École doctorale IAEM Lorraine

Optimization methods for multi-level lot-sizing problems

THÈSE

présentée et soutenue publiquement le 22 septembre 2017

pour l'obtention du

Doctorat de l'Université de Lorraine
(mention automatique)

par

Guillaume Goisque

Composition du jury

Président :	Anass Nagih
Rapporteurs :	Albert Wagelmans Bernard Penz
Examinateurs :	Safia Kedad-Sidhoum Céline Gicquel
Directeur de thèse :	Christophe Rapine

Laboratoire de Génie Industriel, de Production et de Maintenance (EA 3096)
Université de Lorraine

Acknowledgements

This thesis has been prepared within the LGIPM laboratory (Génie Industriel, Production et Maintenance).

I want to thank especially my thesis director, Christophe Rapine, for his great quality supervision. I thank him for his continuous support, his precious advice, and the countless hours spent on our collaborative work.

I am very grateful to the rapporteurs of the thesis, Mr Wagelmans Albert and Mr Penz Bernard.

I also thank the other members of the jury: Mrs Gicquel Céline, Mrs Kedad-Sidhoum Safia and Mr Nagih Anass.

I would also like to extend my gratitude to the members of the LGIPM who helped me support this important work.

Finally, I thank my family and my friends for their needed support and encouragement.

Contents

French abstract of the thesis

We provide a French abstract of the thesis to comply with the regulation of Université de Lorraine. The rest of the document is written in English.

Méthodes d'optimisation pour la gestion de stocks multi-échelon

Introduction

Dans un contexte de mondialisation, la gestion de chaînes logistiques est un problème toujours plus important. L'augmentation des capacités de transport, la libéralisation des échanges à une échelle mondiale, obligent les entreprises à sans cesse réduire leurs coûts de production et de distribution afin de faire face à la compétition. La gestion des stocks est un sujet de première importance : les coûts résultant du stockage des produits représentent des sommes considérables, et il est crucial d'éviter toute perturbation sur la chaîne, de sorte que la demande des clients puisse être satisfaite dans les délais. Dans ce contexte, la gestion des stocks vise à proposer des outils efficaces et des méthodes d'optimisation pour minimiser ces coûts. La planification de production concerne l'organisation des flux physiques de composants ou de produits finis le long d'une chaîne logistique. Ce système peut être un atelier de production de grande taille pour lequel il est nécessaire de répondre à des questions telles que comment acheminer les composants le long de la chaîne, quand et quelle quantité produire, quand et quelle quantité commander auprès du fournisseur, et où et quand stocker. La production et la distribution de ces produits sont souvent effectuées par lots, ce qui permet d'effectuer d'importantes économies d'échelles. Les systèmes de production peuvent être modélisés comme un ensemble de niveaux sur lesquels s'effectuent le processus de fabrication. Ce processus se décompose en un flux de produits, allant d'un niveau à un autre, suivant une suite de procédures telles que des processus d'assemblage, de production ou de distribution. Les niveaux au sommet de la chaîne commandent des matières premières ou différents composants auprès d'un fournisseur extérieur. Les niveaux en fin de chaîne doivent satisfaire la demande des clients.

Dans le domaine de la recherche opérationnelle, le problème du dimensionnement de lots, ou lot-sizing, introduit par Wagner and Whitin, 1958 traite de planification de production ou de réseaux de distribution intégrant des stocks. Ce problème vise à organiser le processus global tout en minimisant tous les coûts impliqués dans le système. Dans des modèles avec un horizon de temps discret et fini, le temps est décomposé en un ensemble de périodes de temps avec leurs propres paramètres et demande de clients. Les modèles de lot-sizing peuvent représenter des systèmes de planification de production, pour lesquels chaque niveau est une étape dans le processus de production, généralement une machine. Lancer la production à une période peut générer des coûts dus aux besoins de main d'oeuvre, à la consommation d'énergie occasionnée par le démarrage de la machine, au transport entre deux machines, etc. Pour cette raison, des coûts de lancement de production sont considérés dans la plupart des modèles étudiés dans la littérature. Ces coûts sont payés chaque fois que la production est lancée à une période donnée, quel que soit le nombre d'unités produites. Bien sûr les coûts de production dépendent aussi de la quantité produite, avec les coûts de production unitaire. A chaque niveau, il est possible de stocker des unités avant de les livrer au niveau suivant. Dans la plupart des études sur le sujet, un coût de stockage unitaire est pris en compte lorsque qu'une unité et stockée d'une période à un autre. Les coûts de stockage peuvent modéliser, entre autres, la réduction du capital de l'entreprise, la détérioration des produits stockés, le coût de location d'espaces de stockage. Les problèmes de lot-sizing peuvent aussi gérer des réseaux de distribution, lorsqu'un ou plusieurs entrepôts appartiennent à une même compagnie. Chaque entrepôt passe commande et fournit d'autres niveaux du réseau, des grossistes aux détaillants qui doivent livrer les clients. Les coûts de lancement peuvent représenter des coûts de transport ou des coûts de traitement de commande. Des coûts de commande unitaires sont considérés et chaque entrepôt possède sa propre zone de stockage.

Dans nos recherches, nous avons principalement considéré des problèmes de lot-sizing sur plusieurs niveaux organisés en série, c'est à dire, que chaque niveau a un unique fournisseur et un unique niveau à fournir. Le premier niveau commande auprès d'un fournisseur extérieur et le dernier niveau doit satisfaire le client. Un autre sujet abordé dans cette thèse est la considération de capacités de production, ou de volume de commande limité, afin de mieux refléter la réalité. En effet, une machine possède nécessairement une capacité de production limitée durant une période de temps donnée. D'un point de vue distribution, cela peut représenter des capacités de transport ou de manutention.

Etude bibliographique

Le problème de lot-sizing avec capacités (CLSP pour capacitated lot-sizing problem) a été introduit par Florian and Klein, 1971. Ils proposent un algorithme exact de programmation dynamique pour des capacités stationnaires et des fonctions de coûts concaves. Van Hoesel and Wagelmans, 1996 ont ensuite proposé une amélioration de leur résultat. Van Vyve, 2007 présente une version du problème avec des contraintes des livraisons par lots. Tous ces articles ont pour hypothèse des capacités stationnaires dans le temps, Florian, Lenstra, and Rinnooy Kan, 1980a et Bitran and Yanasse, 1982a ayant montré que le problème avec des capacités variables dans le temps était NP-difficile, même dans de nombreux cas particuliers.

Quelques articles traitent également du cas multi-niveaux avec capacités de production : Kaminsky and Simchi-Levi, 2003 pour le cas à deux niveaux et capacités de production sur chacun des deux niveaux, mais sans coût fixe de lancement de production. Sargut and Romeijn, 2007 ont considéré le problème sur deux niveaux, avec capacités de production au premier niveau uniquement. Van Hoesel et al., 2005 ont considéré le cas multi-niveaux et capacités de production seulement au premier niveau, proposant un algorithme polynomial en T, mais exponentiel en N, T étant le nombre de périodes et N le nombre de niveaux. Hwang, Ahn, and Kaminsky, 2013 ont résolu ce dernier problème en proposant un algorithme polynomial à la fois polynomial en T et en N. Hwang, Ahn, and Kaminsky, 2016 ont proposé plusieurs algorithmes en temps polynomial pour le problème du lot-sizing a deux niveaux avec capacités au premier niveau et des coûts concaves de production et de stockage. Hwang, Ahn, and Kaminsky, 2016 ont considéré un problème sur un nombre N de niveaux en séries avec capacités. Ils proposent un algorithme exact en temps polynomial, mais dont la complexité devient rapidement trop importante avec la taille de l'instance.

Contributions de la thèse

La plupart des recherches sur des approches exactes pour des problèmes de lot-sizing multi-niveaux en série avec des capacités de production concernent des réseaux avec des capacités au premier niveau uniquement. Cela permet de modéliser des cas réalistes, par exemple si la restriction vient des mesures protectrices concernant les matières premières, si la capacité du fournisseur extérieur est limitée, ou si le premier niveau représente le niveau de production avec une capacité, tandis que les autres niveaux sont des niveaux de distribution. Cependant, dans de nombreux cas, le problème de capacité peut survenir au sein de la chaîne d'approvisionnement, à des niveaux intermédiaires. Si on considère une société de taille moyenne particulièrement, à moins qu'elle se fournisse en matériaux rares, on peut supposer que l'offre du marché est assez riche pour satisfaire ses besoins. Les capacités à l'intérieur du réseau peuvent provenir de restriction au niveau des ressources matérielles ou humaines : la capacité d'une machine, ou la taille d'un camion ou d'un conteneur dans un réseau de distribution. C'est pourquoi cette thèse vise à étudier des cas plus généraux avec des capacités à chaque niveau.

Nous avons tout d'abord étudié un problème du lot-sizing sur deux niveaux en série, avec des capacités aux deux niveaux. On suppose que les capacités sont identiques et stationnaires au deux niveaux. Ce modèle peut représenter un atelier de production équilibré, sur lequel les machines ont la même capacité. Cela peut aussi représenter une chaîne logistique dans laquelle un constructeur s'approvisionne auprès d'un fournisseur extérieur, délivre les biens manufacturés à un détaillant, qui les livre aux clients. Les capacités identiques décrivent alors le fait que les mêmes véhicules effectuent la livraison aux deux niveaux. Contrairement à Kaminsky and Simchi-Levi, 2003, on considère des coûts de lancement de production, non-croissants avec le temps au premier niveau, et généraux au second. Les coûts de production et de stockage unitaires sont linéaires et respectent une nouvelle structure de coût, appelée non-spéculative chemin, généralisant la structure de coût non-spéculative classique, spécifiant qu'il est moins cher de produire aux deux niveaux à une période t, que d'anticiper la production à l'un des niveaux. On introduit une nouvelle classe de politiques, appelée politique doublement imbriquée, dans laquelle une production fractionnaire au premier niveau à une période donnée entraine de facto une production fractionnaire au second niveau à cette période. Réciproquement, une production saturée au second niveau entraine une production saturée au premier niveau. Notre approche de résolution est basée sur une décomposition d'une solution point extrême en un ensemble de composantes connexes. On montre que sous nos hypothèses, une composante connexe peut à son tour être décomposée en sous-plans indépendants, lorsqu'on considère le second niveau uniquement. On propose un algorithme exact en temps polynomial sous ces hypothèses, avec une complexité de $O(T^5)$. Avec des hypothèses de non-spéculation échelon, la complexité est réduite à $O(T^3)$.

Dans le chapitre suivant, nous étendons ces résultats au cas avec un nombre général de niveaux N en série. De plus, on considère qu'à chaque niveau, le stock est réapprovisionné par le niveau en amont, avec des livraisons par lots (batch deliveries), pour lesquelles un coût fixe est comptabilisé pour chaque lot commandé. On considère aussi une limitation sur le nombre de lots pouvant être commandés à une période et un niveau donnés. La taille des lots est supposée identique et stationnaire à chaque niveau, le problème du lot-sizing sur un niveau avec des livraisons par lots étant NP-difficile avec des tailles de lots dépendant du temps (voir Akbalik and Rapine, 2013). Des tailles de lots identiques peuvent modéliser, dans une chaîne logistique, une flotte de véhicules identiques. Contrairement à l'étude précédente, les coûts unitaires de production et de stockage respectent l'hypothèse classique de non-spéculation échelon, ce qui signifie que ce chapitre est un complément mais pas un substitut du précédent. Pour autant que nous le sachions, le statut de ce problème est ouvert. Nous mettons en évidence des propriétés dominantes de politique optimale, et fournissons un algorithme qui est polynomial à la fois en la taille de l'horizon de temps et au nombre de niveaux. Nous réutilisons le concept de basis path introduit par Hwang, Ahn, and Kaminsky, 2013 pour proposer une décomposition du problème en composantes connexes restreintes à un sous-ensemble de niveaux. La propriété doublement imbriquée introduite dans le chapitre précédant est étendue au cas multi-niveaux : les lots pleins commandés au premier niveau sont aussi commandés aux niveaux en amont aux mêmes périodes, et si un lot fractionnaire est commandé à une période et un niveau donnés, un lot fractionnaire est aussi commandé aux niveaux suivants à cette période. Finalement, notre modèle est étendu pour incorporer des coûts de lancement de production au premier niveau.

Nous étudions ensuite des extensions du problème étudié dans le chapitre précédent, pour lequel nous proposons des résultats de complexités et des algorithmes d'approximation. Nous considérons tout d'abord le problème du lot-sizing multi-niveaux sans capacité, avec des livraisons par lots dont la taille dépend du niveau, ce qui est pertinent en pratique, puisque différents types de véhicules peuvent être utilisés selon le niveau. Nous prouvons que ce problème est NP-difficile. Nous proposons ensuite un algorithme de 2-approximation, pour des tailles de lots dépendant du temps et du niveau, ce qui est un résultat intéressant étant donné la complexité d'un tel problème. Pour cette approximation, nous introduisons une méthode consistant à encadrer les coûts d'approvisionnement à chaque niveau par deux fonctions affines avec lesquelles on peut résoudre le problème en temps polynomial. Un autre résultat de complexité est donné pour le problème du lot-sizing multi-niveaux avec des capacités dont les valeurs dépendent du niveau. Finalement, un algorithme d'approximation est proposé pour le problème du lot-sizing multi-niveaux avec des capacités dont les valeurs dépendent du niveau, des coûts de lancement de production non-croissants, et une absence de motifs de spéculation.

Pour finir, nous présentons un travail en collaboration avec Ayse Akbalik, de l'Université de Lorraine, qui concerne un problème du lot-sizing avec des limitations périodiques d'énergie. Le système étudié n'est pas en série, mais consiste en un ensemble de machines identiques et parallèles ayant une capacité limitée, chacune consommant un certain montant d'énergie lors de son allumage, et pour chaque unité produite. On considère un coût d'allumage des machines, en plus d'un coût unitaire de production et de stockage, chacun dépendant du temps. On peut noter que ce système possède des points communs avec le problème du lot-sizing multi-niveaux en série avec des livraisons par lots. En effet, la capacité de production du système peut être augmentée en démarrant une machine supplémentaire, ce qui entraine un coût. La similitude avec le modèle précédent avec livraisons par lots réside dans le fait que dans ce dernier, la capacité du système peut être étendue en commandant un lot supplémentaire, ce qui entraine un coût fixe également. La différence étant qu'une machine allumée le reste durant les périodes suivantes, ce qui revient à dire que le coût fixe n'est payé qu'une seule fois pour augmenter la capacité de tout l'horizon de temps jusqu'à extinction de la machine. Outre les décisions classiques aux problèmes du lot-sizing, de combien et à quelles périodes produire, il faut décider du nombre de machine allumer ou éteindre à chaque période. Nous montrons que ce problème est NP-difficile même avec des conditions très fortes sur les paramètres. En revanche, en supposant des paramètres d'énergie stationnaires, nous proposons un algorithme dynamique ayant un temps d'exécution polynomial, et résolvant le problème à l'optimalité en temps $O(N^5T^4)$. On généralise également notre modèle afin d'incorporer des coûts de lancement de production communs, des coûts de fonctionnement, et la prise en compte de la consommation énergétique des machines allumées. Nous montrons que notre algorithme peut être adapté pour résoudre cette version plus générale du problème en temps $O(M^6T^6)$ pour des paramètres d'énergie stationnaires.

Chapter 1

Introduction

In our globalized world, supply chain management is an increasingly important issue. Growing transport capacity, trade liberalization on a global scale, pressure companies to permanently reduce their production and distribution costs in order to face increasing competition. Inventory management is a matter of prime importance: the costs arising from the storage of products represent huge amounts of money, and it is crucial to avoid disruption, so that customer demand may be satisfied in time. In this context, inventory management aims to propose efficient tools and optimization methods to minimize these costs. Inventory management, and more generally production planning, concern the organization of physical flows of components or finished goods in a system. This system can be a large supply chain for which we need to answer questions like when and what to produce, when and what to order from the supplier, and where and when to store. The production and the distribution of these items are often achieved in lots, which allows significant economies of scale. Logistic systems can be modeled as a series of stages or levels. Logistic and production processes can be decomposed into flows of products going from one level to another, proceeding assembly, production or distribution operations. Levels at the top of the chain may order raw materials or other various components from an external supplier. Levels at the end of the supply chain must meet demand from their customers. For example, Figure1.1 represents a logistic network containing a production facility located in China, which supplies a warehouse in Europe, which in turn provides a retailer in Western Europe. The system can also represent a shop floor, where each level represents a machine or a group of parallel machines, and where semi-finite products are stored at each level. Multi-echelon inventory management problems are challenging and complex from a combinatorial optimization point of view, due to their size and the need of coordination between the different levels. Efficient policies are required for these multi-level systems.

In the field of operational research, the lot-sizing problem introduced by Wagner and Whitin, 1958 deals with production planning or distribution networks including inventories. It aims to satisfy the demand of clients while minimizing all costs involved in the system. In models with discrete-time finite horizon, the time decomposes into a set of discrete-time periods with their specific cost parameters and customer demand. Lot-sizing models may represent production systems, for which each level is a step in the production process, usually a machine. Starting the production at a period may cause

FIGURE 1.1: Logistic network on several levels.

some costs due to the need for labor, the energy consumption to turn on the machine, transportation between two machines, etc. For this reason, setup costs are taken into account in most of the models studied in the literature. These costs are paid whenever a production is started, whatever the number of produced units. Of course costs also depend on quantity produced through unit production costs. At each level, it is possible to store some units before being delivered to the next level. In most of the studies, a unit inventory cost or holding cost is charged to carry a unit in stock from one period to another. Inventory costs can model, among other, the reduction of capital of the company, the impairment loss of the goods, the cost of renting storage space. Lot-sizing problems can also deal with distribution networks, when one or more warehouses belong to the same company. Each warehouse orders and provide other levels of the network, from the wholesalers to the retailers which deliver the clients. Setup costs can represent transportation costs or order processing costs. Unit ordering costs are charged and each warehouse has its own storage area. Lot-sizing models address both production systems and logistic systems. We use in this document both terminologies. This thesis aims to study multi-level in series lot-sizing problems with capacities.

Lot-sizing literature is far too wide to propose here an exhaustive review of all variants of the problem. In our research we chose to focus on deterministic models for which all parameters are known including the demand to satisfy, but a large range of studies deals with stochastic models. Moreover, we consider problems in which backlogging is not allowed, that is, the demand must be met on time. We also deal mainly with multi-level lot-sizing problems with levels organized in series, that is, each level has a unique provider and a unique level to supply. First level purchases from an external supplier and the last level must satisfy the client (see Figure 1.3). Another topic in this thesis concerns the consideration of production capacities, or limited order volumes, in order to better reflect reality. Indeed, a machine necessarily has a limited processing capability during a given period of time. In a distribution system, vehicles have also limited capacities and warehouse can have limited handling capacities. In the next section we present a brief overview of the literature of single-item lot-sizing problems. For a complete and recent survey on single-item dynamic lot-sizing problems, see Brahimi et al., 2017 and Pochet and Wolsey, 2006. We focus in this thesis on models with a discrete time horizon, but many studies deal with models with a continuous time horizon (see Roundy, 1985, Muckstadt and Roundy, 1993).

1.1 Lot-sizing models

The single-level uncapacitated lot-sizing problem ($ULSP$) introduced by Wagner and Whitin, 1958 aims to determine an inventory management policy minimizing the total costs to satisfy a known demand over a discrete-time finite horizon discretized into T periods. It is possible to keep some units in inventory from one period to another. At each period t it must be decided which quantity is ordered and how many units are stored from period t to period $t+1$. An order at a period involves a time-dependent fixed setup cost. Unit ordering and inventory costs are also time-varying. The demand must be entirely satisfied on time by placing orders from an external supplier at some periods. The goal is to find a compromise between frequent ordering, which involves large setup costs, and ordering more rarely, implying important inventory costs. The problem can also represent a machine producing units to satisfy the demand, with the setup costs reflecting the cost of preparing the machine to produce, and with an inventory for the manufactured products. The following notations are used to model the $ULSP$:

T	Length of the planning horizon;
d_t	Demand in period t, $t \in \{1, .., T\}$;
K_t	Setup cost at period t;
p_t	Unit ordering cost at period t,
h_t	Unit holding cost at period t;

The decision variables are, for each period $t \in \{1, .., T\}$:

x_t	Amount of unit ordered at period t;
y_t	Production indicator (binary variable) at period t;
s_t	Stock level at the end of period t;

The $ULSP$ can be formulated as follow:

$$\min \sum_{t=1}^{T} (K_t y_t + p_t x_t + s_t h_t) \tag{P}$$

subject to

$$x_t + s_{t-1} = d_t + s_t, \ \forall t \in \{1, .., T\}, \tag{1.1}$$

$$x_t \leq \sum_{t'=t}^{T} d_{t'} y_t, \ \forall t \in \{1, .., T\}, \tag{1.2}$$

$$s_0 = 0, \tag{1.3}$$

$$x_t \geq 0, s_t \geq 0, y_t \in \{0, 1\}, \ \forall t \in \{1, .., T\} \tag{1.4}$$

The objective function minimizes the setup, ordering and inventory costs. Constraints (1.1) correspond to the flow conservation, that is, at each period, the entering stock plus the amount ordered is equal to the demand to serve plus the outgoing stock. Constraints (1.2) ensure that an order at period t incurs a setup costs. Constraint (1.3) imposes a null inventory at the beginning of the horizon. Other formulations more efficient have been proposed, see Pochet and Wolsey, 2006 and Brahimi et al., 2017.

Wagner and Whitin, 1958 use a dynamic programming (DP) approach based on a time decomposition to solve the problem optimally. A property is said to be dominant for a problem if it admits at least an optimal solution which exhibits this property. A classical approach for finding an optimal solution consists in finding some dominance properties for the problem, which allows to reduce the solution space of a problem focusing on the solutions verifying the properties. The class of ZIO (Zero-Inventory Ordering) policies is introduced in Wagner and Whitin, 1958: a policy is ZIO if productions/orders only occur at periods without entering stock. Wagner and Whitin, 1958 show that ZIO policies are dominant for the single-level uncapacitated lot-sizing problem with concave cost functions.

A regeneration point is a period t without entering stock, that is, $s_{t-1} = 0$. A subplan is a set of consecutive periods $(u, ..., v)$ such that the inventory between periods u and $v - 1$ is always positive and u and v are two (consecutive) regeneration points. In other words, $s_{u-1} = 0$, $s_{v-1} = 0$ and $s_t > 0$ $\forall t \in \{u, ..., v - 2\}$. Each subplan is independent from the other periods of the problem. Consequently, for given periods u and v, the optimal cost of subplan (u, v) can be evaluated independently. A classical method for finding the optimal solution of a single-level problem consists in evaluating the minimum cost of every of all the $O(T^2)$ possible subplans in a first phase. A shortest path problem is then solved, in the acyclic graph whose nodes represent the periods from 1 to T, and whose arcs represent the costs of the subplans defined by the two nodes they are adjacent to. The minimum cost path between node 1 and a fictive node $(T + 1)$ represents an optimal (ZIO) solution of the problem and can be computed in $O(T^2)$, given that the minimum costs of each subplan are yet computed.

A classical assumption is to consider non-speculative motives for the unit ordering and holding costs, also called the Wagner-Within property. This property prevents speculation, that is, without considering the setup costs, it is preferable to order as late as possible. In other words, it is more expensive to order a unit at a period t and to store it until period $t + 1$ rather than to directly ordering this unit at period $t + 1$: $p_t + h_t \geq p_{t+1}$ must hold for any period $t \in \{1, ..., T - 1\}$. This property allows to greatly simplify the structure of extreme point solutions. Wagner and Whitin, 1958 proposed an $O(T^2)$ algorithm for solving this problem for linear variable ordering and holding costs. Several authors proposed an algorithm running in $O(T \log T)$ for general linear cost functions and $O(T)$ without speculative motives (see Federgruen and Tzur, 1991, Wagelmans, Van Hoesel, and Kolen, 1992 and Aggarwal and Park, 1993).

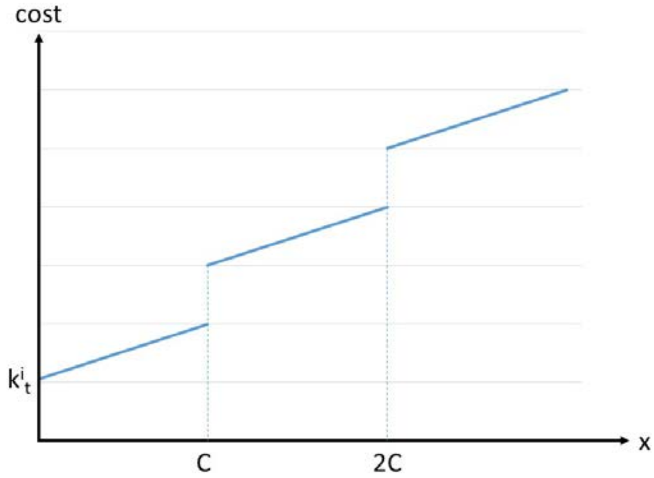

FIGURE 1.2: Full Truck Load (FTL) cost like structure.

Capacitated lot-sizing problems

Our research deals with problems having a limited production capacity, that is, it is assumed that the amount of units which can be produced at a given period is limited, which is clearly a more realistic assumption. We are focusing here on hard capacities, in contrast to soft capacities (batch deliveries) presented in the next section. At each period t, the amount x_t produced is limited by a capacity C_t. The capacitated lot-sizing problem (CLSP) was introduced by Florian and Klein, 1971. They proposed a dynamic programming algorithm to solve this problem in time $O(T^4)$ with a stationary capacity and concave cost functions. Using the concept of subplan they give a characterization of an extreme point solution which drastically reduce the set of solutions which are likely to be optimal. Van Hoesel and Wagelmans, 1996 developed an efficient algorithm in $O(T^3)$ for CLSP, assuming a stationary capacity, concave production costs and linear holding costs. The algorithm proposed by Van Vyve, 2007 can solve this problem under linear cost functions and non-speculative motives in $O(T^2 \log T)$. Chung and Lin, 1988 designed an $O(T^2)$ algorithm for the special case of non-increasing setup costs, unit production costs and non-decreasing capacities. On the opposite, if the capacity is time-varying, CLSP is NP-hard, even in many special cases (see Florian, Lenstra, and Rinnooy Kan, 1980a and Bitran and Yanasse, 1982a).

Lot-sizing problem with batch deliveries

Capacities are also present in distribution networks due to vehicle sizes. However, when the capacity of a vehicle is reached, usually, another one can be ordered. That is, the fleet of vehicles is often constituted of several vehicles, eventually belonging to a third-party logistics. To deal with such situations, more general cost structures were introduced, like the Full Truck Load (FTL) and the Less than Truck Load (LTL) procurement costs, when products are delivered by batches from a supplier. Of course, the fleet of vehicles or the number of containers can be limited, in this case a limit m_t is considered on the number of batches which can be ordered at a period t. It can be seen as an extension of the capacitated lot-sizing problem whose capacity can be extended if the setup cost is paid again. For this reason, batch deliveries are often referred to as soft capacities. Assuming a batch size of C, the cost of ordering x units in the FTL case is equal to $K_t + \lceil x/C \rceil k_t + p_t x$ if $x > 0$, and is null otherwise. Hence, k_t represents a fixed cost per batch, paid for each ordered batch, whenever it is full or not. In the LTL case, the setup cost k_t is charged only for full batches and an additional cost is paid depending on the actual number of units in the truck partially loaded, determined by a freight cost function. K_t is a general setup cost for ordering some units in period t. Figure 1.2 represents the shape of the curve with $K_t = 0$. The single-level lot-sizing problem with time-dependent batch sizes has been shown to be NP-hard by Akbalik and Rapine, 2013. Regarding the issue of identical batch sizes, Van Vyve, 2007 studied a single-level lot-sizing problem with constant batch size and linear ordering and holding costs. He considers time-varying ordering capacities by limiting the maximal number m_t of available batches in each period. He proposes a $O(T^3)$ algorithm for the general case with backlogging and with null setup costs ($K_t = 0$). In the absence of speculative motives the complexity drops to $O(T^2 \log T)$. Li, Hsu, and Xiao, 2004 proposed an $O(T^3 \log T)$ algorithm for the problem with non-decreasing LTL freight cost functions and non-decreasing concave holding costs. Akbalik and Rapine, 2012 proposed two polynomial time algorithms for the constant capacitated lot sizing problem with batch deliveries when production capacities and batch sizes are constant and assuming a Wagner-Whitin cost structure.

Multi-level in series lot-sizing problems

In this thesis, we focus on multi-echelon inventory problems. These may represent distribution systems over several warehouses located geographically apart from each other. Transferring goods between two entities may involve different transportation modes such as a fleet of trucks or container barges. For structures with levels in series, which are studied in the next three chapters, each level gets units from its upstream level and must satisfy the orders of its downstream level. Figure 1.3 provides a schematic representation of a system in series with N levels, setup costs K_t^i, unit ordering costs p_t^i and inventory costs h_t^i at each level $i \in \{1, ..., N\}$ and period $t \in \{1, .., T\}$. The multi-level in series lot-sizing problem can also model serial assembly line, when an item must go through several machines to obtain a final product delivered to the client. At each level semi-finished goods can be stored at a cost h_t^i. An important notion in multi-echelon systems is the echelon holding cost. The echelon holding cost at a level i is defined as the difference between the holding costs at level i and level $i+1$: $h_t'^i = h_t^i - h_t^{i-1}$. It represents the extra cost to pay to move forward a unit in the system. A common assumption is to consider that holding costs increase with the level, which can be due to the

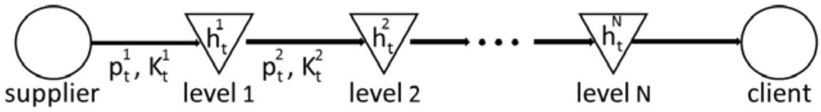

FIGURE 1.3: Representation network of a multi-level in series lot-sizing problem with N levels.

added value at each transportation or manufacturing step. It corresponds to the assumption that $h_t'^{i-1} \leq h_t'^{i}$. In this case it is referred to as positive echelon holding costs. For multi-level problems, the Wagner-Within property has to be written using the echelon holding costs. Specifically, at the first level, it implies that $p_t^1 + h_t^1 \geq p_{t+1}^1$ holds for any period t. At the other levels, for any period t, we must have $p_t^i + (h_t^i - h_t^{i-1}) \geq p_{t+1}^i$. It means that it is always more expensive to move units downstream instead of keeping them at upstream levels. To make the distinction from the classical single-level Wagner-Within property, this cost structure is referred inside this document as the *echelon non-speculative cost structure*.

Zangwill, 1969 proposed an algorithm for the uncapacitated multi-level lot-sizing problem running in time $O(NT^4)$. More recently, Melo and Wolsey, 2010 presented a dynamic algorithm for the 2-level uncapacitated problem with a time complexity in $O(T^2 logT)$.

Kaminsky and Simchi-Levi, 2003 were the first to propose an efficient algorithm for the 2-level case with capacities at both levels and an intermediary transportation level. Capacities are time-dependent and they assume positive echelon holding costs and no speculative motives at both levels. However, in their model, production setup costs are null. As a consequence, the production at the second level is performed as late as possible, subject to the capacity constraints, which allows them to reduce the problem into a single-level CLSP and to solve it in time complexity $O(T^4)$.

Sargut and Romeijn, 2007 considered the 2-level capacitated lot-sizing problem allowing backlogging and subcontracting, with (stationary) production capacities only at the first level. They proposed different polynomial time algorithms for special cases. Van Hoesel et al., 2005 developed an algorithm for the multilevel lot-sizing problem with production capacities, with general concave cost functions, which is polynomial if the number N of levels is fixed, but whose complexity grows exponentially with N. Considering more restrictive cost structures, they were able to propose an algorithm that stays polynomial even if N is part of the input. Finally, using the novel concept of basis path, Hwang, Ahn, and Kaminsky, 2013 developed the first polynomial algorithm for the multilevel lot-sizing problem with production capacities with general concave costs and the number N of levels being part of the inputs. Hwang, Ahn, and Kaminsky, 2016 proposed several polynomial time algorithms for the 2-level capacitated lot-sizing problem with stationary capacities at the first level and concave production and inventory costs. Notice that all these studies consider a limited capacity only at the first level of the chain.

Very recently, Ahmed et al., 2016 considered the minimum concave cost flow problem over a two-dimensional grid network. The multi-level lot-sizing problem is a special case of this problem: a N-level serial lot-sizing problem can be modeled by a $(N+1)$-level 2-dimensional grid network. For the capacitated case, they proposed a polynomial-time algorithm for the problem with a fixed number of levels and a fixed number of different finite capacity values. However, the time complexity of their algorithm is in $O(N^{4KN-4K+1}T^{4KN+4N-4K-3})$, where K is the number of different capacity values and N is the number of levels. Hence, although this is a strong theoretical result, the algorithm is of limited practical use.

Few other articles on multi-level lot-sizing problems deal with different network structures such as the One-Warehouse Multi-Retailer problem (OWMR) and the Joint Replenishment Problem (JRP). It consists in a distribution network composed of one warehouse which orders from an external supplier, and which provides a set of different retailers. The difference between the two problems lies in the fact that the warehouse cannot hold inventory in the JRP. Each retailer must meet the demand of its client. So far, limited research has been done on these problems in their discrete-time version. Chan et al., 2002 have shown that both problems are NP-hard and study the class of zero-inventory-ordering (ZIO) policies. Levi et al., 2008 proposed a 3.6-approximation algorithm for a transportation cost structure, assuming time-independent (but retailer-dependent) batch sizes. In Gayon et al., 2017, the authors improved this result to a 2-approximation algorithm for OWMR with FTL cost structures, allowing non null setup costs and a capacity constraint on a special set of retailers.

There are also some studies on multi-level lot-sizing problems with inventory bounds, modeling a limit on the physical size of the stock, that is, the capacity constraint is on the inventory variable. Hwang et al., 2013 considered a multi-level lot-sizing problem where one of the levels has a stationary inventory bound. They proposed an exact algorithm, based on the concept of basis path, in time complexity $O(LT^7)$. Recently, Phouratsamay, Kedad-Sidhoum, and Pascual, 2016 considered a 2-level lot-sizing problem with inventory bounds. They provide a polynomial dynamic programming algorithm for the case with bounds at the first level. With bounds at both levels, they prove that the problem is NP-hard and propose a pseudo-polynomial time algorithm.

Other studies focus on polyhedral approaches by proposing extended formulations of the problem, using valid inequalities, including among others Zhang, Küçükyavuz, and Yaman, 2012 which proposed an effective branch-and-cut algorithm for capacitated multi-item, multilevel problems, and Van Vyve, Wolsey, and Yaman, 2014 which studied several variants of the 2-level lot-sizing problem with constant capacity at both levels. They provide an extended formulation for the problem with constant production capacities at both levels but they did not test its performance.

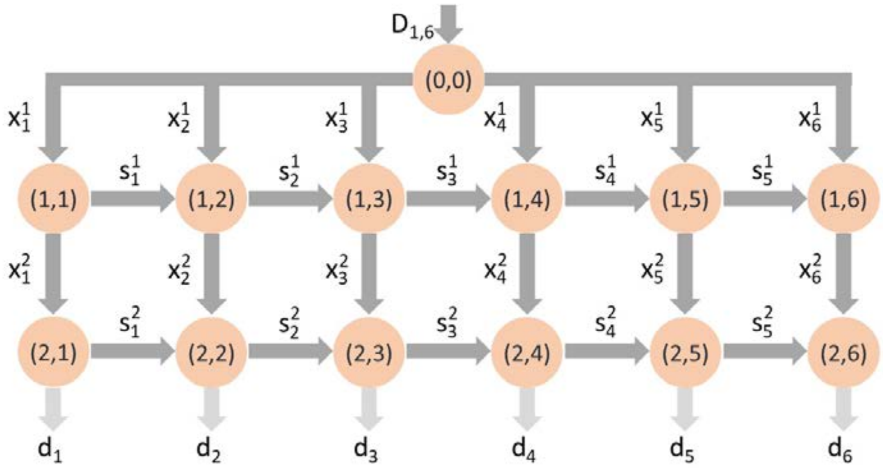

FIGURE 1.4: Network representation of a multi-level in series lot-sizing problem with $N = 2$ levels and $T = 6$ periods.

List of abbreviations for multi-level in series lot-sizing problems

Throughout this document, we use several abbreviations to describe the multi-level lot-sizing problems addressed, listed below:

M-ULSP	multi-level uncapacitated lot-sizing problem.
M-CLSP	multi-level lot-sizing problem with hard capacities at each level.
M-LSP-PC	multi-level lot-sizing problem with production capacity at the first level only.
M-ULSP-B	multi-level uncapacitated lot-sizing problem with batch deliveries.
M-LSP-B	multi-level lot-sizing problem with batch deliveries and with an upper limit on the number of batches of each order.

We namely focus on stationary and identical capacity/batch size, except in Chapter 4.

1.2 Network flow representation

Lot-sizing problems can be modelled as minimum-cost flow network problems. As minimum-cost-flow problems are easy to visualize, we abundantly use the representation of a policy as a flow in this document. For multi-level in series lot-sizing problems with hard capacities, we can define the following network: a node (i, t) is associated to each level i and each period t, for $i \in \{1, ..., N\}$ and $t \in \{1, ..., T\}$. For each level $i \in \{1, ..., N\}$ matches a row of nodes $(i, 1)$, $(i, 2)$,..., (i, T) gathering all information about the decision at level i. Similarly, for each $t \in \{1, ..., T\}$ matches a column of nodes $(1, t)$,

$(2,t)$,..., (N,t). Two nodes (i,t) and $(i,t+1)$ are linked by an arc which corresponds to an inventory flow between periods t and $t+1$ at level i, and whose unit cost is h_t^i. Two nodes $(i-1,t)$ and (i,t) are linked by capacitated arcs which corresponds to a production flow at level i at period t, and whose cost is equal to the production cost at period t and level i. We denote by $D_{t,t'} \equiv d_t + \cdots + d_{t'}$ the cumulative demand between periods t and t'. A source node $(0,0)$, connected to the nodes of the first level, corresponds to the external supplier, with $D_{1,T}$ units available. Finally, each node (N,t) has a positive demand d_t. The objective is to route all the units available at the source node to the sinks at minimal cost. As an example, Figure 1.4 represents a problem with 2 levels and 6 periods of time. To represent lot-sizing problems with soft capacities, i.e., obeying a batch delivery cost structure (with null setup costs), a network with an arc for each batch is defined, each with a capacity and a cost corresponding to those of the batch. Notice that the size of the resulting network is not polynomially bounded since the number of batches which can potentially be used at a period t and a level i is equal to $\lceil D_{t,T}/B_t^i \rceil$. We can prevent exponentially large networks by using binary representation of capacities: only $\log \lceil D_{t,T}/B_t^i \rceil$ arcs are requested at each node (i,t) of the network. The k^{th} arc representing the ordering of 2^{k-1} batches.

The models we study have a concave objective function and a feasible region defined by linear constraints, that is, are polyhedron and thus have an extreme point optimal solution. Recall that an arc is said to be *free* if it is not saturated and it its flow is positive. A period with a production which is not saturated is called a fractional period. We also say that an arc is *full* or *saturated* if its flow is equal to its capacity. A well-known dominant property states that for problems with concave cost functions, the network induced by the free arcs contains no cycle in an extreme point solution (see Zangwill, 2013 and Ahuja, Magnanti, and Orlin, 1993). For single-level problems, this means that there can be at most one fractional period on the same subplan. If the problem is uncapacitated, this production must therefore fulfill all demand of the subplan. As a result, productions are only carried out at periods without entering stock ($s_{t-1} = 0$), that is, the extreme point solution are ZIO. On multi-level problems, it is also possible to decompose an extreme point solution into independent elements using connected components. A connected component is a set of nodes such that each of these nodes are linked by a flow. We use such decomposition through this document to design efficient dynamic programming approaches.

1.3 Contributions of the thesis

Most of research efforts on exact approach for multi-level in series lot-sizing problems with production capacities concern networks with capacities at the first level only (Sargut and Romeijn, 2007, Van Hoesel et al., 2005, Hwang, Ahn, and Kaminsky, 2013). This can model realistic cases, for example if the restriction comes from protective regulations on raw materials, if the external supplier capacity is limited, or if the first level represents a production level, while other levels are distribution levels. However, in many cases, the capacity bottlenecks may arise within the supply chain, at some intermediate levels. Considering a mid-sized company in particular, unless it needs to procure scarce products, one can assume that the market offer should be rich enough to satisfy its needs. The capacities inside the network may come from material or human resource restrictions: the capacity of a machine, or the size of a truck or a container in a distribution network. For these reasons, this thesis aims to study the more general case with capacities at each level.

In Chapter 2 we study a 2-level in series lot-sizing problem with capacities at both levels. We assume that capacities are identical and stationary at both levels. This model may represent a well-balanced flowshop production line, where the machines have the same capacity. It may also represent a supply chain in which the manufacturer gets supplies from an external supplier, and provides the retailer which delivers the goods to the client with a single vehicle. Identical capacities describe a situation where the same type of vehicle performs the shipments at both levels. In contrast to Kaminsky and Simchi-Levi, 2003, we consider time-dependent setup costs, which are non-increasing at the first level and unrestricted at the second one. The unit production and holding costs are linear and follow a new cost structure, called *path non-speculative*, generalizing the classical non-speculative cost structure, which specifies that it is less expensive to produce at both levels at a period t, rather than to anticipate the production at some levels. We introduce a new class of policies, called double-nested policies, in which a fractional production at the first level at a given period involves a fractional production at the second level at this period. Conversely, a full production at the second level causes a full production at the first level. The approach of resolution is based on a decomposition of an extreme point solution into connected components. We show that under our hypothesis, a connected component can in its turn be decomposed into independent subplans considering the second level only. That is, the problem is reduced to a single-level CLSP. We propose an exact polynomial time algorithm under these assumptions, in time complexity $O(T^5)$. Assuming echelon non-speculative motives, the time complexity is reduced to $O(T^3)$. This problem is a particular case of a grid network of three levels and one finite capacity value. Consequently, the algorithm proposed by Ahmed et al., 2016 allows to solve the problem we introduced in time complexity $O(T^{17})$. Even if the time complexity of the algorithm is polynomial, it is quite prohibitive to use in practice. In contrast, our problem is a way less general, but the assumptions we make allow us to propose a more efficient polynomial time algorithm, applicable to some practicable problems, even for a large number of periods.

In Chapter 3 we extend these results to the case with a general number N of levels in series. Moreover, we consider that at each level, the inventory is replenished from its upstream level using batch deliveries, where a fixed cost is incurred for each batch ordered. We also consider a limitation on the number of batches that can be ordered at a given period. Batch sizes are assumed identical and stationary at each level, the single-level problem being NP-hard with time-dependent batch sizes (see Akbalik and Rapine, 2013). Identical batch sizes may model in a supply chain a fleet of identical vehicles.

Contrary to the previous chapter, unit production and holding costs follow the classical echelon non-speculative motives at every levels, which means that the results of this chapter is a complement and not a full generalization of Chapter 2. As far as we know, the status of this multi-level capacitated lot-sizing problem with batch deliveries is open. We highlight some structural properties of an optimal policy, and provide a dynamic algorithm which is polynomial both in the length of the planning horizon and in the number of levels. We reuse the concept of basis path introduced by Hwang, Ahn, and Kaminsky, 2013 to propose a decomposition of the problem into induced connected components restricted to a subset of levels. The double-nested property introduced in the previous chapter is extended to the multi-level case: full batches ordered at a level are also ordered at its upstream level, and an ordering period is fractional at a level, then it is also a fractional ordering period at its downstream level. Finally, we extend our model to incorporate non null setup costs at the first level in addition to the fixed costs per batch.

We consider in Chapter 4 generalizations of the problem studied in Chapter 3, for which we propose complexity results and approximation algorithms. We first consider the multi-level uncapacitated lot-sizing problem with batch deliveries and level-dependent batch sizes (M-ULSP-B), that is, each level orders using a specific batch size B^i. This model is relevant in practice, as different types of vehicles can be used depending on the level. We establish that this problem is NP-hard. We provide then a simple 2-approximation algorithm for M-ULSP-B, with time-dependent and level-dependent batch sizes, which is an interesting result considering the complexity of this problem. For this approximation, we use a method consisting in sandwiching the procurement costs of each level in each period between two affine functions. Another complexity result is given for the multi-level lot-sizing problem with level-dependent (hard) capacities C^i (M-CLSP). Finally, an approximation algorithm is proposed for M-CLSP with level-dependent capacities, non-increasing setup costs and non-speculative motives. Notice that under these assumptions, the complexity of the problem is open. We are currently trying to reuse this idea of sandwiching to other problems, through an on-going collaboration with Albert Wagelmans and Wilco van den Heuvel, from Erasmus University Rotterdam.

Chapter 5 is a collaborative work with Ayse Akbalik, from University of Lorraine, which deals with a single-item lot sizing problem under a periodic energy limitation. The system here is not in series, but consists in identical and parallel capacitated machines, each one consuming a certain amount of energy when being switched on and when producing. The originality of this model lies in the fact that we consider a limit of the amount of energy that can be consumed in each period by the production system. Considering the energy consumption is an important issue in many industries, and is in accordance with the context of energy aware production and environmental sustainability. We consider a cost for starting-up the machines, in addition to a unit production cost and a unit holding cost, all being time-dependent. Notice that this system shows similarities with the multi-level in series lot-sizing problem with batch deliveries. Indeed, the production capacity of the system can be extended by turning on an additional machine, incurring a starting cost. This is similar in a sense to batch deliveries where the capacity can be extended by ordering additional batches with a fixed cost. The difference is that a machine turned on once may remain running during the next periods, that is, a start-up cost is paid only once to increase the capacity on the whole remaining horizon. Besides the classical lot sizing decisions of how much and in which periods to produce, we have to decide the number of machines to switch on and to switch off in each period. We show that this problem is NP-hard even under restricted conditions. In contrast, assuming stationary energy parameters, we propose a polynomial time dynamic programming algorithm to solve to optimality the problem in time $O(M^5T^4)$, with M being the number of machines and T the number of periods considered in the planning horizon. We also generalize our model to incorporate joint setup costs, running costs and running energy consumption due to machines that are not turned off. We show that we can adapt our algorithm to solve this generalized version of the problem in time complexity $O(M^6T^6)$ for stationary energy parameters.

Finally, Chapter 6 concludes the thesis, recalling our main results and providing some guidelines for further research.

Chapter 2

An efficient algorithm for the 2-level capacitated lot-sizing problem with identical capacities at both levels

This chapter deals with a 2-level in series lot-sizing problem with identical and stationary production capacities at both levels. We use inside this chapter a production terminology, since we consider hard capacities, unlike the next chapter where batch deliveries are considered. We propose an exact dynamic algorithm running in $O(T^5)$, which significantly reduces the time complexity of the algorithm proposed by Ahmed et al. (2014) Ahmed et al., 2016 (see literature review in the introductory chapter). In addition, we exhibit a new original class of policies, called double-nested policies, that we prove dominant for this problem. The nice and simple structure of double-nested policies, where both levels produce at full capacity in a synchronous way, can certainly be reused in future works. Finally, this work also introduces the new path non-speculative cost structure, which generalizes in a simple manner the classic echelon non-speculative cost structure of the literature (see Section1.1). In particular, the non-speculative path cost structure is verified by any non-negative holding costs when production costs are stationary. This chapter is based on an article published in the European Journal of Operational Research (Goisque and Rapine, 2017a).

2.1 Introduction

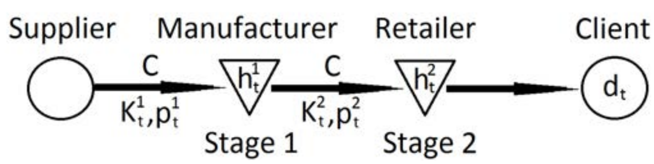

FIGURE 2.1: Serial supply chain with two production levels.

We consider a serial structure composed of two production levels in series with identical and stationary production capacities. For convenience, we call the first level *the manufacturer* and the second level *the retailer*, see Figure 2.1. This problem is a natural extension of the capacitated single-item lot-sizing problem (CLSP), well-studied in the literature since the seminal paper of Florian and Klein (1971) Florian and Klein, 1971. CLSP, on a single level, is known to be NP-hard if the capacity is time-varying. For this reason, we restrict ourselves to the case of stationary capacities. In addition, we assume that the capacities are identical at both levels, that is, $C_t^1 = C_t^2 = C$ for all periods t. This problem may model the realistic situation of a well-balanced flowshop production line, where each machine has the same capacity. It may also model a supply chain in series, where setup costs represent the fixed transportation cost paid to order units from an external supplier to the manufacturer, and from the manufacturer to the retailer, see Figure 2.1. In order to propose a quite efficient algorithm, we consider that the setup costs are non-increasing at the manufacturer level, that is, $K_t^1 \geq K_{t+1}^1$ for all periods t. We also assume that production and holding costs are following the path non-speculative cost structure defined below.

Path non-speculative cost structure

In this chapter we consider a cost structure which is slightly more general than the echelon non-speculative cost structure. To introduce this cost structure, let us define q_{ab}^t as the cost associated with the production and the storage of a unit that is processed at the manufacturer at period a, at the retailer at period b, to satisfy a demand at period t. Cost q_{ab}^t is hence the production and holding costs associated with the path from node $(1, a)$ at the manufacturer to demand node $(2, t)$, via production node $(2, b)$ at the retailer, see Figure 2.2. We require that costs q_{ab}^t are non-negative. We say that a cost structure is *path non-speculative* if the following properties hold:

(1) $q_{ab}^t \geq q_{tt}^t$ for all periods a, b, t such that $0 \leq a \leq b \leq t \leq T$

(2) $q_{ab}^t \geq q_{a'b}^t$ for all periods a, a', b, t such that $0 \leq a \leq a' \leq b \leq t \leq T$

The first condition stipulates that for a given demand period t, the cheapest unit path consists in producing directly at period t both at the manufacturer and the retailer level. Figure 2.2 illustrates this condition: any grey path, which represents the costs incurred by a unit produced at period a at the manufacturer and at period b at the retailer to satisfy demand at period t, is more expensive than the black path, which represents the situation where a unit is directly produced at period t at both levels. The second condition specifies that for a unit supplying the demand at a given period t and produced at a given period b at the retailer, the cheapest path consists in producing it as late as possible at the manufacturer. Notice that considering costs q_{ab}^t defined on paths is more general than considering costs defined edge by edge independently. It can among others capture the case of perishable goods (for instance, if a unit cannot be produced in advance more than a certain number P of periods, we can simply set $q_{ab}^t = \infty$ for any periods such that $t - a > P$). It can also be used to encapsulate backlogging costs, by allowing a and b to be greater than t.

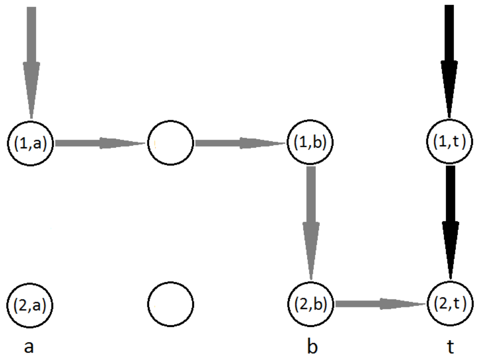

FIGURE 2.2: Illustration of the cost structure.

We now show that this new cost structure generalizes the echelon non-speculative cost structure, that is, a cost structure obeying the echelon non-speculative properties also satisfies the path non-speculative properties. Consider a cost structure with unit production and holding costs satisfying the echelon non-speculative properties. The cost associated with the production and the storage of a unit produced at the manufacturer at period a, at the retailer at period b, to satisfy demand at period t is equal to $q^t_{ab} \equiv p^1_a + h^1_a + \cdots + h^1_{b-1} + p^2_b + h^2_b + \cdots + h^2_{t-1}$. The second condition of path non-speculative motives is equivalent to have echelon non-speculative motives at the manufacturer level, and thus is satisfied. To establish that the first condition is also satisfied, consider any periods a, b and t, with $a \leq b \leq t$. Using the echelon non-speculative property at the retailer level, we have $p^2_b + h^2_b + \cdots + h^2_{t-1} \geq h^1_b + \cdots + h^1_{t-1} + p^2_t$. It implies that $q^t_{ab} \geq p^1_a + h^1_a + \cdots + h^1_{t-1} + p^2_t$. Using now the echelon non-speculative property at the manufacturer level, we obtain finally that $q^t_{ab} \geq p^1_t + p^2_t \equiv q^t_{tt}$. Hence, the cost structure is also path non-speculative.

On the contrary, some cost structures may satisfy the path non-speculative properties and not the echelon non-speculative properties. One noticeable case corresponds to stationary production costs. In this situation, if production costs are stationary, the path non-speculative properties are verified by *any* non-negative holding costs, while echelon non-speculative properties require non-negative echelon holding costs, that is, $h^1_t \leq h^2_t$ for all period t.

Organization of the chapter.

Section 2.2 provides a formulation of the problem and presents the structure of extreme points. Section 2.3 introduces some dominance properties that allows us to restrict to what we call the *double-nested policies*. Based on these properties, we present in section 2.4 a dynamic programming algorithm to solve the problem in time complexity $O(T^5)$ for path non-speculative cost structures, and in time complexity $O(T^3)$ for echelon non-speculative cost structures. Finally, section 2.5 concludes the chapter.

2.2 Problem Formulation and structure of extreme points

We present in this section a mixed integer linear formulation for the $2-$level capacitated lot-sizing problem with identical capacities at both levels (2-CLSP-CC). We summarize below the different notations used in the chapter:

C Production capacity;
K_t^i Setup cost at period t at level i, $t \in \{1,..,T\}$, $i \in \{1,2\}$;
p_t^i Unit production cost at period t at level i, $t \in \{1,..,T\}$, $i \in \{1,2\}$;
h_t^i Unit holding cost at period t at level i, $t \in \{1,..,T\}$, $i \in \{1,2\}$;

The decision variables are, for each period $t \in \{1,..,T\}$ and level $i \in \{1,2\}$:

x_t^i Amount of production at period t at level i;
y_t^i Production indicator (binary variable) at period t at level i;
s_t^i Stock level at the end of period t at level i;

2-CLSP-CC can be formulated as follows:

$$\min \sum_{t=1}^{T} (K_t^1 y_t^1 + K_t^2 y_t^2 + p_t^1 x_t^1 + p_t^2 x_t^2 + s_t^1 h_t^1 + s_t^2 h_t^2) \tag{P}$$

subject to

$$x_t^1 + s_{t-1}^1 = x_t^2 + s_t^1, \; \forall t \in \{1,..,T\}, \tag{2.1}$$

$$x_t^2 + s_{t-1}^2 = d_t + s_t^2, \; \forall t \in \{1,..,T\}, \tag{2.2}$$

$$x_t^i \le \min\{C, \sum_{t'=t}^{T} d_{t'}\} y_t^i, \; \forall t \in \{1,..,T\}, \forall i \in \{1,2\} \tag{2.3}$$

$$s_0^1 = s_0^2 = 0 \tag{2.4}$$

$$x_t^i \ge 0, \; \forall t \in \{1,..,T\}, \forall i \in \{1,2\} \tag{2.5}$$

$$s_t^i \ge 0, \; \forall t \in \{1,..,T\}, \forall i \in \{1,2\} \tag{2.6}$$

$$y_t^i \in \{0,1\}, \; \forall t \in \{1,..,T\}, \forall i \in \{1,2\} \tag{2.7}$$

The objective function minimizes the setup, production and inventory costs. Constraints (2.1) and (2.2) correspond to the flow conservation at each production level. Constraints (2.3) ensures that a production at period t cannot exceed the capacity and incurs a setup costs. Constraint (2.4) imposes a null inventory at the beginning of the horizon.

Structure of Extreme Points

Figure 2.3 provides a network representation of a 2-CLSP-CC instance over an horizon of $T = 6$ periods: A node (i, j) represents level i at period j and the arcs between the nodes indicate the possible flows between two nodes and the corresponding production/holding costs (see Section1.1). As an illustration, let us consider the following instance (I) over a time horizon of 6 periods. The capacity at each level is equal to $C = 10$; the setup costs are $K_t^1 = 100$ at the manufacturer level and $K_t^2 = 1$ at the retailer level

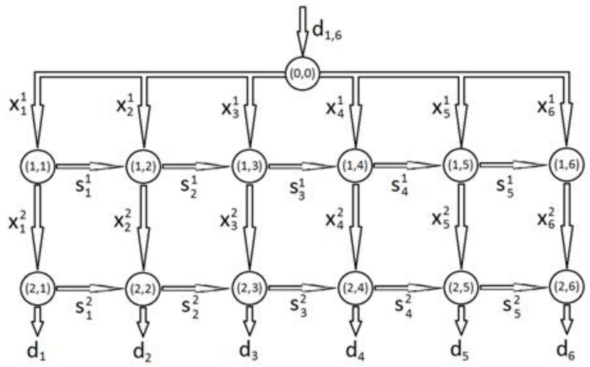

FIGURE 2.3: The network representing a 2-level lot-sizing-problem with $T = 6$ periods

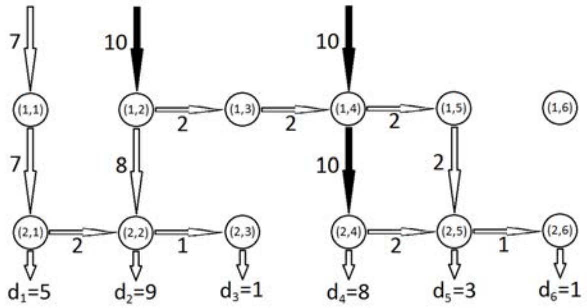

FIGURE 2.4: Representation of the optimal solution of instance (I). Full arcs are represented in bold, and correspond to saturated productions periods, while empty arcs represent the free arcs (non-saturated productions or inventory).

$\forall t \in \{1, .., T\}$. Unit costs of production are null at both levels ($p_t^1 = 0$ and $p_t^2 = 0\ \forall t \in \{1, .., T\}$). Holding costs are defined as follow: $h_t^1 = 2$ for $t \in \{2, 3, 4\}$, $h_t^2 = 1$ for $t \in \{1, 2, 4, 5\}$ and $h_1^1 = 100$, $h_5^1 = 100$, $h_3^2 = 100$. Finally, the demand to satisfy is given by the vector $d = [5, 9, 1, 8, 3, 1]$. Figure 2.4 represents a solution for instance (I): an arc between two nodes (i_1, j_1) and (i_1, j_2) illustrates a flow between two periods j_1 and j_2 at level i_1, i.e. the amount of inventory stored between these the two periods. Similarly, an arc between two nodes (i_1, j_1) and (i_2, j_1) represents a production at level i_2 in period j_1. Observe that instance (I) obeys the path non-speculative cost structure since the unit production costs are null, but not the echelon non-speculative cost structure since the echelon holding cost of the first periods is negative.

It can be easily checked that the solution drawn in Figure 2.4 is the unique optimal solution of instance(I), and thus is an extreme point. Notice that the graph induced by the free arcs is acyclic. However, it is relevant to observe that the graph corresponding to an extreme point solution of the problem can contain cycles, like the one between nodes $(1, 4)$, $(1, 5)$, $(2, 4)$ and $(2, 5)$, due to the full arcs. Such structures are non-trivial and may be difficult to analyze. We present in the next section some additional dominance properties, based on our assumptions on the cost structure, which will be useful to characterize the possible structures of an optimal solution.

2.3 Properties of double-nested policies

In this section we present some structural properties for 2-CLSP-CC, satisfied by path non-speculative cost structures. These properties on what we call the double-nested policies, will enable us to propose an efficient dynamic programming algorithm in the next section. We assume for the rest of the chapter that unit costs of production and inventory costs follow the cost structure defined in section 2.1. Recall that setup costs at the first level are non-increasing in time. Also notice that we do not make any assumption on h_t^1 and h_t^2. In particular, we do not restrict to positive echelon holding costs. Be aware that, although the non-speculative property is expressed on paths, costs are defined on arcs.

First, since we consider linear holding costs, we can assume that the demand in each period is lower than or equal to the capacity at the retailer. Indeed, if at a period t we have $d_t > C$, any feasible policy has to carry at least $(d_t - C)$ units in stock at the retailer to satisfy the demand in period t, incurring a holding cost of $h_{t-1}^2(d_t - C)$. Hence, we can transfer the excess of demand to the preceding period $t-1$ without changing the set of feasible solutions, and decreasing the cost of any policy by a fixed term (see van Hoesel and Wagelmans Van Hoesel and Wagelmans, 1996).

We state the classical following proposition:

Property 1 *There exists an optimal solution such that at most one fractional period occurs in each manufacturer subplan. If there exists, the fractional period is the first period of the subplan.*

Proof. The proposition relies on the fact that for concave costs, there exists an optimal flow with no cycle of free arcs. Since a positive inventory is carried on the arcs inside a subplan, we can have at most one free arc connecting the source of the network to a period of the subplan. This establishes that at most one fractional production period may occur. To establish that it corresponds to the first period, we can notice that postponing the production of one unit at a period a to a later period a' where the setup cost is already paid cannot increase the cost of the policy since $q_{ab}^t \geq q_{a'b}^t$ with our cost structure. This exchange is feasible if a' is not saturated and the inventory level is positive between a and a'. As a consequence, if the fractional period occurs at a period $\bar{a}$ which is not the first period u of the subplan, we can postpone some units of u, till either $\bar{a}$ becomes saturated, or another regeneration period appears between u and $\bar{a}$. The result follows. □

A policy is said to be *nested* if the retailer produces in each period the manufacturer produces. It means that there is no incentive for the manufacturer to anticipate the production of an order of the retailer. The following property holds:

Property 2 (Dominance of nested policies) *There exists an optimal policy which is nested, that is,* $x_t^1 > 0$ *implies* $x_t^2 > 0$*.*

Proof. Let us consider an optimal solution having a period a with a production $x_a^1 > 0$ at the manufacturer and no production at the retailer. As a consequence, we have $s_a^1 \geq x_a^1$. We show that this production can be postponed at a subsequent period without increasing the cost of the solution. Let us focus on the amount of units stored in periods subsequent to a. As long as both levels do not produce, the stock does not evolve. If both levels produce, Property 1 implies that the manufacturer produces at full capacity (the period is not the first of a manufacturer subplan). As both capacities are equal, whatever the quantity produced at the retailer, the stock at the end of the period is at least equal to the stock at its beginning. Hence, there exists a period a' subsequent to a with a production at the retailer and no production at the manufacturer, and such that the amount of stock between a and a' is always greater than or equal to x_a^1. It is thus possible to postpone the production at the manufacturer from period a to a' while keeping a feasible policy. Recall that we restrict ourselves to the case of non-increasing setup costs at the manufacturer level. Moreover, $q_{ab}^t \geq q_{a'b}^t$ due to our cost structure. As a consequence, the production at period a can be postponed to period a' without increasing the cost of the solution, and thus the resulting policy is still optimal. □

For the same reasons, there exists an optimal policy such that $s_t^1 < C$ for each period t, that is, the stock level at the manufacturer is always lower than its capacity: a stock greater than C implies that a production can be postponed. Notice that this property is not verified in general at the retailer.

In a network representation, a connected component is a subgraph in which any two vertices are connected by a path. Consider an optimal nested policy π. Its flow partitions the network associated with our problem into a set of connected components. For instance, in Figure 2.4, the solution is a unique connected component. Consider such a connected component. It is constituted at each level of a set of consecutive periods. Let r be the first period of the component at the manufacturer level, and let s be the last period of the connected component at the retailer level. Necessary period r corresponds to the first period with a manufacturer production, and thereby also to the first period with a retailer production as the policy is nested. Hence, r is also the first period of the component at the retailer level. Considering now period s, it corresponds to the last period whose demand is satisfied by units produced inside the component. Again, since the policy is nested, if there is a production at the manufacturer in a period $t \in [r, s]$, the units produced cannot be used to satisfy the demands inside another connected component. Thus each period $t \in [r, s]$ at the manufacturer level either belongs to the connected component, or is isolated, that is, is a connected component by itself (like node $(1, 6)$ in Figure 2.4). By convention, we consider that isolated period t of $[r, s]$ belongs to the connected component. As a consequence, the connected component can be described only with the two periods r and s.

A connected component $[r, s]$ decomposes at each level into a series of *subplans*. We turn now our attention to the retailer level. Our first dominance is the counterpart of the nested dominance at the manufacturer, see Property 2. Basically it states that if a production occurs in period t at the retailer, then the manufacturer also produces in period t. However, this property holds only for full production periods:

Property 3 *There exists an optimal policy such that each full production period at the retailer is also a full production period at the manufacturer, that is,* $x_t^2 = C \Rightarrow x_t^1 = C$.

Proof. The property is a direct consequence of Property 1, and the fact that the stock level at the manufacturer is always lower than its capacity: let t be a period which contradicts the Property, that is, $x_t^2 = C$ and $x_t^1 < C$. If $x_t^1 > 0$, s_{t-1}^1 must be strictly positive to balance the flows, which contradicts Property 1. If $x_t^1 = 0$, the C units produced at the retailer at period t must come from the inventory. Thus $s_{t-1}^1 = C$, which contradicts the fact that the stock level at the manufacturer is always lower than its capacity. □

Definition 1 *A policy is said double-nested if the retailer produces whenever the manufacturer does, and the manufacturer produces at full capacity whenever the retailer does. That is,* $x_t^2 = 0 \Rightarrow x_t^1 = 0$ *and* $x_t^2 = C \Rightarrow x_t^1 = C$

Properties 2 and 3 imply that double-nested policies are dominant for our problem. Such double-nested policies have a very nice structure. In particular, if the production at the retailer is not fractional, then the production at the manufacturer and at the retailer are equal. As a consequence, in any double-nested policy, the amount of inventory at the manufacturer is only modified at a period with a fractional production at the retailer. This gives rise to the following property:

Property 4 *There exists an optimal double-nested policy such that there is no unit in stock at the manufacturer level between any two consecutive retailer fractional periods belonging to the same retailer subplan.*

Proof. Consider an optimal double-nested policy. We can choose it such that no cycle of free arcs appears in the flow. Let k and l be two consecutive fractional production periods in a retailer subplan (u, v). At the manufacturer level, the stock level s_k^1 at the end of period k is equal to the entering stock level s_{l-1}^1 at period l. Since the stock level is positive at the retailer level between k and l by definition of a subplan, a positive stock at the end of period k at the manufacturer level would create a cycle of free arcs in the flow, which contradicts our choice of the optimal policy. □

The fact that at most one fractional period can take place in each manufacturer subplan holds under very general cost structure, namely for concave costs. Under our cost structure, using double-nested policies, we can state a property stronger than Property 1:

Theorem 1 *There exists an optimal solution such that at most one fractional period occurs at the manufacturer level in each connected component. Moreover, under our cost structure, the fractional period is located at the first period of the connected component.*

Proof.: Consider a policy which is double-nested and an extreme point solution. For the sake of contradiction, assume k and l to be two fractional production periods at the manufacturer located in a connected component $[r, s]$. We will prove that the entering stock and the outgoing stock at the highest level (that is, at level one if such a stock exists, otherwise at level two) of each period between k and l are either at the same level or linked by a free arc, thus creating a path of free arcs between k and l. For each period $t \in \{k+1, ..., l-1\}$, since we are inside a connected component, there is a at least one entering stock, and one outgoing stock. Let us consider the entering stock at the highest level in t and the outgoing stock at the highest level in t. If both flows are at the same level, periods $t-1$ and $t+1$ are linked by free arcs. Otherwise, if one of the flow is at the manufacturer while the other one is at the retailer, the amount of inventory at period t changes, which implies that there is a fractional production at the retailer in t. As a result, in each case the two flows are at the same level or linked by a free arc. As the highest outgoing flow at period t corresponds to the highest entering flow in $t+1$, there is a path of free arcs linking periods $k+1$ and $l-1$ of the connected component. Moreover, periods k and l are connected to this path as there is a fractional production at both levels at these periods. Hence, we have a cycle of free arcs with the source, which contradicts that the policy is an extreme point solution. This contradicts our choice of the policy as an extreme point solution.
For the second part of the theorem, recall that under our cost structure, disregarding the setup costs, there is no incentive to anticipate a production. Let w be a fractional production period at the manufacturer in a connected component (r, s). Due to the nested property and to Property 3, a fractional production also occurs at the retailer in w. As a consequence, an additional unit can be produced in period w, both at the manufacturer and the retailer level. For the sake of contradiction, assume that $w > r$. Since $s_w^1 = 0$ due to Property 1, we must have $s_w^2 > 0$ inside the connected component. Consider one unit in stock and let b be the period where it has been produced at the retailer and a be the period where it has been produced at the manufacturer. Due to our cost structure, as $a \leq b < w$, we have $q_{ab}^w \geq q_{ww}^w$. Hence, we can increase this way the production in period w, without increasing the production costs, until one capacity is saturated, at the retailer or at the manufacturer level. Since we have $x_w^1 \geq x_w^2$, the capacity at the manufacturer is saturated first. As a result, if a fractional production exists in a connected component of an optimal solution, it must be the first period of the connected component. □

A retailer subplan is more involved than a manufacturer subplan, for which we know that at most one fractional period may occur, and only in the first period (see in Figure 2.4). However, as a direct consequence of the

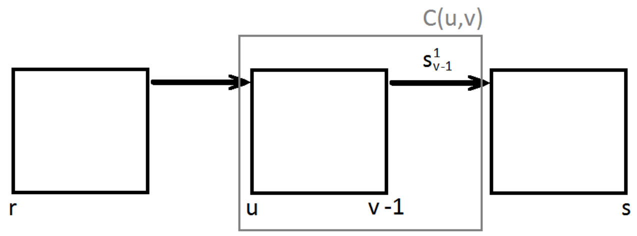

FIGURE 2.5: Schematic illustration of the cost of a retailer subplan (u, v) in a connected component (r, s)

preceding property, we can assert that at most 2 fractional periods may occur in a retailer subplan:

Property 5 *There exists an optimal policy such that each retailer subplan contains at most two fractional periods.*

Proof. We consider an optimal policy satisfying the preceding dominances. Let (u, v) be a retailer subplan, and assume for the sake of the contradiction that j, k and l, with $u \leq j < k < l \leq v - 1$, correspond to consecutive fractional production periods. Since there is no stock at the manufacturer level between periods j and k and between periods k and l (Property 4), the conservation of the flow implies that $x^1_k = x^2_k$. As a consequence, period k is a fractional period at the manufacturer, that is, $0 < x^1_k < C$. But obviously period k cannot be the first period of the connected component, thus Definition 1 implies that $x^1_k \in \{0, C\}$, which contradicts Property 4
□

In the next section, we use the dominance properties of the double-nested policies to develop a polynomial time dynamic programming algorithm.

2.4 A Dynamic Programming Algorithm

A solution of problem 2-CLSP-CC decomposes into a set of connected components. Recall that a connected component can be represented by its first and its last period, so there are $O(T^2)$ possible connected components. A classical way to find an optimal solution of a given instance is to evaluate the minimal cost of all the $O(T^2)$ possible connected components, and then to solve a shortest path problem, in the acyclic graph whose nodes represent the periods from 1 to T, and whose arcs represent the costs of the connected components defined by the two nodes they are adjacent to. The minimum cost path between node 1 and a fictive node $(T + 1)$ represents an optimal solution of the problem and can be computed in $O(T^2)$, given that the minimum costs of each connected component are yet computed.

2.4.1 Computation of the minimum cost of a connected component

From now on, we focus on the computation of the optimal cost of the connected components. In the following, we consider a connected component defined by its first period r and its last period s. We denote by $\mathscr{C}[r,s]$ its optimal cost. To find an optimal policy on the connected component $[r,s]$, the principle of our algorithm is to decompose it in its turn into a sequence of retailer subplans. Notice that by definition of a connected component, there is no entering stock at period r and no outgoing stock at period s. Hence, period r is the first period of a retailer subplan and s is the last period of a retailer subplan. Thus such a decomposition into retailer subplans exists. Consider a retailer subplan (u,v) inside the connected component. We denote by $\mathscr{C}(u,v)$ its cost in an optimal policy. In this cost, in addition to the production and holding costs at the retailer level over the periods u up to $v-1$ (included), we also account for the production and holding costs incurred at the manufacturer level between periods u to $v-1$, see a schematic representation Figure 2.5. Again, if the costs $\mathscr{C}(u,v)$ are known for all the possible retailer subplans (u,v) inside the connected component $[r,s]$, the optimal cost $\mathscr{C}[r,s]$ of the connected component can be computed as a shortest path problem. However, to use this approach, there are two difficulties we have to overcome:

1. If period u is by definition a regeneration point at the retailer level, it is not a regeneration point at the manufacturer level, except if $u = r$. Hence, $\mathscr{C}(u,v)$ depends on the entering stock level s^1_u and the outgoing stock level s^1_{v-1} at the manufacturer. Nevertheless, we show that the amount of entering stock at the manufacturer at the beginning of a retailer subplan (u,v) inside a connected component $[r,s]$ is fixed in a dominant double-nested policy, making the optimal cost of a retailer subplan independent of the evaluation of the other retailer subplans of the connected component.

2. Evaluating cost $\mathscr{C}(u,v)$ is still a 2-level capacitated lot-sizing problem. The main idea of our algorithm is to reduce this problem to a single echelon lot-sizing problem, without considering the manufacturer level. As a consequence, algorithms of the literature for CLSP can be used to compute efficiently $\mathscr{C}(u,v)$.

The second point, the computation of the optimal cost of a retailer subplan, is detailed in Section 2.4.3. We show that it can be performed in time $O(T^2)$ for a given retailer subplan. We explain now the first point, that is, how the cost of a given connected component can be determined using a decomposition into retailer subplans. Recall that we consider a connected component $[r,s]$. We first show that the cost $\mathscr{C}(u,v)$ of an optimal policy over a retailer subplan (u,v) can be determined regardless of the others subplans, that is, that the entering stock level s^1_{u-1} at the manufacturer is fixed in a dominant policy. If $u = r$, we have clearly $s^1_{u-1} = 0$. If $r < u$, due to Theorem 1, no fractional production period may occur at the manufacturer level between period u and the end of the connected component. Thus, the total amount $X^1_{u,s}$ produced at the manufacturer during these periods is a multiple of capacity C. Writing the conservation of the flow, we have $s^1_{u-1} + X^1_{u,s} = D_{u,s}$. Modulo C the equality boils down to $s^1_{u-1} \equiv D_{u,s} \pmod{C}$. For short we denote by σ^s_u the quantity $D_{u,s} \bmod C$. As stressed by the notation, quantity σ^s_u only depends on periods u and s. Since the number of units in stock at the manufacturer level is lower than C in every periods of a dominant policy, we obtain from the previous discussion that:

$$s^1_{u-1} = \begin{cases} 0 & \text{if } u = r \\ \sigma^s_u \equiv D_{u,s} \bmod C & \text{if } u > r \end{cases}$$

The first retailer subplan (r, v) of the connected component has a special structure, since period r is the only fractional period of the connected component. Notice that the fractional quantity produced is σ^s_r. We denote by $\mathscr{C}^s_{\text{FST}}(r, v)$ the optimal cost of such a first subplan, and by $\mathscr{C}^s_{\text{IN}}(u, v)$ the optimal cost of a subplan with a positive entering stock σ^s_u. For a given period s, we can compute in time complexity $O(T^2)$ the shortest path $\mathcal{C}_{(u,s]}$ from each period u to period s in the acyclic graph with costs $\mathscr{C}^s_{\text{IN}}(u, v)$ on the arcs. The optimal cost $\mathscr{C}[r, s]$ of a connected component $[r, s]$ can then be determined in time $O(T)$ as $\min\{\mathscr{C}^s_{\text{FST}}(r, v) + \mathcal{C}_{(v,s]} \mid r \leq v \leq s\}$. Hence, assuming that the $O(T^3)$ values for the optimal cost of the retailer subplans are known, the optimal cost of a connected components $[r, s]$ can be computed $O(T^2)$ time complexity.

2.4.2 Time complexity of the algorithm

The overall time complexity of the algorithm can be analyzed as follows:

- We have $O(T^3)$ costs $\mathscr{C}^s_{\text{FST}}(r, v)$ to evaluate, for all possible triples of periods r, v, s, with $r \leq s$ and $r < v \leq s + 1$, which corresponds to the optimal costs of the first retailer subplan of a connected component.
- We have $O(T^3)$ costs $\mathscr{C}^s_{\text{IN}}(u, v)$ to evaluate, for all possible triples of periods u, v, s, with $u \leq v$ and $u < v \leq s + 1$, which corresponds to the optimal costs of the other retailer subplans. Notice that these costs are independent of the first period r of their connected component.

Once the $O(T^3)$ values for the optimal cost of the retailer subplans are computed, the optimal cost of all the possible connected components $[r, s]$ can be evaluated in time $O(T^3)$. The optimal planning can then be determined by a shortest path algorithm in time $O(T^2)$ as explained in the beginning of this section. Hence, the time complexity of our algorithm is dominated by the computation of the optimal costs of all the possible retailer subplans. More precisely, if the optimal cost of a given retailer subplan can be determined in time complexity at most $f(T)$, then the overall complexity of our dynamic approach is in $O(T^3 f(T))$. It is hence essential to compute these costs efficiently. We detail in the next section how we can reduce this computation to a one-level CLSP, taking advantage of the structure of the double-nested policies.

2.4.3 Computation of the minimum cost of a retailer subplan

In this section, we consider a retailer subplan (u, v) inside a connected component $[r, s]$. We show that the problem of evaluating the minimum cost $\mathscr{C}(u, v)$ of the retailer subplan can be reduced to a single-echelon capacitated lot-sizing problem, CLSP. This is due to the fact that, roughly speaking, in a double-nested policy, the two levels are synchronous: Except at the fractional periods of the retailer, both levels produce the same amount in each period, either 0 or C units, see Definition 1.

We focus first on the case where $r < u$ and $v - 1 < s$, which implies that there is an entering and an outgoing stock level at the manufacturer level ($\sigma_u^s \sigma_v^s > 0$). It corresponds to the case of a subplan located between two other subplans, which is the most complex structure. Other structures are presented in Appendix A. Recall that a fractional production at the manufacturer level can only occur at the first period of the connected component, see Corollary 1. Hence, only full production periods can take place at the manufacturer in (u, v). Moreover, there exists an optimal policy such that each retailer subplan contains at most two fractional periods, see Property 5. Let k and l, $k \leq l$, be the two fractional periods at the retailer, with $k = l$ if only one fractional production occurs inside the subplan, and by convention $k = l = v$ if no fractional production occurs. We show that the optimal cost $\mathscr{C}^s_{\text{IN}}(u, v)$ of the subplan can be computed as

$$\mathscr{C}^s_{\text{IN}}(u, v) = G(k, l) + H(k, l) \tag{2.8}$$

where $G(k, l)$ is the minimum cost of a single-level CLSP problem, and $H(k, l)$ accounts for the holding costs incurred at the manufacturer level in an optimal solution. For convenience, production costs at fractional periods k and l are also incorporated in $H(k, l)$, since we show in the following that quantities x_k^i and x_l^i are fixed at both levels when k and l are fixed. We detail first how this cost $H(k, l)$ can be determined, depending on the number of fractional periods occurring at the retailer level inside the subplan. We have only 3 cases to distinguish.

Case with two fractional periods inside the subplan

We consider first the case where two fractional periods k and l, $k < l$, occur in the subplan (u, v). Recall that for any other period t different of k and l, we have $x_t^1 = x_t^2$, and either t is a full production period, or no production occurs at period t. We have also established that there is no unit in stock at the manufacturer between the two fractional periods of a retailer subplan, see Property 4. In particular, we have $s_k^1 = s_{l-1}^1 = 0$. It implies that:

- As $0 < x_k^2 < C$ and $s_k^1 = 0$, no production can occur at the manufacturer in period k, that is, $x_k^1 = 0$. Indeed, period k cannot be a fractional production period at the manufacturer (Theorem 1). It follows that $x_k^2 = s_{k-1}^1$. Moreover, since both levels produce the same quantity in period u up to $k - 1$, the amount of stock at the manufacturer cannot evolve. It

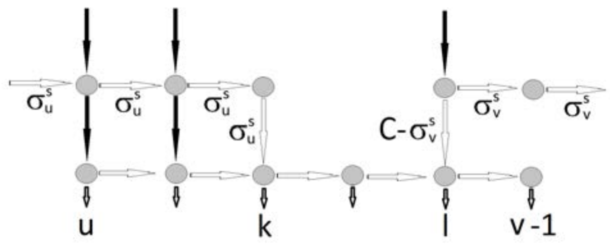

FIGURE 2.6: Retailer subplan with two fractional retailer production periods. Full productions are represented with bold arcs.

follows that exactly σ_u^s units are carried in stock from period $u-1$ up to k.

- Since $s_{l-1}^1 = 0$, a full production must occur in period l at the manufacturer to supply the fractional production of the retailer. As a consequence, there is a strictly positive amount of units stored at the manufacturer at the end of period l, equal to $s_l^1 = C - x_l^2$. Similarly, this amount is carried in stock till period v, and thus must be equal to σ_v^s. As a consequence, we have $x_l^2 = C - \sigma_v^s$.

Figure 2.6 illustrates the situation inside the subplan. Clearly, the holding costs incurred at the manufacturer can be easily deduced from periods k and l. Recall that the quantities produced at the fractional periods at the retailer are also fixed. From the previous discussion, we have $x_k^2 = \sigma_u^s$ and $x_l^2 = C - \sigma_v^s$. The cost $H(k,l)$ is thus equal to:

$$H(k,l) = \sigma_u^s(h_u^1 + \cdots + h_{k-1}^1) + \sigma_v^s(h_l^1 + \cdots + h_{v-1}^1) + K_k^2 + p_k^2(\sigma_u^s) \\ + K_l^1 + K_l^2 + p_l^1(C) + p_l^2(C - \sigma_v^s)$$

Case with a single fractional period inside the subplan

Let us now assume that there is a single fractional period at the retailer, occurring at period k, see Figure 2.7. In a way similar to the previous case, the entering stock σ_u^s is carried at the manufacturer from period u to k and the outgoing stock σ_v^s is carried at the manufacturer from periods k to v. Considering the amount of production x_k^2 at the retailer in the fractional period, two different cases have to be distinguished. If σ_u^s is greater than σ_v^s, then the entering stock in period k at the manufacturer is greater than its outgoing stock. It implies that k is not a production period at the manufacturer, and we have $x_k^2 = \sigma_u^s - \sigma_v^s$, see Figure 2.7. Consequently, the holding cost incurred at the manufacturer plus the production costs at period k sum up to:

$$H(k,k) = \sigma_u^s(h_u^1 + \cdots + h_{k-1}^1) + \sigma_v^s(h_k^1 + \cdots + h_{v-1}^1) + K_k^2 + p_k^2(\sigma_u^s - \sigma_v^s)$$

Otherwise, if σ_u^s is lower than σ_v^s, period k is necessarily a full production period at the manufacturer. The conservation of the flow implies that $\sigma_u^s +$

$C = \sigma_v^s + x_k^2$, see Figure 2.8. Thus we have: $x_k^2 = C + \sigma_u^s - \sigma_v^s$. Consequently, the holding cost incurred at the manufacturer plus the production costs at period k are equal to:

$$H(k,k) = \sigma_u^s(h_u^1+\cdots+h_{k-1}^1)+\sigma_v^s(h_k^1+\cdots+h_{v-1}^1)+K_k^1+p_k^1(C)+K_k^2+p_k^2(C+\sigma_u^s-\sigma_v^s)$$

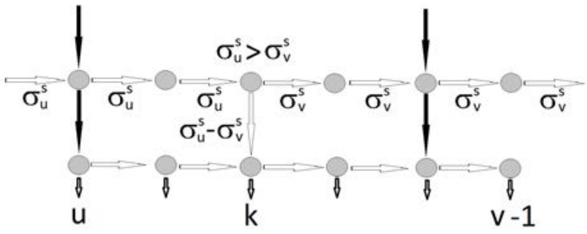

FIGURE 2.7: Retailer subplan with one fractional retailer production period and no manufacturer production in k

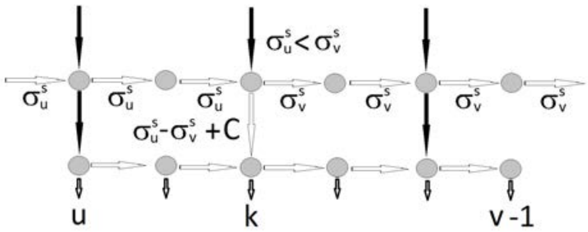

FIGURE 2.8: Retailer subplan with one fractional retailer production period with a manufacturer production in k

Case with no fractional period inside the subplan

Let us finally assume that there is no fractional production at the retailer inside the subplan (u, v). Recall that by convention we have $k = l = v$. Since in each period the same amount is produced at the manufacturer and at the retailer, the stock level at the manufacturer does not vary inside the subplan. Hence, we must have $\sigma_u^s = \sigma_v^s$ and $H(v, v)$ boils down to:

$$H(v,v) = \sigma_u^s(h_u^1 + \cdots + h_{v-1}^1)$$

Reduction to a single-level CLSP problem

We have shown that, if the location of the fractional productions at the retailer are known in a subplan, then the amount of units produced at the fractional periods and all the quantities stored at the manufacturer along $(u, v-1)$ are also known, due to the fact that the other production periods are full production periods simultaneously at the retailer and at manufacturer levels in a double-nested policy. We now explain how to compute the cost represented by $G(k,l)$ in Equation 2.8.

To determine in which periods full productions take place in an optimal policy, we reduce the problem to a single-level capacitated lot-sizing problem in the following way. Assuming that we know the fractional periods, we can decrement the demands which are totally or partially satisfied by the units produced at the fractional retailer production periods. Here we assume that units produced at a fractional period k are used to fulfill the closest demands, that is, d_k, then d_{k+1} if $x_k^2 > d_k$, and so on. In the reduced problem, we prevent a policy to order in periods k and l by setting a very large setup cost. The setup cost for another period t is defined as $K_t' = K_t^1 + K_t^2$, since t is necessarily a (full) production period both at the manufacturer and at the retailer in a double-nested policy. We have obtained a single echelon lot-sizing problem, forgetting about the manufacturer level, with full production periods to be determined to satisfy the remaining demand. This problem is a discrete CLSP, which can be solved in time complexity $O(T)$ with the greedy algorithm presented by van Hoesel and Wagelmans (1996) Van Hoesel and Wagelmans, 1996. Since we have $O(T^2)$ possible couples to consider for the fractional periods k and l, the optimal cost $\mathscr{C}_{\text{IN}}^s(u,v)$ of the subplan can be computed in time $O(T^3)$.

Improvement of time complexity

It is possible to reduce the complexity by a factor T by using the algorithm presented by van Hoesel and Wagelmans in 1996 Van Hoesel and Wagelmans, 1996. They proposed an $O(T^3)$ algorithm for CLSP with a constant capacity, concave production costs and linear holding costs. The algorithm is in two phases: the first one evaluates the minimum cost for all subplans (u,v), $1 \leq u \leq v \leq T$ and the second phase applied a shortest path algorithm to find the optimal solution of the problem. We focus here on the first phase. Our idea is to fix only one fractional period, and to let the algorithm of Van Hoesel and Wagelmans, 1996 determine optimally the location of the second one, if any. Notice that the authors define a subplan as a set of consecutive periods $u,..,v-1$ such that at most one fractional production occurs. It implies that (u,v) is still a subplan after the production of the fixed fractional period has been discarded from the demands. The principle of their algorithm is to compute in time $O(T)$ a first optimal solution for the subplan (u,v) when the fractional production is fixed in a period t. From this solution, they establish that it is possible to evaluate in constant time the optimal solution when the fractional solution occurs in $t-1$. As a result, the minimum cost of the subplan can be found in time $O(T)$ by varying the fractional production period from v down to u. We apply their result to a subplan having two fractional production periods by fixing the first fractional period k: the fractional production x_k^2 is discarded of the demands, and k is made unavailable for production. For a fixed fractional period k, the best location for the remaining fractional period l can be found in $O(T)$, by varying l from v down to $k+1$. The computation is repeated for each possible location of k. As there are T possible positions for k, it incurs an overall complexity in time $O(T^2)$ to determine the optimal position of k and

l for a given subplan. If subplan (u, v) has less than two fractional periods, the algorithm of Van Hoesel and Wagelmans, 1996 directly determines the optimal cost of the subplan in time $O(T)$.

In Appendix A, we present the other possible structures of retailer subplans and how to evaluate them. In each case, the optimal cost of the subplan can be determined in time at most in $O(T^2)$. Since there is $O(T^3)$ different configurations of retailer subplans to be evaluated, the overall complexity of the algorithm to deliver an optimal policy over the time horizon is in $O(T^5)$. We state the following result:

Theorem 2 *If our cost structure follows the path non-speculative motives, the 2 level in series lot-sizing problem with identical and stationary capacities can be solved in time $O(T^5)$*

Case with echelon non-speculative motives

We discuss here the particular case of echelon non-speculative motives. Recall that at the retailer level, this cost structure requires that $p_t^2 + (h_t^2 - h_t^1) \geq p_{t+1}^2$ for any period t. Notice that the condition is written with the echelon holding cost $(h_t^2 - h_t^1)$, but does not require it to be positive. As proved in section 2.1, it is a particular case of the path non-speculative cost structure. Thus all the properties established previously remain valid. However, under an echelon non-speculative cost structure, there are stronger properties on the structure of an optimal solution:

Property 6 *Under echelon non-speculative motives, there exists an optimal solution such that at most one fractional period occurs at the retailer level in each retailer subplan. Moreover, the fractional period is located at the first period of the retailer subplan.*

Proof. To see this, it is sufficient to observe that there exists an optimal solution such that the retailer produces a fractional quantity only at periods for which the entering stock level is null, that is, at regeneration points. Indeed, consider a fractional period k at the retailer, and assume that $s_{k-1}^2 > 0$. Let us consider a unit stored at the retailer between periods $k-1$ and k, and let t be the period at which it was produced at the retailer. Due to non-speculative motives at the retailer, $p_t^2 + h_t^2 + h_{t+1}^2 + ... + h_{k-1}^2 \geq h_t^1 + h_{t+1}^1 + ... + h_{k-1}^1 + p_k^2$. It is thus possible to postpone the production of this unit at the retailer from period t to period k without increasing the global cost of the solution. This process can be repeated until the production at the retailer in k is saturated, or until s_{k-1}^2 becomes null. □

With this property the time required to compute the cost of a retailer subplan is reduced to $O(T)$ since only the full production periods need to be determined. Hence, the time complexity to compute the optimal cost of a retailer subplan drops to $O(T)$ with echelon non-speculative motives, instead of $O(T^2)$ assuming only path non-speculative motives. Moreover, we can notice that it is not necessary to know the last period of the connected component to which each subplan belongs. This is due to the fact that, in a given subplan (u, v), all the production periods located between periods $u + 1$ and $v - 1$ are full productions. Hence, the value of the fractional production in u is necessarily equals to $D_{u,v-1} \bmod C$. As a consequence, the optimal cost of a retailer subplan (u, v) can be determined, in time complexity $O(T)$, independently of its connected component. The overall time complexity of our algorithm is reduced to $O(T^3)$. We have the following result:

Theorem 3 *If our cost structure follows the echelon non-speculative motives, the 2 level in series lot-sizing problem with identical and stationary capacities can be solved in time* $O(T^3)$

2.5 Conclusion

In this chapter, we present a 2-level production-in-series lot-sizing problem with identical capacities at both levels. We define a new cost structure generalizing the echelon non-speculative cost structure. We highlight the particular structures of an optimal solution, introducing the double-nested policies, and show how the problem can be reduced to a single-level lot-sizing on a given retailer subplan. This reduction is possible due to the fact that the production periods at both levels are nearly synchronous, except for the fractional production periods, which are limited on a retailer subplan. We propose a dynamic programming algorithm based on the construction of an optimal solution from the set of every possible retailer subplans. The overall time complexity of our algorithm is $O(T^5)$ for path non-speculative cost structure, and in $O(T^3)$ for echelon non-speculative cost structure, which makes it, we believe, of practicable use.

In the next chapter, we study a lot-sizing problem with a general number of level, and we consider procurement costs following a FTL structure, with batch deliveries.

Chapter 3

The multi-level in series lot-sizing problem with batch deliveries

This chapter extends the results of the previous chapter towards two directions: the number of levels is part of the input and the procurement costs follow a FTL structure, that is, deliveries are made by batches. Moreover, the number of batches which can be ordered at each period is limited. This problem is denoted as the multi-level lot-sizing problem with batch deliveries and with an upper limit on the number of batches of each order (M-LSP-B). We have presented the previous chapter using a production terminology. Here, we switch to a distribution terminology which fits better with the cost structure with batch deliveries. Under a specific cost structure with identical and stationary batch sizes, we propose an exact dynamic algorithm running in $O((N^2 + N \log T)T^3)$. Our approach is based on a decomposition of a solution into induced connected components, called boxes, which can be evaluated independently, and reuses the concept of basis path. This chapter is derived from an article submitted to the journal Operations Research and currently in a major revision process (Goisque and Rapine, 2017b).

3.1 Introduction

The multi-level lot-sizing problem with batch deliveries consists in a system of N levels organized in series, typically a supply chain, with an external supplier at the beginning of the chain, and a client with a deterministic periodic demand at the end. The levels are numbered from 1, the most upstream level, to N, the most downstream level where the demand is to be satisfied. Each level i can carry units in stock, from one period to the other, and orders its units from its upstream level $i-1$, using batch deliveries of size C (see Section1.1), identical and stationary at each level. At a level i and a period t, the limit of batches which can be ordered is equal to m_t^i. We think that this problem captures the essence of a large number of complex logistics systems, as the shipments between different storage facilities are often carried out using a common container size. Each transportation of a container operated by a truck incurred a cost which depends mainly on the traveled distance and on the volume. Hence, shipment costs can be quite accurately modeled in many cases by a fixed cost per container plus a unit cost per product transported, resulting in a lot-sizing problem with batch deliveries. The limitation of the size of the shipments that we incorporate in our model allows to reflect practical situations with a transportation capacity, due to the limited number of trucks in the fleet, or with a facility capacity, due to limited available resources for handling operations, or due to a limited number of docks to receive inbound trucks. This multi-level lot-sizing problem with batch deliveries captures two important special cases. In the case where $m_t^i = 1$ for all levels i and all periods t, that is, at most one batch can be ordered, we have a multi-level capacitated lot-sizing problem (M-CLSP). Switching to a production terminology, this problem can model a production line of N machines in series, each one with the same hard production capacity C, and where the fixed cost per batch can be identified with a setup cost of production, that is, if level i decides to produce a positive quantity at period t, a setup cost k_t^i is to be paid. Notice that Chapter 2 deals with the 2-CLSP problem. In the case where all the m_t^i have large values (larger than the total remaining demand $d_t + \cdots + d_T$), we have a multi-level uncapacitated lot-sizing problem with batch deliveries (M-ULSP-B), which can model a supply chain where items are transported using identical shipping containers or trucks of capacity C. As many items as needed can be shipped in each order, but a fixed cost k_t^i is incurred for each vehicle/container used. The batch size C is often referred in the literature as a soft capacity. As far as we know, the status of both problems, with hard or soft capacities, is open when the number N of levels is part of the inputs.

Very few results are known for multi-level lot-sizing problems with batch deliveries. We establish in this chapter that the multi-level lot-sizing problem M-LSP-B can be solved in polynomial time under the following 3 assumptions on the parameters:

(A1) Unit ordering and holding costs follow a *non-speculative motives echelon cost structure* (see Section1.1). At the first level the assumption is similar to that used in single-level problems: $p_t^1 + h_t^1 \geq p_{t+1}^1$ must hold for any period $t \in \{1, ..., T-1\}$. At the other levels, the condition is slightly modified to take into account the holding cost incurred at level $i-1$ when postponing the order: Inequality $p_t^i + (h_t^i - h_t^{i-1}) \geq p_{t+1}^i$ must hold for all $i \in \{2, ..., N\}$, $t \in \{1, ..., T-1\}$. The absence of speculative motives is a classical assumption in literature, also known as the Wagner-Whitin cost structure, which states that, considering only the unit ordering and holding costs, it is always preferable to order as late as possible at each level;

(A2) At each level, the fixed costs per batch are non-increasing with time, except for the last level for which they are unrestricted. That is, $k_t^i \geq k_{t+1}^i$ for each level $i < N$ and each period $t < T$;

(A3) At each period t, the limit m_t^i of the number of batches that can be ordered is non-increasing with the index level. That is, the maximal number of batches a level i can order is smaller than or equal to the maximal number of batches its upstream level can order. Notice that we do not require the m_t^i's to be monotone relatively to the periods.

To the best of our knowledge, we are the first to propose an algorithm which is both polynomial in the length of the planning horizon and in the number of levels, for the multi-level lot-sizing problem with capacities (hard or soft) at each level. Without the assumptions $A1$, $A2$ and $A3$, the complexity status of problem M-CLSP with identical hard capacities is open. Not only we prove that the problem is polynomial under our assumptions, but the algorithm is really attractive for a practical use, due to its low time complexity and its simplicity. In addition, we extend our model to incorporate non-null setup costs at the first level of the network. Since the first level may represent orders to an external supplier (or the production stage in a distribution network), it is particularly relevant to be able to model more general cost structure at this level. Our approach reuses the concept of basis path introduced by Hwang, Ahn, and Kaminsky, 2013 inside a new decomposition of an optimal flow into connected components on networks restricted to a subset of levels.

Organization of the chapter. Next section presents formulations of the problem together with some dominant properties implied by our assumptions $A1$, $A2$, $A3$. In §3.3, we study the connected components of a policy induced on each of the networks restricted to the last downstream levels, introducing the concepts of box, basis path and gate. We show in §3.4 that an optimal solution admits a decomposition into boxes that can be computed efficiently by a dynamic programming approach. Finally, in §3.5, our results are extended to a more general cost structure with non-null setup costs at the first level. §3.6 concludes the chapter.

3.2 Problem Formulation and Structural Properties

This section provides a mixed integer linear formulation for the multi-level lot-sizing problem with batch deliveries and a limit on the number of batches that can be ordered at each period. We then introduce a classical network flow representation of the problem and establish some structural properties of an optimal policy. These properties allow us to decompose an optimal flow in §3.4. We give below the notations used throughout the chapter. We have the following parameters:

N	number of levels of the distribution network; levels are indexed from 1 to N;
C	size of a batch;
m_t^i	maximal number of batches that can be ordered at period t at level i;
k_t^i	fixed cost of a batch ordered at period t at level i;

We also use as a shorthand the following notations:

$H^i_{t,t'} \equiv h^i_t + \cdots + h^i_{t'}$	cumulative holding cost between periods t and t' at level i;
$q^i_t(x) \equiv p^i_t x + \lceil x/C \rceil k^i_t$	cost of ordering x units at period t at level i;

Since we can assume a FIFO discipline, a policy is entirely specified by the amount of units ordered in each period at each level. We say that a period t at a level i is an *ordering period* if a positive amount is ordered. As the problem is deterministic, we can consider w.l.o.g. that the leadtimes are null: units ordered at a period t are instantaneously available. For convenience, we also manipulate the number of units carried in stock at each level. For each period $t \in \{1,..,T\}$ and each level $i \in \{1,..,N\}$, we introduce the following decision variables:

x^i_t	amount of units ordered in period t at level i;
y^i_t	number of batches ordered in period t at level i;
s^i_t	stock level at the end of period t at level i;

3.2.1 Problem Formulation

The multi-level in-series lot-sizing problem with batch deliveries (M-LSP-B) can be formulated as follows:

$$\min \sum_{t=1}^{T} \sum_{i=1}^{N} (k^i_t y^i_t + p^i_t x^i_t + h^i_t s^i_t)$$

subject to

$$x^i_t + s^i_{t-1} = x^{i+1}_t + s^i_t, \ \forall t \in \{1,..,T\}, i \in \{1,..,N-1\}, \tag{3.1}$$

$$x^N_t + s^N_{t-1} = d_t + s^N_t, \ \forall t \in \{1,..,T\}, \tag{3.2}$$

$$x^i_t \leq C y^i_t, \ \forall t \in \{1,..,T\}, i \in \{1,..,N\}, \tag{3.3}$$

$$y^i_t \leq m^i_t, \ \forall t \in \{1,..,T\}, i \in \{1,..,N\}, \tag{3.4}$$

$$s^i_0 = 0, \ \forall i \in \{1,..,N\}, \tag{3.5}$$

$$x^i_t \geq 0, \ \forall t \in \{1,..,T\}, \forall i \in \{1,..,N\} \tag{3.6}$$

$$s^i_t \geq 0, \ \forall t \in \{1,..,T\}, \forall i \in \{1,..,N\} \tag{3.7}$$

$$y^i_t \in \mathbb{Z}^+, \ \forall t \in \{1,..,T\}, \forall i \in \{1,..,N\} \tag{3.8}$$

The objective function minimizes the ordering costs and the inventory costs. Constraints (3.1) and (3.2) correspond to the flow conservation at each level. Constraints (3.3) ensures that sufficiently many batches are used to ship the units ordered, while Constraints (3.4) limits the number of batches effectively used by the maximum number of batches that it is possible to ship. Constraint (3.5) imposes a null inventory at the beginning of the horizon.

In the minimum-cost flow network representation, each arc has a capacity of C, a unit cost of p^i_t and a fixed cost k^i_t to be paid if a positive flow is rooted through the arc. Notice that the number of arcs in the network between node $(i-1,t)$ and (i,t) must be as large as $\lceil D_{t,T}/C \rceil$, since all the remaining demand can be ordered at this period in a policy (see Section 1.2). Hence, the size of the network may not be polynomially bounded in the size of the inputs. This is not an issue, since we use the flow representation of the problem to demonstrate the properties of an optimal solution, and not in our resolution algorithm, based on dynamic programming. A source node $(0, 0)$, connected to the nodes of the first level, corresponds to the external supplier, with $D_{1,T}$ units available. Finally, each node (N, t) has a positive demand d_t. The objective is to route all the units available at the source node to the sinks at the minimal cost.

3.2.2 Structural properties

Consider a flow in the network defined in §3.2.1. For ordering arcs, a free arc corresponds to a *fractional* batch, that is, a batch that is neither empty nor full. By extension, we say that an ordering period is fractional if it orders at least one fractional batch. Notice that any holding arc with a positive flow is *free*, since holding arcs are uncapacitated. In the following, $(x \bmod C)$ represents the modulo of x by C, that is, the rest of x in the Euclidean division by C. We also denote by $\lfloor x \rfloor_C$ the quotient in the Euclidean division of x by C. Hence, $x = C\lfloor x \rfloor_C + (x \bmod C)$.

We give in this section some structural properties derived from our assumptions on the cost structure, see §3.1. The first property stipulates that there exists an optimal policy such that each node of the network has at most one free entering arc. It is named the *single free source* property. As a consequence, the flow representation of an optimal policy is an out-forest on the free arcs. Another way to state this property is that a node with an entering stock cannot order a fractional batch:

Property 7 (Single free source) *There exists an optimal policy such that fractional orderings only occur at periods with no entering stock, that is,* $(x_t^i \bmod C)s_{t-1}^i = 0 \ \forall i = 1, .., N, \forall t = 1, .., T$.

Proof. Consider an optimal flow, and assume that the property is not verified. Let (i, t) be the most bottom left node of the network such that $(x_t^i \bmod C)s_{t-1}^i > 0$, in the sense that the property is true at any node (j, u) with $u > t$ or with $u = t$ and $j > i$. Consider a unit entering in stock at period t at level i. Let $\tau < t$ be the period at which this unit has been ordered at level i. Due to non-speculative motives, postponing the ordering of this unit at level i from period τ to period t cannot increase its cost. The resulting policy is still feasible as one batch ordered at i at period t was fractional. This interchange can be repeated until only full batches are ordered at level i at period t, or until the amount of entering units in stock at level i at period t drops to zero. As the interchange does not affect the periods subsequent to t and the levels downstream i, we can apply the transformation at upstream levels and precedent periods. □

The next properties are the corner stones of our analysis. Their correctness is established into a joint proof given in Appendix B. The first property stipulates that there exists an optimal policy such that, at each level except the last one, the amount of units carried in stock from one period to another is always lower than C.

Property 8 *There exists an optimal policy obeying Property 7 such that all the stock levels are lower than C at the $N-1$ first levels of the network. That is, $s_t^i < C$ $\forall i = 1, \ldots, N-1, \forall t = 1, \ldots, T$.*

Notice that Properties 7 and 8 induce some restrictions on the number of batches ordered at two consecutive levels at the same period. Indeed, for a level $i < N$, in a period t where $x_t^i < x_t^{i+1}$, there is necessarily an entering stock at node (i, t) in order to satisfy entirely the order of its downstream level. Due to Property 7, this implies that x_t^i is a full ordering period. In addition, since s_{t-1}^i is lower than C, node (i, t) must order at least as many full batches as node $(i+1, t)$. Conversely, if $x_t^i \geq x_t^{i+1}$, we also have $\lfloor x_t^i \rfloor_C \geq \lfloor x_t^{i+1} \rfloor_C$, due to the monotony of $\lfloor . \rfloor_C$. As a consequence, in a policy obeying Properties 7 and 8, we can assert that if a level orders β full batches (and possibly one fractional batch) at a period t, then its upstream level orders at least β full batches at the same period. Turning our attention to fractional orders, we prove in appendix that if (i, t) is a fractional ordering period, then its downstream level $(i+1, t)$ also is. We have the following property:

Property 9 *There exists an optimal policy obeying Properties 7 and 8 such that, (i) for any period, all the full batches ordered at a level are also ordered at its upstream level, and (ii) if an ordering period is fractional at a level, then it is also a fractional ordering period at its downstream level. That is, for all periods $t \in \{1, ..T\}$ and all levels $i \in \{1, .., N-1\}$, we have:*

$$\lfloor x_t^i \rfloor_C \geq \lfloor x_t^{i+1} \rfloor_C \ \textit{and} \ (x_t^i \bmod C) \neq 0 \ \Rightarrow \ (x_t^{i+1} \bmod C) \neq 0$$

Considering the whole network, Property 9 induces a form of inheritance for full and fractional orderings over the levels. More precisely, for any given period, if level i orders only full batches, then all its upstream levels up to level 1 also order only full batches at this period. Moreover, they must order exactly the same number of full batches, otherwise their outgoing stock level would be larger to or equal to C, contradicting Property 8. Conversely, if level i orders one fractional batch, then all its downstream levels down to level N also order one fractional batch. We call this structure of policy *Full Up / Fractional Down*, as full batches ordered at a level go up to the first level, and fractional batch ordered at a level goes down to the last level (though the number of units in the batch may vary from one level to another). We highlight this structure in the following corollary:

Corollary 1 (Full-Up/Fractional-Down) *There exists an optimal policy obeying Properties 7 and 8 such that, for each level i and each period t:*

- *If x_t^i is a full ordering, then x_t^j is a full ordering for all upstream level $j \leq i$, and $x_t^j = x_t^i$*

- *If x_t^i is a fractional ordering, then x_t^j is a fractional ordering for all downstream level $j \geq i$*

In addition, if period t is an ordering period at level i, then period t is also an ordering period at all downstream levels $j \geq i$.

The last assertion is immediate if x_t^i is a fractional ordering. Otherwise, x_t^i is a full ordering. In particular, this implies that at least C units are ordered. Due to Property 8, node $(i+1, t)$ must order a positive quantity to assert that the stock level at the end of period t at level i is lower than C. The result follows by direct induction.

We say that a policy is *dominant* if it satisfies the Properties 7, 8 and 9. We have established in this section that there exists an optimal and dominant policy. We present in the next section a decomposition of dominant policies into induced connected components. Based on the structural properties established in this section, we exhibit a particular structure of the components, which will allow us to propose a polynomial time dynamic programming algorithm.

3.3 Induced Connected Components

In this section we consider an optimal policy π satisfying Properties 7, 8 and 9, see §3.2. In its flow representation, policy π induces a set of connected components. Connected components are a natural way to decompose a policy, since each connected component is independent of another one. However, for problem M-LSP-B, the connected component of a dominant policy seems too complicated to be studied directly. We need to decompose further the problem. One alternative would be to consider the connected components induced by the free arcs of a policy: Due to Property 7, each such component is an out-tree. We use in this article a new decomposition. Our originality is to consider the connected components of the flow induced on each of the networks restricted to the last $N-i+1$ levels, for $i = 1, \dots, N$.

More precisely, consider a fixed level $i < N$, and a connected component $\mathscr{C}$ induced on the network restricted to levels $\{i, i+1, \dots, N\}$. For short, we say that $\mathscr{C}$ is a connected component induced on level i. We restrict our attention to non-trivial connected components, that is, components that do not contain only a single node. Notice that, by definition, any two nodes of the connected component are linked by a path, and that there is no connection between a node of $\mathscr{C}$ and the other nodes of the network restricted to levels $\{i, i+1, \dots, N\}$ which are not in $\mathscr{C}$: connected component $\mathscr{C}$ can be linked with other nodes only by ordering arcs at level i.

Let l be the first period of the connected component $\mathscr{C}$ at level i. That is, node (i, l) belongs to $\mathscr{C}$ and, if l is not the first period of the time horizon, for any period $t < l$, node (i, t) belongs to another connected component induced on level i. We call period l a *regeneration point of the network* at level i, and we denote by $\mathscr{R} = \{r_1, \dots, r_n\}$ the set of regeneration points of the network at level i. For convenience, we add to $\mathscr{R}$ a fictive period $r_{n+1} = T + 1$ at the end of the time horizon. Notice that, by definition, node (i, l) has no entering stock. Since we assume that $\mathscr{C}$ is non-trivial, node (i, l) has necessarily an entering flow, and thus l is an ordering period at level i. Due to Corollary 1, this implies that period l is also an ordering period at all downstream levels. As a consequence, each nodes (j, l) belongs to $\mathscr{C}$ for $j = i, \ldots, N$. In addition, none of these nodes can have an entering stock, otherwise, since $(0, 0)$ is the only source in the network, a unit in stock at the beginning of period l at a level $j \geq i$ would have been necessarily ordered at a period $t < l$ at level i, contradicting the fact that l is the first node of $\mathscr{C}$ at level i. Hence, at a regeneration point of the network, we have a straight separation with the previous connected component induced on level i. This implies that two non-trivial connected components on level i cannot have a node located at the same period. More precisely, if r is the next regeneration point of the network at level i, with possibly $r = T + 1$ if $\mathscr{C}$ is the last non-trivial connected component induced on level i, we can assert that all the nodes of $\mathscr{C}$ lie between periods l and $r - 1$. Hence, in our network representation, nodes of $\mathscr{C}$ are included in a rectangle whose four angles corresponds to nodes (i, l), $(i, r - 1)$, (N, l) and $(N, r - 1)$. That is, the rectangle is delimited on top by level i, below by level N where the demand is to be served, and on its left and right hand sides by successive regeneration points of the network, namely l and r. We call such a rectangle a *box*, and we introduce the following definition:

Definition 2 (Box) *A box $B^i(l, r)$ corresponds to the network delimited vertically by periods l and r and horizontally by levels i and N, that is, containing the nodes $\{(j, t) \mid i \leq j \leq N \text{ and } l \leq t < r\}$, such that l and r are two successive regeneration points of the network at level i.*

Due to our definition, box $B^i(l, r)$ contains all the nodes of the non-trivial connected component $\mathscr{C}$ starting at node (i, l), plus eventually some isolated nodes, that is, nodes without entering nor outgoing flow. In Figure 3.1, the set of nodes and arcs inside the red rectangle represents box $B^i(l, r)$. Recall that there is no inventory flow entering the box at period l, and no outgoing inventory flow at period $r - 1$. Hence, nodes of $B^i(l, r)$ can be linked with another node outside the box only via an ordering arc at level i at a period occurring between l and $r - 1$, that is, somehow, via the top of the box. In Figure 3.1, $B^i(l, r)$ is connected to other nodes of the policy through the ordering arc in (i, l) for example.

The remainder of this section provides a set of properties resulting from Properties 7, 8 and 9, available within a box in an optimal policy. We also introduce the concept of *basis path* and *gate* to describe the structure of a box: As detailed in §3.4, these elements naturally decompose a box at level i into a set of boxes at downstream levels.

3.3.1 Basis path

In their paper, Hwang, Ahn, and Kaminsky, 2013 introduce the new notion of basis path of a connected component. Considering a network representation of a solution, they define a regeneration network (s_1, s_2, t_1, t_2) as a connected component of the subnetwork induced by the free arcs, involving periods between s_1 and s_2 at the first level and periods t_1 and t_2 at the last level. The basis path of (s_1, s_2, t_1, t_2) corresponds to the unique path linking the nodes $(1, s_1)$ and $(1, s_2)$. For a box $B^i(l,r)$, we introduce the following definition of its basis path: It is defined as the unique path of free arcs linking node (i,l) to the latest node of the box at level i with an entering stock. In addition, we show in the next property that the basis path of a box $B^i(l,r)$ always remains at level i.

Property 10 *Let w be the last period in $\{l+1,\dots,r-1\}$ such that $s^i_{w-1} > 0$. If no node of the box at level i has an entering stock, we set by convention $w = l$. In box $B^i(l,r)$, there exists a unique path of free arcs linking node (i,l) and (i,w), and all the nodes of this path are located at level i.*

Definition 3 (Basis path) *The basis path of box $B^i(l,r)$ is defined as the unique path $((i,l),(i,l+1)\dots,(i,w))$ of free arcs linking node (i,l) and (i,w).*

Proof. Let consider a box $B^i(l,r)$. If $w = l$, the property is immediate. Hence, assume that there exists a node at level i inside the box with an entering stock. To establish Property 10, we simply demonstrate that units are carried in stock at level i at all the periods between period l and period w. This proves the existence of a path of free arcs, namely $((i,l),(i,l+1)\dots,(i,w))$, linking node (i,l) and (i,w). Its uniqueness is due to the fact that the free arcs of an optimal solution constitute an out-forest. For the sake of contradiction, assume that there exists a period t, $l < t < w$, with no entering stock at level i. We can choose t to be the last period before w with no entering stock. We thus have $s^i_{t-1} = 0$ and $s^i_t > 0$. Due to Corollary 1, the amount of inventory at a level $i < N$ can be modified only at periods ordering a fractional batch at level $i+1$. Consequently, period t must be a fractional ordering period at level $i+1$. Again Corollary 1 implies that period t is a fractional ordering period also at all its downstream levels. Since a fractional ordering period cannot have an entering stock (Property 7), we have $s^j_{t-1} = 0$ at all levels $j = i,\dots,N$. Hence, nodes (i,l) and (i,w) cannot be connected by a path (of free or saturated arcs) in the network restricted to levels $\{i, i+1, .., N\}$. Since node (i,w) has an entering stock, this is not an isolated node, that is, it belongs to a non-trivial component of level i. This contradicts the fact that node (i,l) and (i,w) belongs to the same box $B^i(l,r)$ □

Figure 3.1 represents a Box $B^i(l,r)$. The basis path is colored in green, and corresponds to the part located at level i of the unique path linking periods l and $r-1$. It lies between periods l and w. Notice that in the case where the basis path is of length 0, that is, $w = l$, we are in the situation where no units are carried in stock at level i inside the box. Hence, level i plays the role of a transhipment level, where orders received from the upstream level are immediately shipped integrally to the downstream level. Basically, "nothing" happens in the box $B^i(l,r)$ at level i, and we can instead focus on box $B^{i+1}(l,r)$, or at a box at a higher level $j > i+1$, with a non-null basis path.

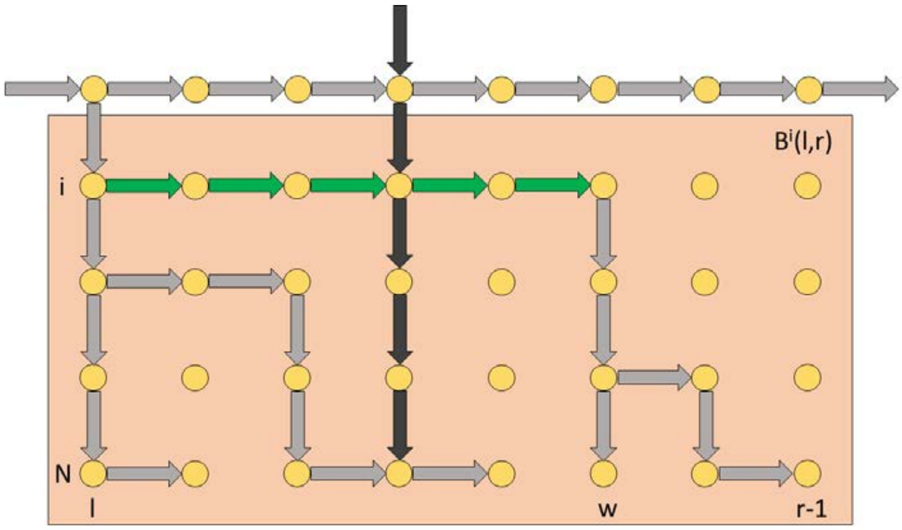

FIGURE 3.1: Box $B^i(l,r)$ on levels $i, i+1, ..., N$ between periods l an r. Black arcs correspond to full ordering while grey arcs are fractional ordering. Green arcs represent the basis path of the box

3.3.2 Gate

Consider a box $B^i(l,r)$ of an optimal dominant solution π. Let (i,τ) be a node situated on the basis path of the box, such that τ is an ordering period at level $i+1$. Notice that Corollary 1 implies that τ is also an ordering period at all downstream levels $\{i+2, i+2, .., N\}$. Hence we have a set of orders linking the basis path to the last level of the network. In addition, if there is no entering stock at period τ at each of the levels $i+1, i+2, \dots, N$, we obtain the splitting element to decompose vertically a box. We introduce the following definition:

Definition 4 (Gate) *The set of nodes $\{(i,\tau), (i+1,\tau), \dots, (N,\tau)\}$ is called a gate of box $B^i(l,r)$ at period τ if period τ belongs to the basis path of the box, and for all downstream levels $j = i+1, \dots, N$, we have $x^j_\tau > 0$ and $s^j_{\tau-1} = 0$.*

Observe that each fractional ordering of level i placed at a period τ such that (i,τ) belongs to the basis path, defines de facto a gate, since a fractional ordering propagates at downstream levels (Corollary 1), and implies a null entering stock at these levels (Property 7). Notice that period w, the last period of the basis path, always define a gate if the basis path is not null since there is necessarily a fractional ordering in $(i+1,w)$. The first period l of the basis path also always defines a gate. The important property of a gate is that, due to its definition, there is one and only one entering flow into the gate, and this flow enters at level i. For the gate at period l, this flow corresponds to the units ordered by node (i,l) ; for a gate at a period $\tau > l$, this flow corresponds to the entering stock of node (i,τ). Hence, nodes of $B^i(l,r)$ located at periods prior to τ and nodes located at subsequent periods are linked only by the flow on the basis path. In Figure 3.2, the two gates of

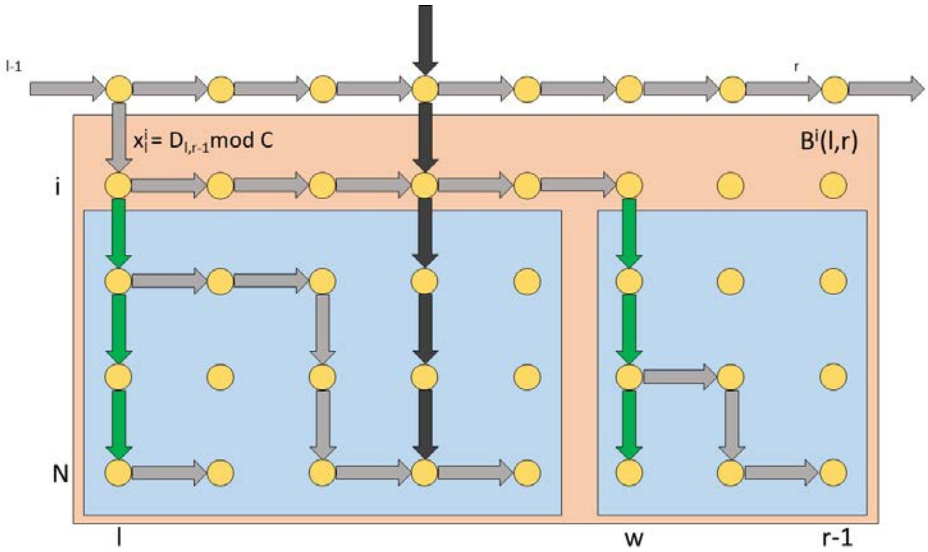

FIGURE 3.2: The gates of Box $B^i(l,r)$, situated at periods r and w, are represented in green and define two boxes on level $i+1$: $B^{i+1}(l,w)$ and $B^{i+1}(w,r)$

$B^i(l,r)$ are colored in green: one is situated at period l and the other at period w. Using these two gates, box $B^i(l,r)$ can be decomposed in two parts, as depicted in Figure 3.2, where each part is highlighted by a blue rectangle. These two rectangles are effectively linked by (and only by) the basis path of $B^i(l,r)$ on the network restricted to levels $\{i,\ldots,N\}$. Said differently, if there is a gate at period τ at level i, then there is a regeneration point of the network at level $i+1$. As a consequence, two consecutive gates at periods τ and τ' in Box $B^i(l,r)$ define a box $B^{i+1}(\tau,\tau')$. This shows how an optimal policy can be decomposed level by level, starting with boxes on the N levels, and decomposing them into boxes on levels $\{2,..,N\}$, and repeating this process in a recursive way. This will be fully detailed in §3.4.

The remainder of this section shows that we can determine the size of the single fractional batch ordered at the highest level of a box $B^i(l,r)$, that is, linking the box to its upstream level $i-1$. Moreover, we prove that the amount of inventory on the basis path can be easily deduced, knowing the gates of the box.

3.3.3 Fractional flow at the top of a box

In this section, we consider the flow of a box at its top level, that is, both the flow entering the box via an ordering arc at level i, and the flow routed on the basis path (at level i) via inventory arcs. Firstly, let us focus on the quantities ordered at level i inside the box. We can notice that all the ordering arcs at level i correspond to full ordering periods, except eventually for period l. Indeed, if a period τ is a fractional ordering period at level i, then τ is also a fractional ordering period at all downstream levels, due to Corollary 1. Since a fractional ordering period has no entering stock (Property 7), such a period τ would disconnect the flow of a dominant solution induced in the network restricted to levels $\{i, \ldots, N\}$. In other words, such a period τ would be a regeneration point of the network at level i, and thus, would correspond to the first period of a box at level i. We have the following property, that allows us to determine the quantity ordered in the fractional batch, if any, in the first period of a box:

Property 11 (Entering fractional flow of a box) *In a box $B^i(l, r)$, there is at most one fractional ordering period at level i. If there exists, this fractional ordering period is located at period l, and the quantity ordered in the fractional batch is equal to $D_{l,r-1} \bmod C$.*

Proof. The proof is quite immediate since any other ordering period at level i must order only full batches, that is, $x^i_t \pmod C \equiv 0$ for any period $t = l+1, \ldots, r-1$. Besides, the whole demand $D_{l,r-1}$ of a box $B^i(l, r)$ must be entirely satisfied by the units ordered at level i between periods l and $r-1$, as these are the only flows entering the Box. Writing the conservation of the flow modulo C, the result follows. □

Secondly, let us consider the flow routed on the basis path of the box. Let $\tau > l$ be a period defining a gate of the box. The inventory flow at level i between periods $\tau-1$ and τ is the only flow linking the nodes of $B^i(l, r)$ located at periods prior to τ, to the nodes of $B^i(l, r)$ located at periods subsequent to τ, including τ. The cumulative demand $D_{\tau,r-1}$ between periods τ and $r-1$ must be entirely satisfied by the entering stock in (i, τ) plus the units ordered at level i between periods τ and $r-1$, which is a multiple of C, as the only fractional ordering period at level i in box $B^i(l, r)$ occurs necessarily at period l (Property 11). Property 8 stipulates that the stock level is lower than C in a dominant policy. Hence, we obtain, similarly to Property 11, that the entering stock of the gate is $s^i_{\tau-1} = D_{\tau,r-1} \bmod C$. As the amount of inventory can be modified only at periods with a fractional ordering at the downstream level $i+1$, the stock on the basis path between two gates does not evolve. Evaluating the entering stock at each gate gives the value of the flow on the basis path, as stated in the following property:

Property 12 (Flow on the basis path) *Let τ and τ' be two periods defining two successive gates. For all periods $t = \tau, \ldots, \tau'-1$, the stock level s^i_t on the basis path is identical and equal to $D_{\tau',r-1} \bmod C$*

3.3.4 Number of batches ordered at the top of a box

In this section, we compare the difference in the number of batches ordered at two consecutive levels. Due to Corollary 1, at a given period t, all the full batches ordered at level $i+1$ are also ordered at level i. Our decomposition consists in evaluating boxes at level i, using boxes at level $i+1$. When connecting boxes with upstream level, we must keep track of the number of batches resulting from the structure of the boxes to ensure that limits m^i_t are not exceeded. We have the following property:

Property 13 *For all levels $i < N$ and for all periods t, we have $\lceil x_t^{i+1} \rceil_C - 1 \leq \lceil x_t^i \rceil_C \leq \lceil x_t^{i+1} \rceil_C + 1$. In addition, $\lceil x_t^i \rceil_C > \lceil x_t^{i+1} \rceil_C$ implies that t is a regeneration period at level i.*

Proof. We can write x_t^i and x_t^{i+1}, the orders at period t at levels i and $i+1$, as $x_t^i = \beta C + \alpha$ and $x_t^{i+1} = \beta' C + \alpha'$, with $\beta \in \mathbb{N}$, $\beta' \in \mathbb{N}$, $0 \leq \alpha < C$ and $0 \leq \alpha' < C$. Notice that $\lceil x_t^i \rceil_C = \beta$ if and only if $\alpha = 0$, and equals $\beta + 1$ otherwise. Corollary 1 states that, at period t, all the full batches ordered at level $i+1$ are also ordered at level i, that is, $\beta \geq \beta'$, which implies that $\lceil x_t^i \rceil_C \geq \beta \geq \lceil x_t^{i+1} \rceil_C - 1$. Which proves the left part of the inequality.

According to Property 8, the inventory at level i is always lower than or equal to $C-1$, since i is not the last level. Notice that $s_t^i \geq x_t^i - x_t^{i+1}$. It implies that $C - 1 \geq (\beta - \beta')C + (\alpha - \alpha')$.

- If $\alpha = 0$, that is, $\lceil x_t^i \rceil_C = \beta$, it gives $(\beta - \beta')C \leq C - 1 + \alpha'$. If $\alpha' = 0$, it implies that $\beta = \beta' = \lceil x_t^{i+1} \rceil_C$. If $\alpha' > 0$, it implies that $\beta \leq \beta' + 1 = \lceil x_t^{i+1} \rceil_C$. In any case, we have $\lceil x_t^i \rceil_C \leq \lceil x_t^{i+1} \rceil_C$.
- If $\alpha > 0$, that is, $\lceil x_t^i \rceil_C = \beta + 1$, it gives $(\beta - \beta')C \leq C - 1 + \alpha' \leq 2(C-1)$. If $\alpha' = 0$, it implies that $\beta = \beta' = \lceil x_t^{i+1} \rceil_C$. If $\alpha' > 0$, it implies that $\beta \leq \beta' + 1 = \lceil x_t^{i+1} \rceil_C$. This is the only case where $\lceil x_t^i \rceil_C$ can be greater than $\lceil x_t^{i+1} \rceil_C$. Notice that $\alpha > 0$ corresponds precisely to the case where t is a regeneration period at level i. In any case, we have $\lceil x_t^i \rceil_C \leq \lceil x_t^{i+1} \rceil_C + 1$.

As a result, in both cases $\lceil x_t^i \rceil_C \leq \lceil x_t^{i+1} \rceil_C + 1$, which proves the left part of the inequality. □

This property states that the difference in the number of batches ordered on two consecutive levels at any period is at most one. Moreover, the number of batches ordered at level i at period t can be greater than the number ordered at level $i+1$ only if t is a regeneration period at level i. As a result, the number of batches ordered at a given period t can only increase from one level to its upstream level, at levels i such that t is the first period of a box at level i. In the next section, we show how the decomposition of a box along its gates and its basis path into boxes of its downstream level allows to determine entirely an optimal flow.

3.4 Box decomposition of a policy

Consider an optimal solution following all the properties introduced in §3.2. This solution decomposes into a set of connected components. Consider one of these connected components: It is included into a unique box at level 1, say box $B^1(l, r)$. Figure 3.3 represents an optimal policy over a network of 4 levels and 8 periods. The policy has a single connected component, included in box $B^1(1, 9)$. On the figure, free arcs are colored in light grey and saturated arcs are colored in dark grey. Property 11 specifies that only the first period l of a box can be a fractional ordering period at its top level, and that $x_l^1 \bmod C = D_{l,r-1} \bmod C$. In Figure 3.3, the only fractional order at level 1 occurs at period 1, and the amount ordered in the fractional batch is equal to $D_{1,8} \bmod C$. This arc is represented in green to highlight the fact that we know its flow modulo C.

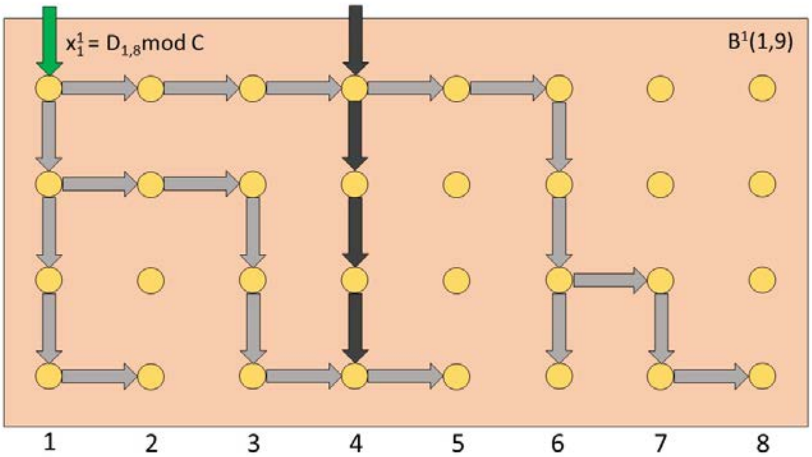

FIGURE 3.3: The flow of an optimal policy on a network of $N = 4$ levels and $T = 8$ periods. Free arcs are represented in light grey and saturated arcs are in dark grey.

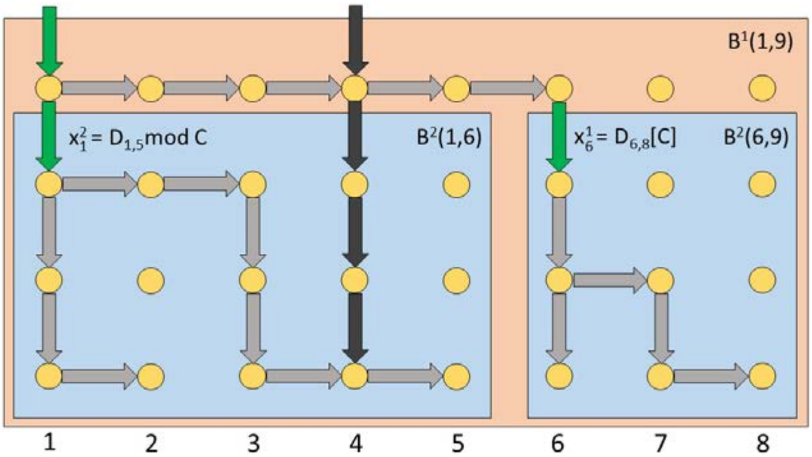

FIGURE 3.4: Decomposition of $B^1(1,9)$ into two boxes $B^2(1,6)$ and $B^2(6,9)$ at level 2. The limits of the boxes correspond to the gates of box $B^1(1,9)$

Consider now the basis path, as defined in Subsection 3.3.1. It has been shown in §3.3.2 that, for each fractional ordering period τ at a level $i+1$ such that node (i,τ) belongs to the basis path, the solution is split on levels $(i+1,...,N)$ by a gate into two independent parts, only linked by the basis path. Decomposing a box along its basis path and its gates makes boxes

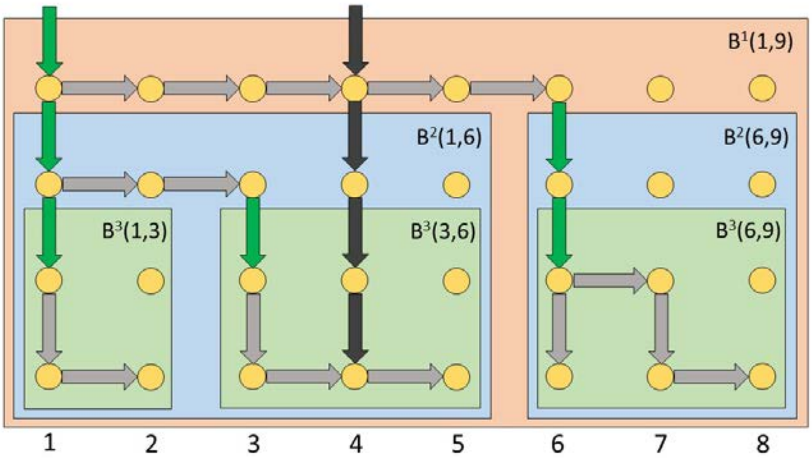

FIGURE 3.5: Decomposition of boxes $B^2(1,6)$ and $B^2(6,9)$ at level 2

appear on the downstream level. That is, a box can be decomposed into a set of boxes at its downstream level, linked by the arcs of its basis path and of its gates. This process is illustrated in Figure 3.4: The basis path of box $B^1(1,9)$ is path $((1,1),(1,2),..,(1,6))$. It admits two gates, situated respectively at period 1 and at period 6, for which a fractional ordering occurs at level 2. When considering the network restricted to levels $\{2,\dots,N\}$, Figure 3.4 shows how the policy decomposes in two boxes, $B^2(1,6)$ and $B^2(6,9)$. Notice that the amount ordered in the fractional batch entering $B^2(1,6)$ at node $(2,1)$ is $D_{1,5} \bmod C$, and the amount ordered in the fractional batch entering $B^2(6,9)$ at node $(2,6)$ is $D_{6,8} \bmod C$. Notice also that, knowing the gates of box $B^1(1,9)$, we know the flow rooted on the basis path. It is equal to $D_{6,8} \bmod C$ on the example.

Repeating this process dynamically, a dominant policy can be decomposed into a set of independent boxes at each level, each box being itself decomposed in the same way. Figure 3.5 shows the decomposition of box $B^2(1,6)$ into two boxes $B^3(1,3)$ and $B^3(3,6)$ at level 3. For box $B^2(6,9)$, its basis path is null, and thus its decomposition results in only one Box $B^3(6,9)$ at level 3. Finally, the decomposition at level 3 (see Figure 3.6) results into a single box at the last level for $B^3(1,3)$ and $B^3(3,6)$, respectively box $B^4(1,3)$ and box $B^4(3,6)$. Box $B^3(6,9)$ is decomposed into two boxes, namely box $B^4(6,7)$ and box $B^4(7,9)$. Notice that the value modulo C of the all the green arcs in Figure 3.6 are imposed by the box decomposition.

At the last level, a box $B^N(l,r)$ coincides (almost) with the classical notion of *subplan*, that is, a sequence of periods such that all the periods, except the first and the last ones, have a positive entering stock level. The only distinction is that a box also incorporates isolated nodes, which corresponds to periods with neither an entering nor an outgoing stock level. In the next section, we show how the decomposition into boxes of an optimal policy can be found using a dynamic programming approach.

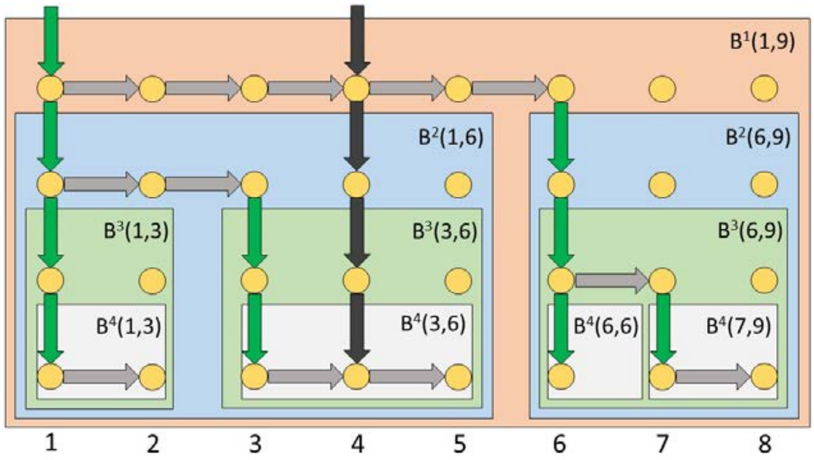

FIGURE 3.6: Decomposition of boxes $B^3(1,3)$, $B^3(3,6)$ and $B^3(6,9)$ at level 3

3.4.1 A Dynamic Programming Algorithm for the cases $m = +\infty$ and $m = 1$.

In this section we propose an exact dynamic algorithm to solve M-LSP-B for its two particular cases, namely the case with soft capacities, $m = +\infty$, and the case with hard capacities, $m = 1$. The idea of our algorithm is to compute dynamically the decomposition into boxes of an optimal policy, that is, the decomposition inducing the lowest cost. Consider a level $i < N$ and a box $B^i(l,r)$ of the decomposition of a dominant optimal policy π. We define cost $\mathcal{C}^i(l,r)$ as the cost incurred by all the flows rooted inside the box on ordering and inventory arcs, plus the cost of full orders at level i incurred at all the upstream levels. More precisely, if at a period $t \in \{l,\dots,r-1\}$, an order is constituted of β full batches at level i, we account in $\mathcal{C}^i(l,r)$ the cost of ordering these batches at levels $j = i, i-1, \dots, 1$ at period t. Notice that, due to Corollary 1, we can assert that in policy π these β full batches are indeed ordered in period t also at all levels $j < i$. Let $r_1 = l \le r_2 \le \dots \le r_n = w$ be the gates of the box, where w is the last period of the basis path. For convenience we define $r_{n+1} = r$. With this notation, box $B^i(l,r)$ is decomposed at level $i+1$ into a set of n boxes, $\mathscr{B} = \{B^{i+1}(r_k, r_{k+1}) \mid 1 \le k \le n\}$. Assume that the optimal cost $\mathcal{C}^{i+1}(u,v)$ of each box $B^{i+1}(u,v) \in \mathscr{B}$ is known. We detail how cost $\mathcal{C}^i(l,r)$ can be determined dynamically. For that, consider the flow inside box $B^i(l,r)$. It can be partitioned into:

- The flows induced on levels $\{i+1,\dots,N\}$. These flows are routed inside the boxes of level $i+1$, and hence, their costs are already accounted in
$$\sum_{(u,v):B^{i+1}(u,v)\in\mathscr{B}} \mathcal{C}^{i+1}(u,v) \tag{3.9}$$
- The flows rooted on the basis path. Consider a period t of the basis

path. Recall that due to Property 10, the basis path lies at level i between period l and period w. Period t is necessarily situated between two gates u and v, such that $u \leq t < v$. Said differently, period t is included in box $B^{i+1}(u, v)$ at level $i + 1$. Due to Property 12, we can assert that the stock level s^i_t at the end of period t is equal to $D_{v,r-1} \bmod C$. Hence, the total holding costs paid along the basis path of box $B^i(l, r)$ is equal to:

$$\sum_{(u,v):B^{i+1}(u,v)\in\mathscr{B}} H^i_{u,v-1}(D_{v,r-1} \bmod C) \tag{3.10}$$

where $H^i_{u,v-1} \equiv h^i_u + \cdots + h^i_{v-1}$ denotes the sum of the holding costs at level i between periods u and $v - 1$.

- The flows corresponding to orders at level i. Let us first focus on periods $t \notin \mathscr{R}$, that is, that are not a gate of the box. Consequently, period t is a full ordering period at level $i + 1$. Due to Corollary 1, period t is also a full ordering period at level i and Property 8 implies that the same amount is ordered at both levels. As a consequence, these costs are already accounted in cost $\mathcal{C}^{i+1}(u, v)$ of the box $B^{i+1}(u, v)$ where t belongs. That is, they already appear in Equation 3.9

- Finally, the flows corresponding to the orders at the gates. Let $u \in \mathscr{R}$ be a gate, and v be the next gate, with possibly $v = r$ if $u = w$. Recall that the quantity ordered in the fractional batch at period u at level $i + 1$ is imposed by box $B^{i+1}(u, v)$. Precisely, we have $x^{i+1}_u = C\lfloor x^{i+1}_u \rfloor_C + (D_{u,v-1} \bmod C)$, see Property 11. For short, let us denote by $\beta = \lfloor x^{i+1}_u \rfloor_C$ the number of full batches ordered at level $i + 1$. Notice that the cost of ordering these β batches at all the upstream levels is yet accounted in cost $\mathcal{C}^{i+1}(u, v)$. Due to Property 12, the entering and the outgoing stock level at period u on the basis path is also imposed, equal respectively to $s^i_{u-1} = (D_{u,r-1} \bmod C)$ and to $s^i_u = (D_{v,r-1} \bmod C)$. Writing the conservation of the flow at node (i, u), we obtain that:

$$x^i_u = x^{i+1}_u + s^i_u - s^i_{u-1} = \beta C + (D_{u,v-1} \bmod C) + (D_{v,r-1} \bmod C) \\ - (D_{u,r-1} \bmod C) \tag{3.11}$$

We can remark that the quantity x^i_u ordered is clearly a multiple of C, which complies with the fact that only full batches can be ordered at level i at a period $t > l$. Since the stock level is lower than C in a dominant policy, only the two values βC and $(\beta+1)C$ are admissible for x^i_u. this implies that either the same number of full batches are ordered at period u at levels i and $i + 1$, or an additional full batch is ordered at level i. The ordering cost of this additional full batch must be accounted at level i and at all the upstream levels. For short, let us denote by Q^i_t the cost of ordering a full batch at all the levels $j = 1, \ldots, i$ at a period t, which is equal to:

$$Q^i_t \equiv \sum_{j=1}^{i} k^j_t + \sum_{j=1}^{i} p^j_t C$$

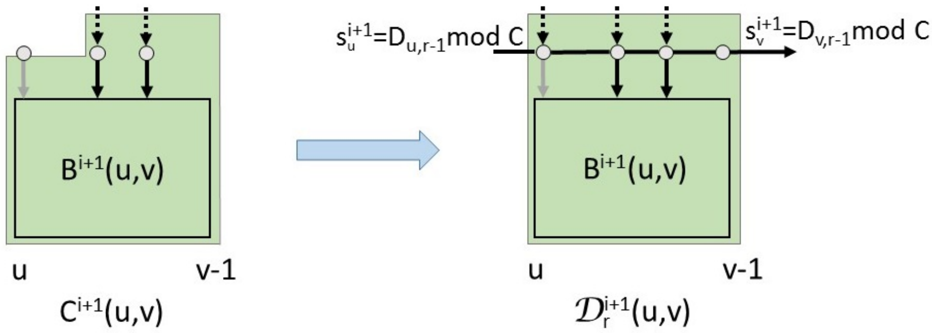

FIGURE 3.7: Schematic representation of a box $B^{i+1}(u,v)$: cost $\mathcal{C}^{i+1}(u,v-1)$ on the left, and its augmented cost if its belong to a box $B^i(l,r)$, with its basis path and an eventual full batch on level i at period u

Notice that all the quantities Q_t^i can be precomputed in time $O(NT)$. Let us also denote by $\delta_r^i(u,v)$ the quantity:

$$\delta_r^i(u,v) \equiv \lfloor (D_{u,v-1} \bmod C) + (D_{v,r-1} \bmod C) - (D_{u,r-1} \bmod C) \rfloor_C$$

which corresponds to the number (0 or 1) of additional batches ordered at level i relatively to level $(i+1)$. Notice that δ depends on the right hand side limit r of the box, but not on the left hand side limit l. With these notations, the cost incurred at period u in $\mathcal{C}^i(l,r)$ due to the possible ordering of an additional full batch, is equal to $Q_u^i \delta_r^i(u,v)$.

For period l, the first period of the box, we are in a similar situation, except that l can be a fractional ordering period at level i. We must also ensure that the quantity ordered in the fractional batch is equal to $(D_{l,r-1} \bmod C)$, and should be accounted only at level i in cost $\mathcal{C}^i(l,r)$. Hence, the additional cost incurred by the orders at the gates is equal to:

$$q_l^i(D_{l,r-1} \bmod C) + \sum_{(u,v):B^{i+1}(u,v)\in\mathscr{B}} Q_u^i \delta_r^i(u,v) \tag{3.12}$$

Notice that, in the case of hard capacities ($m_t^i = 1$, $\forall i \in \{1,...,N\}$, $\forall t \in \{1,...,T\}$), if $\delta_r^i(u,v) = 1$, a full batch must be ordered at period u, in addition to the fractional batch. As a consequence, the limit of the number of batches is exceeded, so the box is unfeasible. It means that there is no such box observing the dominant properties described in Section 3.2. In this case, we set $\mathcal{D}_r^{i+1}(u,v) = +\infty$, which prevents using this box in the solution. We can conclude that cost $\mathcal{C}^i(l,r)$ induced by box $B^i(l,r)$ is equal to the sum of the quantities defined in Equations 3.9, 3.10 and 3.12.

We write the expression of $\mathcal{C}^i(l,r)$ in this way to make apparent that this cost decomposes itself into the costs of the boxes at level $i+1$. More precisely, knowing the right side period r of the box at level i, for each box $B^{i+1}(u,v)$

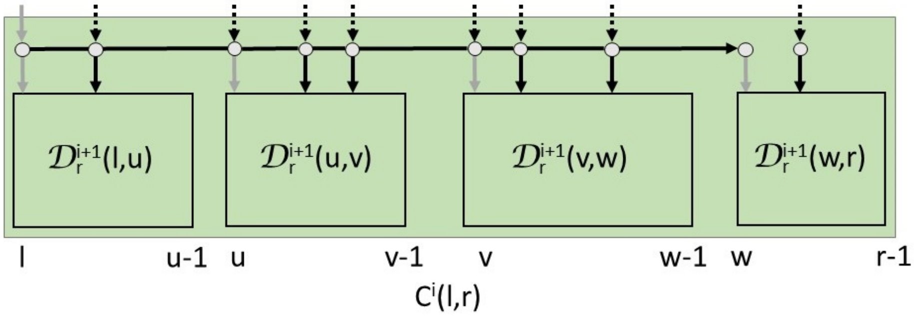

FIGURE 3.8: Representation of the optimal cost of a box $B^i(l,r)$, as a combination of the augmented costs of boxes on level $i+1$.

at level $i+1$, we can associate the following augmented cost:

$$\mathcal{D}_r^{i+1}(u,v) \equiv \mathcal{C}^{i+1}(u,v) + H_{u,v-1}^i(D_{v,r-1} \bmod C) + Q_u^i \delta_r^i(u,v) \tag{3.13}$$

The left part of Figure 3.7 illustrates a box $B^{i+1}(u,v)$. The green frame represents the cost of the box, plus the costs incurred at the upper levels. Knowing that this box is included in a box $B^i(l,r)$, the augmented costs $\mathcal{D}_r^{i+1}(u,v)$ can be deduced easily. With this notation, cost $\mathcal{C}^i(l,r)$ of box $B^i(l,r)$ is equal to the sum of the cost $\mathcal{D}_r^{i+1}(u,v)$ over all the boxes $B^{i+1}(u,v)$ of its decomposition, plus the cost $q_l^i(D_{l,r-1} \bmod C)$ of ordering eventually a fractional batch at period l. Figure 3.8 provides the representation of cost $\mathcal{C}^i(l,r)$ of box $B^i(l,r)$, composed by 4 boxes at level $i+1$: $B^{i+1}(l,u)$, $B^{i+1}(l,u)$, $B^{i+1}(v,w)$, $B^{i+1}(w,r)$. Of course, we do not know the decomposition of box $B^i(l,r)$. However, knowing the augmented cost of all possible boxes $B^{i+1}(u,v)$, we can compute $\mathcal{C}^i(l,r)$ as a shortest path problem in the directed graph whose vertices represent periods from 1 to r. An arc (u,v) indicates that box $B^{i+1}(u,v)$ belongs to the optimal partition. Its associated length is precisely $\mathcal{D}_r^{i+1}(u,v)$. The shortest path in the graph between period 1 and r provides the optimal cost $\mathcal{C}^i(l,r)$ of all the boxes at level i with r as its right hand side: It suffices to add to the length of the path between period l and r the cost $q_l^i(D_{l,r-1} \bmod C)$ of the fractional ordering at period l. Recall that the augmented cost of a box $B^{i+1}(u,v)$ includes the costs of full batches ordered at all its upstream levels. Through to Corollary1 ensures that they are ordered in this period. It could be argued that it may imply an unfair comparison with "short" boxes which may order no full batch. However, each possible path used to build a box at level i necessarily have the same number of full batches ordered at level i. We have the following lemma:

Lemma 1 *Knowing the costs $\mathcal{C}^{i+1}(u,v)$ of all the possible boxes at level $i+1$, the costs $\mathcal{C}^i(l,r)$ of all the possible boxes at level i can be computed in time $O(T^3)$, plus a precomputation step in $O(NT)$*

Proof. From the previous discussion, we only need to check that this computation can be performed in time $O(T^3)$. Consider a period r. Notice that the graph where the shortest path problem is solved for period r is a directed acyclic graph with r vertices. Hence, the shortest path between periods 1 and r can be found in time $O(T^2)$. The computational effort to build the graph is also in $O(T^2)$: Basically, for each arc (u, v), $1 \leq u < v \leq r$, we have to compute the augmented cost $\mathcal{D}_r^{i+1}(u, v)$ from the cost $\mathcal{C}^{i+1}(u, v)$ of box $B^{i+1}(u, v)$, see Equation 3.13. This can be achieved in constant time if we precompute the different quantities $H^i_{t,t'}$, $D_{t,t'}$ for all periods t, t', $1 \leq t \leq t' \leq T$. This precomputation clearly requires at most $O(T^2)$ operations. Finally, we can also precompute the quantities Q_t^j for all $t = 1, \dots, T$ and all $j = 1, \dots, N$. This precomputation can be done in time $O(NT)$. We account separately this precomputation step in the time complexity of the lemma, since it can be done once for the whole network, and hence is dominated in the final complexity of our algorithm. $\square$

Lemma 1 shows that the costs of the boxes can be computed from one level to the next upstream level in an efficient way. The cost of an optimal policy can be computed as a shortest path problem at level 1 given the costs $\mathcal{C}^1(l, r)$ of all the possible boxes. Equivalently, it can be defined as the optimal cost of a box $B^0(1, T)$. The basis on the induction corresponds to solve the problem at level N. Notice that this is a single-level lot-sizing problem, identical to the one studied in Van Vyve, 2007. The author proposes a polynomial time algorithm in time $O(T^3)$, allowing the backlog of demand. However, our cost structure slightly differs, due to the fact that in $\mathcal{C}^N(u, v)$, each full batch ordered at a period t is accounted for its ordering cost Q_t^N through all the levels, while a fractional batch is accounted only for its actual ordering cost at the last level. This difficulty can be easily overcome, since a fractional ordering only occurs in the first period of a box. Hence, assuming a FIFO discipline, we can discard the quantity $(D_{u,v} \bmod C)$ ordered in the fractional batch of a box $B^N(u, v)$ from the demand of period u (and of the following periods if necessary), to boil down with a lot-sizing problem where only full batches can be ordered. Van Vyve, 2007 proposes a very efficient algorithm in time $O(T \log T)$ for this problem. We have the following lemma:

Lemma 2 *The cost $\mathcal{C}^N(u, v)$ of all the possible boxes at level N can be computed in time $O(T^3 \log T)$. If the setup costs at the last level are also non-increasing, this complexity reduces to $O(T^3)$.*

Proof. For each possible box $B^N(u, v)$, with $1 \leq u < v \leq T + 1$, we can compute its cost as explained above using the algorithm of Van Vyve, 2007. Since there are $O(T^2)$ boxes to consider, the result follows. When setup costs at the last level are assumed non-speculative motives, the problem becomes quite straightforward. Due to non-speculative motives and non-increasing setup costs at each level, the cost Q_t^N is also non-increasing with t. Hence, each unit must be ordered as late as possible, with respect to its demand and to the restriction on the maximum number of batches at each period. We can greedily consider the periods starting from period $v - 1$ down to period u, ordering one full batch each time the unmet demand exceeds C. This greedy algorithm can be clearly implemented in linear time for each possible box. □

We give in Algorithm 1 a sketch of our algorithm, where the main steps have already been detailed in the two previous lemmas: Lemma 1 for the induction step, and Lemma 2 for the basis of the induction at level N. We can conclude this section with our main result, given in Theorem 4:

Algorithm 1 Algorithm Box Decomposition M-ULSP-B/M-CLSP

// Precomputation step
Compute quantities $D_{u,v}$ for all u, v, $1 \leq u \leq v \leq T$
Compute quantities Q_u^i for all u, i, $1 \leq u \leq T$ and $1 \leq i \leq N$
// Basis of the induction on the last level
Solve the single-level lot sizing problems at level N for all possible boxes $B^N(u, v)$, $1 \leq u < v \leq T+1$ to obtain costs $\mathcal{C}^N(u, v)$
// Induction on the levels
for $i = N - 1 \to 1$ **do**
  Compute quantities $H_{u,v}^i$ for all u, v: $1 \leq u \leq v \leq T$
  Determine costs $\mathcal{C}^i(l, r)$ for $1 \leq l < r \leq T+1$ as a shortest path problem on the augmented costs $\mathcal{D}_r^{i+1}(u, v)$ of its downstream level $(i+1)$
end for
Determine the cost of an optimal policy as a shortest path problem on the costs $\mathcal{C}^1(l, r)$ of the boxes at level 1.

Theorem 4 *Problems M-ULSP-B and M-CLSP can be solved in time complexity $O((N+\log T)T^3)$ under our assumptions, see §3.1. Assuming non-increasing setup costs at the last level, the time complexity is reduced to $O(NT^3)$.*

Proof. The algorithm computes the optimal cost $\mathcal{C}^N(u, v)$ of all possible boxes at level N in time complexity $O(T^3 \log T)$ or $O(T^3)$ when setup costs are non-increasing. Then the optimal cost of all the boxes at upper level can be computed dynamically (Lemma 1). This computation requires $O(T^3)$ operations at each level, plus a precomputation step in $O(NT)$ for quantities Q_t^j, which is dominated in the final complexity. The optimal cost of a policy can be computed in $O(T^2)$ as a shortest path over the boxes of level 1. □

This complexity is somehow surprisingly low. In particular, it depends only linearly on the number N of levels. This allows to model quite precisely a practical distribution network, taking into account all the physical storage points (inbound, intermediate, outbound, ...), without jeopardizing the resolution time of the problem. In the next section, we provide a detailed example of the execution of the dynamic algorithm on a small instance of M-CLSP.

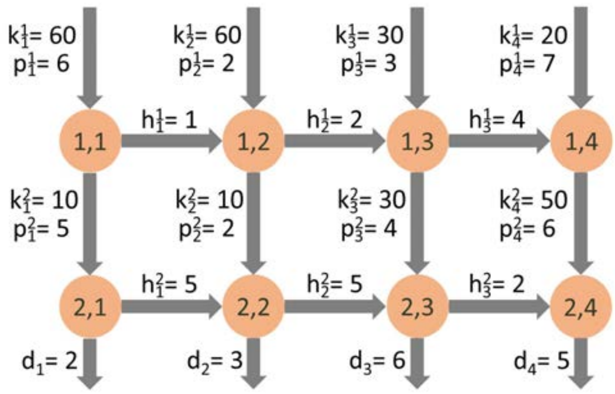

FIGURE 3.9: Example parameters

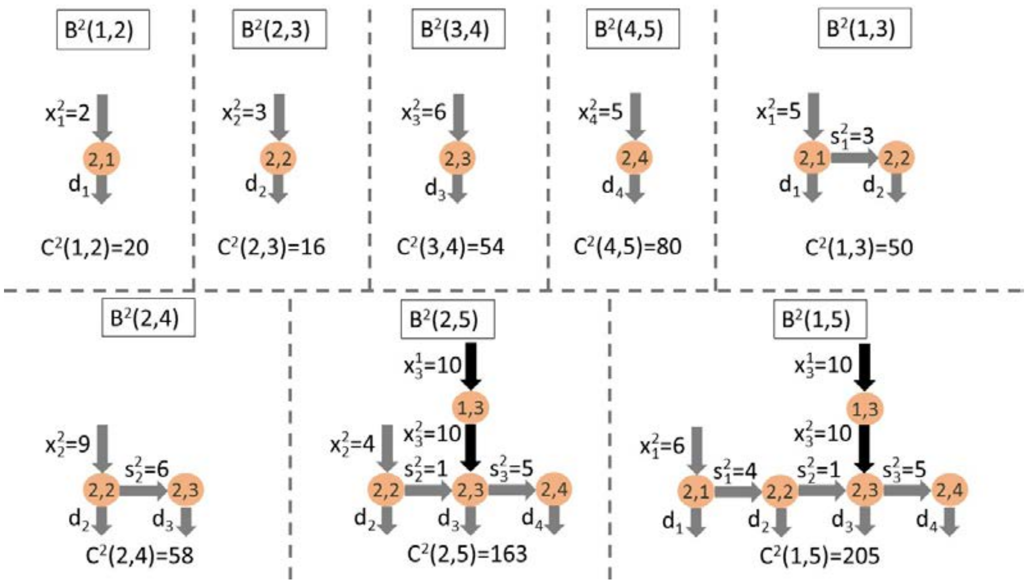

FIGURE 3.10: The different boxes $B^2(u, v)$ for the last level

Execution of the dynamic algorithm through an example

To illustrate the behavior of our algorithm, we consider a 2-level problem with 4 periods of time. The costs k_t^i, p_t^i and h_t^i, as well as the demand d_t, are given in Figure 3.9. We consider that all the m_t^i are equal to 1.

The first phase of our algorithm consists in solving the single-level lot sizing problems at level $N = 2$ for all possible boxes $B^2(u, v)$, $1 \leq u < v \leq 5$ to obtain costs $\mathcal{C}^2(u, v)$. The different boxes are represented in Figure 3.10. Consider the evaluation of $B^2(1, 5)$ as an example. The cumulative demand from period 1 to 4 is equal to 16, thus the value of the fractional batch ordered in period 1 must be equal to 6 (Property 11).

Some boxes $B^2(u, v)$ are not considered because the amount of the fractional batch ordered at period u is lower that the demand d_u. As a result, an additional full batch needs to be ordered in u, which is impossible since $m_u^1 = 1$ (for instance $B^2(3, 5)$). These boxes are ignored (we set an infinite cost to them) since they cannot fit with the dominant properties describe in Section 3.2.

The purpose of the next phase of the algorithm is to evaluate the optimal costs $\mathcal{C}^1(l, r)$ for each possible Box $B^1(l, r)$, for $1 \leq l < r \leq 5$ at level 1. It begins with the determination of the augmented costs $\mathcal{D}^2_r(u, v)$, for all $1 \leq u < v \leq r \leq 5$. Recall that $\mathcal{D}^2_r(u, v)$ is computed from $\mathcal{C}^2(u, v)$ using 3.13, adding the holding costs on the basis path, plus the extra full batch at the first period if requested. Figure 3.11 shows the boxes considered with their augmented costs when the value of r is equal to 5. It means that these values of $D^2_5(u, v)$ can be used to evaluate boxes $B^1(l, r)$ when $r = 5$. For example, let us evaluate $\mathcal{C}^1(1, 5)$, the minimum cost of box $B^1(1, 5)$. To do that, we construct a shortest path problem as described in Section 3.4.1 and illustrated in Figure 3.12. The shortest path is of length 205, and decomposed box $B^1(1, 5)$ into boxes $B^2(1, 2)$ and $B^2(2, 5)$ (see Figure 3.13). Adding the cost of the fractional batch ordered in the first period at level 1, it results that $C^1(1, 5) = 283$.

The same process is used to evaluate all boxes $B^1(l, r)$. Figure 3.14 provides illustration of every possible boxes obtained. Once again some boxes are not considered since they are not feasible (the limit $m^i_t = 1$ is exceeded at their first period).

Finally, a shortest path problem is defined using these boxes and their respective costs (see Figure 3.15). Solving this shortest path problem provides the optimal solution of the problem, which is composed by the unique box $B^1(1, 5)$, with a cost of 283.

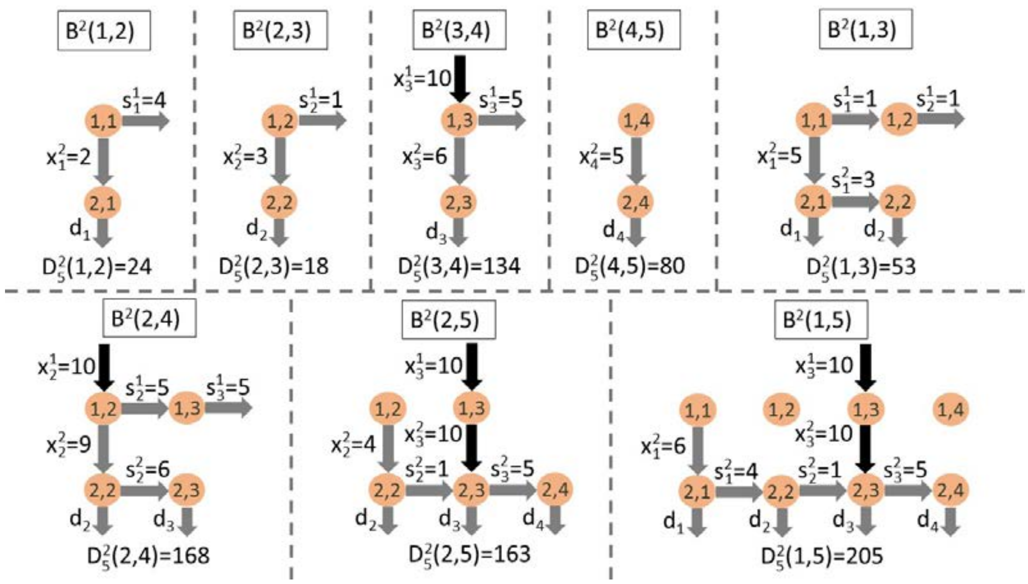

FIGURE 3.11: The augmented costs $D^2_5(u, v)$ of boxes $B^2(u, v)$, if included in box $B^1(1, 5)$

In the next section, we explain how our algorithm can be adapted to work with a general limitation on the number of batches.

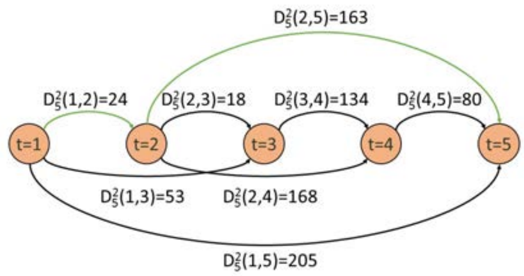

FIGURE 3.12: Shortest path problem for finding the optimal decomposition of $B^1(1,5)$

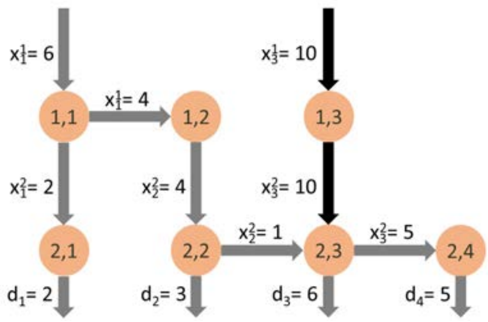

FIGURE 3.13: Flow of $B^2(1,5)$ with cost $\mathcal{C}^2(1,5)$

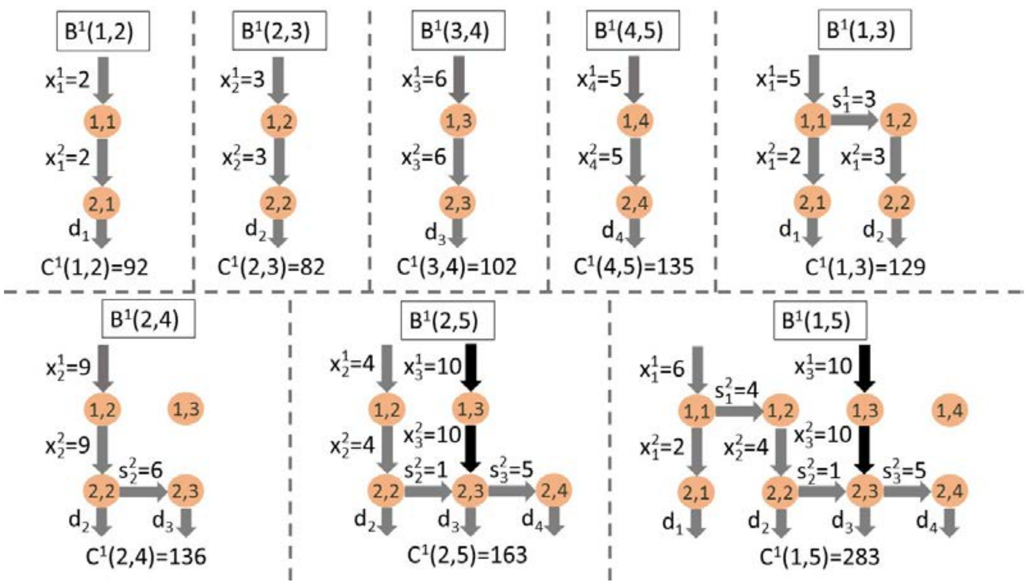

FIGURE 3.14: The different boxes $B^1(u,v)$ of the first level

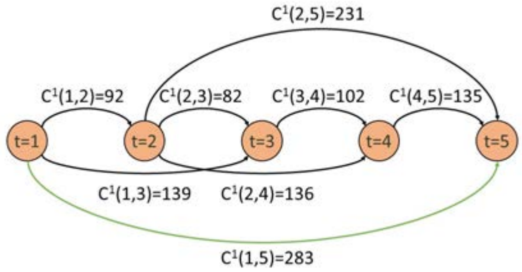

FIGURE 3.15: Shortest path problem for finding the optimal solution

3.4.2 A Dynamic Programming Algorithm for the case with a limitation on the number of batches.

First of all, let us explain briefly why the dynamic algorithm needs to be adapted to work correctly in the case where the number of batches which can be ordered is limited. Notice that the dominant structure with boxes described in Section 3.3, as well as the box decomposition detailed in Section 3.4 remain valid. However, algorithm of Section 3.4.1 must be modified to consider the limits m_t^i. This algorithm works recursively, starting with the evaluation of the optimal cost of the boxes at level N, then using these optimal boxes to evaluate boxes at level $N-1$, and so on until the first level. As explain in Section 3.3.4 (Property13), the number of batches ordered at a period t, from one level to the next one, is the same, except at the regeneration periods where an additional batch must possibly be ordered. As we assumed that m_t^i is non increasing ($m_t^{i+1} \leq m_t^i$), it results that we only need to check the first box of each decomposition. More precisely, let $B^i(l,r)$ be a box, and $B^{i+1}(l,v)$, with $l < v \leq r$, be the first box of its decomposition. Box $B^{i+1}(l,v)$ has a fractional batch entering at level $i+1$ at period l, whose amount is given by $(D_{l,v-1} \bmod C)$. At level i and period l, the outgoing stock, that is, the basis path of $B^i(l,r)$, has a value of $(D_{v,r-1} \bmod C)$. If the value of $(D_{l,v-1} \bmod C) + (D_{v,r-1} \bmod C)$ is larger than $(D_{l,r-1} \bmod C)$, that is, the value of the fractional batch of $B^i(l,r)$ at level i, then an additional full batch must be ordered at period l at level i, as well as at all the upstream levels. However, if the constraint on the number of batches ordered at level $i+1$ is saturated, and if $m_l^i = m_l^{i+1}$, it is then impossible to order this additional batch at level i. The resulting solution is not feasible. One might think of setting a infinite value to the augmented cost of $B^{i+1}(l,v)$ when it is used in box $B^i(l,r)$, since the resulting box would be unfeasible, as in the case $m_t^i = 1$ of the previous section. However, we show through an example that an optimal solution may indeed use a sub-optimal box $B'^{i+1}(l,v)$ to get an optimal decomposition of box $B^i(l,r)$.

We consider a problem on two levels and four periods for which all the parameters are stationary, except k_t^i, where only the first period has a null cost (see Figure 3.16). We have $m_t^i = 2\ \forall i \in \{1,..,N\}$, $\forall t \in \{1,..,T\}$, that

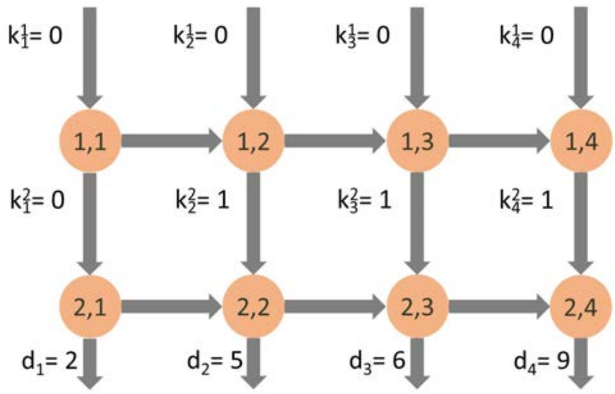

FIGURE 3.16: Parameters values of the instance ($N = 2, T = 4$)

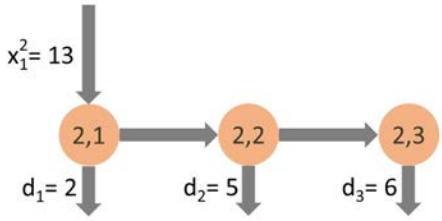

FIGURE 3.17: Optimal flow for box $B^2(1, 4)$

is, up to 2 batches can be ordered at any point of the network. The size of a batch is equal to 10. When evaluating the minimum cost of box $B^2(1, 4)$, all the orders are placed at the first period, which is feasible (see Figure 3.17). A quantity $x_1^2 = 13 = D_{1,4}$ is ordered, which represents two batches (one full, one fractional). If one wants to use this box inside box $B^1(1, 5)$, an additional batch must be ordered in $(1, 1)$, which exceeds the limit (see Figure 3.18) and leads to an unfeasible solution, where $x_1^1 = 22 > 20$. Nevertheless, there exists another solution which can be obtained by postponing the order of a full batch to period 2 (see Figure 3.19). The resulting solution could not have been found by the algorithm, since the resulting box at level $i+1$ has a higher cost than the box in Figure 3.17.

In this new version of the algorithm, adapted to the case with a limitation on the number of batches which can be ordered, the subproblems consisting in evaluating the boxes are overconstrained with a modified value of m_t^i. Instead of evaluating the minimum cost $\mathcal{C}^i(u, v)$ of a box $B^i(u, v)$, for given values of i, u, and v, we evaluate boxes $B^i(u, v)\langle\mu\rangle$, defined as the box $B^i(u, v)$ with a modified value of m_u^i equal to μ. Similarly, $\mathcal{C}^i(u, v)\langle\mu\rangle$ defines the minimum cost of the box $B_j^i(u, v)\langle\mu\rangle$. Regarding the value of μ that we need to consider, μ is clearly upper bounded by m_t^i. For its lower value, μ must verify that $\mu \geq \lceil d_u/C \rceil$, so that the limit on the number of batches allows the demand at period u to be satisfied. In the same way, μ must be greater than 0 since it must be possible to order at least a fractional batch at period u (u is the first period of the induced connected component, and thus its entering stock is null). Finally, at most one additional batch can be ordered from one level to the previous one at a given period (see Property 13). It is thus useless

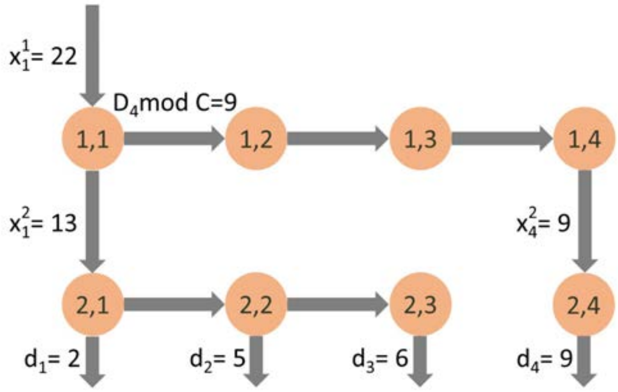

FIGURE 3.18: The decomposition of box $B^1(1,5)$ using box $B^2(1,4)$ is not feasible, since the limit on m_1^1 is exceeded.

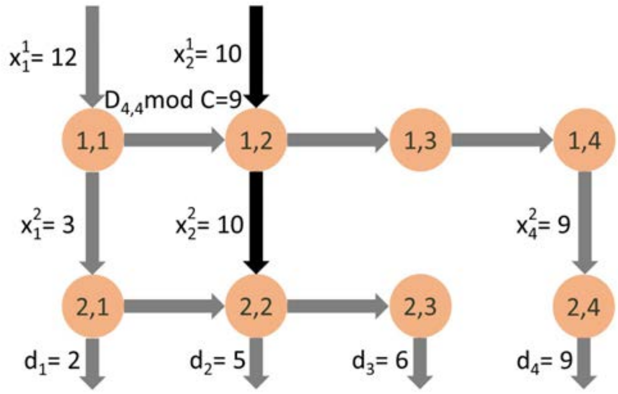

FIGURE 3.19: $B'^1(1,5)$ which is feasible and respects the dominance properties.

to evaluate a box $B^i(u,v)\langle\mu\rangle$ for a value of $\mu \leq m_t^i - (i-1)$. That is, if at most $m_t^i - (i-1)$ batches are ordered at level i, then at most $m_t^i - j + 1 \leq m_t^j$ batches are ordered at any upstream level j : the policy respects the constraints. As a consequence, at most $O(N)$ different values of μ need to be considered.

Consider a box $B^i(l,r)\langle\mu\rangle$. The value of $D_{l,r-1} \bmod C$ corresponds to the amount of the fractional batch ordered at period l at level i. Let $B^{i+1}(l,v)\langle\mu'\rangle$ be the first box at level $i+1$ inside $B^i(l,r)$. The amount of the fractional batch in $(l,i+1)$ is necessarily equal to $D_{l,v-1} \bmod C$. The amount of outgoing stock at level i at period t is equal to $D_{v,r-1} \bmod C$). If $(D_{l,v-1} \bmod C + D_{v,r-1} \bmod C) > (D_{l,r-1} \bmod C)$, we have $\lceil x_l^i \rceil_C = \lceil x_l^{i+1} \rceil_C + 1$ (see paragraph 3.3.4), that is, one more batch is ordered at level i. It implies that $\mu' = \mu - 1$, such that $B^{i+1}(l,v)\langle\mu'\rangle$ does not order more than $\mu - 1$ batches at period l. Otherwise, if $D_{l,v} \bmod C \geq D_{l,r} \bmod C$, we have $\lceil x_l^i \rceil_C = \lceil x_l^{i+1} \rceil_C$, and thus $\mu' = \mu$.

As a result, for given values of i, l and r, two types of boxes are used at level $i+1$ to evaluate the set of boxes $B^i(l,r)\langle\mu\rangle$ which may be used in the optimal solution of the problem:

- boxes $B^{i+1}(u,v)\langle m_u^{i+1}\rangle$ for values of u greater than l and for all v such that $l < v \leq r$. For these boxes, the number of full batches ordered at level i cannot be larger than the number of batches at level $i+1$, during the evaluation of the augmented costs. Consequently, the constraint on the number of batches cannot be violated (see Property 13).

- boxes $B^{i+1}(l,v)\langle\mu'\rangle$ for all v such that $l < v \leq r$, that is, the set of boxes at level $i+1$ whose first period is l and whose number of batches ordered at level i may exceed the number of batches ordered at level $i+1$. If $m_l^{i+1} < m_l^i$, even if an additional batch must be ordered in (i,l), the constraint on m_l^i cannot be exceeded. The box $B^{i+1}(l,v)\langle m_l^{i+1}\rangle$ is thus considered. If $m_l^{i+1} \geq m_l^i$ and if $\delta_r^i(l,v) = 0$, the constraint may be exceeded for values of m_l^{i+1} greater than m_l^i. The box $B^{i+1}(l,v)\langle m_l^i\rangle$ is thus used. Finally, if $m_l^{i+1} \geq m_l^i$ and if $\delta_r^i(l,v) = 1$, the constraint m_l^i may be exceeded for values of m_l^{i+1} greater than $m_l^i - 1$. The box $B^{i+1}(l,v)\langle m_l^i - 1\rangle$ is thus used.

For given values of i, r and j, assuming that the costs of every boxes at level $i+1$ are known, the shortest path between periods 1 and r can be evaluated in $O(T^2)$. There are $O(T)$ possible values for r and $O(N)$ possible values for μ. We have the following lemma:

Lemma 3 *Knowing the costs $\mathcal{C}^{i+1}(u,v)$ and $\mathcal{C}^{i+1}(l,v) < \mu' >$ of all the possible boxes at level $i+1$, the costs $\mathcal{C}^i(l,r)\langle\mu\rangle$ of all the possible boxes at level i can be computed in time $O(NT^3)$, plus a precomputation step in $O(NT)$*

For level N, for determining the costs of the different boxes, we start by evaluating $B^N(u,v)\langle m_u^i\rangle$ for given values u and v. If a feasible solution exists, $\mathcal{C}^N(u,v)\langle m_u^N - 1\rangle$ is then evaluated, and so on until μ reaches its minimum value, or until there is no feasible solution for the tested value of j. It is possible to evaluate $\mathcal{C}^N(u,v)\langle\mu\rangle$ in $O(T \log T)$ time using the algorithm of Van Vyve, 2007. At most N values of μ are then evaluated, so the overall process requires $O((N \log T)T)$ time. Since there are $O(T^2)$ possible values of u and v, we have the following lemma:

Lemma 4 *The cost $\mathcal{C}^N(u,v)\langle\mu\rangle$ of all the possible boxes at level N can be computed in time $O(NT^3 \log T)$.*

Finally, we have the following theorem:

Theorem 5 *Problem M-LSP-B can be solved in time complexity $O((N^2 + N \log T)T^3)$ under our assumptions, see §3.1.*

3.5 Setup costs at the first level

In this section, we generalize the cost structure at the first level, by considering a fixed setup cost. Specifically, at each period t where units are ordered at the first level, a fixed cost K_t is paid, in addition to the fixed costs per batch and the unit ordering cost. This cost structure may allow to model important practical situations. Indeed, while the different levels may represent an internal distribution network between facilities of a company or of an integrated supply chain inside a region, the external supplier can be situated overseas. For short to medium range distances, that is, inside the region, containers are most often shipped by trucks. The costs incurred by a shipment hence mainly boils down to a cost for each truck used, similar to our cost structure with fixed cost per batch. For long range distance, containers usually travel by boat, resulting in a different cost structure. The first level may also represent the manufacturing stage where the product is realized, and then shipped, via the downstream distribution chain, to its final customers. Again, manufacturing activities may result in a different costs structure than transportation activities. The setup cost may model here a classical preparation cost of the machines for the production in a period. In the rest of this section, we call the first level the *manufacturer* level, to distinguish it from the other levels, called *distribution* levels. For the distribution levels, we assume the same cost structure as previously, that is, a FTL cost structure with null setup cost. In addition, we focus on the unbounded case at the manufacturer level, that is, $m_t^1 = \infty$ for all periods t.

Notice that introducing non-null setup costs changes dramatically the structure of a dominant optimal policy at the first level. In particular, Properties 8, 9 and Corollary 1 do not hold anymore, since a policy is incited to group the ordering of batches in order to avoid to pay for too many setups. Hence, the stock level at a period can be larger than C, and the ordering of a full batch at level 2 does not lead anymore to order it at the same period at the manufacturer level. We can only rely on Property 7. This implies that each subplan at the first level contains at most one fractional batch, and this batch is located in the first period of the subplan. Hence, all the other orders inside a manufacturer subplan are composed only of full batches. On the opposite, the structure of a dominant policy remains unchanged for the distribution levels: The connected component induced on the network restricted to levels $\{2, \ldots, N\}$ still admits a decomposition into boxes, as described in §3.3

and §3.4. Our idea is, quite classically, to decompose a policy according to its subplans (u, v) at the manufacturer level. However, we cannot solve directly this problem as a single lot-sizing problem, since ordering decisions at the manufacturer level clearly incurred costs and constraints at the distribution levels. We can remark that if the decomposition into boxes at the distribution levels is known for an optimal policy, then the cost incurred by a subplan (u, v) at the manufacturer level can be easily evaluated. Again, we do not know what is the decomposition into boxes of an optimal policy. Instead, we show that for a given subplan at level 1, only a few number of boxes of level 2 need to be known in order to evaluate the cost of the subplan in the whole network. These boxes correspond basically to the ones containing the extremities of the subplan.

We start by giving a dominant property on the manufacturer level. This property is similar to Property 8, except that it applied only to the ordering periods:

Property 14 *There exists an optimal solution such that the entering stock level of each ordering period at the manufacturer level is lower than C.*

Proof. Let t be an ordering period, included in a manufacturer subplan (u, v). Consider the case where $s^1_{t-1} \geq C$. By a simple interchange argument, we show that we can obtain another optimal policy satisfying the property at period t. Let t' be the last ordering period occurring before t. Since no batch is ordered between periods t' and t, we must have $s^1_{t'} \geq s^1_{t-1} \geq C$. Notice that at least one full batch is ordered at period t': Otherwise t' orders only a fractional batch, and hence t' must be a regeneration point (Property 7). This contradicts the fact that at least C units are in stock at the end of period t'. It is thus possible to postpone the ordering of a full batch from period t' to period t at the manufacturer. The solution remains feasible and its cost does not increase due to our assumption of non-speculative motives and non-increasing fixed cost per batch. □

Consider a manufacturer subplan (u, v), $1 \leq u < v \leq T+1$, in a dominant optimal policy π. Let k and k' be two consecutive ordering periods inside the subplan, with $u \leq k < k' < v$. Notice that any period t belongs to a unique box $B^2(\tau, \tau')$ of level 2 in policy π, with $\tau \leq t < \tau'$. In particular, periods k, k' and $v-1$ belong to some boxes of level 2 in the decomposition of π. We denote by $g_{[a,b][a',b'][e,f]}(k, k')$ the minimum cost associated with all the ordering and inventory flows in the network between period k (included) and period k' (excluded), such that boxes $B^2(a, b)$, $B^2(a', b')$ and $B^2(e, f)$ belongs to the decomposition of π and $a \leq k < b$, $a' \leq k' < b'$ and $e \leq v-1 < f$. In other words, these are the three boxes on level 2 containing periods k, k' and $v-1$, respectively. Notice that in some cases two out of these three boxes can be identical, for example if the same box spans the time interval $[k, k']$ or $[k', v-1]$. We can also have a single box ($a = a' = e$ and $b = b' = f$) if it contains all the periods $k, \ldots, v-1$. We claim that for given values k, k', a, b, a', b', and f, we are able to evaluate $g_{[a,b][a',b'][e,f]}(k, k')$ in polynomial time.

We consider the case where k is not the first period of the subplan (u, v) and neither k nor k' coincide with the beginning of their box, e.g., we have

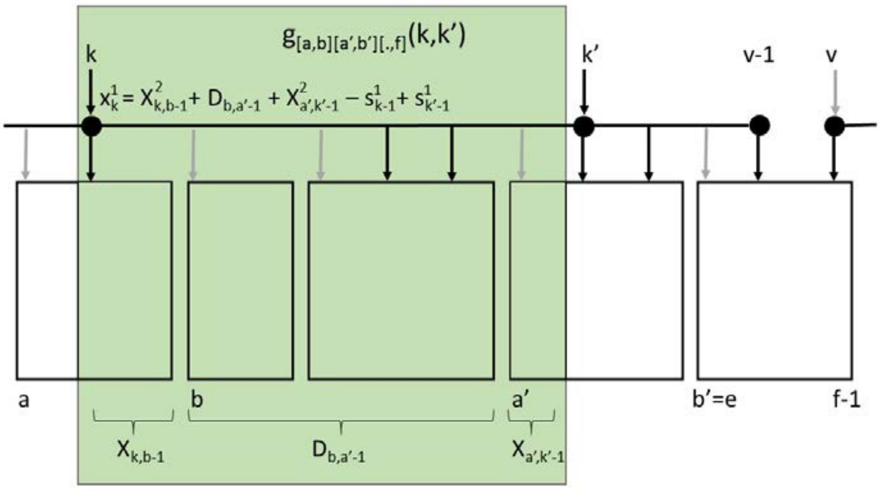

FIGURE 3.20: Representation of $g_{[a,b][a',b'][e,f]}(k, k')$ (green box). It includes all costs associated with periods between k and $k' - 1$.

$a < k < b$, $a' < k' < b'$. We also assume that v is not the first period of a box, see Figure 3.20. We establish here that the entering stock at periods k (and k') at the manufacturer level is fixed by the box decomposition at level 2 of policy π. Since k is not the first period of (u, v), all orders at the first level between period k and period $v - 1$ are constituted only of full batches. Let $X^i_{t,t'} = x^i_t + \cdots + x^i_{t'}$ denote the cumulative orders at level i between periods t and t'. We thus have $X^1_{k,v-1} \bmod C = 0$. Consider now the orders at level 2. Recall that a fractional batch can be ordered only in the first period of a box. It follows that we have $X^2_{v,f-1} \bmod C = 0$ since $e < v \leq f - 1$, and $X^2_{k,b-1} \bmod C = 0$ since $a < k$ due to our hypothesis. Observe that some fractional batches can be ordered at level 2 in time interval $\{b, \ldots, v - 1\}$, as the decomposition of π may be composed of (many) other boxes. Consider the network M induced by nodes $(1, k), \ldots, (1, v - 1)$ at the manufacturer level and all the nodes (j, t) for $j = 2, \ldots, N$ and $t = b, \ldots, f - 1$ at the distribution levels. Since period b is a regeneration point of the network, no stock is carried from period $b - 1$ to period b at the distribution levels. Hence, the entering flow of M is equal to $s^1_{k-1} + X^1_{k,v-1} + X^2_{v,f-1}$. Again, since there is no stock carried from period $v - 1$ to v at level 1 and from period $f - 1$ to f at the distribution levels, the outgoing flow of M is equal to $X^2_{k,b-1} + D_{b,f-1}$. Writing the conservation of the flow modulo C, and using Property 14, we obtain that:

$$s^1_{k-1} = D_{b,f-1} \bmod C \tag{3.14}$$

As a result, the amount of entering stock in node $(1, k)$ corresponds to the fractional part of the cumulative demand $D_{b,f-1}$ between periods b and $f - 1$. That is, the entering stock of an ordering period at the manufacturer is fixed by the regeneration point of the network at level 2 following this period and the next regeneration point of the network at level 2 following the end of the subplan. Similarly, the entering stock level of node $(1, k')$ is equal to $s^1_{k'-1} = D_{b',f-1} \bmod C$.

Since a, b, a' and b' are fixed, the flows rooted in boxes $B^2(a, b)$ and $B^2(a', b')$, and in particular values of $X_{k,b-1}$ and $X_{a',k'-1}$, are known. Moreover, demand of the time interval $[b, k'-1]$ must be supplied by units ordered at period k at the manufacturer, or already in stock at the manufacturer at the end of period $k-1$. It is thus possible to deduced the amount of units ordered in k. We have:

$$x^1_k + s^1_{k-1} = X^2_{k,b-1} + D_{b,a'-1} + X^2_{a',k'-1} + s^1_{k'-1} \tag{3.15}$$

Again, the quantity ordered at period k at the manufacturer is imposed by the boxes at level 2 containing period k and the next ordering period k'. Particular cases, e.g., the situation where $k = u$ or $k = a$, are very similar and thus are omitted here.

To evaluate $g_{[a,b][a',b'][e,f]}(k, k')$, it remains to determine how to satisfy the demand on time interval $[b, a'-1]$, using boxes on levels 2. Notice that the decomposition of policy π on level 2 may involve a large number of boxes between b and $a'-1$. Assuming that the minimum cost of each box has already been evaluated, using the same method presented above (with a slight adjustment in order to not propagate full ordering costs at the first level) their holding costs incurred on the first level can be easily deduced, since all the units are carried from period k, the last ordering period before b. Hence, the holding cost incurred at level 1 by a unit ordered at a period $t \in [b, a'-1]$ at level 2 is equal to $H^1_{k,t}$. It is then possible to use again a shortest path algorithm to determine the optimal set of boxes supplying demand of $[b, a'-1]$. We can notice that this decomposition does not depend on periods k and k', but solely on the right hand side b and left hand side a' of the boxes containing these periods, since whatever the effective period k, the units are carried in stock from period $b-1$ to the period when they are ordered at level 2. Consider that boxes $B^2(a, b)$, $B^2(a', b')$ and period f are fixed. To evaluate all the costs $g_{[a,b][a',b'][e,f]}(k, k')$ for $a \le k < b$ and $a' \le k' < b'$, we start by finding the optimal decomposition in boxes at level 2, taking into account the holding cost incurred by carrying units from period $b-1$. This can be achieved in time complexity $O(T^2)$, assuming that all the boxes have been precomputed using Algorithm 1. Then cost $g_{[a,b][a',b'][e,f]}(b-1, a')$ can be evaluated in time $O(N)$, accounting the cost of the flows in the network at period $b-1$. The cost for all $k \in [a, b-1]$ can be determined iteratively, as the cost for a period k can be deduced from the cost for period $k+1$ in $O(N)$. We can proceed in a symmetric way to evaluate the cost at all period k'. We have the following Lemma:

Lemma 5 *All the costs $g_{[a,b][a',b'][.,f]}(k, k')$ can be computed in time $O(NT^7)$*

The case where periods k, k', and $v-1$ are contained in only two boxes, or a single one, is somehow easier and is not detailed here. Once these costs have been determined, we compute classically the cost of all the possible subplans (u, v) at the manufacturer level. Consider a given period $v-1$ and a given period f. We construct the acyclic directed graph where each node is triple (k, a, b) associated with an ordering period k and a box $B^2(a, b)$ containing k. The length of the arc from (k, a, b) to another node (k', a', b') is equal to $g_{[a,b][a',b'][.,f]}(k, k')$. Such an arc exists if $b \leq a'$ or $a = a'$ and $b = b'$. The shortest path in this graph can be computed in time $O(T^6)$ and provide the optimal cost of all the subplan (u, v), for $1 \leq u < v$ for all the boxes $B^2(a, b)$ and $B^2(e, f)$ that may contain u and v, respectively. Since we have $O(T^2)$ periods v and f to consider, the cost of all the subplans can be obtained in time $O(T^8)$. Finally, we can determine the optimal cost of a policy by solving a shortest path problem on the directed graph where each node (u, a, b) represents a regeneration point u and its associated box $B^2(a, b)$ at level 2. An arc represents the cost of a subplan. Since we have $O(T^6)$ arcs, this problem can be solved in time $O(T^6)$. We have the following result:

Theorem 6 *Problem M-LSP-B with setup costs at the first level can be solved in time complexity* $O(NT^7 + T^8)$

3.6 Conclusion

In this chapter, we establish that under some assumptions on the parameters, the multi-level lot-sizing problem with batch deliveries M-LSP-B is polynomially solvable even if the number of levels is part of the input. Our approach is based on the decomposition of an optimal policy into a set of independent induced connected components. This box decomposition relies on the main property that each full batch ordered at a level is ordered at the same period at all the upstream levels, while a fractional ordering propagates at all the downstream levels. The overall time complexity of our algorithm is in $O((N^2 + N \log T)T^3)$. This low time complexity clearly makes the algorithm of practical use even for large supply chains.

In the next chapter we study more general multi-level lot-sizing problems with capacities. We provide NP-hard results for M-ULSP-B with level-dependent batch sizes and M-CLSP with level-dependent capacities. Approximations algorithms are given for M-ULSP-B where batch sizes are both time-dependent and level-dependent and for M-CLSP with level-dependent capacities.

Chapter 4

Approximation algorithms and complexity results for multilevel lot-sizing problems with capacities

In the previous chapter we proposed an exact algorithm running in $O((N^2 + \log T)T^3)$ time for the multi-level lot-sizing problem with batch deliveries. We considered a model with identical and stationary batch sizes, and with a limitation on the number of batches which can be ordered at a period. Echelon non-speculative motives were assumed at each level, as well as non-increasing setup costs, except at the last level. In this chapter, we consider more general M-ULSP problems where the capacities/size of the batches may differ from one level to another. We prove that the M-ULSP-B with level-dependent batch sizes and the M-CLSP with level-dependent capacities are both NP-hard. We propose then 2-approximation algorithms, respectively for the M-ULSP-B with level-dependent and time-dependent batch sizes and for the M-CLSP with level-dependent capacities with echelon non-speculative motives and non-increasing setup costs.

4.1 Introduction

As in the previous chapter, we consider both the multi-level lot-sizing problem with either soft and hard capacities, but with non-identical capacities/batch sizes and with setup costs. Again, we consider problem M-LSP-B with batch deliveries, which encapsulates both cases by introducing an upper limit m_t^i on the number of batches of each order. The multi-level in-series lot-sizing problem with batch deliveries can be formulated as follows:

K_t^i fixed setup cost for ordering a positive amount at period t at level i;
B_t^i size of a batch at period t at level i;
y_t^i order indicator (binary variable) at period t at level i;

$$\min \sum_{t=1}^{T} \sum_{i=1}^{N} (K_t^i y_t^i + \lceil \frac{x_t^i}{B_t^i} \rceil k_t^i + p_t^i x_t^i + s_t^i h_t^i) \tag{P}$$

subject to

$$x_t^i + s_{t-1}^i = x_t^{i+1} + s_t^i, \ \forall t \in \{1,..,T\}, i \in \{1,..,N-1\}, \tag{4.1}$$

$$x_t^N + s_{t-1}^N = d_t + s_t^N, \ \forall t \in \{1, .., T\}, \tag{4.2}$$

$$x_t^i \leq m_t^i B_t^i y_t^i, \ \forall t \in \{1, .., T\}, i \in \{1, .., N\}, \tag{4.3}$$

$$s_0^i = 0, \ \forall i \in \{1, .., N\}, \tag{4.4}$$

$$x_t^i \geq 0, \ \forall t \in \{1, .., T\}, \forall i \in \{1, .., N\} \tag{4.5}$$

$$s_t^i \geq 0, \ \forall t \in \{1, .., T\}, \forall i \in \{1, .., N\} \tag{4.6}$$

$$y_t^i \in \{0, 1\}, \ \forall t \in \{1, .., T\}, \forall i \in \{1, .., N\} \tag{4.7}$$

We focus on the 2 important special cases: $m_t^i = +\infty$, that is, the uncapacitated problem M-ULSP-B with batch deliveries, and $m_t^i = 1$, that is, the capacitated version M-CLSP (see Section3.1). For time-dependent batch sizes or production capacities, both problems are known to be NP-hard even on a single level (see Akbalik and Rapine, 2013, Florian, Lenstra, and Rinnooy Kan, 1980a and Bitran and Yanasse, 1982a). We first prove that the multi-level lot-sizing problem with batch deliveries is NP-hard in the case of level-dependent batch sizes. That is, each level i uses a stationary batch size B^i but batch sizes may differ from one level to another. For instance, it may be relevant to deal with distribution networks using different transportation modes at each level. In the chapter, we provide a 2-approximation algorithm in the case of batch sizes which are both time-dependent and level-dependent. We then turn our attention to the problem with hard capacities, that is, $m_t^i = 1$ $\forall t \in \{1, .., T\}$, $i \in \{1, .., N\}$, when the capacities are level-dependent. We prove that this problem is also NP-hard. Finally, we propose an approximation algorithm for the latter problem with echelon non-speculative motives and non-increasing setup costs. Our approximation algorithms use a simple approximation technique which could be reused for other problems.

4.2 Complexity of the multi-level uncapacitated lot-sizing problem with level-dependent batch sizes

We consider in this section M-ULSP-B where batch sizes are stationary but level-dependent (B^i), and without limitation in the number of batches, that is, $m_t^i = \infty$ $\forall t \in \{1, .., T\}$, $i \in \{1, .., N\}$. We show that this problem can be reduced to a single-level lot-sizing problem with time-dependent batch sizes. The latter has been shown to be NP-hard by Akbalik and Rapine, 2013. We have the following theorem:

Theorem 7 *M-ULSP-B with batch sizes which are stationary but level-dependent is NP-hard when N is part of the input, even with null setup costs, null unit ordering costs, and assuming echelon non-speculative cost structure.*

Proof. Consider an instance I of a single-level problem with time-dependent batch sizes on a time horizon of T periods. For each period t, let B_t be the size of a batch and k_t denotes the fixed cost per batch ordered. The other

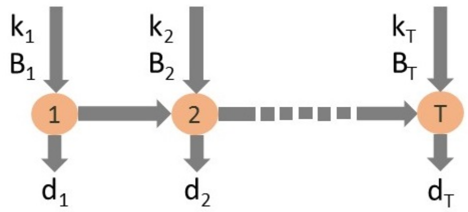

FIGURE 4.1: Network representation of instance I.

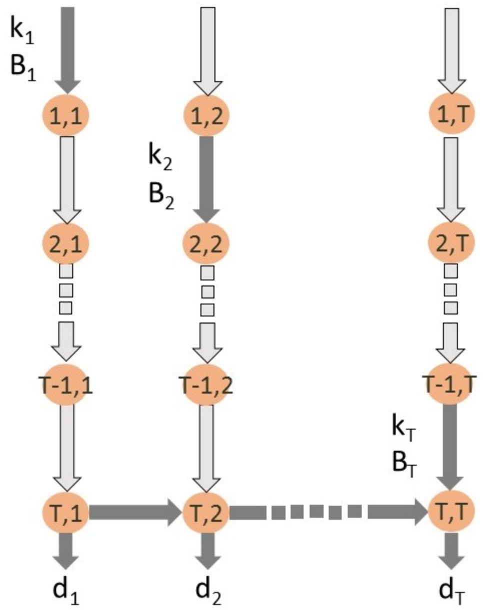

FIGURE 4.2: Network representation of instance I'. Arcs with an infinite cost are not represented. Light colored arcs have a null ordering cost.

parameters, that is, the setup costs and the unit ordering and holding costs are null ($K_t = 0$, $p_t = 0$ and $h_t = 0$) (see Figure 4.1). Akbalik and Rapine, 2013 showed that this problem is NP-hard. We associate to I an instance I' of the multi-level lot-sizing problem as follows: I' also covers a time horizon of T periods and has a number $N = T$ of levels. Setup costs, unit ordering costs are null. Moreover, all unit holding costs are set to ∞ except at the last level where they are null. It results that I' observes the echelon non-speculative cost structure. At level $i \in \{1, ..., N\}$, for all $t \in \{1, ..., T\} \backslash \{i\}$, the fixed cost of a batch is null. For $t = i$, the fixed costs per batch correspond to those of instance I, that is, $k_t^i = k_t$. The demand at the last level is the same as in instance I. Figure 4.2 provides a network representation of I'. Comparing Figure 4.1 and Figure 4.2, it is easy to see that both instances are equivalent: the costs of ordering and routing x units from the source node to the last level are the same in both problems, and the only way to store units is at the last level, at the same cost. As a result, any feasible policy for I can be readily converted into a feasible policy for I', of the same cost, and conversely. Since I is a NP-hard problem, it implies that the multi-level lot-sizing problem with level-dependent batch sizes is also NP-hard when N is part of the input, even with null setup costs, null unit ordering costs, and assuming echelon non-speculative cost structure. □

4.3 A 2-approximation algorithm for the multi-level lot-sizing problem with batch deliveries

In this section we propose a 2-approximation algorithm for M-ULSP-B where batch sizes are time-dependent and level-dependent (B_t^i). Recall that an λ-approximation algorithm for a minimization problem is a polynomial algorithm which, for any instance of the problem, returns a solution whose value is not greater than λ times the optimal solution value. Our approximation algorithm is remarkably simple but, considering the complexity of the problem, a 2-approximation algorithm is a very interesting result. It consists in solving a relaxation of the problem where the FTL ordering costs are replaced by affine costs functions. To get a performance guarantee, we want the affine cost function to "sandwich" the FTL ordering cost, has explained in the following.

Let consider a M-ULSP-B problem (P). Recall that the procurement costs are given by $q_t^i(x) = K_t^i + \lceil x/B_t^i \rceil k_t^i$, $\forall t \in \{1,..,T\}$, $i \in \{1,..,N\}$. Let us consider a M-ULSP problem (P') without batch deliveries, whose parameters are similar to (P) except the procurement costs. We have the same number of levels and periods, and holding costs $h_t'^i$ are equal to h_t^i $\forall t \in \{1,..,T\}$, $i \in \{1,..,N\}$. For all possible values of i and t, we want to find an affine function $r_t^i(x)$ such that $r_t^i(x) \leq q_t^i(x) \leq \beta r_t^i(x)$ holds whatever the quantity x ordered (see in Figure 4.3). Since $r_t^i(x)$ is an affine procurement cost, an optimal solution π' of (P') can be found in $O(NT^4)$ using the algorithm of Zangwill, 2013. Since $r_t^i(x) \leq q_t^i(x)$, (P') is a relaxation of (P), and hence its optimal policy π' provides a lower bound. Since the only difference between (P) and (P') is the procurement cost, that is, both problems have the same set of feasible solutions, π' is feasible for (P). Moreover, as $r_t^i(x) \leq q_t^i(x) \leq \beta r_t^i(x)$, the cost incurred by π' in (P) is at most β times greater than the optimal solution of (P). As a result, we obtain a β-approximation of M-ULSP-B where batch sizes are time-dependent and level-dependent running in time complexity $O(NT^4)$.

We now have to find an affine function $r(x)$ such that the value of β is as small as possible. Let us define the following affine function $r_t^i(x)$, $\forall t \in \{1,..,T\}$, $i \in \{1,..,N\}$, where $0 \leq \alpha \leq 1$ is a given parameter:

$$r_t^i(x) = \begin{cases} 0 & \text{if } x = 0 \\ K_t^i + \alpha k_t^i + \frac{(1-\alpha)k_t^i}{B_t^i} x & \text{if } x > 0 \end{cases}$$

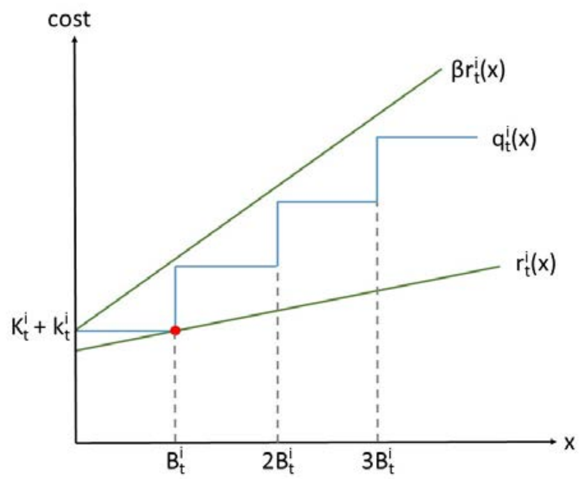

FIGURE 4.3: Representation of $r_t^i(x)$, $q_t^i(x)$ and $\beta r_t^i(x)$.

We claim that for any parameter $\alpha \in [0,1]$, we have $r_t^i(x) \leq q_t^i(x)\ \forall\ x \geq 0$. For any positive value of x, we have, on one hand

$$q_t^i(x) \geq K_t^i + x\frac{k_t^i}{B_t^i}, \text{ since } \lceil . \rceil \text{ is a non-increasing function.}$$

and, on the other hand

$$q_t^i(x) \geq K_t^i + k_t^i, \text{ since } x > 0$$

By multiplying the first inequality by $(1-\alpha)$ and the second inequality by α, we obtain that:

$$(1-\alpha)q_t^i(x) + \alpha q_t^i(x) \geq (1-\alpha)(K_t^i + x\frac{k_t^i}{B_t^i}) + \alpha(K_t^i + k_t^i)$$

which gives:

$$q_t^i(x) \geq K_t^i + \alpha k_t^i + \frac{(1-\alpha)k_t^i}{B_t^i}x = r_t^i(x)$$

Hence, for any value of $\alpha \in [0,1]$, function $r_t^i(x)$ is upper bounded by FTL cost $q_t^i(x)$. We now have to determine the value of β verifying $q_t^i(x) \leq \beta r_t^i(x)$, and to choose the value of parameter α such that β is minimized. Since $q_t^i(x) \leq \beta r_t^i(x)$ must hold for any value of x, in particular we must have:

- $K_t^i + k_t^i \leq \beta(K_t^i + \alpha k_t^i)$, that is, the inequality holds for $x \to 0$
- $k_t^i/B^i \leq \beta(1-\alpha)k_t^i/B^i$, that is, the inequality holds for $x \to +\infty$

Notice that $q_t^i(x)$ is under the affine function $q_t'^i(x) = (K_t^i + k_t^i) + k_t^i/B^i x$. The two previous inequalities in fact ensure that $q_t^i(x) \leq \beta r_t^i(x), \forall x$, since $\beta r_t^i(x)$ is affine. As a result, the minimum value of β can be found by solving the following non-linear problem:

$$\begin{array}{ll} min & \beta \\ s.t. & \beta(K_t^i + \alpha k_t^i) \geq K_t^i + k_t^i \\ & \beta(1-\alpha) \geq 1 \\ & 0 < \alpha \leq 1 \end{array} \tag{4.8}$$

Figure 4.3 might help in representing the constraints. A feasible solution is given by:

$$\alpha = \frac{k_t^i}{K_t^i + 2k_t^i}, \beta = \frac{K_t^i + 2k_t^i}{K_t^i + k_t^i}$$

It corresponds to the solution obtained when both constraints of problem 4.8 are saturated. The value of β is clearly always lower than 2, so we obtain a 2-approximation algorithm. Notice that better a posteriori performance guarantee can be obtained depending on the parameters of the instance, for instance, if the setup cost is always greater than the fixed cost per batch ($K_t^i \geq k_t^i$), which seems a plausible assumption, the algorithm provides a solution at most $3/2$ times the optimal.

Theorem 8 *The multi-level lot-sizing uncapacitated lot-sizing problem with batch deliveries can be approximated within a ratio 2 in $O(NT^4)$ time complexity.*

4.4 Multilevel capacitated lot-sizing problem with level-dependent capacities

We consider now the multi-level capacitated lot-sizing problem (M-CLSP) with level-dependent (hard) capacities C^i, that is, $m_t^i = 1$ $\forall t \in \{1, .., T\}$, $i \in \{1, .., N\}$. Notice that CLSP with time dependent capacities is NP-hard even on a single level. Ahmed et al., 2016 proposed a polynomial time algorithm for M-CLSP when the number of different capacities is constant. They prove that the problem is NP-hard when source and sinks are in more than two rows. We consider the case with N being part of the input. As a result the number of different capacities is not constant and their result does not stand for our problem. In fact, we prove that M-CLSP with level-dependent capacities is NP-hard:

Theorem 9 *The M-CLSP with level-dependent capacities C^i is NP-hard when N is part of the input, even if the C^i are non-increasing with the levels.*

Proof. The reduction is made from the partition problem. Recall that an instance of Partition is constituted of a list of n integers a_i. We assume, w.l.o.g., that $a_1 \geq a_2 \geq ... \geq a_N$. Denoting by A the quantity $\sum_i ai/2$, it is asked whether there exists a subset $S \subseteq \{1, ..., n\}$ such that $\sum_{i \in S} ai = A$. For short, for a given subset S, we denote by $a(S)$ the sum of the a_i's for $i \in S$. We also denote by A_k the sum of the k first a_i's: $A_k = a_1 + a_2 + ... + a_k = a(\{1, ..., k\})$. We transform an instance I of Partition into an instance $\tau(I)$ of M-CLSP as follows:

- To each element a_i there are i time periods associated. The planning horizon is thus constituted of $T = \sum_{i=1}^{n} i = n(n+1)/2$ periods. The demand at the last period is equal to A. Other demands are defined below.
- There are $N = n$ levels. The capacity of level i is $C^i = A_{N-i+1}$. Notice that capacities are level-dependent but stationary. We also have $C^1 = 2A \geq C^2 \geq ... \geq C^N = a_1$.
- The setup costs are set to $+\infty$ except at the last level, where they are null, and at some periods specified below.
- The holding costs are set to $+\infty$ except at some periods, specified below.
- It is asked if a solution of cost at most A exists.

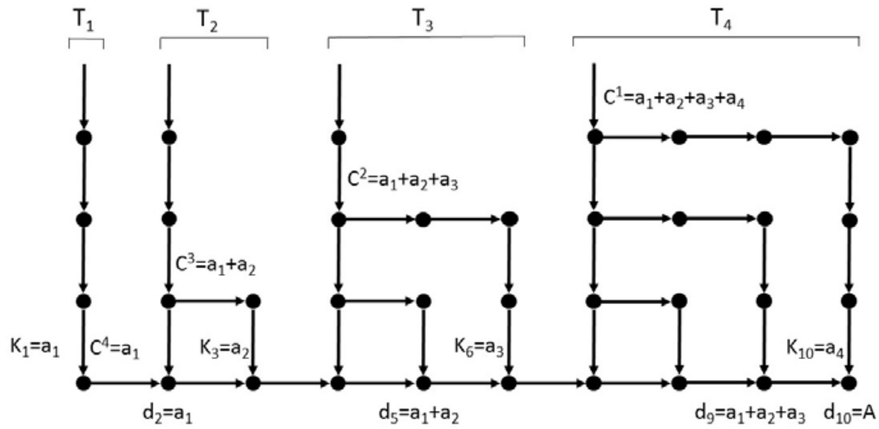

FIGURE 4.4: Example of transformation of a partition instance with 4 elements, into a M-CLSP with level-dependent capacities. Only arcs with a finite cost are represented.

A network representation of an instance associated to $n = 4$ elements $\{a_1, a_2, a_3, a_4\}$ is given in Figure 4.4. We define the remaining costs of the instance such that, if we remove the set of arcs with an infinite cost, the holding arcs at the last level, the remaining arcs form a forest, with a tree T_i associated to the i periods of each a_i, see Figure 4.4. All the setup costs and holding costs of the arcs of T_i are null, except at level N, for the last period of the tree, whose setup cost is equal to a_i. To sum-up our construction, for the i periods associated with a_i:

- T_i spans on the i periods,
- The penultimate period has a demand equal to A_{i-1},
- Setup cost of the last period at level N is equal to a_i. All other setup costs of the arcs of T_i are null.

Notice that the structure of tree T_i is identical to T_{i-1} with an additional branch. By induction, it is easy to see that the bottleneck of T_i is imposed by the ordering arc of level $N-i+1$ through which all units rooted on the tree must transit. Its capacity is equal to A_i. At the next level ($N-i+2$), the units must split in two parts: the first part is brought for free until the last level using the $i-1$ first periods of the tree, identical to T_{i-1}. Notice that a limit of A_{i-1} units is imposed by the capacity at level $N-i+2$, which corresponds to the demand associated with the tree. Therefore, we can assume that the demand of the tree is fully satisfied using these arcs. The additional units must use the ordering arcs of the tree at the last period spanned by T_i. This path has a cost a_i due to the setup cost at the last level at this period. The capacity of the path is given by the smallest level capacity $C^N = a_1$. However, the difference between C_{N-i+1} and C_{N-i+2} being equal to a_i, at most a_i units can be routed on this path in a dominant solution. Since the capacities at each level are greater than or equal to a_1, and $a_i \leq a_1$, it is indeed possible to route a_i units through this path. Hence, the tree T_i can route A_{i-1} units to the last level at null cost, and possibly a_i additional units, involving a cost a_i, if it used its last branch. Since the demand of each tree T_i is equal to A_{i-1}, a feasible policy also satisfies the demand at the last period, with a subset of trees using their last period to route units to satisfy the demand A. Let $S \subseteq \{1, ..., n\}$ be such a subset of indices, such that $i \in S$ if and only if units are rooted on the last branch of tree T_i. We must have $a(S) \geq A$, so that the demand at period T is satisfied. However, to obtain a solution which costs less than A, we must also have $a(S) \leq A$. As a result, we must have $a(S) = A$. Hence, the instance of the partition problem is positive if and only if the instance of the M-CLSP is positive.

□

4.5 Approximation algorithm for the multi-level lot-sizing problem with level-dependent capacities

We consider a multi-level lot-sizing problem with level-dependent capacities C^i. We assume that the capacities are non-increasing with the level, that is, $C^1 \geq C^2 \geq ... \geq C^N$. We also assume that setup costs K_t^i are non-increasing in time for a given level, that is, $K_t^i \geq K_{t+1}^i$ for every levels and every periods. Finally, holding costs h_t^i and unit ordering costs p_t^i are echelon non-speculative at each level, that is, $p_t^i + (h_t^i - h_t^{i-1}) \geq p_{t+1}^i$. Notice that the complexity of this problem is open since our complexity result for level-dependent capacities (Section 4.2) is not valid with echelon non-speculative motives and non-increasing setup costs. Without loss of generalities, we can assume that the demand at a period t is lower than C^N. Otherwise, as explained in Section 2.3, we can transfer the excess of demand to the preceding period $t-1$ without changing the set of feasible solutions, and decreasing the cost of any policy by a fixed positive term. This transformation cannot affect the efficiency of the approximation algorithm, since adding the same positive amount to two solution values can only reduce their ratio.

Our idea is to relax the problem (P) by removing the capacity constraints and using FTL procurement costs, with batch of sizes C^i. We then relax again the resulting problem by replacing the FTL procurement costs by affine functions. The optimal solution of this problem is then modified, without increasing its cost, such that it becomes feasible for problem (P).

Consider a M-ULSP-B problem (P') with FTL procurement costs $b_t^i(x) = \lceil x/C^i \rceil K_t^i + p_t^i x$ for $x > 0$, and null if $x = 0$. That is, we consider batch deliveries, with batch of size C^i and a fixed cost per batch K_t^i. Notice that for any ordered quantity $x \leq C^i$, we have $b_t^i(x) = q_t^i(x)$. Hence, (P') is clearly a relaxation of (P). In particular, if OPT′ is the optimal solution of (P'), we have OPT′≤OPT.

To build a feasible solution π' for (P'), as in Section 4.3, we replace the FTL procurement cost $b_t^i(x)$ by an affine procurement cost $r_t^i(x) = K_t^i/2 + (p_t^i + K_t^i/2C)x$ for $x > 0$, and null if $x = 0$. The optimal solution π'' of this second relaxation can be found in time $O(NT^4)$ with the algorithm of Zangwill, 2013 when using $r_t^i(x)$ instead of $b_t^i(x)$. Since we have $r_t^i(x) \leq b_t^i(x) \leq 2r_t^i(x)$ for any ordered quantity x, the cost $\mathscr{C}'(\pi'')$ of π'' in (P') is at most 2OPT′ which is lower than or equal to 2OPT. We show that π'' can be transformed into a feasible solution π for (P), such that the cost $\mathscr{C}(\pi)$ of π in (P) is not greater than $\mathscr{C}'(\pi'')$. This transformation, as explained below, can be done in time complexity $O(NT)$. It results that $\mathscr{C}(\pi) \leq$ 2OPT, and thus we obtain a 2-approximate solution for (P) in time $O(NT^4)$.

We now detail how π'' is modified to obtain a feasible solution for (P) without increasing the cost of the solution. Let x'' be the quantity ordered in policy π''. Notice that $x_t''^i$ can be larger than C^i. We define x the quantity ordered in policy π by induction, simply postponing the units exceeding the actual capacity C^i. Start with $x = x''$. Let t be the first period such that $x_t^N > C^i$. Postponing the order of $x_t^N - C^i$ units does not make the solution unfeasible since the demand at a given period is lower than or equal to C^N. Moreover, the cost of the solution can only decrease since we assume non-increasing setup costs and echelon non-speculative motives. Finally, the number of batches ordered does not increase and can even decrease if two fractional batches are gathered at period $t + 1$. By reiterating this process at all subsequent periods, the resulting solution π is feasible at the last level for (P).

The same process can be performed at level $N - 1$. Considering the first period t such that $x_t^{N-1} > C^{i-1}$, the order of $x_t^{N-1} - C^{i-1}$ units are postponed, without increasing the cost of the solution. Since the demand of level N is lower than or equal to C^N, through the shifts of the last step, the resulting solution is feasible for (P). Repeating the process at each period, and then at all previous levels, the resulting solution π is feasible for (P) and is cheaper than π'' for (P'). The transformation can be performed in time complexity $O(NT)$. We thus have the following theorem:

Theorem 10 *The multi-level lot-sizing problem with level-dependent capacities,* *non-increasing setup costs, and echelon non-speculative motives can be approximated within a ratio* 2 *in* $O(NT^4)$ *time complexity.*

4.6 Conclusion

In this chapter, we proved that M-ULSP-B with level-dependent batch sizes and M-CLSP with level-dependent capacities are both NP-hard. We propose an approximation algorithm for the multi-level lot-sizing problem with batch deliveries, which consists in replacing the procurement costs by an affine function. Based on this result, a 2-approximation algorithm is also proposed for the M-CLSP with level-dependent capacities under some assumptions on the parameters. Since the complexity of the latter problem is open, it could be a matter for future research. Our approximation method, using sandwich affine (or concave) functions, could also be used for other problems whose complexity arises from the structure of the objective function.

Chapter 5

Energy-aware lot sizing problem: Complexity analysis and exact algorithms

This chapter deals with a single-item lot sizing problem under a periodic energy limitation. In contrast to the other chapters, the system studied here is not in series. We consider identical and parallel capacitated machines which can be turned on and off according to the production requirements. This system shows similarities to the multi-level in series lot-sizing problem with batch deliveries since the production capacities can be extended by switching machines on, which results in additional costs. However, machines turned on may remain on at the next periods, that is, a start-up cost is paid only once to increase the capacity on the whole remaining horizon. Besides the classical lot sizing decisions of how much and in which periods to produce, we have to decide the number of machines to switch on and to switch off in each period. In addition, the models studied here take into account the energy consumed in the production process. We provide complexity results and propose polynomial algorithms for the case with stationary energy parameters. This chapter is based on an article wrote with Ayse Akbalik, from University of Lorraine.

5.1 Introduction

Energy-efficiency in production planning becomes more and more appealing for researchers and practitioners. According to Biel and Glock, 2016, "*In 2010, the industrial sector was responsible for 39.4% of the overall energy consumption and this latter largely originates from manufacturing industries.*" The same authors mention that the aim of energy-efficient production planning models is not only to take into account the classical metrics such as the minimization of overall cost or completion time, but also to consider energy-aware factors such as energy related constraints, energy cost or energy consumption minimization, etc. There is also a change in the consumers' behavior, with a higher sensitivity to the environmental impacts of the industrial activities (pollution, energy consumption, etc.). For couple of years, numerous companies have thus begun to rethink and optimize their production processes in order to produce at lower cost, but also more ecologically and with a lower energy consumption.

We consider in this chapter a lot-sizing problem taking into account the energy consumption as a hard constraint, in a capacitated machines environment. In addition to an extensible production capacity, dependent on the number of machines running in a given period (which is a decision variable in our problem), we consider a limit of the amount of energy that can be consumed in each period by the production system. The different activities responsible for consuming energy that we consider in this chapter include the start-up of the machines, the production of goods, and keeping the machines running, either they do produce or are idle. The aim is to decide when and how much to produce, when and how many machines to turn on or to turn off, in order to minimize the total cost, respecting the amount of energy available in each period. The problem studied is thus in accordance with the context of energy aware production and environmental sustainability. We call this energy aware lot sizing problem *energy*-LSP in the rest of the chapter.

Problem *energy*-LSP in a parallel production system was first introduced by Rapine et al., 2016a; Rapine et al., 2016b. They propose a very efficient $O(T \log T)$ algorithm for a restricted version assuming that start-up costs are stationary and that only one activity (start-up or production) consumes energy. In this article we extend the model in different directions to render it more realistic: First, we consider time-varying cost parameters, including start-up costs but also non-null joint production setup costs, to be paid in each period where production occurs, and running costs, to be paid for each machine that is not turned off. Second, we consider that all activities may have a non-negligible energy consumption. It means that, in each period one has to arbitrate how the available amount of energy is to be shared between the start-up of machines, which increases the production capacity of the system, and the effective production of units. In addition, we also consider a running energy consumption, that represents the energy consumed by a machine that is turned on, whenever producing or not. Under this quite general framework, we establish that the problem is NP-hard if some energy parameters are time-varying, even on a single resource with non-null setup or running costs. In contrast, we show that the problem is polynomially solvable if all the energy parameters, that is, periodic amount of available energy, start-up and unit consumptions, are stationary. Our approach is based on dynamic programming and provides an $O(M^6T^6)$ exact algorithm for the most general case studied in this chapter. To the best of our knowledge, there is no other theoretical studies on this integrated lot sizing and energy issues, except Rapine et al., 2016a and Rapine et al., 2016b.

Organization of the chapter.

The chapter is organized as follows. In Section 5.2, we present relevant studies in both lot sizing and energy-efficient production planning problems. We first study a restricted version of the problem, with null setup cost and null running cost/consumption. The problem description is given in Section 5.3 via a mixed integer programming formulation. In Section 5.4, we establish that problem *energy*-LSP is NP-hard, even with null production and null holding costs. A polynomial time algorithm, based on dynamic programming, with a time complexity in $O(M^5T^4)$, is proposed in Section 5.5 under stationary energy parameters. In Section 5.6, different extensions are studied, namely under running cost, joint setup cost and running energy consumption. For different extensions NP-hardness results are established and the previous dynamic programming algorithm is slightly modified to solve the problem with stationary energy parameters to optimality. We finally conclude in Section 5.7.

5.2 Literature review

The problem studied in this article can be positioned at the intersection of the single-item dynamic lot sizing problem (LSP) and energy-efficient manufacturing. In this section, relevant studies published in both domains are presented, together with the very few studies at their intersection.

The single-item LSP aims to determine how much and in which periods to produce in order to satisfy a deterministic and discrete demand over a given time horizon, while minimizing the total production and storage costs. The reader can refer to Wagner and Whitin, 1958 for a seminal paper, and to Florian, Lenstra, and Rinnooy Kan, 1980b and to Bitran and Yanasse, 1982b for the first complexity analysis on the capacitated LSP. For more details on the different extensions and the methods proposed for this well-known production planning problem, refer to Brahimi et al., 2006, Pochet and Wolsey, 2006. In the literature, most of the existing problems in production planning focus on the minimization of production and holding costs. However, in order to respect the new environmental standards and energy consumption issues, more and more theoretical and practical applications integrate them within the optimization of the production planning (see Gahm et al., 2016 and Biel and Glock, 2016).

The aim of this chapter is to integrate the energy constraints into the lot sizing problem. In the related literature, some studies integrate explicitly the energy cost in the optimization problem (see Özdamar and Birbil, 1999, Uzel, 2004, Tang, Che, and Liu, 2012 and Ding et al., 2016), but they suppose that the available energy is unlimited. We propose here an approach assuming a certain limit on the energy level in each period, which also limits the quantity that can be produced. Notice that a limit on the amount of available energy is also considered by Artigues, Lopez, and Haït, 2013, Nattaf, Artigues, and Lopez, 2015, Nattaf et al., 2016 and Ngueveu, Artigues, and Lopez, 2016 for scheduling problems, by Schultz, Sellmaier, and Reinhart, 2015 for a short term production control problem and by Masmoudi et al., 2017 for a single-item capacitated LSP. In Masmoudi et al., 2017, the authors consider a flow-shop system with a maximum allowable energy level, as well as an electricity price in their objective function. The flow-shop system considered in Masmoudi et al., 2017 makes the problem quite different than ours with deadlines to respect. Moreover, the authors propose heuristics to solve their problem, while we theoretically study our models, analysing their complexity and proposing polynomial time exact algorithms. A very recent study from Giglio, Paolucci, and Roshani, 2017 integrates energy consumption issues into the lot sizing and scheduling decisions in a multi-item, multi-machine job-shop environment, with additional backlogging and remanufacturing assumptions. The capacitated machines consume a certain amount of energy when being idle and when producing units with a normal or with an accelerated mode. All the later energy issues are modeled via costs of energy consumption into the objective function. The authors propose a relax-and-fix heuristic to cope with this integrated problem. There are also several papers dealing with industrial case studies on energy issues in production planning problems (see Artigues, Lopez, and Haït, 2013, Santos and Almada-Lobo, 2012, Waldemarsson, Lidestam, and M., 2013 and Zhao, Ierapetritou, and Rong, 2016). For a recent review on energy-efficient scheduling issues in manufacturing, refer to Gahm et al., 2016 and for recent reviews on energy-efficient production planning to Biel and Glock, 2016.

The majority of the papers published in the domain of energy-efficient production planning consists in energy-efficient machine scheduling problems (see Biel and Glock, 2016). To the best of our knowledge, there are only a few studies in the literature coupling energy issues with discrete lot sizing problem: Masmoudi et al., 2017, Giglio, Paolucci, and Roshani, 2017 and Rapine et al., 2016a. In Masmoudi et al., 2017 and Giglio, Paolucci, and Roshani, 2017, the authors consider respectively flow-shop and job-shop systems where they integrate some energy cost or constraints and propose heuristics to solve the related complex problems (see the details above). As mentioned earlier, the problem we study in this chapter consists in an extension of the energy lot sizing problem studied in Rapine et al., 2016a. The main differences between the two models are the time-dependent start-up costs we consider here instead of stationary start-up costs considered in Rapine et al., 2016a, and the fact that both the production of units and the start-up of machine may have a non-null energy consumption. We also consider a more general cost structure, including running cost, setup cost and running energy consumption of the machines.

5.3 Problem formulation

The system we study is constituted of M parallel, identical and capacitated machines that can be started at any period respecting the energy restriction. We say that a machine is *running* if it is not turned off. That is, a running machine may either be producing units, or simply idle, ready to produce. We assume through the chapter that the amount of energy available in each period is stationary, and denoted by E. Note that, in our problem, the capacity limit related to the available energy amount is hard, whereas the capacity limit related to the capacitated machines is soft, since the available production capacity is not known in advance and is dependent on the number of running machines. In particular, we can take the decision to switch on more machines, if less than M machines are currently turned on, in order to increase the production capacity. We denote by m_t the number of machines that are running, and thus available for production, during period t. Initially we assume that all the machines are turned off, that is, $m_0 = 0$. In each period t, we have to decide:

- How many machines m_t^+ to turn on and how many machines m_t^- to turn off.
- Which quantity x_t to produce to satisfy the demand or/and to store in inventory.

The aim is to minimize the total cost over a finite horizon of length T to satisfy a deterministic demand d_t in each period t, without backlogging, while respecting the production capacity and the limit on the amount of energy available. The cost of a production planning includes the cost to start the machines, the cost to produce units and the cost to carry units in stock. The parameters used are listed below:

d_t:	demand in period t
c_t:	unit production cost in period t
h_t:	unit holding cost to carry a unit from period t to $t+1$
$f_t(k)$:	cost to turn on k machines on period t
U:	capacity of a machine
M:	number of machines in the system
E:	amount of energy available in each period
p_t:	unit energy consumption to produce one unit in period t
w_t:	energy consumption to start a machine in period t

The *energy*-LSP is formulated as the following mixed-integer program:

$$
\begin{aligned}
\min \quad & \textstyle\sum_{t=1}^{T}(f_t(m_t^+) + c_t x_t + h_t s_t) & & \\
s.t. \quad & s_{t-1} + x_t = s_t + d_t & \forall t \in \{1..T\} \quad & (1) \\
& x_t \le U m_t & \forall t \in \{1..T\} \quad & (2) \\
& p_t x_t + w_t m_t^+ \le E & \forall t \in \{1..T\} \quad & (3) \\
& m_t = m_{t-1} + m_t^+ - m_t^- & \forall t \in \{1..T\} \quad & (4) \\
& m_t \le M & \forall t \in \{1..T\} \quad & (5) \\
& s_t \ge 0, x_t \ge 0, m_t \in \mathcal{Z}^+, m_t^+ \in \mathcal{Z}^+, m_t^- \in \mathcal{Z}^+ & \forall t \in \{1..T\} \quad & (6)
\end{aligned}
$$

Constraint (1) is the classical material balance between the produced, stored and satisfied units. In each period t the production is limited by two constraints: production capacity constraint (2) and energy restriction constraint (3). Finally, constraint (4) represents the total number of machines running in each period, taking into account the machines switched on and switched off at the beginning of the period. Constraint (5) stipulates that this number of machines running cannot exceed the number of machines of the system. The feasibility domains are given by constraint (6).

We assume through the chapter that production costs follow non-speculative motives, also called Wagner-Whitin (WW) cost structure. Non-speculative motives imply that for any period t, we have $c_t + h_t \geq c_{t+1}$. In other words, producing and storing one unit in a period t has a higher cost than producing it later in period $t+1$. Under WW costs, it is dominant to produce as late as possible. Notice that, w.l.o.g., we can consider that the holding costs are null, by substituting unit production costs c_t with $\tilde{c}_t = c_t + \sum_{u=t}^{T} h_u$. This transformation is possible due to the linearity of the holding costs. For the details, see Pochet and Wolsey, 2006. WW cost assumption implies that $\tilde{c}_t \geq \tilde{c}_{t+1}$ holds for all periods, in other words, modified unit production costs $\tilde{c}_t$ are non-increasing over time.

5.4 Complexity result

In this section, we establish that *energy*-LSP is computationally difficult if the unit energy consumption parameter p_t is time-dependent, even if other energy parameters are stationary and most of the cost parameters are null. Next section will demonstrate that *energy*-LSP is polynomially solvable if all the energy parameters are stationary, which set quite precisely the frontier between hard and easy problems for *energy*-LSP.

Theorem 11 *If the number M of machines is part of the instance, problem energy-LSP is NP-hard even with null production cost ($c = 0$) and null holding cost ($h = 0$), and with stationary energy parameters E and w.*

Proof. The reduction is made from the PARTITION problem. Recall that an instance of PARTITION is constituted of a list of n integers a_i. Denoting by A the quantity $\sum_i a_i/2$, it is asked whether there exists a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} a_i = A$. For short we denote by $a(S)$ the sum of the a_i's for $i \in S$. Notice that we can restrict to instances such that all the a_i are lower than A, otherwise the answer is trivial. We transform an instance I of PARTITION into an instance $\tau(I)$ of *energy*-LSP as follows:

- We have $n + 2$ periods, indexed from 0 to $T = n + 1$. The only positive demand appears in the last period, with $d_T = nA$
- The capacity of a machine is $U = A$. The number of machines is equal to $M = n$. Unit production and holding costs are null.
- The amount of energy available in each period is $E = \Pi_{i=1,n}(A - a_i)$
- The amount of energy required to start a machine is $w = E$
- The amount of energy required to produce one unit at period t is equal respectively to: $p_0 = 1$, $p_t = E/(A - a_t)$ for $t = 1, \ldots, n$, and $p_T = 0$. Notice that for $t = 1, \ldots, n$, due to the limited amount E of energy, at most $(A - a_t)$ units can be produced.
- The start-up cost function is linear, with $f_0(1) = 0$, $f_t(1) = a_t$ for $t = 1, \ldots, n$, and $f_T(1) = A + 1$

- It is asked whether a solution of cost at most A exists.

The following table gives an overview of the time-dependent parameters p_t and f_t for the instance $\tau(I)$:

Period t	0	$1, \dots, n$	T
p_t	1	$E/(A - a_t)$	0
$f_t(1)$	0	a_t	$A{+}1$

We show that $\tau(I)$ is positive if and only if instance I is positive. Consider any planning x, and define S as the subset of periods of $\{1, \dots, n\}$ where a machine is started. Since the energy needed to start a machine is $w = E$, at most one machine can be started in each period, and in this case nothing can be produced, except for the last period T. Notice that starting a machine at period 0 is clearly dominant, and starting a machine at period T is prohibited for a planning of cost at most A. Hence, the number of machines available to produce at the last period is equal to $1 + |S|$. Let us denote by X the cumulative production of periods $0, \dots, n$, and let $\bar{S} = \{1, \dots, n\} \backslash S$. Notice that a production before the last period can only occur in periods of $\bar{S}$. In these periods, at least one machine is on (the one started in period 0). Hence, production is limited by the amount of energy available. For the planning to be feasible, we must have $X + x_T \geq d_T = nA$. We have:

- $X = \sum_{i \in \bar{S}} x_i \leq \sum_{i \in \bar{S}} (A - a_i) = |\bar{S}|A - a(\bar{S})$. Hence, we have $X \leq (n - |S|)A + (a(S) - 2A)$
- $x_T \leq (1 + |S|)A$ since $1 + |S|$ machines are on at the last period

As a result, on one hand, for a planning to be feasible, we must have the inequality $(n-1)A + a(S) \geq nA$, or written differently, $a(S) \geq A$. On the other hand, for a planning to cost at most A, we must have $a(S) \leq A$, since starting a machine in a period $i \in S$ incurs a cost $f_i(1) = a_i$. We can conclude that instance I is positive, as S defines a partition of value A. Conversely, if S is a partition of value A, it is easy to check that a planning starting a machine at period 0 and at each period of S, is feasible and costs at most A. □

5.5 A polynomial time algorithm for stationary energy parameters

In this section we propose an exact $O(M^5T^4)$ time dynamic programming algorithm for the important case when all the energy parameters are stationary over time. More precisely, we assume that, in each period, an amount E of energy is available, the unit energy consumption is equal to p and the energy consumption to start a machine is equal to w. Notice that the cost parameters of the problem are allowed to be time-varying, under the restriction that the unit production and holding costs follow a non-speculative motive. Considering the formulation of *energy*-LSP given in Section 5.3, we call a period $C-$saturated if it saturates the production capacity constraint (2), and $E-$saturated if it saturates the energy constraint (3). If a period t is $C-$saturated, by definition we have $x_t = m_t U$, whereas if t is $E-$saturated, we have $x_t = (E - w m_t^+)/p$. Notice that both m_t and m_t^+ are decision variables of the problem. Following the classical terminology of lot-sizing problems, we also called a period *a regeneration point* if its entering inventory is null. We have the following property:

Property 15 *In a dominant solution, each period t is either:*

- *a regeneration point, that is,* $s_{t-1} = 0$
- *or* $C-$*saturated, that is,* $x_t = m_t U$
- *or* $E-$*saturated, that is,* $p x_t + w m_t^+ = E$

A period may eventually be at the same time a regeneration point, a $C-$*saturated period and/or a* $E-$*saturated period.*

Proof. Let us consider an optimal solution π' and let t be the last period which complies with none of these possibilities. Consider one unit entering in stock in period t, and let t' be the production period of this unit. Since both capacities are not reached (t being neither $C-$saturated nor $E-$saturated), and due to non-speculative motives (delaying a production is not more expensive), it is possible to postpone the production of this unit from period t' to period t without increasing the cost of the solution or rendering it unfeasible. This operation can be iterated until the stock vanishes at the beginning of period t or t becomes $C-$saturated or $E-$saturated. As a result, Property 15 becomes valid at period t and remains valid at the subsequent periods. The process can be repeated until Property 15 is valid at each period. The result follows. $\square$

In lot sizing domain, a very common problem-solving approach consists in decomposing the entire horizon into independent subplans (u, v), solving each subplan separately and then constructing the optimal solution via a shortest path algorithm. Recall that a subplan is defined as the set of periods between two consecutive regeneration points. By definition of a subplan (u, v), the entering stock level in periods u and v is null, and in any period t inside the subplan the stock level is positive. Very classically, we compute the optimal cost of each possible subplan (u, v) using dynamic programming. However, this cost clearly depends on the number m_{u-1} of machines on at the beginning of period u, which depends on prior decisions, and on the number m_{v-1} of machines on at the beginning of period v, which impact decisions on subsequent periods. For this reason, to render the cost of a subplan independent of the rest of the horizon, we fix the number of machines on at the beginning of the regeneration periods. That is, we compute the optimal cost of a policy for all pairs of periods (u, v), $1 \leq u < v \leq T+1$ and all pairs of integers (k, l), $0 \leq k, l \leq M$, assuming that k machines, respectively l, are running at the beginning of period u, respectively v. Let us denote by

$\mathscr{S}(u,k,v,l)$ the optimal cost of such a subplan. To compute the optimal cost of a policy, we consider the graph whose nodes are labeled by pairs (u,k), for $1 \le u \le T+1$ and $0 \le k \le M$, and where an arc between a node (u,k) and a node (v,l) exists if $u < v$. The length of the arc $((u,k),(v,l))$ is precisely $\mathscr{S}(u,k,v,l)$. We also add a leaf node S, which is a successor of each node $(T+1,k)$, for $0 \le k \le M$. The length of any arc $((T+1,k),S)$ is null. Notice that this graph has $O(TM)$ nodes. Since this graph is acyclic, the shortest path between root node $(1,0)$ and leaf node S can be computed in a time linear in the number of arcs, and thus quadratic in the number of nodes. The length of this shortest path corresponds to the cost of an optimal policy over the time horizon $1,\ldots,T$. As a consequence, assuming that all the costs $\mathscr{S}(u,k,v,l)$ are known, the optimal cost of a policy can be computed in time $O(M^2T^2)$. We detail now how the cost $\mathscr{S}(u,k,v,l)$ of a subplan can be determined using dynamic programming.

Optimal cost of a subplan

Let us consider a subplan (u,v), such that k machines, respectively l, are running at the beginning of period u, respectively v, in an optimal solution. Let us denote by $\mathscr{B} = \{u+1,\ldots,v-1\}$ the periods inside the subplan. Due to Property 15, each of these periods is either $C-$saturated or $E-$saturated, or both. We denote by $\mathscr{B}_C = \{t \in \mathscr{B} \mid t \text{ is } C\text{-saturated}\}$ and by $\mathscr{B}_E = \{t \in \mathscr{B} \mid t \text{ is } E\text{-saturated and not } C\text{-saturated}\}$. It clearly defines a partition of set $\mathscr{B}$. Let $t \in \{u+1, v-1\}$ be a period inside the subplan. Assume that we know the value of the following quantities in an optimal planning:

- m: number of machines running at the beginning of period t, before one decides how many machines m_t^+ to start in this period, that is, $m = m_{t-1}$.
- N_C: the sum of the number of the machines running during each period of $\mathscr{B}_C$ over the time horizon $\{t,\ldots,v-1\}$: $N_C = \sum_{i\in\mathscr{B}_C:i\ge t} m_i$
- N_E^+: the sum of the number of machines started over the periods of $\mathscr{B}_E$ over the time horizon $\{t,\ldots,v-1\}$: $N_E^+ = \sum_{i\in\mathscr{B}_E:i\ge t} m_i^+$
- n_E: the number of periods of $\mathscr{B}_E$ over the time horizon $\{t,\ldots,v-1\}$, that is, $n_E = |\mathscr{B}_E|$

We claim that the entering stock level s_{t-1} of period t in an optimal planning is fixed for a given vector (t, m, N_C, N_E^+, n_E). To see why, let us write the flow conservation on time horizon $\{t,\ldots,v-1\}$. We have:

$$\begin{aligned} D_{t,v-1} + s_{v-1} &= s_{t-1} + \textstyle\sum_{i=t}^{v-1} x_i \\ &= s_{t-1} + \textstyle\sum_{i\in\mathscr{B}_C:i\ge t} x_i + \sum_{i\in\mathscr{B}_E:i\ge t} x_i \\ &= s_{t-1} + \textstyle\sum_{i\in\mathscr{B}_C:i\ge t} m_i U + \sum_{i\in\mathscr{B}_E:i\ge t}(E - m_i^+ w)/p \\ &= s_{t-1} + N_C U + (n_E E - N_E^+ w)/p \end{aligned}$$

Since $s_{v-1} = 0$ as v is a regeneration period, we obtain that:

$$s_{t-1} = D_{t,v-1} - N_C U - (n_E E - N_E^+ w)/p \tag{5.7}$$

We denote by $\sigma_{t-1}(v, m, N_C, N_E^+, n_E)$ this quantity. Notice that σ_{t-1} can be computed in constant time if the cumulative demand $D_{t,v-1}$ has been pre-computed. Vector (t, m, N_C, N_E^+, n_E) represents the state vector of our dynamic programming algorithm. For fixed values v and l, we compute $\mathscr{C}(t, m, N_C, N_E^+, n_E)$ the minimal cost of a policy on the time horizon $\{t, \ldots, v-1\}$, such that the inventory level at the end of period $v-1$ is null and the number of running machines is equal to l. To express this cost, we distinguish between two cases, depending whether t belongs to $\mathscr{B}_C$ or $\mathscr{B}_E$.

$t \in \mathscr{B}_C$: t is a $C-$saturated period

If m' is the number of machines running during period t, by definition the quantity produced is equal to $m'U$. In this case, $(m' - m)^+$ machines are started at the beginning of period t, incurring a cost of $f((m' - m)^+)$ and consuming an amount $w(m' - m)^+$ of energy. For the planning to be feasible, the energy constraint must be satisfied, which implies that $pm'U + w(m' - m)^+ \leq E$ must hold. We must also have $m'U + s_{t-1} \geq d_t$ in order to satisfy the demand at period t. The minimal possible cost of a policy on the remaining time horizon $\{t+1, \ldots, v-1\}$ is by definition $\mathscr{C}(t+1, m', N_C - m', N_E^+, n_E)$. Let us denote by $\gamma_C(m, m')$ the cost incurred at period t, including the start-up cost of the machines and the variable production cost. We set by convention $\gamma_C(m, m') = +\infty$ if the energy constraint is violated or the demand cannot be satisfied in a full production period. We obtain the formula:

$$\gamma_C(m, m') = \begin{cases} +\infty & \text{if } m'U + \sigma_{t-1}(v, m, N_C, N_E^+, n_E) < d_t \\ +\infty & \text{if } pm'U + w(m' - m)^+ > E \\ f_t((m' - m)^+) + \tilde{c}_t m'U & \text{otherwise} \end{cases}$$

We must have by suboptimality:

$$\mathscr{C}(t, m, N_C, N_E^+, n_E) = \min_{m'=0,\ldots,M} \{\gamma_C(m, m') + \mathscr{C}(t+1, m', N_C - m', N_E^+, n_E)\} \tag{5.8}$$

$t \in \mathscr{B}_E$: t is not a $C-$saturated period but an $E-$saturated period

Again, knowing the number m^+ of machines started in period t, the production x_t is fixed, since we have $px_t + wm_t^+ = E$. We can enumerate all the possible values of m' as previously, with $m^+ = (m' - m)^+$. For a given value of m', the amount x_t produced is equal to $(E - w(m' - m)^+)/p$. For the planning to be feasible, we must satisfy demand d_t and ensures that the quantity x_t produced is non-negative. It results that we must have $d_t \leq s_{t-1} + x_t$ and $x_t \geq 0$. The optimal cost on the remaining time horizon till period v is then equal to $\mathscr{C}(t+1, m', N_C, N_E^+ - (m' - m)^+, n_E - 1)$. Again we introduce cost

$\gamma_E(m,m')$ incurred by period t if it saturates the energy constraint. We have:

$$\gamma_E(m,m') = \begin{cases} +\infty \ \text{ if } (E - w(m'-m)^+)/p + \sigma_{t-1}(v,m,N_C,N_E^+,n_E) < d_t \\ +\infty \ \text{ if } E - w(m'-m)^+ < 0 \\ f_t((m'-m)^+) + \tilde{c}_t(E - w(m'-m)^+)/p \ \text{ otherwise} \end{cases}$$

It results that

$$\mathscr{C}(t,m,N_C,N_E^+,n_E) = \min_{m'=0,\dots,M} \{\gamma_E(m,m') + \mathscr{C}(t+1,m',N_C,N_E^+ - (m'-m)^+, n_E - 1)\} \quad (5.9)$$

Dynamic programming algorithm

For a given regeneration point v and a given number l of machines running at the beginning of period v, we compute cost $\mathscr{C}(t,m,N_C,N_E^+,n_E)$ for all possible state vectors such that $t \leq v$ using Equations (5.8) and (5.9), by taking the minimal of both expressions. The basis of the induction is given in period v by setting:

$$\mathscr{C}(v,m,N_C,N_E^+,n_E) = \begin{cases} 0 & \text{for vector } (v,l,0,0,0) \\ +\infty & \text{otherwise} \end{cases} \quad (5.10)$$

The computation of each cost $\mathscr{C}(t,m,N_C,N_E^+,n_E)$ according to Equations (5.8) and (5.9) requests to compare $O(M)$ different values of m', where each computation for a given m' can be performed in constant time. The number of state vectors to consider is bounded by $(M+1)^3T^3$: t and n_E can take at most T different values, m is clearly bounded by $(M+1)$ and N_C by $(M+1)T$. Parameter N_E^+ can also take at most $(M+1)$ different values, since at most M machines can be started in a period. Recall that in our setting there is no incentive to shut down a machine. It results that the number of starts over the time horizon is bounded by M, since a machine is started at most once. Hence, the computation of all the costs $\mathscr{C}(t,m,N_C,N_E^+,n_E)$ for a given pair (v,l) requires $O(M^4T^3)$ operations.

Recall that we want to compute the cost $\mathscr{S}(u,k,v,l)$ of all possible subplan (u,v), with k and l being the number of machines running at the beginning of period u and respectively v. Assume that all the costs $\mathscr{C}(t,m,N_C,N_E^+,n_E)$ have been computed for pair (v,l). Again, if we know the state vector $(u+1,m,N_C,N_E^+,n_E)$ reached at period $u+1$, we can deduce the quantity x_u produced in period u and the number m_u^+ of machines started in period u. Indeed, we have $x_u = d_u + \sigma_u(v,m,N_C,N_E^+,n_E)$ and, clearly, $m_u^+ = (m-k)^+$. In order to be feasible, the production must respect the capacity available and the energy limitation. With our convention that the cost $\gamma_R(k,m)$ is infinite if the constraints are violated in period u, we have:

$$\gamma_R(k,m) = \begin{cases} +\infty \ \text{ if } d_u + \sigma_u(v,m,N_C,N_E^+,n_E) > mU \\ +\infty \ \text{ if } p(d_u + \sigma_u(v,m,N_C,N_E^+,n_E)) + w(m-k)^+ > E \\ f_u((m-k)^+) + \tilde{c}_u(d_u + \sigma_u(v,m,N_C,N_E^+,n_E)) \ \text{ otherwise} \end{cases}$$

This cost $\gamma_R(k,m)$ can be computed in constant time if the value σ_u of the stock level is known. The cost $\mathscr{S}(u,k,v,l)$ of the subplan starting at period u with k machines running can be obtained as:

$$\mathscr{S}(u,k,v,l) = \min\{\gamma_R(k,m) + \mathscr{C}(u+1,m,N_C,N_E^+,n_E) \\ \mid 0 \leq m \leq M,\ 0 \leq N_C \leq MT,\ 0 \leq N_E^+ \leq M, 0 \leq n_E \leq T\} \quad (5.11)$$

Hence, computing $\mathscr{S}(u,k,v,l)$ for a given subplan can be performed in time $O(M^3T^2)$. Computing this cost for all the subplans (u,k,v,l) represents a computation effort in $O(M^5T^4)$, since we have in total $O(M^2T^2)$ subplans to consider. Notice that this complexity dominates the computation complexity in $O(M^4T^3)$ of determining the costs $\mathscr{C}(t,m,N_C,N_E^+,n_E)$. The final complexity of the algorithm is thus in $O(M^5T^4)$ time. We have the following result:

Theorem 12 *Problem energy-LSP can be solved in polynomial time in $O(M^5T^4)$ if energy parameters p, w and E are stationary.*

Recall that Theorem 11 shows that the problem becomes NP-hard if the unit production energy consumption p is allowed to be time-varying. In the next section, we generalize the problem by considering a joint set-up cost for production, and a running cost and a running energy consumption incurred by each machine which is not turned off.

5.6 Extensions

In the previous sections, there was no incentive to shut down a machine due to the cost structure we considered. This may be quite unrealistic, since an idling machine that is not switched off may incur some costs, and eventually may consume some energy. Also, only start-up costs are considered in the previous sections, whereas starting a production (on a machine already running) usually incurs a fixed set-up cost. In this section we extend our results by including the following parameters in our model:

r_t: running cost, incurred by each machine running during period t. This cost is incurred whether the machine is producing or is idle. Notice that an additional cost of $\tilde{c}_t x$ is still to be paid if a machine produces x units during period t.

g_t: running energy consumption, which represents the energy consumed by a machine running during period t. In the same way, this energy is consumed whenever the machine is not turned off. Notice that if a machine produces x units during period t, it consumes an additional $p_t x$ amount of energy.

K_t: joint set-up cost for producing in period t, whatever the number of machines used.

We first provide complexity results considering separately each of these three new parameters. We show that problem *energy*-LSP becomes NP-hard on a single resource if one of these parameter is not null. Then we show how the polynomial time algorithm given in Section 5.5 can be modified to fit with this more general problem when energy parameters are stationary.

5.6.1 Complexity result for non-null running costs

We consider that, for each machine running during period t, a cost r_t has to be paid, whether it produces or not. The costs incurred at a given period t where a quantity x_t is produced in the system with m_t machines running, are now equal to $f_t(m_t^+) + c_t x_t + h_t s_t + r_t m_t$, where m_t^+ is as before the number of machines started at the beginning of the period, and s_t the stock level at the end of the period. Karmarkar, Kekre, and Kekre, 1987 proposed an $O(T^2)$ time algorithm for a special case of our problem, considering a single uncapacitated machine and without energy consideration. Next theorem establishes that problem *energy*-LSP with running costs is NP-hard even for a single machine. This result strengthens Theorem 11 where the number M of machines is part of the inputs.

Theorem 13 *Problem energy-LSP with running costs is NP-hard, even for a single machine and null start-up energy consumption* $(w = 0)$.

Proof. The reduction is made from the capacitated lot-sizing problem (CLSP), which has been shown to be NP-hard by Florian, Lenstra, and Rinnooy Kan, 1980b. In a CLSP instance, the amount that can be produced in each period is limited by a capacity C'_t. We also have a set-up cost K'_t to be paid if a positive quantity is produced in period t. Let c'_t, h'_t and d'_t be the unit production cost, unit holding cost and the demand of CLSP instance I', respectively, on a time horizon of T' periods. It is asked whether a feasible schedule of total cost at most Z exists. The complexity proof of Florian, Lenstra, and Rinnooy Kan, 1980b allows us to assume that $h'_t = 0$ and that the c'_t are non-increasing, that is, they obey a non-speculative cost structure. We can also restrict ourselves to the instance with non-null capacities. We transform instance I' of CLSP into an instance I of *energy*-LSP as follows:

- We have $T = 2T'$ periods, indexed from 1 to T.
- We have a single machine of stationary capacity $U = \max_t C_t$.
- The amount of energy available in each period is $E = \Pi_{t=1,\dots,T} C_t$
- The amount of energy required to start a machine is null, $w = 0$
- It is asked if a feasible planning of cost at most Z exists

In the instance I of *energy*-LSP, we distinguish between odd and even periods. We set the value of the different parameters as follows:

t	d_t	f_t	c_t	h_t	p_t	r_t
$2t'-1$	$d_{t'}$	$K_{t'}$	$c_{t'}$	0	$E/C_{t'}$	0
$2t'$	0	0	0	0	0	$Z+1$

Notice that in each odd period $2t'-1$, the production is limited by $E/p_t = C_{t'}$ due to the limited amount of energy available. Basically, an odd period $t = 2t'-1$ in instance I plays the role of period t' in instance I', whereas even periods force to shut-down the machine, since the running cost r_t in these periods exceeds Z. It is immediate to check that I is positive if and only if instance I' is positive. □

5.6.2 Complexity result for non-null running energy consumption

We consider that a machine that is not turned off consumes some energy even if it does not produce. This running energy consumption parameter reflects practical situation where a non-negligible amount of energy is required to keep a machine ready or on standby, for instance to maintain the temperature of a furnace. It involves that in each period, one has now to arbitrate how to share the available amount E of energy between 3 activities, namely the start of machines, the effective production and keeping the machines on running. In the formulation given in Section 5.3, the energy constraint (3) at a period t becomes $p_t x_t + w_t m_t^+ + g_t m_t \leq E$. Again, the problem becomes NP-hard on a single machine when considering running energy consumption:

Theorem 14 *Problem energy-LSP with running energy consumption is NP-hard, even for a single machine and null start-up energy consumption ($w = 0$).*

We can use the same reduction as in the proof of Theorem 13. In order to force the shut-down of the machine during the even periods we simply set the running consumption parameter g_t equal to $E + 1$.

5.6.3 Complexity result for non-null joint setup costs

In lot-sizing literature, it is common to consider that a positive production at a period incurs a setup cost, representing typically the efforts and materials requested to prepare the machines. We assume here a joint setup cost for the system: A positive production at a period t incurs a fixed setup cost K_t, whatever the amount produced and the number of running machines. Hence, if $x_t > 0$, the costs to be paid for period t are $f_t(m_t^+) + c_t x_t + h_t s_t + K_t$. Otherwise, if no production takes place, the costs reduce to $h_t s_t$. We have the following result:

Theorem 15 *Problem energy-LSP with a joint setup cost is NP-hard, even on a single machine with null start-up energy consumption ($w = 0$) and null start-up cost ($f = 0$).*

Proof. Once again, the reduction is made from the capacitated lot-sizing problem. Using the same notations as in the proof of Theorem 13, the instance I' of CLSP is transformed into an instance I of *energy*-LSP as follows:

- We have $T = T'$ periods, indexed from 1 to T.
- We have a single machine of stationary capacity $U = \max_t C'_t$.
- The amount of energy available in each period is $E = \Pi_{t=1,\ldots,T} C'_t$
- The amount of energy required to start the machine is null, $w = 0$
- The cost to start the machine is null, $f_t = 0$
- The unit energy consumption is $p_t = E/C'_t$
- The setup costs are identical to the setup costs of instance I', $K_t = K'_t$

Notice that, in instance I of *energy*-LSP, the quantity x_t that can be produced in period t is limited by $\min\{U, E/p_t\} = C'_t$. It results that instance I is positive if and only if instance I' of CLSP is positive. □

Bitran and Yanasse, 1982b establish other NP-hardness results for different variants of CLSP. In particular, the authors show that even with null holding and null unit production costs (and hence obeying non-speculative motives), problem CLSP remains NP-hard when the setup costs and the production capacities are both non-increasing or both non-decreasing over time. With the previous transformation, this result immediately translates to *energy*-LSP: Problem *energy*-LSP is NP-hard even in the special case of null holding and null unit production costs, if joint setup costs are non-increasing and unit energy consumption are non-decreasing, or vice-versa.

5.6.4 Polynomial time algorithm for running cost, joint setup cost and running energy consumption case

We finally consider the general version of *energy*-LSP including all three parameters, that is, running cost, joint setup cost and running energy consumption. We show that when all the energy parameters p, w, g and E are stationary, the problem is still polynomially solvable. Introducing a binary variable y_t to indicate the periods with a positive production, the formulation of the problem can be written as follows:

$$
\begin{array}{llrr}
\min & \sum_{t=1}^{T}(f_t(m_t^+) + c_t x_t + h_t s_t + r_t m_t + K_t y_t) & & \\
s.t. & s_{t-1} + x_t = s_t + d_t & \forall t \in \{1..T\} & (11) \\
& x_t \leq U m_t & \forall t \in \{1..T\} & (12) \\
& p x_t + w m_t^+ + g m_t \leq E & \forall t \in \{1..T\} & (13) \\
& m_t = m_{t-1} + m_t^+ - m_t^- & \forall t \in \{1..T\} & (14) \\
& m_t \leq M & \forall t \in \{1..T\} & (15) \\
& x_t \leq U M y_t & \forall t \in \{1..T\} & (16) \\
& s_t \geq 0, x_t \geq 0, y_t \in \{0,1\}, m_t \in \mathcal{Z}^+, m_t^+ \in \mathcal{Z}^+, m_t^- \in \mathcal{Z}^+ & \forall t \in \{1..T\} & (17)
\end{array}
$$

Constraint (16) forces variable y_t to be equal to 1 if the quantity x_t produced in period t is positive. Notice that UM is an upper bound of the quantity that can be produced in any period. Recall that we assume non speculative motives, that is, $\tilde{c}_t \equiv c_t + (h_t + \cdots + h_T)$ is non-increasing over time.

One important difference with the basic model introduced in Section 5.3 is that now each running machine incurs a cost and consumes energy, even if it is not producing units. As a consequence, an optimal solution may have to switch off some machines in a period to start them again latter, in order to save their running costs and to limit their energy consumption. Another difference is that, due to economies of scale induced by the joint setup costs, there can be periods with null production even if the number of running machines is positive and with a demand to satisfy. As a result, Property 15 must be slightly modified for our generalized model.

Property 16 *In a dominant solution, each period t is either:*

- *a regeneration point, that is, $s_{t-1} = 0$*
- *or $C-$saturated, that is, $x_t = m_t U$*
- *or $E-$saturated, that is, $p x_t + w m_t^+ + g m_t = E$*
- *or a null production period, that is, $x_t = 0$*

It implies that a period t with both a positive entering stock ($s_{t-1} > 0$) and a positive production ($x_t > 0$) is either $C-$saturated or $E-$saturated. The proof is similar to the proof of Property 15, by considering a period t with a positive production (hence the joint setup cost is already paid for): Due to the non-speculative cost structure, postponing the production of one unit carried in the entering stock to period t cannot increase the cost of the solution. As a consequence of Property 16, only the first period of a subplan can have a production that does not saturate constraints (12) and (13). Note that a null production period may also occur if all the machines are switched off. This is a special case of a $C-$saturated period, with $m_t = 0$ and $x_t = m_t U = 0$.

We explain in the remainder of this section how the dynamic programming algorithm proposed in Section 5.5 can be slightly modified to solve this more general problem, to obtain the following result:

Theorem 16 *Problem energy-LSP with running costs, joint setup costs and running energy consumption can be solved in polynomial time in $O(M^6T^6)$ if energy parameters p, w, g and E are stationary.*

We point out the modifications by following the structure of Section 5.5 where the initial version of our dynamic programming algorithm is presented. Recall that we compute the optimal cost $\mathscr{S}(u,k,v,l)$ of each possible subplan, using a state vector (t,m,N_C,N_E^+,n_E) to determine the optimal costs $\mathscr{C}(t,m,N_C,N_E,N_E^+,n_E)$ over the time horizon $\{t,\dots,v-1\}$ inside a given subplan.

Optimal cost of a subplan (u,v)

The main modification of the algorithm is the enlargement of the state vector by the addition of a new component N_E, which represents the sum of the number of machines running during each period of $\mathscr{B}_E$ over the time horizon $\{t,\dots,v-1\}$, that is, $N_E = \sum_{i\in\mathscr{B}_E:i\geq t} m_i$. Notice that N_E is the counterpart of N_C for the periods of $\mathscr{B}_E$. This new component is necessary to be able to evaluate the entering stock level of a period in our dynamic programming algorithm, as explained below. Consider the state vector (t,m,N_C,N_E,N_E^+,n_E) associated with a period t inside the subplan (u,v). Using the conservation of the flow over the time interval $\{t,\dots,v-1\}$, we have:

$$\begin{aligned} D_{t,v-1} + s_{v-1} &= s_{t-1} + \textstyle\sum_{i=t}^{v-1} x_i \\ &= s_{t-1} + \textstyle\sum_{i\in\mathscr{B}_C:i\geq t} x_i + \sum_{i\in\mathscr{B}_E:i\geq t} x_i \\ &= s_{t-1} + \textstyle\sum_{i\in\mathscr{B}_C:i\geq t} m_i U + \sum_{i\in\mathscr{B}_E:i\geq t} (E - m_i^+ w - m_i g)/p \\ &= s_{t-1} + N_C U + (n_E E - wN_E^+ - gN_E)/p \end{aligned}$$

Since $s_{v-1} = 0$, as v is a regeneration point, we obtain that the entering stock level at period t is equal to:

$$s_{t-1} = D_{t,v-1} - N_C U - (n_E E - wN_E^+ - gN_E)/p$$

Slightly abusing the notation, we denote by $\sigma_{t-1}(v,m,N_C,N_E,N_E^+,n_E)$ this quantity.

Another modification of the algorithm is that, due to Property 16, a period inside a subplan (u,v) may be never C-saturated nor E-saturated, but a null-production period. Hence, when computing cost $\mathscr{C}(t,m,N_C,N_E,N_E^+,n_E)$, we now need to distinguish between 3 cases, depending whether t belongs to $\mathscr{B}_C$, $\mathscr{B}_E$ or is a null production period.

$t \in \mathscr{B}_C$: t **is a** C**-saturated period.**

Since the modified energy constraint $pm'U + w(m'-m)^+ + gm' \leq E$ must hold, the new formula to evaluate $\gamma_C(m,m')$ taking into account the setup and running costs is:

$$\gamma_C(m,m') = \begin{cases} +\infty & \text{if } m'U + \sigma_{t-1}(v,m,N_C,N_E,N_E^+,n_E) < d_t \\ +\infty & \text{if } pm'U + w(m'-m)^+ + gm' > E \\ f_t((m'-m)^+) + \tilde{c}_t m'U + K_t \mathbf{1}_{\{m'\}} + rm' & \text{otherwise} \end{cases}$$

Where $\mathbf{1}_{\{z\}}$ is the indicator function, equals to 1 if $z > 0$ and to 0 otherwise. It results that:

$$\begin{aligned} &\mathscr{C}(t,m,N_C,N_E,N_E^+,n_E) \\ &\qquad = \min_{m'=0,\ldots,M} \{\gamma_C(m,m') + \mathscr{C}(t+1,m',N_C-m',N_E,N_E^+,n_E)\} \end{aligned}$$

$t \in \mathscr{B}_E$: t **is a** $E-$**saturated period and not a** $C-$**saturated period**

Again some changes appear: the amount x_t produced is now equal to $(E - wm_t^+ - gm')/p$. Hence $\gamma_E(m,m')$ is given by:

$$\gamma_E(m,m') = \begin{cases} +\infty & \text{if } (E - w(m'-m)^+ - gm')/p \\ & \quad +\sigma_{t-1}(v,m,N_C,N_E,N_E^+,n_E) < d_t \\ +\infty & \text{if } E - w(m'-m)^+ - gm' < 0 \\ f_t((m'-m)^+) + \tilde{c}_t(E - w(m'-m)^+ - gm')/p & \\ \quad +K_t \mathbf{1}_{\{E-w(m'-m)^+-gm'\}} + rm' & \text{otherwise} \end{cases}$$

It results that:

$$\begin{aligned} &\mathscr{C}(t,m,N_C,N_E,N_E^+,n_E) \\ &= \min_{m'=0,\ldots,M} \{\gamma_E(m,m') + \mathscr{C}(t+1,m',N_C,N_E-m',N_E^+-(m'-m)^+,n_E-1)\} \end{aligned}$$

$t \notin \mathscr{B}_C \cup \mathscr{B}_E$: t **is a null production period**

The value of x_t is by definition equal to 0. We must ensure that the entering stock level is greater than or equal to the demand in t for the planning to be feasible. Notice that it is possible to start some machines during period t and to keep some other running, even if no unit is produced. Hence we must also ensure that the energy constraint is not violated. The cost $\gamma_N(m,m')$ incurred by a null production period t is given by:

$$\gamma_N(m,m') = \begin{cases} +\infty & \text{if } \sigma_{t-1}(v,m,N_C,N_E,N_E^+,n_E) < d_t \\ +\infty & \text{if } w(m'-m)^+ + gm' \geq E \\ f_t((m'-m)^+) + rm' & \text{otherwise} \end{cases}$$

The optimal cost on the remaining time horizon till period v is equal simply to $\mathscr{C}(t+1, m', N_C, N_E, N_E^+, n_E)$. It results that

$$\mathscr{C}(t, m, N_C, N_E, N_E^+, n_E) = \min_{m'=0,\dots,M} \{\gamma_N(m, m') + \mathscr{C}(t+1, m', N_C, N_E, N_E^+, n_E)\}$$

Complexity of the algorithm.

The number of state vectors $(t, m, N_C, N_E, N_E^+, n_E)$ to consider in a given subplan is bounded by $((M+1)^4 T^5)$. Indeed, the new component N_E can take at most $(M+1)T$ values, in the same way as N_C. Moreover, the feasible set of N_E^+ is also enlarged. Recall that N_E^+ is the sum of the number of machines started in each period of $\mathscr{B}_E$ over $\{t, \dots, v-1\}$. Without running costs nor running energy consumption, we only have M possibilities, since a machine is started at most once in a dominant planning. This is not true anymore in our extension: In a period one can switch off all the machines and later the same machines can be switched on again. Hence, the number of possible values for N_E^+ increases to $(M+1)T$ possibilities. As a consequence, the computation of all the costs $\mathscr{C}(t, m, N_C, N_E, N_E^+, n_E)$ for a given pair (v, l) requires $O(M^5T^5)$ operations. Considering all pairs (v, l), the final complexity becomes $O(M^6T^6)$. Recall that the optimal cost $\mathscr{S}(u, k, v, l)$ of a subplan starting at period u with k machines running, is computed using the expression:

$$\begin{aligned}\mathscr{S}(u, k, v, l) = \min\{&\gamma_R(k, m) + \mathscr{C}(u+1, m, N_C, N_E, N_E^+, n_E) \\ &\mid 0 \le m \le M,\ 0 \le N_C, N_E, N_E^+ \le MT, 0 \le n_E \le T\}\end{aligned}$$

Where $\gamma_R(k, m)$ is defined by:

$$\gamma_R(k,m) = \begin{cases} +\infty & \text{if } d_u + \sigma_u(v, m, N_C, N_E, N_E^+, n_E) > mU \\ +\infty & \text{if } p(d_u + \sigma_u(v, m, N_C, N_E^+, n_E)) + w(m-k)^+ + gm > E \\ f_u((m-k)^+) + \tilde{c}_u(d_u + \sigma_u(v, m, N_C, N_E^+, n_E)) & \\ \qquad + K_u \mathbf{1}_{\{d_u + \sigma_u(v, m, N_C, N_E^+, n_E)\}} + r_u m & \text{otherwise} \end{cases}$$

Hence, for a given subplan, cost $\mathscr{S}(u, k, v, l)$ can be computed in time complexity $O(M^4T^4)$. Since we have $O(M^2T^2)$ 4-uplet (u, k, v, l) to consider, we obtain the overall complexity claimed in Theorem 16. We can remark that with null running energy consumption ($g = 0$), we do not need to incorporate N_E in the state vector. In this case the complexity of the algorithm reduces to $O(M^5T^5)$.

5.7 Conclusion

In this chapter, we have investigated a new energy-aware lot-sizing problem, where the amount of energy available in each period is limited. In the system under study, different activities consume energy, from turning on a machine, keeping it on running, to the effective production of goods. Hence, in each period, one has to arbitrate the use of energy between the increase of system capacity by starting up machines and keeping them running, and the effective production of units. We have established that problem *energy*-LSP is NP-hard, even in very restricted cases with null production cost and null holding costs. We have also shown that the problem is polynomially solvable if all energy consumption parameters are stationary. We have proposed an $O(M^6T^6)$ algorithm based on dynamic programming for the most general case studied in this article, including running costs, joint setup costs and running energy consumptions.

Chapter 6

Conclusion

In this thesis we have studied several multi-level lot-sizing problems with capacities. Such problems are particularly important in supply chain management and production flow problems, which require effective inventory management. We have been focusing on problems with capacities at each level, which reflects the fact that some restrictions usually occur in practice, due to transportation or warehouse handling capacities.

In Chapter 2, we have considered a 2-level in series lot-sizing problem with identical capacities at both level. A new cost structure called path non-speculative has been introduced, generalizing the non-speculative property of Wagner-Whitin. We have defined a class of new policies, named double-nested, which states that if the second level orders at full capacity, the first level also does. We established that double-nested policies are dominant for our problem. We have proposed an exact algorithm running in $O(T^5)$ time when setup costs are non-increasing at the first level. Assuming echelon non-speculative motives at both levels, the complexity is reduced to $O(T^3)$.

We think that our algorithm could be used as the basis of approximation algorithms or heuristics to solve problems with different capacity values C^1 and C^2. Such problems can be divided in two cases depending on the values of C^1 and C^2. In the case where $C^1 > C^2$, the manufacturer has a greater production capacity than the retailer. This invalidates the dominance of double-nested policies (Property 3) since the productions will tend to be gathered at the manufacturer, to get economies of scale and to reduce the number of periods for which a setup cost is paid. A simple heuristic can be designed by setting the capacity at the retailer as $\bar{C}^1 = C^2$. The problem is then over-constrained, which implies that the resulting solution is feasible for the initial problem. It is then possible to re-optimize the solution by fixing all productions at the retailer in the solution, and by regrouping productions at the manufacturer by solving a single level lot-sizing problem with the true manufacturer capacity C^1. With a greater capacity at the retailer, similarly to the previous case, Property 2 is invalidated since productions at the retailer will tend to be gathered. We can consider the same two-phase approach by setting $\bar{C}^2 = C^1$. It would be interesting to study through numerical experiments if this two-phase simple heuristic provides efficient solutions.

In Chapter 3, we have considered the M-LSP-B with a number of levels N which is part of the input, and for which deliveries are done using batches of size C, typically the capacity of a container. The number of batches which can be ordered is also limited. We have proposed an exact algorithm running in $O((N^2 + \log T)T^3)$ under some particular cost structure. This model extends Chapter 2, however, our algorithm does not work without considering non-speculative motives at each level, so the problem considered of second chapter is not a particular case of this one.

The low complexity of our algorithm encourages us to think that some of our assumptions on the parameters and the cost structure may be relaxed to deal with more general distribution networks, as we did in §3.5 when considering non-null setup costs at the first level. One direction of future researches would be to extend our algorithm by allowing backlogging of the demand, as studied by Van Vyve, 2007 on a single level. Another direction is to consider that the size of a batch may be different from one level to another. As far as we know, the status of the problem with a fixed number of different batch sizes is open. Finally, another extension would be to consider inventory bounds on the number of batches that can be stored at a level from one period to another, in addition to the limit of the number of batches one can order. Observe that under our assumptions, we have established that less than C units are carried in stock in a dominant policy at any level, except the last one. Hence, introducing an inventory bounds at the last level seems particularly relevant. Finally, one can investigate if our approach considering induced connected components can be used to approximate more general distribution networks or different structures of distribution not in series. While the class of policies admitting a box decomposition is certainly not dominant anymore, we can study if, under some assumptions, it may lead to a solution with proven performance guarantee.

In Chapter 4 we have provided NP-hardness results for M-ULSP-B when batch sizes are stationary and level-dependent, as well as for M-CLSP with level-dependent capacities. We have then proposed a 2-approximation algorithm for M-ULSP-B where batch sizes are time-dependent and level-dependent. The principle of the algorithm is to sandwich the FTL procurement cost by two affine functions. This process has been reused to approximate the M-CLSP with level-dependent capacities C^i, but under the assumptions of non-increasing setup costs and non-speculative motives.

The complexity of the latter problem is, however, an open question. It could be an interesting track for further research. The sandwich method could be reused for other problems with complex cost functions. We are currently trying to answer these questions, in collaboration with Albert Wagelmans and Wilco van den Heuvel, from Erasmus University Rotterdam.

Finally, in Chapter 5, we have studied a single-level lot-sizing problem under a periodic limit on energy consumption. The system is composed of identical machines working in parallel. At each period it must be decided how many machines to switch on or off. Turning a machine on increases the production capacity of the system, but incurs a start-up cost and consumes energy. Therefore, this shows a kind of similarity with problems with batch deliveries, except that the capacity is maintained at the subsequent periods. In this problem we consider that energy is a scarce resource, that is, in each period the amount of available energy is limited. NP-hardness results were provided under restrictive assumptions, and a polynomial time exact dynamic algorithm was proposed, in $O(M^5T^4)$ time, assuming stationary energy parameters. The model is also generalized to take in account joint setup costs, running costs and running energy consumption due to machines that remains on. In the literature, very few studies integrate energy issues into the lot sizing problem from a theoretical point of view. This work is among one of the first attempts to study such production problem.

Among the different perspectives of this work, one can investigate if efficient approximation algorithms can be developed for *energy*-LSP with time-dependent energy parameters. Exact approaches for the problem, through polyhedral studies and the design of new cuts, would also be of interest. One extension of our model is to distinguish among more than two possible states for a machine (either turned off or running in our study). In many situations, an idle machine can be put on a standby or sleep state, where it consumes less energy than in a running state, but may require an additional warm-up cost/energy consumption to resume. There can also exist different production modes on a machine, each one with a given processing speed and a given energy consumption. That is, the production capacity of a machine can be modulated with its energy consumption level. It would be interesting to see if our model can be extended to capture such different energy consumption levels/states of the machines.

More generally, energy consumption limitations could be introduced on multi-level lot-sizing problems for further research. Production lines or supply chains may be subject to periodic limitation on the amount of available energy, for instance to reduce the emission of greenhouse gases and pollution impact. The study of such problems would be an interesting perspective as a merge of the different chapters of this thesis.

Appendix A

The other possible structures of retailer subplans, §2.4.3

A.1 Subplan located at the beginning of a connected component

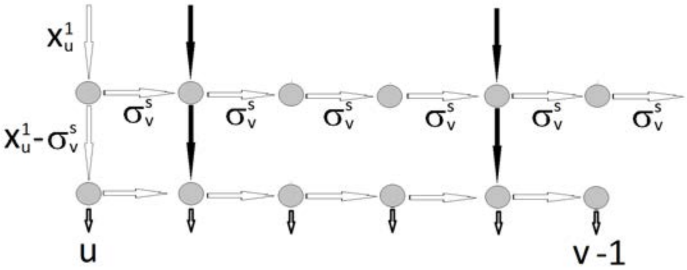

FIGURE A.1: Subplan located at the beginning of a connected component with $x_u^2 < x_u^1$

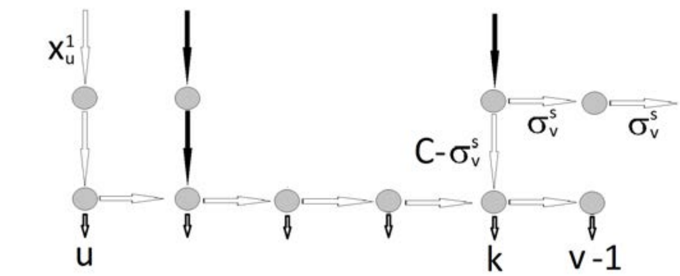

FIGURE A.2: Subplan located at the beginning of a connected component with $x_u^2 = x_u^1$

We consider a subplan (u, v) such that $u = r$ and $v - 1 < s$, that is (u, v) is the first (but not last) subplan of a connected component $[r, s]$. There are two different possible structures for such a subplan, depending on the values of x_u^1 and x_u^2.
Recall that the amount produced at the manufacturer at period u is $x_u^1 = \sigma_u^s \equiv D_{u,s} \bmod C$ and the amount of outgoing stock at period v is

$\sigma_v^s \equiv D_{v,s} \bmod C$. Notice that $x_u^2 \leq x_u^1$ and thus x_u^2 is fractional if $x_u^1 < C$. There are two possible cases for the value of x_u^2:

Case $x_u^2 < x_u^1$ (see Figure A.1). It implies that x_u^2 is fractional and $s_u^1 > 0$, whether u is a full production or not. As a result, there can't be another fractional production at the retailer between periods from $u+1$ to v due to Property 5. All other periods, from $u+1$ to v are either full production periods at both levels (due to Property 4), or periods with no production at both level (due to nested policy). Consequently the stock is maintained until v and we have $s_u^1 = \sigma_v^s$ and $x_u^2 = x_u^1 - \sigma_v^s$. Hence the amount of stock at the manufacturer is fixed so the inventory costs at the manufacturer, the production costs at period u as well as their induced inventory costs at the retailer can be directly computed. The cost $H(u,u)$ is thus equal to:

$$H(u,u) = \sigma_v^s(h_u^1 + \cdots + h_{v-1}^1) + K_u^1 + p_u^2(x_u^1) + K_u^2 + p_u^2(x_u^1 - \sigma_v^s)$$

After decrementing the demands totally or partially satisfied by the units produced in u, the problem consists in satisfying the remaining demand at a minimum cost by locating periods with full productions at both levels. It can be reduced to a discrete $CLSP$ problem and can be solved in $O(T)$.

Case $x_u^2 = x_u^1$ (see Figure A.2). It implies that $s_u^1 = 0$. As (u,v) is not the last retailer subplan of (r,s), there must be an outgoing stock at the end of the subplan. Hence a fractional production must occur at a period k located between periods $u+1$ and v. x_k^2 has to be supplied by a full production in k (Property 3), thus there is a stock at the end of period k. This stock is maintained until period v and we have $s_k^1 = \sigma_v^s$ and $x_k^2 = C - \sigma_v^s$. In a similar way to the first case, inventory costs at the manufacturer as well as the production costs in u and in k can be directly computed and the demands satisfied by units produced in u and k decremented. The cost $H(u,k)$ is thus equal to:

$$H(u,k) = \sigma_v^s(h_k + \cdots + h_{v-1}) + K_u^1 + p_u^1(x_u^1) + K_u^2 + p_u^2(x_u^1) + K_k^1 + p_k^1(C) + K_k^2 + p_k^2(C - \sigma_v^s)$$

The problem consists then in satisfying the remaining demand at a minimum cost by locating periods with full productions at both levels. It can be reduced to a discrete CLSP problem and can be solved in $O(T)$. To find the value of k which minimize the cost of (u,v), the subplan have to be evaluated for values of k from $u+1$ to v. It can be done in linear time with the algorithm of van Hoesel and Wagelmans (1996) once the subplan has been evaluated for a given value of k.

To find the minimum cost of a retailer subplan (u,v) located at the beginning of a connected component, the algorithm pick the minimum value of the optimal subplans found for the two possible structures presented above. Hence the overall total complexity is in $O(T)$.

Subplan located at the end of a connected component

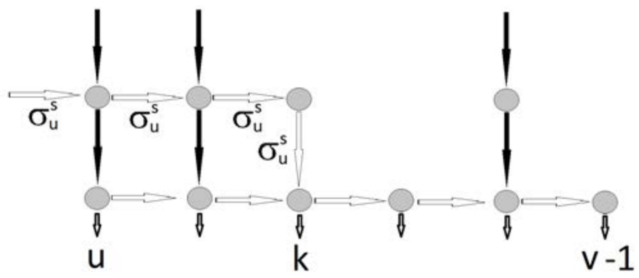

FIGURE A.3: Subplan located at the end of a connected component

We consider the last (but not first) subplan (u, v) of a connected component $[r, s]$, with $r < u$ and $v - 1 = s$. Such a subplan has an entering stock σ_u^s at the manufacturer in u and no outgoing stock in $v-1$ (see Figure A.3). Since $\sigma_u^s \neq \sigma_v^s$, it follows that a fractional retailer production must occur at a period k located between u and $v - 1$ and we have $x_k^2 = \sigma_u^s$. There can't be a second fractional production period at the retailer on the subplan because it would create a positive inventory which would necessarily be maintained until the end of the subplan while the outgoing stock must be null. Once again fixed costs are immediately determined and demands satisfied by units produced in k are decremented. The cost $H(k, k)$ is thus equal to:

$$H(k, k) = \sigma_u^s(h_u^1 + \cdots + h_{k-1}^1) + K_k^2 + p_k^2(\sigma_u^s)$$

The problem consists then in satisfying the remaining demand at a minimum cost by locating periods with full productions at both levels which can be reduced to a discrete CLSP problem and can be solved in $O(T)$.
To find the value of k which minimize the cost of (u, v), the subplan have to be evaluated for values of k from u to v. It can be done in linear time with the algorithm of van Hoesel and Wagelmans (1996) once the subplan has been evaluated for a given value of k.
Notice that such a subplan can be treated as a an inside subplan, that is a subplan with positive entering and outgoing stock, by simply stating that $\sigma_v^s = 0$.

Subplan forming an isolated connected component

We consider a retailer subplan forming an isolated connected component, with $u = r$ and $v - 1 = s$. Such a subplan has neither entering stock in u nor outgoing stock in $v - 1$ (see Figure A.4). There are fractional productions at both levels in u (except if $D_{u,v} \bmod C \equiv 0$), which are $x_u^1 = x_u^2 \equiv D_{u,v} \bmod C$ and the other productions are full productions at both levels and are placed

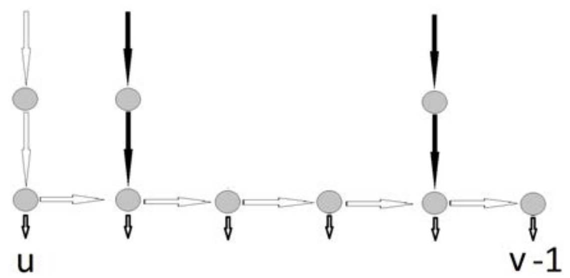

FIGURE A.4: Retailer subplan forming an isolated connected component

according to the greedy algorithm from van Hoesel in $O(T)$. The cost $H(u,u)$ is thus equal to:

$$H(k,k) = K_u^1 + p_u^1(x_u^1) + K_u^2 + p_u^2(x_u^1)$$

Another fractional production period at the retailer would create a positive inventory which would necessarily be maintained until the end of the subplan while the outgoing stock must be null.
Notice that such a subplan can be treated as a an inside subplan, that is a subplan with positive entering and outgoing stock, by simply stating that $\sigma_u^s = 0 = \sigma_v^s = 0$.

Appendix B

Proof of the dominance of FUFD policies, §3.2

In this appendix, we prove that policies obeying Properties 7, 8 and 9 are dominant for problem M-LSP-B. Consider an optimal policy π observing Property 7. We have to establish that we can choose π such that, for any level $i < N$ and for any period $t > 1$, the two following conditions hold:

$(C1)$ $s^i_{t-1} < C$

$(C2)$ $(x^i_t \bmod C) \neq 0 \;\Rightarrow\; (x^{i+1}_t \bmod C) \neq 0$

As noticed in §3.2, the fact that $\lfloor x^i_t \rfloor_C \geq \lfloor x^{i+1}_t \rfloor_C$ is then a direct consequence of Property 7 and of the upper bound given by Condition $(C1)$ on the stock level. Assume that policy π does not satisfy Property 8 or Property 9. Let t be the last period such that one of the two conditions $(C1)$ or $(C2)$ is violated, and let $i \in \{1, ..., N-1\}$ be the most upstream level such that $s^i_{t-1} \geq C$ or $x^i_t \pmod C \neq 0$ and $x^{i+1}_t \pmod C = 0$. Observe that both cases cannot occur simultaneously, since $s^i_{t-1} \geq C$ obviously implies that there is an entering stock at node (i, t), and $x^i_t \pmod C \neq 0$ means by definition that (i, t) is a fractional ordering period. As a consequence, since policy π satisfies Property 7, at least one of the two conditions $(C1)$ and $(C2)$ are verified in each node. We distinguish between 2 cases, depending if Condition $(C1)$ or Condition $(C2)$ is violated at node (i, t).

Case 1. Assume that Condition $(C1)$ is violated at (i, t), that is, we have $s^i_{t-1} \geq C$. Firstly, we show that there exists a path between the source node $(0, 0)$ and node (i, t), going forward in time, such that the flow on this path is always greater than or equal to C. The units in stock entering node (i, t) must have been ordered at some previous periods at level i, since we assume no initial inventory. Let $t' < t$ be the last ordering period at level i before t. Since no units are ordered between t' and t, we clearly have $s^i_{t'} \geq s^i_{t-1} \geq C$. In particular, the in-going flows of node (i, t') sum-up to at least C units. Observe that if only a fractional batch is ordered at period t' ($x^i_{t'} < C$), the entering stock at (i, t') need to be positive to obtain an outgoing stock of C or more units, contradicting Property 7. Hence, we can assert that at least one full batch is ordered at node (i, t'). As a result, at least C units must enter node $(i-1, t')$. In the same way, the last ordering period $t'' \leq t'$ at level $i-1$

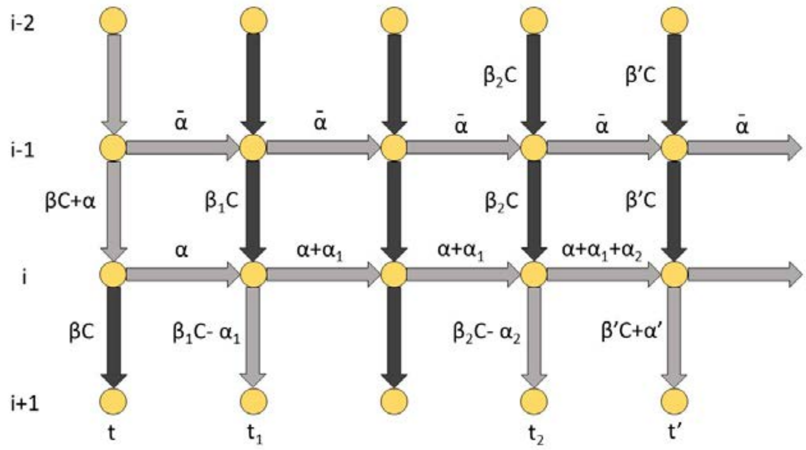

FIGURE B.1: Initial solution π: the nested property is violated in (i,t)

orders at least one full batch. We can exhibit this way, going up till level 1, a path σ with at least C units of flow rooted on each arc in solution π.

Secondly, we show that part of this flow can be rerooted, without increasing the total cost of the policy. We consider the alternate policy π' where C units of flow along path σ are rerouted on path $((0,0),(1,t),\dots,(i,t))$. It corresponds to postpone the ordering of one full batch along σ to period t at each level $j=1,\dots,i$. Due to our assumptions $(A1)$ on echelon non-speculative motives and $(A2)$ on the monotony of the fixed cost per batch, see §3.1, the cost of policy π' cannot be greater than the cost of policy π. We must check that policy π' is feasible, which boils down to verify that the number of batches ordered at period t at a level $j \le i$ does not exceed the upper bound m_t^j. Consider policy π. Due to our choice of period t, we have $s_{t-1}^i \ge C$ and $s_t^i < C$. It results that $x_t^{i+1} > x_t^i$. Moreover, due to Property 7, period t is a full ordering period at level i. Hence, we have $x_t^i = \beta C$ for some integer β. The conservation of the flows at node (i,t) implies that level $i+1$ orders at least one more batch than level i at period t, and thus we have $m_t^{i+1} \ge \beta + 1$. Due to our assumption $(A3)$, it results that $m_t^j \ge \beta+1$ for all $j=1,\dots,i$. Now, due to our choice of i, all the nodes (j,t) for $j<i$ verify conditions $(C1)$ and $(C2)$, and have an outgoing stock lower than C. As a consequence, only full batches are ordered at all the upstream levels, and we have $x_t^j \le \beta C$. We can conclude that one additional batch can be ordered at each level without violating the constraint on the maximum number of batches.

Hence, we have obtained a feasible policy π', of cost at most the cost of π. Observe that policy π' still obeys Property 7 since only full batches have been postponed. Also observe that policy π' is identical to π for periods subsequent to t, and thus conditions $(C1)$ and $(C2)$ are verified in policy π' by all the nodes after period t and by all the nodes at period t at the levels $\{1,\dots,i\}$, level i included.

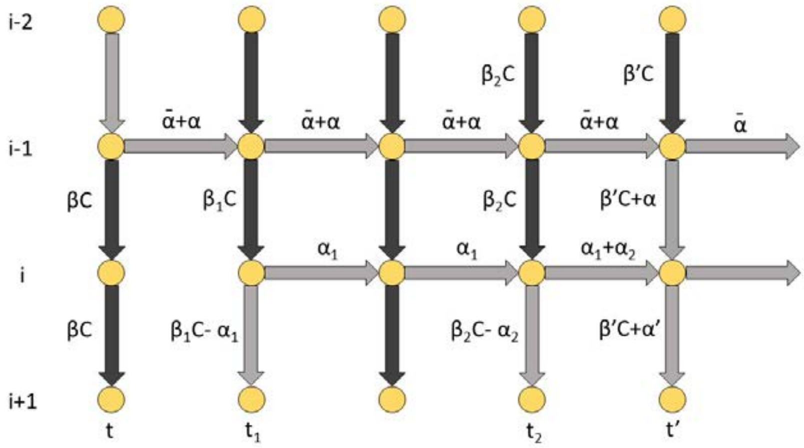

FIGURE B.2: Solution π_1: the fractional batch in (i, t) is shifted to period t'. Property 7 becomes violated in (i, t')

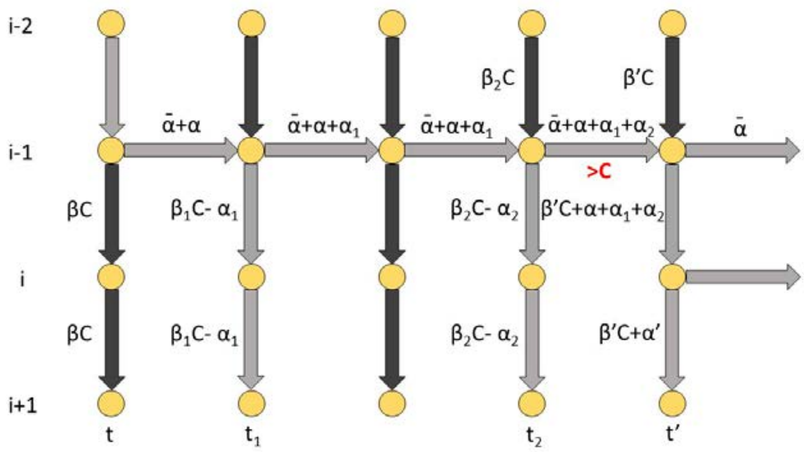

FIGURE B.3: Solution π_2: part of the orders at level i and periods t_1 and t_2 are shifted to period t'. The entering stock in (i, t') vanishes.

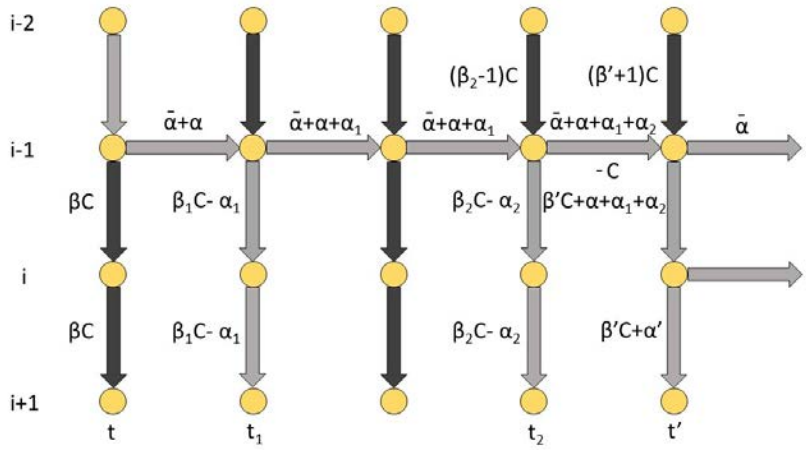

FIGURE B.4: Solution π_3: the order of one full batch is postponed from t_2 to t' at levels $(1, ..., i-1)$

Case 2. Assume that Condition $(C2)$ is violated at (i, t), that is, $x_t^i \bmod C > 0$ and $x_t^{i+1} \bmod C = 0$ (see Figure B.1). We can write $x_t^i = \beta C + \alpha$, with $\beta \in \mathbb{Z}^+$ and $0 < \alpha < C$, which corresponds to set $\beta = \lfloor x_t^i \rfloor_C$ and $\alpha = x_t^i \bmod C$. On one hand, due Property 7, there is no entering stock in (i, t). Hence, the conservation of the flow at node (i, t) implies that $x_t^{i+1} \leq x_t^i$. On the other hand, due to our choice of t, we have $s_t^i < C$, which implies that $x_t^{i+1} > x_t^i - C$. Since period t is a full ordering period at level $i+1$, we thus have $x_t^{i+1} = \beta C$ and $s_t^i = \alpha$.

Consider the first period $t' > t$ such that $x_{t'}^i < x_{t'}^{i+1}$. Observe that for all periods τ, $t < \tau < t'$, the entering stock in (i, τ) is lower than or equal to the outgoing stock. As $s_t^i = \alpha$, this implies that the flow on the inventory arcs between periods t and t' is greater than or equal to α. In particular, $s_{t'-1}^i > 0$, which implies due to Property 7 that the ordering in (i, t') is not fractional, that is, $x_{t'}^i = \beta' C$, with $\beta' \in \mathbb{Z}^+$. Since $x_{t'}^i < x_{t'}^{i+1}$ by our choice of t', one more batch is ordered in $(i+1, t')$ compared to (i, t'). As a consequence, we have $m_{t'}^i \geq m_{t'}^{i+1} \geq \beta' + 1$: It is thus possible to order an additional batch at node (i, t'). We consider the alternate policy π_1 where the ordering of the fractional batch of α units is postponed from node (i, t) to node (i, t'). It consists in rerouting α units of flow on the inventory arcs between periods t and t' from level i to its upstream level $i-1$. The limit of number of batches is respected and thus policy π_1 is feasible. Moreover, since echelon non-speculative motives are assumed at level i and setup costs are non-increasing in time, this postponing cannot increase the cost of the solution.

However, Property 7 may be violated in policy π_1 at node (i, t'), since a fractional batch is now ordered at this node. On the contrary, notice that Corollary 1 is still satisfied, as the order $x_{t'}^{i+1}$ at node $(i+1, t')$ is fractional. To ensure that Property 7 still holds for all periods $\tau > t$, solution π' has to be modified. Consider the initial solution π and let $\{t_1, t_2, .., t_K\}$ be the set of periods located between t and t' such that more units are ordered at level

i than at level $i+1$. In Figure B.1, we have a set of periods $\{t_1, t_2\}$. Notice that periods t_k's correspond to all the ordering periods between t and t' at level i, except the ones for which the same amount is ordered at both levels, which do not modify the inventory level. Due to Property 7, $\{t_1, t_2, .., t_K\}$ are full ordering periods at levels i and fractional ordering periods at level $i+1$, since the difference in the quantity ordered at two consecutive levels at a given period cannot exceed $C-1$ units (Property 8). Hence, for each period $\tau \in \{t_1, t_2, .., t_K\}$, we can assume that $x_\tau^{i+1} = \beta_\tau C - \alpha_\tau$, with $\beta_\tau \in \mathbb{Z}^+$ and $0 < \alpha_\tau < C$. Consequently, an amount of stock α_τ is stored at the end of period τ, and these units are kept in stock until period t'. As a result, if set $\{t_1, t_2, .., t_K\}$ is not empty, Property 7 is violated in π' at node (i, t'), where there is a positive entering stock and a fractional order (see Figure B.2). We consider the alternate solution which postpones the ordering at periods (i, τ) of the α_τ units for all $\tau \in \{t_1, t_2, .., t_K\}$, to order them in (i, t'). Let π_2 denotes this solution (see Figure B.3). After these shifts, we have $x_{t'}^i = \beta' C + \alpha + \alpha_{t_1} + \alpha_{t_2} + ... + \alpha_{t_K}$ in policy π_2. By construction, the amount $\alpha + \alpha_{t_1} + \alpha_{t_2} + ... + \alpha_{t_K}$ is equal to the entering stock level $s_{t'-1}^i$ in policy π : Due to Property 8, this amount is thus lower than C. Consequently, at most $C-1$ additional units are ordered at node (i, t') in policy π_2, and the solution remains feasible. Properties 7 is now satisfied at period t'.

It remains to ensure that Properties 7 and 8 are still satisfied by policy π_2 at level $i-1$ between periods t and t', where the stock level has increased. Firstly, observe that there is no fractional ordering in policy π between periods $t+1$ and t' at level $i-1$. Indeed, a fractional ordering at level $i-1$ would imply, due to Corollary 1, a fractional ordering at level i, which violates Property 7. Consequently, Property 7 is satisfied in π_2 at level $i-1$. Secondly, let us turn our attention to Property 8 at level $i-1$. In policy π_2, the stock level between t and t_1 has increased by α units, between t_1 and t_2 by $\alpha + \alpha_1$ units, and so on until $(i-1, t')$ for which the entering stock has increased by $\alpha + \alpha_{t_1} + \alpha_{t_2} + ... + \alpha_{t_K}$ units. Let us denote by $\overline{\alpha}$ the stock level s_t^{i-1}, outgoing from note $(i-1, t)$, in policy π. Since only full ordering takes place at levels $i-1$ and i between periods t and t', the amount of units in stock does not evolve in policy π, and remains equal to $\overline{\alpha}$ (see Figure B.1). As a result, the amount of inventory in π_2 at level $i-1$ is non-decreasing between t and t', and goes from $\overline{\alpha}$ in t to $\overline{\alpha} + \alpha + \alpha_{t_1} + \alpha_{t_2} + ... + \alpha_{t_K}$. Since it corresponds to values of the stock level in π at periods later than t, we have $\overline{\alpha} < C$ and $\alpha + \alpha_{t_1} + \alpha_{t_2} + ... + \alpha_{t_K} < C$ (Property 8). It can therefore be argued that $\overline{\alpha} + \alpha + \alpha_{t_1} + \alpha_{t_2} + ... + \alpha_{t_K} < 2C$. If this value is in fact lower than C, Property 8 is satisfied at level $i-1$ in policy π_2. Otherwise, let Z be the last period such that its entering stock is lower than C, and its outgoing stock is greater than or equal to C. Notice that Z is necessarily one of the periods t_k. We again modify the policy by postponing the ordering of a full batch from period Z to period t' at level i, as well as at each higher levels $1, \ldots, i-2$. In the example, we have $s_{t_2-1}^{i-1} < C$ and $s_{t_2}^{i-1} \geq C$, thus the ordering of one full batch is postponed from period t_2 to period t' at levels $1, \ldots, i-1$, see Figure B.4. In the new solution, denoted by π_3, the stock level between period t and period t' is now always lower than C at level $i-1$.

Notice that node (i, t) orders only full batches in the final policy. If node $(i-1, t)$ happens to order a fractional batch, the nested property is now violated at this node. However, the process can be reiterated, as well as at the higher levels, until the solution observes the nested property at period t at levels $1, \ldots, i$.

Once the appropriated case has been resolved, the resulting solution still observes Property 7 over the entire time horizon. Moreover, Property 8 and Corollary 1 are now satisfied at period t at levels $1, \ldots, i$ (and at subsequent periods). This process can thus be repeated at period t at each lower levels which doesn't exhibit Property 8 and Corollary 1, and then at earlier periods. The resulting solution is optimal and observes Properties 7 and 8 and Corollary 1.

Bibliography

Aggarwal, A. and J.K. Park (1993). "Improved algorithms for economic lot-size problems". In: *Operations Research* 41.3, pp. 549–571. DOI: 10.1287/opre.41.3.549.

Ahmed, S. et al. (2016). "On the computational complexity of minimum-concave-cost-flow in a two-dimensional grid". In: *SIAM Journal on Optimization To appear*. URL: http://arxiv.org/abs/1602.08515.

Ahuja, R.K., T.L. Magnanti, and J.B. Orlin (1993). *Network Flows: Theory, Algorithms, and Applications*. 1st edition. Prentice Hall.

Akbalik, A. and C. Rapine (2012). "Polynomial time algorithms for constant capacitated single-item lot sizing problem with stepwise production cost". In: *Operations Research Letters* 40, pp. 390–397. DOI: 10.1016/j.orl.2012.05.003.

— (2013). "The single item uncapacitated lot-sizing problem with time-dependent batch sizes: NP-hard and polynomial cases". In: *European Journal of Operational Research* 229.2, pp. 353–363. DOI: 10.1016/j.ejor.2013.02.052.

Artigues, C., P. Lopez, and A. Haït (2013). "The energy scheduling problem: industrial case-study and constraint propagation techniques". In: *International Journal of Production Economics* 143, pp. 13–23. DOI: 10.1016/j.ijpe.2010.09.030.

Biel, K. and C.H. Glock (2016). "Systematic literature review of decision support models for energy-efficient production planning". In: *Computers and Industrial Engineering* 101, pp. 243–259. DOI: 10.1016/j.cie.2016.08.021.

Bitran, G.R. and H.H. Yanasse (1982a). "Computational complexity of the capacitated lot size problem". In: *Management Science* 28.10, pp. 1174–1186. DOI: 10.1287/mnsc.28.10.1174.

— (1982b). "Computational complexity of the capacitated lot size problem". In: *Management Science* 28.10, pp. 1174–1186. DOI: 10.1287/mnsc.28.10.1174.

Brahimi, N. et al. (2006). "Single item lot sizing problems". In: *European Journal of Operational Research* 168.1, pp. 1–16.

Brahimi, N. et al. (2017). "Single-Item Dynamic Lot-Sizing Problems: An Updated Survey". In: *European Journal of Operational Research*. DOI: 10.1016/j.ejor.2017.05.008.

Chan, L.M.A. et al. (2002). "Effective Zero-Inventory-Ordering Policies for the Single-Warehouse Multiretailer Problem with Piecewise Linear Cost Structures". In: *Management Science* 48.11, pp. 1446–1460. DOI: 10.1287/mnsc.48.11.1446.267.

Chung, C.S. and C.H.M. Lin (1988). "An $O(T^2)$ algorithm for the NI/G/NI/ND capacitated lot size problem". In: *Management Science* 34.3, pp. 420–426. DOI: 10.1287/mnsc.34.3.420.

Ding, J.-Y. et al. (2016). "Parallel machine scheduling under time-of-use electricity prices: New models and optimization approaches". In: *IEEE Transactions on Automation Science and Engineering* 13.2, pp. 1138–1154. DOI: 10.1109/TASE.2015.2495328.

Federgruen, A. and M. Tzur (1991). "A simple forward algorithm to solve general dynamic lot-sizing models with n periods in $O(n \log n)$ or $O(n)$ time". In: *Management Science* 37.8, pp. 909–925. DOI: 10.1287/mnsc.37.8.909.

Florian, M. and M. Klein (1971). "Deterministic Production Planning with Concave Costs and Capacity Constraints". In: *Management Science* 18.1, pp. 12–20. DOI: 10.1287/mnsc.18.1.12.

Florian, M., J.K. Lenstra, and A.H.G. Rinnooy Kan (1980a). "Deterministic production planning: algorithms and complexity". In: *Management Science* 26.7, pp. 669–679. DOI: 10.1287/mnsc.26.7.669.

— (1980b). "Deterministic production planning: Algorithms and complexity". In: *Management Science* 26.7, pp. 669–679. DOI: 10.1287/mnsc.26.7.669.

Gahm, C. et al. (2016). "Energy-efficient scheduling in manufacturing companies: A review and research framework". In: *European Journal of Operational Research* 248, pp. 744–757. DOI: 10.1016/j.ejor.2015.07.017.

Gayon, J.-P. et al. (2017). "Fast approximation algorithms for the One-Warehouse Multi-Retailer problem under general cost structures and capacity constraints". In: *Math. Oper. Res. To appear*.

Giglio, D., M. Paolucci, and A. Roshani (2017). "Integrated lot sizing and energy-efficient job shop scheduling problem in manufacturing/remanufacturing systems". In: *Journal of Cleaner Production* 148, pp. 624–641. DOI: 10.1016/j.jclepro.2017.01.166.

Goisque, G. and C. Rapine (2017a). "An efficient algorithm for the 2-level capacitated lot-sizing problem with identical capacities at both levels". In: *European Journal of Operational Research* 261.3, pp. 918 –928. DOI: 10.1016/j.ejor.2017.02.024.

— (2017b). "The multi-level in series lot-sizing problem with batch deliveries". In: *Operations Research*.

Hwang, H.-C., H. Ahn, and P. Kaminsky (2013). "Basis Paths and a Polynomial Algorithm for the Multistage Production-Capacitated Lot-Sizing Problem". In: *Operations Research* 61.2, pp. 469–482. DOI: 10.1287/opre.1120.1141.

Hwang, H.-C., H.-S. Ahn, and P. Kaminsky (2016). "Algorithms for the two-stage production-capacitated lot-sizing problem". In: *Journal of Global Optimization* 65.4, pp. 777–799. DOI: 10.1007/s10898-015-0392-2.

Hwang, H.-C. et al. (2013). "Solving a multi-level lot-sizing problem with inventory bounds". In: *International Workshop on Lot Sizing (IWLS)*. Hannover, Germany.

Kaminsky, P. and D. Simchi-Levi (2003). "Production and distribution lot sizing in a two stage supply chain". In: *IIE Transactions* 35.11, pp. 1065–1075. DOI: 10.1080/07408170304401.

Karmarkar, U., S. Kekre, and S. Kekre (1987). "The Dynamic Lot-Sizing Problem with Startup and Reservation Costs". In: *Operations Research* 35.3, pp. 389–398. DOI: 10.1287/opre.35.3.389.

Levi, R. et al. (2008). "A Constant Approximation Algorithm for the One-Warehouse Multiretailer Problem". In: *Management Science* 54.4, pp. 763–776. DOI: 10.1287/mnsc.1070.0781.

Li, C.-L., V.N. Hsu, and W.-Q. Xiao (2004). "Dynamic lot sizing with batch ordering and truckload discounts". In: *Oper. Res.* 52.4, pp. 639–654.

Masmoudi, O. et al. (2017). "Lot-sizing in a multi-stage flow line production system with energy consideration". In: *International Journal of Production Research* 55.6, pp. 1640–1663. DOI: 10.1080/00207543.2016.1206670.

Melo, R.A. and L.A Wolsey (2010). "Uncapacitated two-level lot-sizing". In: *Operations Research Letters* 38.4, pp. 241–245. DOI: 10.1016/j.orl.2010.04.001.

Muckstadt, J. and R. Roundy (1993). "Chapter 2 Analysis of multistage production systems". In: *Handbooks in Operations Research and Management Science* 4, pp. 59–131. DOI: 10.1016/S0927-0507(05)80182-3.

Nattaf, M., C. Artigues, and P. Lopez (2015). "A hybrid exact method for a scheduling problem with a continuous resource and energy constraints". In: *Constraints* 20.3, pp. 304–324. DOI: 10.1007/s10601-015-9192-z.

Nattaf, M. et al. (2016). "Energetic reasoning and mixed-integer linear programming for scheduling with a continuous resource and linear efficiency functions". In: *OR Spectrum* 38.2, pp. 459–492. DOI: 10.1007/s00291-015-0423-x.

Ngueveu, S.U., C. Artigues, and P. Lopez (2016). "Scheduling under a non-reversible energy source: An application of piecewise linear bounding of non-linear demand/cost functions". In: *Discrete Applied Mathematics* 208, pp. 98–113. DOI: 10.1016/j.dam.2016.03.001.

Phouratsamay, S.L., S. Kedad-Sidhoum, and F. Pascual (2016). "Two-level lot-sizing with inventory bounds". In: *Computing Research Repository* abs/1604.02278. URL: http://arxiv.org/abs/1604.02278.

Pochet, Y. and L.A. Wolsey (2006). *Production Planning by Mixed Integer Programming*. 1st edition. Springer-Verlag New York. DOI: 10.1007/0-387-33477-7.

Rapine, C. et al. (2016a). "Capacity acquisition for the single-item lot sizing problem under energy constraints". In: *Roadef 2016*. Compiègne, France. URL: https://hal.archives-ouvertes.fr/hal-01420958.

— (2016b). "Lot sizing problem with energy constraints". In: *International Workshop on Lot Sizing (IWLS)*. Hannover, Germany.

Roundy, R. (1985). "98%-Effective Integer-Ratio Lot-Sizing for One-Warehouse Multi-Retailer Systems". In: *Management Science* 31.11, pp. 1416–1430. DOI: 10.1287/mnsc.31.11.1416.

Santos, M.O. and B. Almada-Lobo (2012). "Integrated pulp and paper mill planning and scheduling". In: *Computers & Industrial Engineering* 63, pp. 1–12. DOI: 10.1016/j.cie.2012.01.008.

Sargut, F.Z. and H.E. Romeijn (2007). "Capacitated production and subcontracting in a serial supply chain". In: *IIE Transactions* 39.11, pp. 1031–1043. DOI: 10.1080/07408170601091899.

Schultz, C., P. Sellmaier, and G. Reinhart (2015). "An approach for energy-oriented production control using energy flexibility". In: *Procedia CIRP* 29, pp. 197–202. DOI: 10.1016/j.procir.2015.02.038.

Tang, L., P. Che, and J. Liu (2012). "A stochastic production planning problem with nonlinear cost". In: *Computers & Operations Research* 39, pp. 1977–1987. DOI: 10.1016/j.cor.2011.09.007.

Uzel, E. (2004). "A mathematical modeling approach to energy cost saving in manufacturing plant". MA thesis. İzmir Institute of Technology.

Van Hoesel, C.P.M. and A.P.M. Wagelmans (1996). "An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities". In: *Management Science* 42.1, pp. 142–150. DOI: 10.1287/mnsc.42.1.142.

Van Hoesel, S. et al. (2005). "Integrated lot-sizing in serial supply chains with production capacities". In: *Management Science* 51.11, pp. 1706–1719. DOI: 10.1287/mnsc.1050.0378.

Van Vyve, M. (2007). "Algorithms for Single-Item Lot-Sizing Problems with Constant Batch Size". In: *Mathematics of Operations Research* 32.3, pp. 594–613. DOI: 10.1287/moor.1070.0257.

Van Vyve, M., L. Wolsey, and H. Yaman (2014). "Relaxations for two-level multi-item lot-sizing problems". In: *Mathematical Programming* 146.1, pp. 495–523. DOI: 10.1007/s10107-013-0702-8.

Wagelmans, A., S. Van Hoesel, and A. Kolen (1992). "Economic lot sizing: An $O(n \log n)$ algorithm that runs in linear time in the Wagner-Whitin case". In: *Operations Research* 40.1, pp. 145–156. DOI: 10.1287/opre.40.1.S145.

Wagner, H.M. and T.M. Whitin (1958). "Dynamic version of the economic lot size model". In: *Management Science* 5.1, pp. 89–96. DOI: 10.1287/mnsc.5.1.89.

Waldemarsson, M., H. Lidestam, and Rudberg M. (2013). "Including energy in supply chain planning at a pulp company". In: *Applied Energy* 112, pp. 1056–1065. DOI: 10.1016/j.apenergy.2012.12.032.

Zangwill, W.I. (1969). "A backlogging model and a multi-echelon model of a dynamic economic lot size production system—A network approach". In: *Management Science* 15.9, pp. 506–527. DOI: 10.1287/mnsc.15.9.506.

— (2013). "Minimum concave cost flows in certain networks". In: *Management Science* 14.7, pp. 429–450. DOI: 10.1287/mnsc.14.7.429.

Zhang, M., S. Küçükyavuz, and H. Yaman (2012). "A Polyhedral Study of Multiechelon Lot Sizing with Intermediate Demands". In: *Operations Research* 60.4, pp. 918–935. DOI: 10.1287/opre.1120.1058.

Zhao, H., M.G. Ierapetritou, and G. Rong (2016). "Production planning optimization of an ethylene plant considering process operation and energy utilization”. In: *Computers and Chemical Engineering* 87, pp. 1–12. DOI: 10.1016/j.compchemeng.2016.01.002.

Özdamar, L. and S.I. Birbil (1999). “A hierarchical planning system for energy intensive production environments”. In: *International Journal of Production Economics* 58, pp. 115–129. DOI: 10.1016/S0925-5273(98)00076-0.

Abstract

In this thesis we are interested in several multi-level lot-sizing problems taking into account production capacities. We first study a 2-level in series lot-sizing problem with identical and stationary capacities at both levels, for which we propose an exact dynamic algorithm running in polynomial time under some hypothesis. Next chapter extends this result on two main lines: we consider the multi-level in series lot-sizing problem with batch deliveries and with a number of level which is part of the input. We provide a very efficient exact algorithm for this problem, which is polynomial in the number of levels and in the number of periods, based on an original decomposition into induced connected components. Then, we consider more general versions of this problem, for which we provide NP-hardness results when batch sizes or capacities are level-dependent. We propose 2-approximation algorithms for these problems, based on the sandwiching of the objective function by two affine functions. Finally, we study a single-level lot-sizing problem in a system composed of identical machines working in parallel. The originality of this study is to consider a periodic energy limitation. At each period it must be decided how many machines to switch on or off and the volume to be produced and stored. Complexity results are provided, showing that this problem is NP-hard, even under some restrictive assumptions, and an exact dynamic algorithm running in polynomial time is proposed for the case of stationary energy parameters.

Résumé

Dans cette thèse nous nous intéressons à plusieurs problèmes de gestion de stocks, à travers des modèles de dimensionnement de lots sur plusieurs niveaux, en tenant compte de capacités de production. Nous étudions tout d'abord un problème de dimensionnement de lots à deux niveaux en série avec des capacités de production identiques et stationnaires aux deux niveaux, pour lequel proposons un algorithme dynamique exact pouvant résoudre le problème en temps polynomial sous certaines hypothèses. Dans le chapitre suivant nous étendons ce résultat dans deux directions : nous considérons le problème de gestion de stocks sur un nombre quelconque de niveaux en série, et nous considérons des livraisons par lots. Nous présentons un algorithme exact de résolution, polynomial et très efficace, basé sur une décomposition originale en composantes connexes induites. Nous considérons ensuite des versions plus générales de ce problème, en établissant des résultats de NP-complétude lorsque chaque niveau à une capacité ou une taille de lot différentes. Nous proposons pour ces problèmes une 2-approximation, basé sur l'encadrement de la fonction objectif par deux fonctions affines. Pour finir nous étudions un problème sur un seul niveau mais dans un système de production composé de machines identiques fonctionnant en parallèle. L'originalité de ce problème est de considérer une limitation de la consommation énergétique. A chaque période, on doit décider combien de machines allumer ou éteindre, et quel volume produire et stocker. Des résultats de complexité sont proposés, montrant que ce problème est NP-difficile même sous des hypothèses fortes, et un algorithme dynamique exact est présenté pour le cas de paramètres d'énergie stationnaires.