} (resp. s^{it}) is an ordering period of the warehouse (resp. retailer i) in π^u and $r^{it} \leq s^{it} \leq t$. Let r and s be the ordering periods for demand d_t^i in the single-echelon policy π_0 and π_i , respectively. Assume first that $t \geq s^+$: If $s \in R$ then $s^+ = s$ and the pair $[s^+, s^+]$ is valid. If $s \notin R$, s is by construction uncrossed with s^+ in Step 1 of the algorithm. Therefore s^+ is an ordering period of π'_i and the pair $[s^+, s^+]$ is valid. Now consider the case in which $i \in I_J$ and $t < s^+$: By construction in Step 2 of the algorithm π'_i places an order in period $s^- \leq s \leq t$ and the pair $[s^-, s^-]$ is valid. Finally if $i \in I_W$ and $t < s^+$, notice that necessarily from the feasibility of policy π_0 we have $r \leq s^- \leq s$ and the pair $[r, s]$ is valid since periods r and s are both ordering periods of policy π_0 and π'_i , respectively.

Now let us focus on the costs incurred in the final uncrossing solution. Note that for all $i = 1, \dots, N$ the total ordering cost in π'_i is less than twice the total ordering cost in π_i since Step 1 adds at most one new order in π'_i for each order in π_i . On the other hand, no additional order is added to policy π_0 at the warehouse. Therefore we have

$$\mathcal{K}(\pi') \leq \mathcal{K}_0(\pi_0) + 2 \sum_{i=1}^N \mathcal{K}_i(\pi_i) \tag{B.7}$$

It remains to prove that Step 2 and Step 3 do not increase the costs incurred by $\pi_0, \pi'_1, \dots, \pi'_N$ after Step 1. This is clearly the case for Step 2 since it possibly cancels some orders of J -retailers. We now show that Step 3 can only decrease the total holding cost incurred by the system. We focus on the holding costs incurred by the different units,

depending on the item (i.e. which retailer they serve) : Given a demand (i, t) , let r (resp. s) be the period when the warehouse (resp. retailer i) orders d_t^i in the single-echelon policy π_0 (resp. π_i). We now show that the holding cost paid by policy π^u for each unit of demand d_t^i is at most the holding cost paid by π_0 in (S_0) for a unit of demand of item i in period t plus the holding cost paid by π_i in (S_i) for a unit of demand in period t . Depending on the type of retailer i , we can bound the holding cost incurred using the properties of the metric holding cost parameters.

Case 1 : i is a W -retailer. If $t \geq s^+$, the final holding cost incurred by policy π^u to serve a unit of d_t^i is $h_{s^+s^+}^{it}$ and we have from properties (P2) and (P3) :

$$\begin{aligned} h_{s^+s^+}^{it} &\leq h_{ss}^{it} \\ &\leq h_{rt}^{it} + h_{ss}^{it} \end{aligned} \quad (\text{B.8})$$

If $t < s^+$, π^u incurs a holding cost of h_{rs}^{it} for each unit of demand d_t^i and property (SA4) ensures that

$$h_{rs}^{it} \leq h_{rt}^{it} + h_{ss}^{it} \quad (\text{B.9})$$

Case 2 : i is a J -retailer. If $t \geq s^+$, π^u incurs again a holding cost $h_{s^+s^+}^{it}$ to serve a unit of d_t^i . Property (P3) therefore implies

$$\begin{aligned} h_{s^+s^+}^{it} &\leq h_{ss}^{it} \\ &\leq h_{rr}^{it} + h_{ss}^{it} \end{aligned} \quad (\text{B.10})$$

If $t < s^+$, the holding cost incurred by π^u is equal to $h_{s^-s^-}^{it}$ for each unit of demand d_t^i . Using property (P3), we obtain

$$\begin{aligned} h_{s^-s^-}^{it} &\leq h_{rr}^{it} \\ &\leq h_{rr}^{it} + h_{ss}^{it} \end{aligned} \quad (\text{B.11})$$

In all cases, the cost holding cost incurred in the final policy to serve demand d_t^i is lower than the sum of the holding costs incurred in the single-echelon policies π_0 and π_i in systems (S_0) and (S_i) , respectively. Therefore we have from inequalities (B.8)-(B.11) :

$$\mathcal{H}(\pi^u) \leq \mathcal{H}_0(\pi_0) + \mathcal{H}_i(\pi_i) \quad (\text{B.12})$$

We conclude from inequalities (B.7) and (B.12) that the following bound on the total cost incurred by π^u holds :

$$\mathcal{C}(\pi^u) \leq \mathcal{C}_0(\pi_0) + \sum_{i=1}^N \mathcal{C}_i(\pi_i) + \sum_{i=1}^N \mathcal{K}_i(\pi_i)$$

and the lemma follows.

B.5 Proof of Lemma 4.4.5, §4.4.2

Consider the formulation (IP) for the OWMR problem introduced in §4.4.2. Let z_r^{it} be the boolean variable indicating if demand d_t^i is ordered at the warehouse in period r , i.e. $z_r^{it} = \sum_{s:r \leq s \leq t} x_{rs}^{it}$. In a symmetric way, we also introduce $u_s^{it} = \sum_{r:1 \leq r \leq s} x_{rs}^{it}$, the boolean variable indicating if demand d_t^i is ordered at retailer i in period s . Finally, we define c_r^{it} as the minimal holding cost to serve demand d_t^i at retailer i with a warehouse order in period r . That is, $c_r^{it} = H_{rt}^{it}$ if $i \in I_W$ and $c_r^{it} = H_{rr}^{it}$ if $i \in I_J$.

Now due to Property (P2), observe that $c_r^{it} z_r^{it} \leq \sum_{s:r \leq s \leq t} x_{rs}^{it} H_{rs}^{it}$. Also Property (P3) implies that $H_{ss}^{it} u_s^{it} \leq \sum_{r:1 \leq r \leq s} x_{rs}^{it} H_{rs}^{it}$. Hence we can write for each demand point (i, t) that $\sum_{r,s:r \leq s \leq t} x_{rs}^{it} H_{rs}^{it} \geq \frac{1}{2} \sum_{r \leq t} c_r^{it} z_r^{it} + \frac{1}{2} \sum_{s \leq t} H_{ss}^{it} u_s^{it}$. The optimal solution to the following linear integer program (RP) is thus a lower bound of the optimal solution of formulation (IP) (observe that constraints (1) and (1') in formulation (RP) are redundant).

$$\begin{aligned}
 (RP) \left\{ \begin{array}{l}
 \min \quad \sum_{r=1}^T y_r^0 K^0 + \sum_{i=1}^N \sum_{s=1}^T y_s^i K^i + \sum_{i=1}^N \sum_{t=1}^T \left(\frac{1}{2} \sum_{r=1}^t c_r^{it} z_r^{it} + \frac{1}{2} \sum_{s=1}^t H_{ss}^{it} u_s^{it} \right) \\
 \text{s.t.} \quad \sum_{r,s:r \leq s \leq t} x_{rs}^{it} = 1, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T \text{ such that } d_t^i > 0 \\
 \sum_{r:r \leq t} z_r^{it} = 1, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T \text{ such that } d_t^i > 0 \quad (1) \\
 \sum_{s:s \leq t} u_s^{it} = 1, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T \text{ such that } d_t^i > 0 \quad (1') \\
 z_r^{it} = \sum_{s:r \leq s \leq t} x_{rs}^{it}, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall r = 1, \dots, t \quad (2) \\
 u_s^{it} = \sum_{r:1 \leq r \leq s} x_{rs}^{it}, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall s = 1, \dots, t \quad (2') \\
 u_s^{it} \leq y_s^i, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall s = 1, \dots, t \\
 z_r^{it} \leq y_r^0, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall r = 1, \dots, t \\
 z_r^{it}, y_r^0 \in \{0, 1\}, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall r = 1, \dots, t \\
 u_s^{it}, y_s^i \in \{0, 1\}, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall s = 1, \dots, t \\
 x_{rs}^{it} \in \{0, 1\}, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall s = 1, \dots, t, \forall r = 1, \dots, s
 \end{array} \right.
 \end{aligned}$$

Now if we relax constraints (2) and (2'), problem (RP) becomes decomposable. Formulation (RP) therefore reduces to solving $N + 1$ independent problems : One problem (RP_0)

for the warehouse :

$$(RP_0) \left\{ \begin{array}{l} \min \sum_{r=1}^T y_r^0 K^0 + \sum_{i=1}^N \sum_{t=1}^T \sum_{r=1}^t \frac{1}{2} c_r^{it} z_r^{it} \\ \text{s.t.} \quad \sum_{r:r \leq t} z_r^{it} = 1, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T \text{ such that } d_t^i > 0 \\ z_r^{it} \leq y_r^0, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall r = 1, \dots, t \\ z_r^{it}, y_r^0 \in \{0, 1\}, \quad \forall i = 1, \dots, N, \forall t = 1, \dots, T, \forall r = 1, \dots, t \end{array} \right.$$

and one for each retailer $i = 1, \dots, N$, (RP_i) :

$$(RP_i) \left\{ \begin{array}{l} \min \sum_{s=1}^T y_s^i K^i + \sum_{t=1}^T \sum_{s:s \leq t} \frac{1}{2} H_{ss}^{it} u_s^{it} \\ \text{s.t.} \quad \sum_{s:s \leq t} u_s^{it} = 1, \quad \forall t = 1, \dots, T \text{ such that } d_t^i > 0 \\ u_s^{it} \leq y_s^i, \quad \forall t = 1, \dots, T, \forall s = 1, \dots, t \\ u_s^{it}, y_s^i \in \{0, 1\}, \quad \forall t = 1, \dots, T, \forall s = 1, \dots, t \end{array} \right.$$

These problems are precisely the single-echelon lot-sizing problems (\widehat{S}_i) . Integer linear programs (RP_0) and (RP_i) are thus a valid formulation for the problems we solve in Step 1 of our algorithm. As a consequence, $\widehat{\mathcal{C}}_0(\widehat{\pi}_0^*) + \sum_{i=1}^N \widehat{\mathcal{C}}_i(\widehat{\pi}_i^*)$ is a lower bound for (RP) and the result follows.

B.6 Proof of Proposition 4.5.1, §4.5.1

We prove here that any FTL procurement cost and any LTL procurement costs satisfying conditions of Proposition 4.5.1 is linearly sandwiched. Recall that we restrict ourself to a stationary batch size B and a stationary fixed cost per batch k .

First consider a FTL procurement cost : $FLL_t(x) = K_t + \lceil x/B \rceil k$. We define $A_t = K_t + k/2$ and $b = \frac{k}{2B}$. On one hand, for any quantity $x > 0$, we clearly have $FLL_t(x) \geq K_t + \frac{k}{B}x$ and $FLL_t(x) \geq K_t + k$. It results that $FLL_t(x) \geq A_t + bx$. On the other hand $FLL_t(x) \leq K_t + (\frac{x}{B} + 1)k$ implies that $FLL_t(x) \leq 2A_t + 2bx$. Thus $FLL_t(x)$ is linearly sandwiched by $A_t + bx$.

Now consider a LTL procurement cost : $LTL_t(x) = K_t + \lfloor x/B \rfloor k + f(x - \lfloor x/B \rfloor)$, with $k \leq K_t$. Recall that we assume that freight cost function is non-decreasing on $[0, B]$ with $f(0) = 0$ and $f(B) = k$ (in fact we only need $0 \leq f(0)$ and $f(B) \leq k$ in this proof). We

define two constants λ and μ as follows :

$$\lambda = \inf\{f(u) - \frac{k}{2B}u \mid u \in [0, B]\}$$

$$\mu = \sup\{f(u) - \frac{k}{B}u \mid u \in [0, B]\}$$

Consider a positive quantity x . For conciseness we denote by $u(x)$ the quantity $x - \lfloor x/B \rfloor$ ($u(x) \in [0, B]$). We thus have $LTL_t(x) = K_t + \frac{k}{B}(x - u(x)) + f(u(x))$. On one hand we can write that $LTL_t(x) \geq K_t + f(u(x))$. It follows by definition of λ that $LTL_t(x) \geq K_t + \lambda + \frac{k}{2B}x$. On the other hand we have $LTL_t(x) \leq K_t + \frac{k}{B}x + \mu$. We define $A_t = K_t + \lambda$ and $b = \frac{k}{2B}$. From what precedes and the inequality $k \leq K_t$, procurement costs $LTL_t(x)$ are 2-linearly sandwiched if the inequality $\mu - 2\lambda \leq k$ holds.

To show that $\mu - 2\lambda \leq k$, let $\varepsilon > 0$ be a fixed value. By definition of μ , there necessary exists at least one value $y \in [0, B]$ such that $f(y) - y\frac{k}{B} \geq \mu - \varepsilon$. For any $u \leq y$, using the fact that $0 \leq f(u) \leq k$, we have :

$$f(u) - \frac{k}{2B}u \geq -\frac{k}{2B}y \geq \frac{1}{2}(\mu - \varepsilon - f(y)) \geq \frac{1}{2}(\mu - \varepsilon - k)$$

For any $u \geq y$ we have, since f is non-decreasing :

$$f(u) - \frac{k}{2B}u \geq f(y) - \frac{k}{2B}u \geq (\mu - \varepsilon + \frac{k}{B}y) - \frac{k}{2B}u \geq \mu - \varepsilon - \frac{1}{2}k$$

Observe that $\mu \geq 0$ since $f(0) = 0$. It results that $\lambda \geq \frac{1}{2}(\mu - k) - \varepsilon$. Since the inequality holds for any $\varepsilon > 0$, we obtain that certainly $2\lambda \geq \mu - k$, which concludes the proof.

Annexe C

Proofs of chapter 5

C.1 Proof of Lemma 5.3.1

Consider an optimal policy $\pi^{\text{OPT}} = (\pi_0^{\text{OPT}}, \pi_1^{\text{OPT}}, \dots, \pi_N^{\text{OPT}})$ for the OWMR system with backlog. For all $j = 0, \dots, N$, let $\bar{\pi}_j$ be the single-echelon policy that places the same orders as π_j^{OPT} . That is, if π^{OPT} orders demand (i, t) in period r at the warehouse and s at retailer i , $\bar{\pi}_0$ and $\bar{\pi}_i$ orders demand (i, t) in period r and s , respectively.

Note that by definition of $\bar{\pi}_j$ for all j , the ordering costs remain untouched : $\mathcal{K}(\pi^{\text{OPT}}) = \sum_{j=0}^N \mathcal{K}_j(\bar{\pi}_j)$. Therefore we simply have to prove that $\sum_{j=0}^N \Psi_j(\bar{\pi}_j) \leq \Psi(\pi^{\text{OPT}})$ and the proof will follow from the optimality of policies $\hat{\pi}_0^*, \hat{\pi}_1^*, \dots, \hat{\pi}_N^*$. Consider a demand point (i, t) and let r and s be the periods when π^{OPT} orders the demand at the warehouse and at retailer i , respectively. The carrying cost incurred by π^{OPT} for this demand is therefore equal to ψ_{rs}^{it} . On the other hand, the total cost incurred by policies $\bar{\pi}_0$ and $\bar{\pi}_i$ to carry this demand in systems (\hat{S}_0) (resp. (\hat{S}_i)) is equal to $\frac{1}{3}\omega_{rr}^{it}$ (resp. $\frac{2}{3}\psi_{ss}^{it}$) if $i \in I_W$ and $\frac{1}{2}\omega_{rr}^{it}$ (resp. $\frac{1}{2}\psi_{ss}^{it}$) if $i \in I_J$. Hence we want to show that ψ_{rs}^{it} is bounded as follows :

$$\psi_{rs}^{it} \geq \begin{cases} \frac{1}{3}\psi_{rr}^{it} + \frac{2}{3}\psi_{ss}^{it} & \text{if } i \in I_W \\ \frac{1}{2}\psi_{rr}^{it} + \frac{1}{2}\psi_{ss}^{it} & \text{otherwise.} \end{cases} \quad (\text{C.1})$$

which is precisely the sum of the costs incurred by policies $\bar{\pi}_0$ and $\bar{\pi}_j$ to carry demand d_t^i . We now prove that inequality (C.1) holds depending on the nature of retailer i .

If $i \in I_J$, we have $r = s$ since π^{OPT} is optimal and therefore

$$\psi_{rs}^{it} = \psi_{rr}^{it} = \psi_{ss}^{it} \quad (\text{C.2})$$

$$= \frac{1}{2}\psi_{rr}^{it} + \frac{1}{2}\psi_{ss}^{it} \quad (\text{C.3})$$

If $i \in I_W$, we distinguish between two cases, listed below :

$s \leq t$ or $r > t$: Properties (P2) and (P3) ensure that

$$\begin{aligned}\psi_{rs}^{it} &\geq \max \{ \psi_{rt}^{it}, \psi_{ss}^{it} \} \\ &\geq \frac{1}{3}\psi_{rr}^{it} + \frac{2}{3}\psi_{ss}^{it}\end{aligned}\tag{C.4}$$

$r \leq t < s$: We have from properties (P2) and (P3)

$$\begin{aligned}\psi_{rs}^{it} &\geq \max \{ \psi_{rt}^{it}, \psi_{ts}^{it} \} \\ &\geq \frac{1}{3}\psi_{rr}^{it} + \frac{2}{3}\psi_{ts}^{it} \\ &\geq \frac{1}{3}\psi_{rr}^{it} + \frac{2}{3}\psi_{ss}^{it}\end{aligned}\tag{C.5}$$

Hence in all cases, inequality (C.1) is satisfied and we have $\Psi(\pi^{\text{OPT}}) \geq \sum_{j=0}^N \Psi_j(\bar{\pi}_j)$. As a consequence the following inequality holds :

$$\mathcal{C}^* = \mathcal{C}(\pi^{\text{OPT}}) \geq \sum_{j=0}^N \mathcal{C}_j(\bar{\pi}_j)\tag{C.6}$$

Inequality (C.6) and the optimality of policies $\hat{\pi}_j^*$ for problems (\hat{S}_j) then conclude the proof :

$$\begin{aligned}\sum_{j=0}^N \mathcal{C}_j(\hat{\pi}_j^*) &\leq \sum_{j=0}^N \mathcal{C}_j(\bar{\pi}_j) \\ &\leq \mathcal{C}^*\end{aligned}$$

C.2 Proof of Lemma 5.3.2

We first prove that the resulting policy is feasible. Note that for all $i = 1, \dots, N$ and $t = 1, \dots, T$, the pair of orders $[r^{it}, s^{it}]$ chosen by the algorithm to serve d_t^i satisfies $r^{it} \leq s^{it}$. Therefore we simply need to prove that π^u opens $[r^{it}, s^{it}]$. Let r and s be the ordering periods when $\hat{\pi}_0$ and $\hat{\pi}_i$ order to serve demand d_t^i , respectively. If $s \in R$, we have by definition $s^+ = s^- = s$ and the pair $[s^+, s^+]$ is opened as it uses existing orders in the single-echelon policies $\hat{\pi}_0$ and $\hat{\pi}_i$. If $s \notin R$, $\hat{\pi}_0$ does not place a warehouse order in period s , hence π^u places a retailer i order in period s^+ , either from the single-echelon policy $\hat{\pi}_i$ or from Step 1 of the algorithm. In addition if $i \in I_J$, π^u also places a retailer i order in period s^- from Step 2 of the algorithm. Since π^u uses the same warehouse orders as $\hat{\pi}_0$, policy π^u is uncrossed and feasible.

We now prove that the final cost incurred by π^u is bounded. It is straightforward to check that Step 1 of the algorithm at most doubles the ordering costs incurred by policies $\hat{\pi}_i$ since for all $i = 1, \dots, N$, π^u adds at most one order for each existing order in $\hat{\pi}_i$. Moreover

the uncrossing algorithm leaves the orders of $\hat{\pi}_0$ untouched, hence if $I_W \neq \emptyset$ we only have to prove that the following inequalities hold :

$$\Psi(\pi^u) \leq 3 \sum_{j=0}^N \widehat{\Psi}_j(\hat{\pi}_j) \quad (\text{C.7})$$

Let $i = 1 \dots, N$ and $t = 1, \dots, T$ and focus on demand point (i, t) . Let r be the period when $\hat{\pi}_0$ orders d_t^i and s the period when $\hat{\pi}_i$ orders d_t^i in their single-echelon solution. We now bound the final carrying cost incurred in π^u to serve d_t^i . More precisely, we want to show that for a retailer i and periods t, r, s , the following inequality is satisfied :

$$\psi_{r^{it}, s^{it}}^{it} \leq \begin{cases} 3 \cdot \frac{1}{3} \psi_{rt}^{it} + 3 \cdot \frac{2}{3} \psi_{ss}^{it} = \psi_{rt}^{it} + 2\psi_{ss}^{it} & \text{if } i \in I_W \text{ and } r \leq t & (\text{C.8}) \\ 3 \cdot \frac{1}{3} \psi_{rr}^{it} + 3 \cdot \frac{2}{3} \psi_{ss}^{it} = \psi_{rr}^{it} + 2\psi_{ss}^{it} & \text{if } i \in I_W \text{ and } r > t & (\text{C.9}) \\ 2 \cdot \frac{1}{2} \psi_{rr}^{it} + 2 \cdot \frac{1}{2} \psi_{ss}^{it} = \psi_{rr}^{it} + \psi_{ss}^{it} & \text{if } i \in I_J & (\text{C.10}) \end{cases}$$

If $i \in I_J$, the order of retailer i that serves d_t^i is synchronized with a warehouse order either in period s^- or s^+ . Note that by definition we necessarily have $r \leq s^-$ if $r \leq s$ and $r \geq s^+$ if $r > s$. Hence if $r \leq s$, we have from property (P4)

$$\begin{aligned} \psi_{s^- s^-}^{it} &\leq \psi_{rr}^{it} \\ &\leq \psi_{rr}^{it} + \psi_{ss}^{it} \end{aligned}$$

On the other hand if $r > s$, π^u uses the pair $[s^+, s^+]$ to serve d_t^i and from property (P2) and (P3) the following inequality holds

$$\begin{aligned} \psi_{s^+ s^+}^{it} &\leq \begin{cases} \psi_{rr}^{it} & \text{if } r > t \\ \psi_{ss}^{it} & \text{otherwise} \end{cases} \\ &\leq \max \{ \psi_{rr}^{it}, \psi_{ss}^{it} \} \\ &\leq \psi_{rr}^{it} + \psi_{ss}^{it} \end{aligned}$$

Hence if $i \in I_J$ inequality (C.10) is satisfied for all t, r, s .

In the case where i is a W -retailer, we again derive a lower bound for the carrying cost depending on how r, s and t are ordered. If $r \leq s$ and $r \leq t$, the final cost incurred in π^u to serve d_t^i is $\psi_{s^- s^-}^{it}$. Recall that in this case, we have $r \leq s^-$. Therefore we can bound the

carrying cost incurred to serve demand d_t^i using properties (P3), (P4) and (P5) :

$$\begin{aligned} \psi_{s^-s}^{it} &\leq \begin{cases} \psi_{s^-t}^{it} + \psi_{ss}^{it} & \text{if } r \leq s \leq t \\ \psi_{s^-t}^{it} + \psi_{ts}^{it} & \text{if } r \leq t < s \end{cases} \\ &\leq \begin{cases} \psi_{rt}^{it} + \psi_{ss}^{it} & \text{if } r \leq s \leq t \\ \psi_{rt}^{it} + \psi_{ts}^{it} & \text{if } r \leq t < s \end{cases} \\ &\leq \psi_{rt}^{it} + 2\psi_{ss}^{it} \end{aligned}$$

On the other hand if $t < r \leq s$, properties (P3) and (P5) ensure that

$$\begin{aligned} \psi_{s^-s}^{it} &\leq \psi_{ts}^{it} \\ &\leq 2\psi_{ss}^{it} \\ &\leq \psi_{rr}^{it} + 2\psi_{ss}^{it} \end{aligned}$$

Finally if $r > s$, we use properties (P2) and (P3) to bound the final carrying cost :

$$\begin{aligned} \psi_{s^+s^+}^{it} &\leq \begin{cases} \psi_{rr}^{it} & \text{if } t < s^+ \\ \psi_{ss}^{it} & \text{otherwise} \end{cases} \\ &\leq \max \{ \psi_{rr}^{it}, \psi_{ss}^{it} \} \\ &\leq \psi_{rr}^{it} + \psi_{ss}^{it} \end{aligned}$$

We conclude that if $i \in I_W$ inequalities (C.8) and (C.9) are satisfied.

Therefore if π^u opens the pair of orders $[r^{it}, s^{it}]$, inequalities (C.8), (C.9) and (C.10) are always satisfied. As a consequence inequality (C.14) holds and the proof follows :

$$\mathcal{C}(\pi^u) \leq 3 \sum_{j=0}^N \widehat{\Psi}_j(\widehat{\pi}_j) + \mathcal{K}_0(\widehat{\pi}_0) + 2 \sum_{i=1}^N \mathcal{K}_i(\widehat{\pi}_i)$$

To conclude the proof, notice that when $I_W = \emptyset$, inequality (C.10) implies that

$$\Psi(\pi^u) \leq 2 \sum_{j=0}^N \widehat{\Psi}_j(\widehat{\pi}_j)$$

and thus one can modify the upper bound in the special case of the JRP :

$$\mathcal{C}(\pi^u) \leq 2 \sum_{j=0}^N \widehat{\Psi}_j(\widehat{\pi}_j) + \mathcal{K}_0(\widehat{\pi}_0) + 2 \sum_{i=1}^N \mathcal{K}_i(\widehat{\pi}_i)$$

C.3 Proof of Lemma 5.4.1

We prove this inequality in a similar manner as we did for Lemma 5.3.1. Consider an optimal policy $\pi^{\text{OPT}} = (\pi_0^{\text{OPT}}, \pi_1^{\text{OPT}}, \dots, \pi_N^{\text{OPT}})$ for the OWMR system with lost-sales. For all $j = 0, \dots, N$, we define $\bar{\pi}_j$ as the policy that orders in the same periods as π_j^{OPT} and fills or looses demands exactly as the latter does. Obviously, we again have $\mathcal{K}(\pi^{\text{OPT}}) = \sum_{j=0}^N \mathcal{K}_j(\bar{\pi}_j)$. We now prove that $\sum_{j=0}^N \Psi_j(\bar{\pi}_j) \leq \Psi(\pi^{\text{OPT}})$. Consider a retailer i and a demand point (i, t) . If d_t^i is lost in π^{OPT} , then by construction it is lost in $\bar{\pi}_0$ and $\bar{\pi}_i$ and we directly have :

$$\psi^{it} = \frac{1}{2}\psi^{it} + \frac{1}{2}\psi^{it} \quad (\text{C.11})$$

Now assume the demand is filled and let r and s be the periods when π^{OPT} orders d_t^i at the warehouse and at the retailer, respectively. The holding cost incurred by π^{OPT} for d_t^i is equal to ψ_{rs}^{it} . If i is a J -retailer, we have $r = s$ and therefore

$$\psi_{rs}^{it} = \psi_{rr}^{it} = \psi_{ss}^{it} = \frac{1}{2}\psi_{rr}^{it} + \frac{1}{2}\psi_{ss}^{it} \quad (\text{C.12})$$

On the other hand if i is a W -retailer, we can bound this cost using properties (P2) and (P3) :

$$\begin{aligned} \psi_{rs}^{it} &\geq \max \{ \psi_{rt}^{it}, \psi_{ss}^{it} \} \\ &\geq \frac{1}{2}\psi_{rt}^{it} + \frac{1}{2}\psi_{ss}^{it} \end{aligned} \quad (\text{C.13})$$

In both cases, the sum of the holding costs incurred by $\bar{\pi}_0$ and $\bar{\pi}_i$ to serve demand (i, t) in systems (\tilde{S}_0) and (\tilde{S}_i) is lower than or equal to the cost incurred in π^{OPT} to serve the same demand. Therefore from equations (C.11), (C.12) and (C.13) we conclude $\Psi(\pi^{\text{OPT}}) \geq \sum_{j=0}^N \Psi_j(\bar{\pi}_j)$. The optimality of policies $\tilde{\pi}_j^*$ for problems (\tilde{S}_j) then yields :

$$\begin{aligned} \sum_{j=0}^N \mathcal{E}_j(\tilde{\pi}_j^*) &\leq \sum_{j=0}^N \mathcal{E}_j(\bar{\pi}_j) \\ &= \sum_{j=0}^N \mathcal{K}_j(\bar{\pi}_j) + \Psi_j(\bar{\pi}_j) \\ &\leq \mathcal{K}(\pi^{\text{OPT}}) + \Psi(\pi^{\text{OPT}}) \\ &\leq \mathcal{E}^* \end{aligned}$$

C.4 Proof of Lemma 5.4.2

To prove the feasibility of the resulting policy, we refer the reader to the proof of Lemma 5.3.2. Similarly, it is straightforward to see that $\mathcal{K}(\tilde{\pi}^u) \leq \mathcal{K}_0(\tilde{\pi}_0) + 2 \sum_{i=1}^N \mathcal{K}_i(\tilde{\pi}_i)$, since for every order in $\tilde{\pi}_i$ Step 1 of the algorithm places at most one additional order to

uncross the policies. Therefore we only have to prove that

$$\Psi(\tilde{\pi}^u) \leq 2 \sum_{j=0}^N \Psi_j(\tilde{\pi}_j) \quad (\text{C.14})$$

Let $i = 1 \dots, N$ and $t = 1, \dots, T$ and focus on demand point (i, t) . If the d_t^i is lost in $\tilde{\pi}_0$ or $\tilde{\pi}_i$, it incurs a cost of at least $\frac{1}{2}\psi^{it}$ in these policies. Since by construction it is also lost in $\tilde{\pi}^u$, it incurs a cost of exactly ψ^{it} in the final solution. Now assume the demand is filled both in $\tilde{\pi}_0$ and $\tilde{\pi}_i$ and let r and s be the respective ordering periods of these policies for this demand. We want to prove that the final holding cost incurred in $\tilde{\pi}^u$ to serve d_t^i is at most the sum of the holding cost it incurs in policies $\tilde{\pi}_0$ and $\tilde{\pi}_i$. This final cost depends on the type of retailer and on the respective position of r and s :

- If $i \in I_J$ and $r \leq s$, $\tilde{\pi}^u$ orders d_t^i in period s^- at the warehouse and at the retailer for a final cost of $\psi_{s^-s^-}^{it}$. The holding cost incurred by policy $\tilde{\pi}_0$ (resp. $\tilde{\pi}_i$) to serve this demand is equal to ψ_{rr}^{it} (resp. ψ_{ss}^{it}). Since $s^- \geq r$, we have from property (P3) :

$$\begin{aligned} \psi_{s^-s^-}^{it} &\leq \psi_{rr}^{it} \\ &\leq \psi_{rr}^{it} + \psi_{ss}^{it} \end{aligned} \quad (\text{C.15})$$

- If $i \in I_J$ and $r > s$, $\tilde{\pi}^u$ orders the demand in period s^+ at the warehouse and at the retailer for a final cost of $\psi_{s^+s^+}^{it}$. The holding cost incurred by policy $\tilde{\pi}_0$ (resp. $\tilde{\pi}_i$) to serve this demand is equal to ψ_{rr}^{it} (resp. ψ_{ss}^{it}). Since $s^+ \geq s$, we have from property (P3) :

$$\begin{aligned} \psi_{s^+s^+}^{it} &\leq \psi_{ss}^{it} \\ &\leq \psi_{rr}^{it} + \psi_{ss}^{it} \end{aligned} \quad (\text{C.16})$$

- If $i \in I_W$ and $r \leq s$, $\tilde{\pi}^u$ orders the demand in period s^- at the warehouse and in period s at the retailer for a final cost of $\psi_{s^-s}^{it}$. Recall that by definition, we have $r \leq s^-$. The holding cost incurred by policy $\tilde{\pi}_0$ (resp. $\tilde{\pi}_i$) to serve this demand is equal to ψ_{rt}^{it} (resp. ψ_{ss}^{it}). Thus we have from property (P3) and (P4) :

$$\psi_{s^-s}^{it} \leq \psi_{r,s}^{it} \leq \psi_{rt}^{it} + \psi_{ss}^{it} \quad (\text{C.17})$$

- If $i \in I_W$ and $s < r$, $\tilde{\pi}^u$ orders the demand in period s^+ at the warehouse and at the retailer for a final cost of $\psi_{s^+s^+}^{it}$. The holding cost incurred by policy $\tilde{\pi}_0$ (resp. $\tilde{\pi}_i$) to serve this demand is equal to ψ_{rt}^{it} (resp. ψ_{ss}^{it}). Since $s^+ > s$ we have from property (P3) :

$$\begin{aligned} \psi_{s^+s^+}^{it} &\leq \psi_{ss}^{it} \\ &\leq \psi_{rt}^{it} + \psi_{ss}^{it} \end{aligned} \quad (\text{C.18})$$

Note that in inequalities (C.15)-(C.18), the right-hand side corresponds exactly to twice the sum of the holding cost incurred in policy $\tilde{\pi}_0$ and $\tilde{\pi}_i$ to serve demand d_t^i . As a consequence in all cases the holding or penalty cost incurred to serve or loose this demand in $\tilde{\pi}^u$ is at most twice the sum of the costs incurred for the same demand in policies $\tilde{\pi}_0$ and $\tilde{\pi}_i$. We conclude that inequality (C.14) holds and the proof follows.

Bibliographie

- Absi, N., S. Kedad-Sidhoum. 2008. The multi-item capacitated lot-size problem with setup times and shortage costs. *Eur. J. Oper. Res.* **185** 1351–1374.
- Aggarwal, A., J.K. Park. 1993. Improved algorithm for economic lot-size problems. *Oper. Res.* **41**(3) 549–571.
- Aksen, D., K. Altinkemer, S. Chand. 2003. The single-item lot-sizing problem with immediate lost sales. *Eur. J. Oper. Res.* **147** 558–566.
- Aksoy, Y., S.S. Erenguc. 1988. Multi-item inventory models with co-ordinated replenishments : A survey **8**(1) 63–73.
- Anily, S., M. Tzur, L.A. Wolsey. 2009. Multi-item lot-sizing with joint set-up costs. *Math. Programming* **119** 79–94.
- Arkin, E., D. Joneja, R. Roundy. 1989. Computational complexity of uncapacitated multi-echelon production planning problems. *Oper. Res. Letters* **8** 61–66.
- Axsäter, S. 1982. Worst case performance for lot sizing heuristics. *Eur. J. Oper. Res.* **9**(4) 339–343.
- Axsäter, S. 2006. *Inventory Control (second edition)*. Springer.
- Bakker, Monique, Jan Riezebos, Ruud H. Teunter. 2012. Review of inventory systems with deterioration since 2001. *European Journal of Operational Research* **221**(2) 275–284.
- Barbosa, LC, M Friedman. 1978. Deterministic inventory lot size models - a general root law. *Management Sci.* **24**(8) 819–826.
- Bellman, R. 1957. *Dynamic Programming*. Princeton University Press.
- Benkherouf, Lakdere, Brian H. Gilding. 2009. On a class of optimization problems for finite time horizon inventory models. *SIAM J. Control Optim.* **48**(2) 993–1030.
- Bitran, GR, TL Magnanti, HH Yanasse. 1982. Computational complexity of the capacitated lot size problem. *Management Sci.* **28**(10) 1174–1186.
- Bitran, GR, TL Magnanti, HH Yanasse. 1984. Approximation methods for the uncapacitated dynamic lot size problem. *Management Sci.* **30**(9) 1121–1140.
- Brahimi, N., S. Dauzère-Peres, N. M. Najid, A. Nordli. 2006. Single item lot sizing problems. *Eur. J. Oper. Res.* **168** 1–16.
- Chan, L.M.A., A. Muriel, Z.-J.M. Shen, D. Simchi-Levi, C.-P. Teo. 2000. Effective zero-inventory-ordering policies for the single-warehouse multiretailer problem with piecewise linear cost structures. *Management Sci.* **48**(11) 1446–1460.
- De Matteis, J.J., A.G. Mendoza. 1968. An economic lot sizing technique. *IBM Systems Journal* **7**(1) 30–38.

- Donaldson, W.A. 1977. Inventory replenishment policy for a linear trend in demand – an analytical solution. *Operational Research Quarterly* **28**(3) 663–670.
- Federgruen, A., M. Tzur. 1991. A simple forward algorithm to solve the general dynamic lot-sizing models with n periods in $O(n \log n)$ or $O(n)$ time. *Management Sci.* **37**(8) 909–925.
- Federgruen, A., M. Tzur. 1993. The dynamic lot-sizing model with backlogging : A simple $O(n \log n)$ algorithm and minimal forecast horizon procedure. *Naval Research Logistics* **40**(4) 459–478.
- Federgruen, A., M. Tzur. 1994. The joint replenishment problem with time-varying parameters : Efficient, asymptotic and epsilon-optimal solutions. *Oper. Res.* **42** 1067–1087.
- Federgruen, A., M. Tzur. 1999. Time-partitioning heuristics : Application to one-warehouse, multi-item, multi-retailer lot-sizing problems. *Naval Research Logistics* **46** 463–486.
- Federgruen, A., M. Wang. 2013. Inventory models with shelf age and delay dependent inventory costs. *MSOM Conference 2013, Fontainebleau, France*.
- Feige, U. 1998. A threshold of $\log(n)$ for approximating set-cover. *J. of the ACM* **45** 634–652.
- Florian, M., M. Klein. 1971. Deterministic production planning with concave costs and capacity constraints. *Management Sci.* **18**(1) 12–20.
- Florian, M., J.K. Lenstra, A.H.G Rinnooy Kan. 1980. Deterministic production planning : Algorithms and complexity. *Management Sci.* **26**(7) 669–679.
- Garey, M.R., D.S. Johnson. 1979. *Computer and Intractability : A guide to the theory of NP-completeness*. W.H. Freeman.
- Giri, B.C., K.S. Chaudhuri. 1997. Deterministic models of perishable inventory with stock-dependent demand rate and nonlinear holding cost. *Eur. J. Oper. Res.* **105**(3) 467–474.
- Goh, M. 1994. EOQ models with general demand and holding cost functions. *Eur. J. Oper. Res.* **73**(1) 50–54.
- Goyal, S.K., B.C. Giri. 2001. Recent trends in modeling of deteriorating inventory. *Eur. J. Oper. Res.* **134**(1) 1–16.
- Goyal, S.K., B.C. Giri. 2003. The production-inventory problem of a product with time varying demand, production and deterioration rates. *Eur. J. Oper. Res.* **147**(1) 549–557.
- Hariga, M.A. 1994. The lot-sizing problem with continuous time-varying demand and shortages. *J. of Operations Research Society* **45**(10) 827–837.
- Hariga, M.A. 1996. Lot-sizing heuristics for continuous time-varying demand and shortages. *Computers & Oper. Res.* **23**(12) 1211–1217.
- Harris, F.W. 1913. How many parts to make at once. *Factory, the Magazine of management* **10** 135–136.
- Henery, R.J. 1979. Inventory replenishment policy for increasing demand. *J. of Operations Research Society* **30**(7) 611–617.
- Henery, R.J. 1990. Inventory replenishment policy for cyclic demand. *J. of Operations Research Society* **41**(7) 639–643.
- Hoesel, C.P.M. van, A.P.M. Wagelmans. 1996. An $O(t^3)$ algorithm for the economic lot-sizing problem with constant capacities. *Management Sci.* **42**(1) 142–150.
- Hoesel, C.P.M. van, A.P.M. Wagelmans. 2001. Fully polynomial approximation schemes for single-item capacitated economic lot-sizing problems. *Math. Oper. Res.* **26**(2) 339–357.

- Joneja, D. 1990. The joint-replenishment problem : New heuristics and worst case performance bounds. *Oper. Res.* **38**(4) 711–723.
- Kao, E. 1979. A multi product dynamic lot size model with individual and joint setup costs. *Oper. Res.* **27** 279–289.
- Khouja, M., S. Goyal. 2008. A review of the joint replenishment problem literature : 1989-2005. *Eur. J. Oper. Res.* **186**(1) 1–16.
- Kiefer, J. 1953. Sequential minimax search for a maximum. *Proc. Amer. Math. Soc.* **4** 502 – 506.
- Küçükyavuz, S., Y. Pochet. 2009. Uncapacitated lot sizing with backlogging : The convex hull. *Math. Programming* **118**(1) 151–175.
- Levi, R., G. Janakiraman, M. Nagarajan. 2008a. A 2-Approximation algorithm for stochastic inventory control models with lost sales. *Math. Oper. Res.* **33**(2) 351–374.
- Levi, R., A. Lodi, M. Sviridenko. 2008b. Approximation algorithms for the capacitated multi-item lot-sizing problem via flow-cover inequalities. *Math. Oper. Res.* **33**(2) 461–474.
- Levi, R., M. Pal, R.O. Roundy, D.B. Shmoys. 2007. Approximation algorithms for stochastic inventory control models. *Math. Oper. Res.* **32**(2) 284–302.
- Levi, R., R.O. Roundy, D.B. Shmoys. 2006. Primal-dual algorithms for deterministic inventory problems. *Math. Oper. Res.* **31**(2) 267–284.
- Levi, R., R.O. Roundy, D.B. Shmoys, M. Sviridenko. 2008c. A constant approximation algorithm for the one-warehouse multiretailer problem. *Management Sci.* **54**(8) 763–776.
- Levi, R., R.O. Roundy, D.B. Shmoys, V.A. Truong. 2008d. Approximation algorithms for capacitated stochastic inventory control models. *Oper. Res.* **56**(5) 1184–1199.
- Li, C.-L., V.N. Hsu, W.-Q. Xiao. 2004. Dynamic lot sizing with batch ordering and truckload discounts. *Oper. Res.* **52**(4) 639–654.
- Liu, X., F. Chu, C. Chu, C. Wang. 2007. Lot-sizing with bounded inventory and lost-sales. *International J. of Production Res.* **45**(24) 5881–5894.
- Love, S.F. 1972. A facilities in series model with nested schedules. *Management Sci.* **18**(5) 327–338.
- Manne, A.S. 1958. Programming of economic lot sizes. *Management Sci.* **4** 115–135.
- Monahan, J.P. 1984. A quantity discount pricing model to increase vendor profits. *Management Sci.* **30**(6) 720–726.
- Muckstadt, M., R.O. Roundy. 1993. *Analysis of multistage production systems, Handbooks in OR & MS*, vol. 4, Logistics of Production and Inventory, chap. 2. North Holland Amsterdam, 59–131.
- Nahmias, S. 1982. Perishable inventory theory : A review. *Oper. Res.* **30**(3) 680–708.
- Nonner, T., A. Souza. 2009. A 5/3-approximation algorithm for joint replenishment with deadlines. *Proc. of the 3rd Annual International Conference on Combinatorial Optimization and Applications, Huangshan, China* 24–35.
- Nonner, T., M. Sviridenko. 2013. An efficient polynomial-time approximation scheme for the joint replenishment problem. *Proc. 16th Conf. on Integer Programming and Combinatorial Optimization (IPCO'13), Valparaíso, Chile.* 314–323.
- Papadimitriou, C.H. 1993. *Computational Complexity*. Addison-Wesley.
- Pochet, Y., L. Wolsey. 2006. *Production planning by mixed integer programming*. Springer.

- Pochet, Y., L.A. Wolsey. 1988. Lot-size models with backlogging : Strong reformulations and cutting planes. *Math. Programming* **40** 317–335.
- Pochet, Y., L.A. Wolsey. 1994. Polyhedra for lot-sizing with wagner–whitin costs. *Math. Programming* **67** 297–323.
- Porteus, E.L. 2002. *Foundations of Stochastic Inventory Theory*. Stanford University Press.
- Raafat, F. 1991. Survey of literature on continuously deteriorating inventory model. *J. of Operations Research Society* **42** 27–37.
- Resh, M, M Friedman, LC Barbosa. 1976. On a General Solution of the Deterministic Lot Size Problem with Time-Proportional Demand. *Oper. Res.* **24**(4) 718–725.
- Roundy, R.O. 1985. A 98%-effective integer-ratio lot-sizing for one-warehouse multi-retailer systems. *Management Sci.* **31** 1416–1430.
- Roundy, R.O. 1986. A 98%-effective lot-sizing rule for a multi-product, multi-stage production/inventory system. *Math. Oper. Res.* **11**(4) 699–727.
- Roundy, R.O. 1993. Efficient, effective lot-sizing for multi-product, multi-stage production systems. *Oper. Res.* **41** 371–325.
- Sandbothe, R.A., J.L. Thompson. 1990. A forward algorithm for the capacitated lot size model with stockouts. *Oper. Res.* **38**(3) 474–486.
- Schwarz, L. B. 1973. A simple continuous review deterministic one-warehouse n-retailer inventory problem. *Management Sci.* **19** 555–566.
- Shen, Z.-J., J. Shu, D. Simchi-Levi, C.-P. Teo, J. Zhang. 2009. Approximation algorithms for general one-warehouse multi-retailer systems. *Naval Research Logistics* **56** 642–658.
- Silver, E.A. 1979. A simple inventory replenishment rule for a linear trend in demand. *J. of Operations Research Society* **30**(1) 71–75.
- Silver, E.A., H.C. Meal. 1973. A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. *Production and Inventory Management* **14**(2) 64–74.
- Stauffer, G. 2012. Using the economical order quantity formula for inventory control in one-warehouse multiretailer systems. *Naval Research Logistics* **59**(3-4) 285–297.
- Stauffer, G., G. Massonnet, C. Rapine, J.-P. Gayon. 2011. A simple and fast 2-approximation for the one-warehouse multi-retailers problem. *ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 67–79.
- Taft, E.W. 1918. The most economical production lot. *Iron Age* **101** 1410–1412.
- Teng, J.-T. 1994. A note on inventory replenishment policy for increasing demand. *J. of Operations Research Society* **45** 1335–1337.
- Teng, J.-T. 1996. A deterministic replenishment model with linear trend in demand. *Oper. Res. Letters* **19** 33–41.
- Teng, J.-T., H.-J. Chang, C.-Y. Dye, C.H. Hung. 2002. An optimal replenishment policy for deteriorating items with time-varying demand and partial backlogging. *Oper. Res. Letters* **30** 387–393.
- Teng, J.-T., M.-S. Chern, H.-L. Yang. 1997. An optimal recursive method for various inventory replenishment models with increasing demand and shortages. *Naval Research Logistics* **44** 791–806.

- Teng, J.-T., M.-S. Chern, H.-L. Yang, Y.J. Wang. 1999. Deterministic lot-size inventory models with shortages and deterioration for fluctuating demand. *Oper. Res. Letters* **24** 65–72.
- Teng, J.-T., L.-Y. Ouyang, L-H Chen. 2007. A comparison between two pricing and lot-sizing models with partial backlogging and deteriorated items. *International J. of Production Economics* **105**(1) 190 – 203.
- Van den Heuvel, W., A.P.M. Wagelmans. 2010. Worst-case analysis for a general class of online lot-sizing heuristics. *Operations Research* **58** 59–67.
- Vazirani, V.V. 2001. *Approximation Algorithms*. Springer.
- Veinott, A. 1969. Minimum concave cost solutions of leontief substitution models of multi-facility inventory systems. *Oper. Res.* **17** 262–291.
- Wagelmans, A.P.M., C.P.M. van Hoesel, A. Kolen. 1992. Economic lot sizing : an $O(n \log n)$ algorithm that runs in linear time in the wagner-whitin case. *Oper. Res.* **40**(S1) 145–156.
- Wagner, H.M., T. Whitin. 1958. Dynamic version of the economic lot size model. *Management Sci.* **5**(1) 89–96.
- Williamson, D.P., D.B. Shmoys. 2010. *The design of Approximation Algorithms*. Cambridge Press University.
- Wilson, RH. 1934. A scientific routine for stock control. *Harvard business review* **13** 116–128.
- Zangwill, W.I. 1966a. A deterministic multi-period production scheduling model with backlogging. *Management Sci.* **13**(1) 105–119.
- Zangwill, W.I. 1966b. A deterministic multiproduct, multi-facility production and inventory model. *Oper. Res.* **14** 486–507.
- Zipkin, PH. 2000. *Foundations of Inventory Management*. McGraw-Hill, New York.

Résumé :

La gestion de stocks a toujours été une composante majeure de la recherche opérationnelle et de nombreux modèles issus de problématiques industrielles concrètes présentent un intérêt à la fois théorique et pratique. Dans ce cadre, nos travaux se concentrent sur plusieurs problèmes classiques du domaine, pour lesquels on ne connaît pas à ce jour de méthode efficace permettant de calculer une solution optimale. Plus précisément, nous étudions plusieurs modèles déterministes, dans lesquels les demandes des clients sont connues à l'avance, et nous proposons pour chacun d'entre eux des techniques d'approximation qui permettent de construire des solutions approchées en un temps de calcul raisonnable.

Nous considérons d'abord des modèles à temps continu composés d'un entrepôt, dont la demande et les coûts relatifs au stockage des produits dépendent du temps. Nous présentons une technique simple basée sur l'équilibrage des différents coûts encourus par le système pour obtenir des approximations pour une large classe de problèmes de ce type.

Dans un deuxième temps, nous étudions un problème à temps discret NP-difficile, dans lequel un entrepôt central approvisionne plusieurs détaillants faisant face à la demande de leurs clients. Nous introduisons une nouvelle technique de décomposition du problème en sous-systèmes plus simples, faciles à résoudre grâce à des techniques classiques de la littérature. Enfin, nous présentons une méthode de recombinaison originale permettant d'obtenir une politique réalisable pour le problème d'origine à partir des sous-solutions obtenues. L'algorithme ainsi défini peut être adapté à plusieurs extensions du modèle de base incluant des structures de coûts plus générales ou encore offrant la possibilité de servir certaines demandes en retard.

Abstract :

Inventory management has always been a major component of the field of operations research and numerous models derived from the industry aroused the interest of both the researchers and the practitioners. Within this framework, our work focuses on several classical inventory problems, for which no tractable method is known to compute an optimal solution. Specifically, we study deterministic models, in which the demands of the customers are known in advance, and we propose approximation techniques for each of the corresponding problems that build feasible approximate solutions while remaining computationally tractable.

We first consider continuous-time models with a single facility when demand and holding costs are time-dependent. We present a simple technique that balances the different costs incurred by the system and use this concept to build approximation methods for a large class of classical single-echelon problems.

The second part of our work focuses on a discrete time model, in which a central warehouse supplies several retailers facing the final customers demands. This problem is known to be NP-hard, thus finding an optimal solution in polynomial time is unrealistic unless $P=NP$. We introduce a new decomposition of the system into simple subproblems, for which an optimal solution can be computed easily. We then present a recombination method that builds a feasible policy for the original problem from the solutions to these subsystems. The resulting algorithm has a constant performance guarantee and can be adapted to several extensions of the model, including more general cost structures and problems with backlogging or lost-sales.