Université catholique de Louvain
École Polytechnique de Louvain

Mixed Integer Programming Formulations and Heuristics for Joint Production and Transportation Problems

Thèse présentée en vue de l'obtention du grade de
Docteur en Sciences de l'Ingénieur par

Rafael Augusto de Melo

NOVEMBER 2011

Promoteur: Laurence WOLSEY (UCL)
Jury: Per AGRELL (Président, UCL)
Philippe CHEVALIER (UCL)
François GLINEUR (UCL)
Mathieu VAN VYVE (UCL)
Albert WAGELMANS (Erasmus University Rotterdam)

Acknowledgments

I would like to express my gratitude to my advisor Laurence Wolsey for giving me the privilege and for providing me with the means to be his doctorate student. I really appreciate his readiness and willingness to share his knowledge. His kindness is admirable.

I would like to thank the members of my jury Per Agrell, Philippe Chevalier, François Glineur, Mathieu Van Vyve and Albert Wagelmans for their valuable comments which helped to improve this thesis.

I am grateful to Sebastián Urrutia and Celso Ribeiro who encouraged me to pursue a doctorate.

My appreciation to everyone at CORE for the very nice atmosphere. My special thanks to Jean-Sebastién, Santanu, Robert, Joël, Xenia, Peter, Nicolas, Joachim, Olivier, Gauthier, Gilles, Maia, Besik, Dzianis, Anton, Salome, Filomena, Paolo, Ying, Alejandro, Tanja, Hiroshi, Grégory, Margherita, Vlad, Stéphane, Claudia, Mikel, Abdel, Mady, Catherine, Raphaël and Laurent. I am sorry for those I did not remember the name while writing these acknowledgments but my memory is unfortunatelly limited.

I am very thankful to all my friends and family. A very special thanks to Dorottya for all the support and motivation during my last year of studies.

This thesis is dedicated to my parents and my brother.

Abbreviations

DO	Divisible order
FCN	Fixed charge network
GENP	General problem
IO	Indivisible order
JRP	Joint-replenishment problem
JRP-B-ST	Joint-replenishment problem with backlogging and start-up times
LS-U	Uncapacitated lot-sizing problem
LS-C	Capacitated lot-sizing problem
LS-CC	Constant capacitated lot-sizing problem
MC	Multi-commodity
MIP	Mixed integer programming
MWMR-PC-SC0	Multi-warehouse multi-retailer problem without client storage
OWMR	One-warehouse multi-retailer problem
SP	Shortest path
TR	Transportation
TU	Totally unimodular
WW-C	Capacitated Wagner-Whitin lot-sizing problem
WW-CC	Constant capacitated Wagner-Whitin lot-sizing problem
WW-U	Uncapacitated Wagner-Whitin lot-sizing problem
2L-MP-U	Uncapacitated two-level problem with multiple production sites
2L-PT	Two-level production/transportation problem
2L-S/LS-U	Uncapacitated two-level production-in-series lot-sizing problem
2L-S/LS-U-SL	Uncapacitated two-level production-in-series lot-sizing with sales problem

Contents

Chapter 1

Introduction

In this thesis we study different integrated production and transportation problems.

Production planning problems consist in determining when and how much to produce in a production site in order to satisfy certain demands. We deal here only with problems with deterministic time varying demand, meaning that the demand is known beforehand and it varies over a discrete set of time periods.

Supply chain problems are more general problems. In supply chain problems the places in which production takes place (production sites) and the places in which these products are required (clients) are often in different localities. The transportation decision in these problems consists in determining how much to transport in each time period. Therefore the production of goods and their transportation to the customers (or clients) is a common problem faced by many industries and is an essential feature of a supply chain. Often such problems are treated in two separate stages, first a production plan is elaborated and then based on it a transportation plan is created. However, this approach can generate solutions that are far from optimal. In recent years, studies have shown that considerable gains can be obtained when production and transportation are planned in an integrated manner. These integrated production and transportation problems can be seen as two-level problems in which one level represents the production while the other represents the transportation.

Dynamic programming algorithms have been developed for the small number of polynomially solvable problems, but heuristics are among the most used approaches for most of the problems with multiple production sites or clients that are usually NP-Hard. Mixed integer programming has been applied successfully over the years to solve production planning problems mainly through the use of cutting plane algorithms and reformulations. For supply chain problems, on the other hand, much less effort has been made to use mixed integer programming approaches.

We consider several different problems in this thesis. We first study the simplest two-level problem, namely the uncapacitated two-level production-in-series lot-sizing problem that is polynomially solvable. Some NP-Hard extensions are also studied in the subsequent chapters. An extension with multiple clients, the one-warehouse multi-retailer problem is considered next. This in turn leads us to a more general problem with multiple production sites and multiple clients. Finally, a slightly different production and transportation model with orders having different sizes and delivery dates is considered.

For each problem studied the goal is to analyze the structure of the solutions and develop solution methods based on mixed integer programming techniques. We aim at the development of algorithmic methods to solve these problems to optimality whenever possible. In case we cannot solve the problem to optimality the aim is to try to generate good heuristic solutions with some guarantee of the quality of these solutions.

We now give an outline of the thesis. In Chapter 2 we introduce some concepts that will help the reader to understand the results obtained. Definitions and results in mixed integer programming and some results for the simplest single-level production planning model (the uncapacitated lot-sizing problem) are presented along with some definitions and formulations for a variety of supply chain problems appearing in the literature.

In Chapter 3 we study the simplest possible two-level problem, the uncapacitated two-level production-in-series lot-sizing problem. It is known that the problem can be solved in $O(NT^3)$ and the "smallest" extended formulation solving the problem has $O(NT^4)$ variables and $O(NT^3)$ constraints where NT is the number of periods. We propose a new dynamic programming algorithm that runs in $O(NT^2 \log NT)$ and a new compact extended formulation with $O(NT^3)$ variables and $O(NT^2)$ constraints. We also propose a dynamic programming algorithm and a compact extended formulation for an extension in which an additional limited amount can be produced for each period in order to obtain some extra revenue. In this case, our algorithm runs in $O(NT^4)$ and the resulting extended formulation has $O(NT^4)$ variables and $O(NT^3)$ constraints.

In Chapter 4 we study the one-warehouse multi-retailer problem (OWMR), which is a NP-Hard extension of the uncapacitated two-level production-in-series lot-sizing problem. A considerable amount of work has been done with the use of approximation algorithms for the problem and some recent work has been carried out comparing extended formulations. We study possible ways to tackle the problem effectively by using mixed integer programming techniques. We analyze the projection of the multi-commodity formulation (a reformulation in which the demands of different clients in different periods are viewed as different commodities) into the space of the original variables for two special cases, the joint-replenishment problem (a special case of the OWMR in which storage is not allowed in the production site) and the uncapacitated two-level production-in-series lot-sizing problem. We relate its projection to some well known inequalities called dicut inequalities by showing it is composed of only simple dicut inequalities for the joint-replenishment problem but this is not the case for the uncapacitated two-level production-in-series lot-sizing problem. We also consider some valid inequalities in a reduced space of variables. Very limited computational experiments are performed comparing first different reformulations (multi-commodity, transportation and shortest path) for the problem and second the use of valid inequalities with the multi-commodity reformulation and an echelon stock formulation which is obtained by adding reformulations to relaxations of a standard formulation for the problem.

In Chapter 5 we analyze a more general two-level production and transportation problem with multiple production sites. Problems with multiple production sites have not received much attention in the literature and most of the problems treated are very case specific, thus there is not much done in the direction of treating more general problems. Here we treat a general problem with multiple production sites which extends the one-warehouse multi-retailer problem. Relaxations for the problem for which results are available in the literature are identified in order to improve the linear relaxation bounds obtained. We show that some uncapacitated instances with reasonable sizes of a basic problem can often be solved to optimality. On the other hand we show that a hybrid MIP heuristic based on two different MIP formulations permits us to find solutions guaranteed to be within 10% of optimality for instances with limited transportation capacity and/or with additional sales. For instances with big bucket production or aggregate storage capacity constraints the gaps can be larger (up to 40%).

In Chapter 6 we study a slightly different type of production and transportation problem. It was treated in a paper entitled Production and transportation Integration for a Make-to-Order Manufacturing Company with a Commit-to-Delivery Business Mode. In this problem one has orders with different sizes and delivery dates and the transportation is considered to be performed by a third company. The authors of the mentioned paper proposed a formulation that solves a polynomial case of the problem and for a more general NP-Hard version they developed a formulation capable of solving very small instances and a heuristic to deal with larger instances. We develop a mixed integer programming formulation that allows us to solve large instances within a few seconds. An algorithm with a local search phase is proposed to tackle more difficult instances.

In Chapter 7 we summarize the contributions of this thesis and indicate possible directions for further research.

Chapter 2

Basics

The goal of this chapter is to provide the reader with basic concepts that will be used throughout this thesis. In Section 2.1 we introduce some basic concepts in mixed integer linear programming. In Section 2.2 we introduce totally unimodular matrices and present some well-known ways to characterize them. In Section 2.3 we introduce the uncapacitated lot-sizing problem which is a basic problem in the production planning literature and present valid inequalities and a reformulation. In Section 2.4 we give some concepts and results for fixed charge network flow problems of which production planning problems are special cases. In Section 2.5 we introduce the dicut collection inequalities, that are inequalities available for fixed charge network flow problems. In Section 2.6 we give some notation that will be used throughout the thesis. In Sections 2.7 and 2.8 we present basic MIP formulations for a variety of two-level supply chain problems studied in the literature. In Section 2.9 we shortly overview other problem variations.

2.1 Basic Concepts in Mixed Integer Programming

A **mixed integer linear problem** (MIP) is an optimization problem

$$z_{MIP} = \min\{cx + fy : (x, y) \in X\}$$

in which X is the set of feasible solutions with continuous variables x and integer variables y. z_{MIP} denotes the optimal mixed integer solution value.

In case all variables are integer we say that we have an **integer program** (IP)

$$z_{IP} = \min\{fy : (y) \in X'\}.$$

A **polyhedron** P is a subset of $\mathbb{R}^n$ described by a finite set of linear constraints $P = \{x \in \mathbb{R}^n : Ax \geq b\}$. A polyhedron

$$P' = \{(x, y) \in \mathbb{R}^n_+ \times \mathbb{R}^p_+ : Ax + By \geq b\}$$

is a **formulation** for X if $X = P' \cap (\mathbb{R}^n \times \mathbb{Z}^p)$.

The **linear relaxation** of a mixed integer problem $\min\{cx + fy : (x, y) \in X\}$ associated to the formulation P' of X is given by

$$z_{LP} = \min\{cx + fy : (x, y) \in \mathbb{R}^n_+ \times \mathbb{R}^p_+ : Ax + By \geq b\}.$$

z_{LP} denotes the linear relaxation bound.

The **convex hull** of X, denoted $conv(X)$, is the set of all points that are convex combinations of points in X, i.e. $conv(X) = \{x : x = \sum_{i=1}^t \lambda_i x^i, \sum_{i=1}^t \lambda_i = 1, \lambda_i \geq 0$ for $i = 1, \ldots, t$ over all finite subsets $\{x^1, \ldots, x^t\}$ of $X\}$.

The convex hull is a polyhedron if the data is rational. Note that $\min\{cx + fy : (x, y) \in X\} = \min\{cx + fy : (x, y) \in conv(X)\}$.

A **valid inequality** for a set X is an inequality $\alpha x + \beta y \geq \gamma$ with $\alpha \in \mathbb{R}^n$, $\beta \in \mathbb{R}^p$ and $\gamma \in \mathbb{R}$, satisfied by all points in X, i.e. $\alpha\bar{x} + \beta\bar{y} \geq \gamma$ for every $(\bar{x}, \bar{y}) \in X$.

Consider $Q = \{(x, z) \in \mathbb{R}^n_+ \times \mathbb{R}^q : Bx + Gz \geq d\}$. The **projection** of Q into the $x-$space is given by

$$proj_x Q = \{x \in \mathbb{R}^n : \text{ there exists } z \text{ for which } (x, z) \in Q\}.$$

An **extended formulation** for a set $X \subseteq \mathbb{R}^n \times \mathbb{Z}^p$ is a formulation

$$P^E = \{(x, y, z) \in \mathbb{R}^n_+ \times \mathbb{R}^p_+ \times \mathbb{R}^q_+ : Cx + Dy + Ez \geq d\}$$

where

$$proj_{(x,y)} P^E \cap (\mathbb{R}^n_+ \times \mathbb{Z}^p_+) = X.$$

An extended formulation is tight if $proj_{(x,y)} P^E = conv(X)$.

2.2 Totally Unimodular Matrices

Definition 2.1. *A matrix A is **totally unimodular** (TU) if each square submatrix of A has determinant -1, 0, or $+1$.*

Definition 2.2. *A nonempty polyhedron $P \subseteq \mathbb{R}^n$ is said to be integral if each of its nonempty faces contains an integral point.*

We now present a result that characterizes totally unimodular matrices.

Theorem 2.1. *(Hoffman and Kruskal [23]) An integral matrix A is totally unimodular if and only if the polyhedron $\{x : Ax \leq b, x \geq 0\}$ (when it is nonempty) is integral for each integral vector b.*

The next two results give well-known ways to identify totally unimodular matrices.

Proposition 2.2. *(Hoffman and Kruskal [23]) A matrix A is TU if and only if*
(a) the transpose matrix A^T is TU if and only if
(b) the matrix (A, I) is TU.

Theorem 2.3. *(Ghouila-Houri [21]) A matrix A is TU if:*
(a) $a_{ij} \in \{-1, 0, +1\}$ for all i, j,
(b) for any subset M of the rows, there exists a partition (M_1, M_2) of M such that each column j satisfies

$$|\sum_{i \in M_1} a_{ij} - \sum_{i \in M_2} a_{ij}| \leq 1.$$

2.3 Single-item Uncapacitated Lot-sizing Problem

The single-item uncapacitated lot-sizing problem (LS-U) is a basic problem in the production planning literature. There is a planning horizon of NT periods with time varying demands and the goal is to determine a production plan minimizing the total cost while satisfying the demands. The problem data is as follows:

- p_t: unit production cost in period t,
- f_t: fixed setup cost in period t,
- h_t: unit storage cost in period t,
- d_t: demand in period t (we may use d_{kt} to denote $\sum_{j=k}^{t} d_j$).

It is assumed here that there are no initial or end stocks and that the demands are nonnegative. The single item lot-sizing problem is a special case of the fixed charge network flow problem that will be introduced in Section 2.4. The following variables are used in order to model the problem as a mixed integer program:

- x_t: amount produced in period t,
- y_t: is equal to 1 if production occurs in period t and 0 otherwise,

- s_t: amount of stock in the end of period t.

The problem can be formulated by using the described variables as follows:

$$(LS-U)\ \min \textstyle\sum_t p_t x_t + \sum_t f_t y_t + \sum_t h_t s_t \tag{2.1}$$
$$s_{t-1} + x_t = d_t + s_t \quad \forall t, \tag{2.2}$$
$$x_t \le M y_t \quad \forall t, \tag{2.3}$$
$$x, s \in \mathbb{R}_+^{NT},\ y \in [0,1]^{NT}, \tag{2.4}$$
$$y \in \mathbb{Z}^{NT}, \tag{2.5}$$

where M is a very large number. The objective function minimizes the total cost. Constraints (2.2) are balance constraints. Constraints (2.3) set the fixed set-up variables to 1 in case production occurs. Constraints (2.4) and (2.5) are bound constraints on the variables.

We now introduce a form of the valid inequalities for the problem, namely the (l, S) inequalities. Let X^{LS-U} be the set of feasible solutions for LS-U.

Proposition 2.4. *(Barany et al. [4]) Let $1 \le l \le n$, $L = \{1, \ldots, l\}$ and $S \subseteq L$, then the (l, S) inequality*

$$\sum_{j \in L \setminus S} x_j + \sum_{j \in S} d_{jl} y_j \ge d_{1l} \tag{2.6}$$

is valid for X^{LS-U}.

Theorem 2.5. *(Barany et al. [4]) When $s_0 = s_{NT} = 0$, the original constraints (2.2)-(2.4) plus the (l,S) inequalities (2.6) give a complete linear inequality description of $conv(X^{LS-U})$.*

Dynamic programming approaches are often used to tackle polynomial solvable production planning problems. We use this simplest possible production planning problem in order to illustrate the use of a dynamic programming algorithm. We will state some properties of the problem and then present a dynamic programming recursion for it.

Observation 2.1. *Since $s_t = \sum_{k=1}^{t} x_t - d_{1t}$, we can eliminate the stock variables from the objective function (2.1) and alternatively write it as*

$$\sum_t p_t x_t + \sum_t f_t y_t + K, \tag{2.7}$$

where $p_t = p'_t + \sum_{k=t}^{NT} h'_t$ and $K = -\sum_t h_t d_{1t}$.

The following proposition characterizes optimal feasible solutions.

Proposition 2.6. *(Wagner and Whitin [54])*
(1) There exists an optimal solution with $s_{t-1}x_t = 0$ for all t. (There is production only if the stock is zero.)
(2) There exists an optimal solution such that if $x_t > 0$, $x_t = \sum_{i=t}^{t+k} d_i$ for some $k \geq 0$. (If there is production in t, the amount produced exactly satisfies demand for periods t to $t+k$)

Let $\{t_1, t_2, \ldots, t_k\}$ be the set of periods in which production occurs. A basic solution will have the format $[t_1, \ldots, t_2-1], [t_2, \ldots, t_3-1], \ldots, [t_k, \ldots, t_{NT}]$ such that the production for each interval only happens in the production period in the beginning of that interval. Each of these intervals is called a **regeneration interval**.

Observe that we can assume $f_t \geq 0$, since whenever $f_t < 0$ we can add the constant f_t to the objective function and use a new setup cost $\bar{f}_t = 0$. Let $H(k)$ be the optimal cost for periods $1, \ldots, k$. Consider the following forward recursion:

$$H(k) = \min_{1 \leq j \leq k} \{H(j-1) + f_j + p_j d_{jk}\}, \tag{2.8}$$

with $H(0) = 0$.

The problem can be solved by solving recursion (2.8). The optimal solution value is given by $H(NT)$ and the optimal solution can be obtained by keeping track of the periods j chosen at each step until the calculation of $H(NT)$. Observe that choosing a given j in the recursion means that the regeneration interval $[j, k]$ will be part of the solution.

We now present an extended shortest path formulation for the single-item uncapacitated lot-sizing. In this formulation we have variables representing the different regeneration intervals of a solution. We have the following variables:

- ϕ_{kt} is equal to 1 if an amount d_{kt} is produced in period k, and is 0 otherwise.

We formulate the problem using the described variables as follows:

$$(SP-LS-U) \qquad \min \textstyle\sum_t (f_t y_t + p_t x_t) \tag{2.9}$$

$$\textstyle\sum_{t=1}^{NT} \phi_{1t} = 1, \tag{2.10}$$

$$\textstyle\sum_{u=1}^{t-1} \phi_{u,t-1} - \sum_{u=t}^{NT} \phi_{tu} = 0 \quad \forall t \geq 2, \tag{2.11}$$

$$\textstyle\sum_{u=t}^{NT} \phi_{tu} \leq y_t \quad \forall t, \tag{2.12}$$

$$\textstyle\sum_{u=t}^{NT} d_{tu}\phi_{tu} = x_t \quad \forall t, \tag{2.13}$$

$$\phi \in \mathbb{R}_+^{NT \times NT}, \ y \in [0,1]^{NT}. \tag{2.14}$$

Constraints (2.10)-(2.11) are flow conservation constraints. Constraints (2.12) define the fixed setup variables. Constraints (2.13) link the shortest path variables with the original production variables. Denote $Q^{SP-LS-U}$ the set of feasible solutions of SP-LS-U.

Theorem 2.7. *(Eppen and Martin [14]) The linear program*

$$\min\{px + qy : (x, y, \phi) \in Q^{SP-LS-U}\}$$

has an optimal solution with y integer, and thus solves LS-U.

2.4 Network Flow Problems

In this section we will introduce a class of problems of which production planning problems are particular cases. Consider a directed network $G = (N, A)$ with a set N of n nodes and a set A of m arcs (i, j). To each arc $(i, j) \in A$ there is an associated cost c_{ij} as well as a lower bound l_{ij} and a capacity u_{ij}. Each node $i \in N$ has an associated weight b_i representing supply/demand (we consider a node i with $b_i > 0$ a demand node and a node i with $b_i < 0$ a supply node). The **minimum cost network flow problem** is formulated as follows.

$$\min \textstyle\sum_{(i,j)\in A} c_{ij} x_{ij} \tag{2.15}$$

$$\textstyle\sum_{j:(j,i)\in A} x_{ji} - \sum_{j:(i,j)\in A} x_{ij} = b_i \quad \forall i \in N, \tag{2.16}$$

$$l_{ij} \le x_{ij} \le u_{ij} \quad \forall (i, j) \in A, \tag{2.17}$$

where $\sum_{i\in N} b_i = 0$ is a necessary condition for feasibility. The objective function (2.15) minimizes the total cost of the flow. Constraints (2.16) are called the flow conservation constraints. Constraints (2.17) bound the flow at each arc. An example of a network flow problem can be seen in Figure 2.1, in which there are two demand nodes (nodes 2 and 3) and one supply node (node 1).

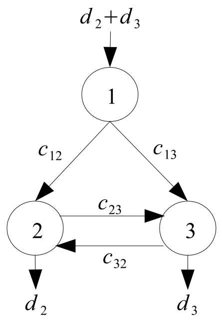


Figure 2.1: Example of a network flow problem

The **fixed charge network flow problem** is the problem in which we additionally impose a fixed cost f_{ij} whenever there is a positive flow on an arc

(i, j).

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij} + f_{ij}y_{ij} \tag{2.18}$$

$$\sum_{j:(j,i)\in A} x_{ji} - \sum_{j:(i,j)\in A} x_{ij} = b_i \quad \forall i \in N, \tag{2.19}$$

$$l_{ij} \le x_{ij} \le u_{ij}y_{ij} \quad \forall (i,j) \in A. \tag{2.20}$$

The single-item uncapacitated lot-sizing problem is an example of fixed charge network flow problem in which there is a source node and the other nodes correspond to periods with their corresponding demands. An illustration of the network of a four period problem is presented in Figure 2.2.

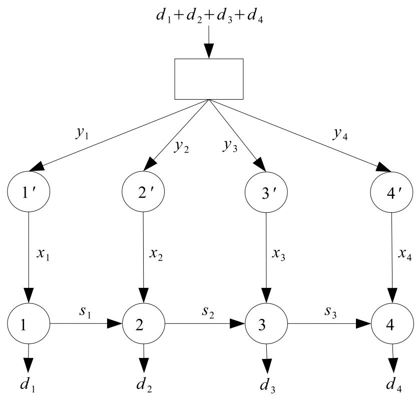


Figure 2.2: Network representation of a four period lot-sizing problem illustrating the fixed charge arcs (corresponding to y variables) separated from the continuous arcs (corresponding to x and s variables)

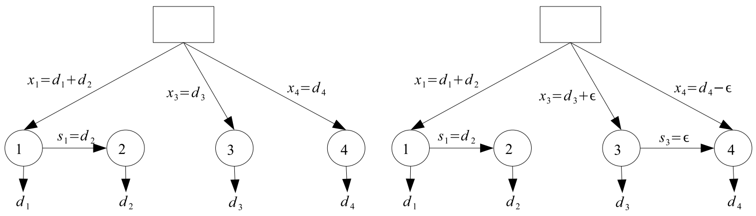
2.4.1 Some Properties of Network Flows

We now present some results on network flow problems that will be useful in later chapters. The first result gives information about the structure of extreme feasible solutions.

Observation 2.2. *(see Zangwill [56]) In a basic or extreme feasible solution of a minimum cost network flow problem, the arcs corresponding to variables with flows strictly between their lower and upper bounds form an acyclic graph.*

Figure 2.3(a) shows an example of a basic feasible solution while Figure 2.3(b) shows an example of a cyclic and thus nonbasic feasible solution.

The next result states the property of decomposability of flows in a network.



(a) Basic feasible solution (b) Cyclic solution

Figure 2.3: Example of basic feasible and cyclic solutions

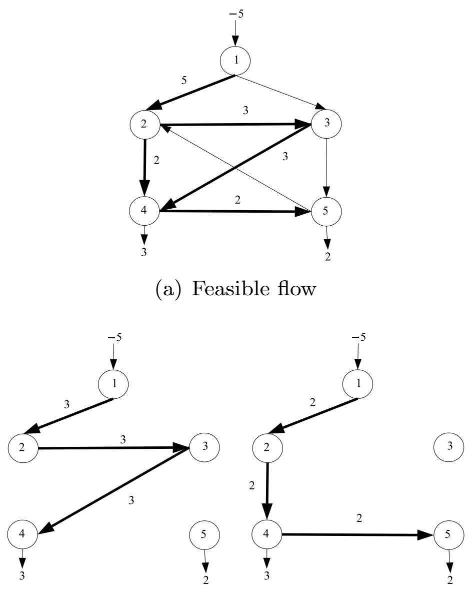
Theorem 2.8. *(see Ford and Fulkerson [19])* ***Flow Decomposition Theorem:*** *Every path and cycle flow has a unique representation as nonnegative arc flows. Conversely, every nonnegative arc flow x can be represented as a path and cycle flow (though not necessarily uniquely) with the following two properties:*
(a) Every directed path with positive flow connects a supply node to a demand node.
(b) At most n+m paths and cycles have nonzero flow; out of these, at most m cycles have nonzero flow.

An illustration of the flow decomposition theorem is depicted in Figure 2.4. The feasible flow in Figure 2.4(a), represented by the bold arcs, can be decomposed in the two paths illustrated in Figures 2.4(b) and 2.4(c).

2.5 Dicut Collection Inequalities

Rardin and Wolsey [41] introduced the dicut collection inequalities for uncapacitated fixed charge network flow problems (FCN). Given a graph $G = (N, A)$ with $A = (F, \bar{F})$ where F represents the set of arcs with fixed charge and $\bar{F}$ represents the set of continuous arcs (when an arc has both fixed and variable costs two arcs are used, one in F and one in $\bar{F}$ as in Figure 2.2), a $t-$**dicut** for $t \in T$ is a set of arcs whose removal from A blocks all flow from the source s to a sink t, where T is the set of considered sinks. A **dicut collection** $\Gamma \triangleq \{\Gamma^t\}_{t\in T}$ is a set of such $t-$dicuts. A **simple dicut collection** is a dicut collection with $|\Gamma^t| \leq 1$, for all $t \in T$, which means that at most one dicut is employed for each sink t. Given the values

- γ^t_{ij} = number of t-dicuts in Γ^t containing the arc (i, j),
- $\gamma_{ij} = \max\{\gamma^t_{ij} : t \in T\}$,
- $\gamma^t = |\Gamma^t|$,



(a) Feasible flow

(b) Path 1 (c) Path 2

Figure 2.4: Example of the decomposition flow theorem

the proposition follows.

Proposition 2.9. *(Rardin and Wolsey [41]) Every inequality*

$$\sum_{(i,j)\in \bar{F}} \gamma_{ij} x_{ij} + \sum_{(i,j)\in F} \sum_{t\in T} d_t \gamma_{ij}^t y_{ij} \geq \sum_{t\in T} \gamma^t d_t$$

derived from a dicut collection $\Gamma \triangleq \{\Gamma^t\}_{t\in T}$ *is valid for the uncapacitated fixed charge network flow problem.*

In Figure 2.5 we have $F = \{y_1, y_2, y_3, y_4\}$ $\bar{F} = \{x_1, x_2, x_3, x_4, s_1, s_2, s_3, s_4\}$. By taking $T = \{1, 2, 3, 4\}$ and the following t-dicuts $\Gamma^1 = \{x_1\}$, $\Gamma^2 = \{x_1, y_2\}$, $\Gamma^3 = \{x_1, y_2, x_3\}$ and $\Gamma^4 = \{x_1, y_2, x_3, y_4\}$, we get the dicut collection inequality

$$x_1 + (d_2 + d_3 + d_4)y_2 + x_3 + d_4 y_4 \geq d_1 + d_2 + d_3 + d_4.$$

Observe that this is a simple dicut inequality.

2.6 Notation used for Supply Chain Problems

A **supply chain** is a set of inter-related operations that provides goods and/or services to end customers. Such operations may include the production, the

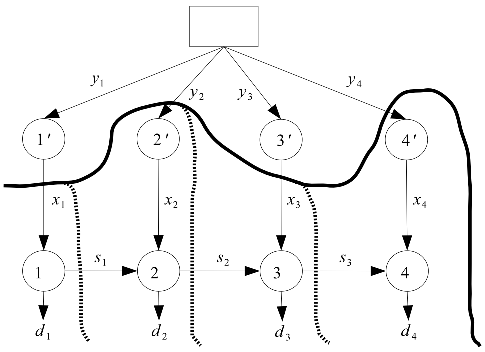


Figure 2.5: Example of dicut collection with t-dicuts $\Gamma^1 = \{x_1\}$, $\Gamma^2 = \{x_1, y_2\}$, $\Gamma^3 = \{x_1, y_2, x_3\}$ and $\Gamma^4 = \{x_1, y_2, x_3, y_4\}$

packing and the transportation of goods to the clients among others.

In the next two sections we will review some particular cases of two-level supply chains, namely some joint production and transportation problems. The purpose of presenting these problems is to familiarize the reader with the sort of problems studied in the literature of joint production and transportation, some of which will be treated specifically in this thesis. These problems have as goal the satisfaction of customer demands while optimizing a certain objective function.

Now we introduce some notation that will be used in the remainder of this thesis. The problem dimensions are denoted by:

- NI: number of items,
- NP: number of production sites,
- NC: number of client areas,
- NT: number of time periods.

Next we introduce some data used to describe the problem and variables that will be used to formulate them as mixed integer programs. The description of the parameters and the variables is very general so that for each specific problem the indices that are not going to be used will be dropped.

There are some costs that may appear in the problems:

- p_t^{0ip}: per unit manufacturing cost of product i at production site p in period t,
- p_t^{1ipc}: cost to transport a unit of product i from production site p to client c in period t,

- f_t^{0ip}: fixed cost of manufacturing product i at production site p in period t,
- f_t^{1pc}: fixed cost of transporting products between production site p and client c in period t,
- h_t^{0ip}: per unit storage cost of product i at production site p in period t,
- h_t^{1ic}: per unit storage cost of product i at client c in period t,
- e_t^{ic}: per unit cost of backlogging product i at client c in period t.

It is possible that some resources are capacity constrained:

- K_0^{ip}: manufacturing capacity for product i in production site p,
- K_1: capacity of each transportation vehicle,
- LK_t^p: manufacturing joint capacity (big bucket) at production site p in period t,
- SK_t^c: upper bound on stocks at client c in period t,
- M: very big number used to determine unlimited capacity.

Different variables can be present in the formulation of the problems:

- x_t^{0ip}: amount manufactured of product i at production site p in period t,
- x_t^{1ipc}: amount transported of product i between production site p and client c in period t,
- y_t^{ip}: is equal to 1 if there is manufacture of product i at production site p in period t,
- Y_t^{pc}: number of vehicles used for transportation between production site p and client c in period t,
- s_t^{0ip}: amount of product i stored at production site p in the end of period t,
- s_t^{1ic}: amount of product i stored at client c in the end of period t,
- r_t^{ic}: amount of product i backlogged at client c in period t,
- v_t^{ic}: amount of product i to be sold at client c in period t.

In the next two sections we will present some two-level problems studied in the literature along with basic formulations as mixed integer programs. In Section 2.7 we present problems with a single production site while in Section 2.8 we overview problems with multiple production sites.

2.7 Single Production Site Supply Chain Problems

2.7.1 Uncapacitated Two-level Production-in-series Lot-sizing Problem

$NI = NP = NC = 1$
Production setup
Transportation setup

The Uncapacitated two-level production-in-series lot-sizing problem (2L-S/LS-U), which will be studied in details in Chapter 3, is a polynomial solvable problem that can be thought of as a production/transportation problem in which a production site is said to be at level zero and a client site at level one. There is a single item with time varying deterministic demands (d_t) over a discrete horizon of NT time periods that have to be satisfied without backlogging. Production and transportation are unlimited and we have the following costs:

- fixed set-up cost (f_t^0/f_t^1) in case production/transportation occurs,
- variable production/transportation cost (p_t^0/p_t^1) that depends on the amount produced,
- storage costs at both the production (h_t^0) and the client site (h_t^1).

The goal is to find a feasible schedule minimizing the total cost. Consider the variables:

- x_t^0: amount manufactured at the production site in period t,
- x_t^1: amount transported between the production site and the client in period t,
- y_t: is equal to 1 if there is production in period t,
- Y_t: is equal to 1 if there is transportation between the production site and the client in period t,
- s_t^0: amount stored at the production site in the end of period t,
- s_t^1: amount stored at the client in the end of period t.

The problem can be formulated as follows:

$$(2L - S/LS - U)$$

$$\min \sum_t (f_t^0 y_t + p_t^0 x_t^0 + h_t^0 s_t^0) + \sum_t (f_t^1 Y_t + p_t^1 x_t^1 + h_t^1 s_t^1) \quad (2.21)$$

$$s_{t-1}^0 + x_t^0 = x_t^1 + s_t^0 \quad \forall t, \quad (2.22)$$

$$s_{t-1}^1 + x_t^1 = d_t + s_t^1 \quad \forall t, \quad (2.23)$$

$$x_t^0 \leq M y_t \quad \forall t, \quad (2.24)$$

$$x_t^1 \leq M Y_t \quad \forall t, \quad (2.25)$$

$$x^0, s^0 \in \mathbb{R}_+^{NT},\ x^1, s^1 \in \mathbb{R}_+^{NT},\ y, Y \in \{0,1\}^{NT}. \quad (2.26)$$

The objective function minimizes the total production, transportation and storage costs. Constraints (2.22) and (2.23) are balance constraints for the production and for the client sites respectively. Constraints (2.24) and (2.25) are setup constraints for the production and for the client sites respectively. Constraints (2.26) are the bound and integrality constraints on the variables. An illustrative example of a fixed charge network, with fixed charges on the production and on the transportation arcs, representing the problem can be viewed in Figure 2.6. The square nodes correspond to the production site and the circular nodes represent the client.

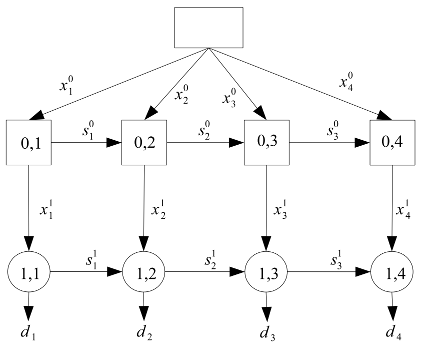


Figure 2.6: Fixed charge network of the uncapacitated two-level production-in-series lot-sizing problem with four periods

The uncapacitated two-level production-in-series lot-sizing problem is a special case of the multi-level production-in-series lot-sizing problem studied in Zangwill [56]. In this more general problem, there are L levels $\{0, 1, \ldots, L-1\}$ such that the production at each level l is subject to a fixed setup cost and to unit production costs and depends on the production of level $l-1$ and demand is present only at level $L-1$.

The standard algorithm for the uncapacitated production-in-series lot-sizing problem is a dynamic programming algorithm due to Zangwill [56] whose running time for $L > 2$ levels is $O(L \times NT^4)$. Using the approach proposed by Eppen and Martin [14, 31], the dynamic programming recursion can be converted into a linear program whose dual provides a tight polynomial size extended formulation for the convex hull of solutions – for the recursion of Zangwill [56] the resulting formulation has $O(L \times NT^4)$ variables and $O(L \times NT^3)$ constraints for general L. In van Hoesel et al. [51], the authors treat two-level and multi-level problems in which there are capacity restrictions at all but the initial production level. Polynomial dynamic programming algorithms are derived when the capacities are constant and it is observed that when $L = 2$, the backward dynamic program of Zangwill [56] runs in $O(NT^3)$. In Lee et al. [25] a two-level problem with more complicated transportation costs is studied and a polynomial dynamic program is derived.

Under certain assumptions on the costs, optimal solutions are known to be nested (i.e. $x_t^0 > 0$ implies $x_t^1 > 0$), see Love [30]. For $L > 2$, the running times of the corresponding algorithms of Zangwill [56] reduce to $O(L \times NT^3)$ in this case. Specifically when $L = 2$, if the costs at level zero satisfy the Wagner-Whitin condition ($p_t^0 + h_t^0 \geq p_{t+1}^0$ for all t) and the fixed costs are non-increasing ($f_t^0 \geq f_{t+1}^0$ for all t), then there is an optimal nested solution.

Various researchers have tried to devise compact tight extended formulations. Using echelon stocks it is shown in Pochet [38] that the (l, S) inequalities from Barany et al. [4] provide a computationally effective approach for multi-level production-in-series problems. Another alternative is to generalize the facility location of Krarup and Bilde [24] or shortest path extended formulations of Eppen and Martin [14] that are known to be tight for the single-level problem. Various authors have observed that these reformulations with $O(NT^2)$ variables are also effective in practice, but even in the two-level case there are instances that are not tight, see Pochet and Wolsey [39].

In Chapter 3, we analyze the uncapacitated two-level production-in-series lot-sizing problem. We present a new dynamic programming algorithm and a tight extended formulation. Denizel et al. [10] have very recently given an alternative network formulation for the problems with two and three levels.

2.7.2 Joint-replenishment Problem

$NI = NP = 1$
Production setup
Transportation setup
No storage at the production site

The Joint-replenishment problem (JRP) is a NP-Hard problem that can be viewed as a multi-client two-level production/transportation problem. There are NC clients with deterministic demand (d_t^c) for each of the NT time periods. Production and transportation for the different clients are unlimited and storage is not allowed at the production site. The costs are:

- fixed production (f_t^0) and transportation costs (f_t^{1c}),
- variable production (p_t^0) and transportation costs (p_t^{1c}),
- holding costs at each client (h_t^{1c}).

The goal is to determine an ordering policy minimizing the total cost and satisfying demands without backlogging. We define the variables:

- x_t^0: amount manufactured at the production site in period t,
- x_t^{1c}: amount transported between the production site and client c in period t,
- y_t: is equal to 1 if there is manufacture at the production site in period t,
- Y_t^c: is equal to 1 if transportation occurs between the production site and client c in period t,
- s_t^{1c}: amount stored at client c in the end of period t.

The problem can be formulated as follows:

$$(JRP)\ \min \textstyle\sum_t (f_t^0 y_t + p_t^0 x_t^0) + \sum_{c,t} (f_t^{1c} Y_t^c + p_t^{1c} x_t^{1c} + h_t^{1c} s_t^{1c}) \quad (2.27)$$
$$x_t^0 = \textstyle\sum_c x_t^{1c} \quad \forall t, \quad (2.28)$$
$$s_{t-1}^{1c} + x_t^{1c} = d_t^c + s_t^{1c} \quad \forall c, t, \quad (2.29)$$
$$x_t^0 \leq M y_t \quad \forall t, \quad (2.30)$$
$$x_t^{1c} \leq M Y_t^c \quad \forall c, t, \quad (2.31)$$
$$s^0, x^0 \in \mathbb{R}_+^{NT},\ s^1, x^1 \in \mathbb{R}_+^{NC \times NT}, \quad (2.32)$$
$$y \in \{0,1\}^{NT},\ Y \in \{0,1\}^{NC \times NT}. \quad (2.33)$$

The objective function minimizes the total production, transportation and storage costs. Constraints (2.28) are balance constraints for the production site while constraints (2.29) are balance constraints for each of the clients. Constraints (2.30) are setup constraints for the production site. Constraints (2.31) are setup constraints for each of the client areas. Constraints (2.32) and (2.33) are bound and integrality constraints on the variables. An illustrative example of a network representing the problem can be viewed in Figure 2.7. We have the production site represented by square nodes and the clients by the circular nodes.

The uncapacitated joint-replenishment problem (JRP) has been studied by numerous authors, see Robinson et al. [42]. It has been shown to be NP-Hard

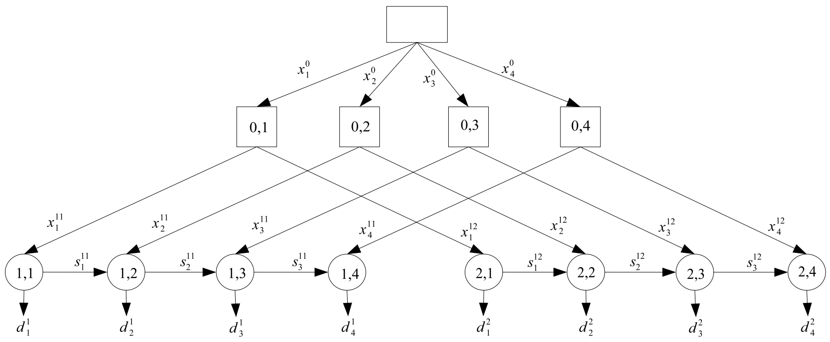


Figure 2.7: Example of the joint-replenishment problem with two clients and four periods

in Arkin et al. [3] by a reduction from the 3-SAT problem. In Federgruen and Tzur [16] the authors elaborated a problem specific exact branch-and-bound algorithm and a partitioning heuristic. Some works propose approximation algorithms for the problem. Levi et al. [27] propose a primal-dual framework for dealing with general deterministic inventory problems. Their approach yields a 2-approximation algorithm when applied to the JRP. Levi et al. [26] present a new 1.8-approximation algorithm based on a linear programming rounding procedure.

Observe that the Joint-replenishment problem can also be considered as a single-level multi-item lot-sizing problem with joint-setup costs.

2.7.3 One-warehouse Multi-retailer Problem

$NI = NP = 1$
Production setup
Transportation setup

The One-warehouse multi-retailer problem (OWMR), which will be studied in more details in Chapter 4, is a generalization of the Joint-replenishment problem in which storage is allowed at the production site. The OWMR is NP-Hard since it is a generalization of the JRP. There is one production site which replenishes multiple (NC) clients over a finite time horizon of NT periods. Each client has a time varying deterministic demand (d_t^c) for each period t in the time horizon. The amount manufactured by the production site and the amount transported from it to the clients are unlimited. We define the variables:

- x_t^0: amount manufactured at the production site in period t,

- x_t^{1c}:amount transported between the production site and client c in period t,
- y_t: is equal to 1 if there is manufacture at the production site in period t,
- Y_t^c: is equal to 1 if transportation occurs between the production site and client c in period t,
- s_t^0: amount stored at the production site in the end of period t,
- s_t^{1c}: amount stored at client c in the end of period t.

The problem can be formulated as follows:

$$(OWMR)$$

$$\min \sum_t (f_t^0 y_t + p_t^0 x_t^0 + h_t^0 s_t^0) + \sum_{c,t} (f_t^{1c} Y_t^c + p_t^{1c} x_t^{1c} + h_t^{1c} s_t^{1c}) \tag{2.34}$$

$$s_{t-1}^0 + x_t^0 = \textstyle\sum_c x_t^{1c} + s_t^0 \quad \forall t, \tag{2.35}$$

$$s_{t-1}^{1c} + x_t^{1c} = d_t^c + s_t^{1c} \quad \forall c,t, \tag{2.36}$$

$$x_t^0 \leq M y_t \quad \forall t, \tag{2.37}$$

$$x_t^{1c} \leq M Y_t^c \quad \forall c,t, \tag{2.38}$$

$$s^0, x^0 \in \mathbb{R}_+^{NT},\ s^1, x^1 \in \mathbb{R}_+^{NC \times NT}, \tag{2.39}$$

$$y \in \{0,1\}^{NT},\ Y \in \{0,1\}^{NC \times NT}. \tag{2.40}$$

The objective function minimizes the total production, transportation and storage costs. Constraints (2.35)-(2.40) are similar to constraints (2.28)-(2.33) with the difference that constraints (2.35) take into account the storage variables at the production site. An illustrative example of a network representing the problem can be viewed in Figure 2.8. We have the production site represented by square nodes and the clients by circular nodes.

Federgruen and Tzur [17] proposed an exact branch-and-bound and a partitioning heuristic for the problem. Levi et al. [26] analyze the problem with a more general cost structure that takes into consideration the total amount of time the product is stored in both the production site and the client. The authors give an 1.8-approximation algorithm based on a linear programming rounding procedure for the problem.

Solyali and Sural [46] compare both theoretically and experimentally MIP formulations for the problem. They could solve to optimality problems with up to $NC = 150$ clients and $NT = 30$ time periods. An obvious extension for the problem is the multi-item version, which is considered in Federgruen and Tzur [17].

In Chapter 4 we will consider the one-warehouse multi-retailer problem.

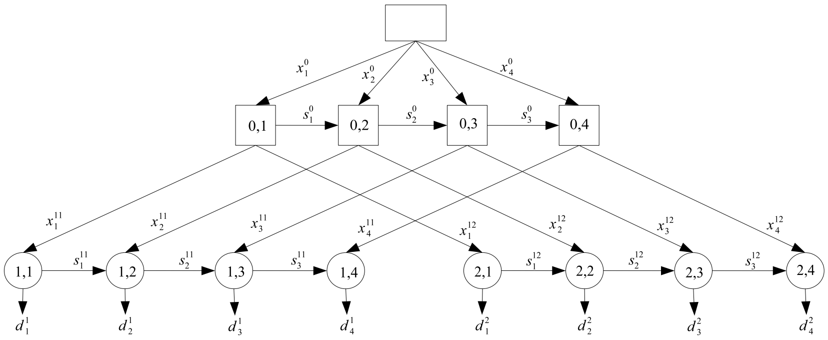


Figure 2.8: Example of the one-warehouse multi-retailer problem with two clients and four periods

2.7.4 The Multi-item Joint-replenishment Problem with Backlogging and Production/Setup Times

$NP = 1$
Production setup
Transportation setup
Capacitated vehicles for transportation
Big bucket capacity at the production site
No storage at the production site
Backlogging

In Özdamar and Yazgaç [35], the authors treat a NP-Hard problem based on a case in which a company produces liquid and power detergents. One production site supplies multiple (NI) items for multiple (NC) clients which have deterministic demands (d_t^{ic}), based on sales forecast from each of them, to be satisfied. Similar to the Joint-replenishment problem, no inventories are held in the production site. The items are divided into different groups (G) such that the items within a group share the same resources in the production site. The following parameters are part of the problem data:

- production setup times (g^i),
- production processing times (o^i),
- limited time capacity (LK_t^G) for producing each group of items at the production site in each period,
- items are transported from the production site to the clients by vehicles with limited capacity (K).

The costs are:

- fixed cost per vehicle to be used in the transportation (f^{1c}),
- storage (h^{1ic}) costs,
- backlogging costs (e_t^{1ic}) at the clients (backorders costs are higher than inventory costs).

We define the variables:

- x_t^{0i}: amount manufactured of product i at the production site in period t,
- x_t^{1ic}: amount transported of product i between the production site and client c in period t,
- y_t^i: is equal to 1 if there is manufacture of product i at the production site in period t,
- Y_t^c: number of vehicles used for transportation between the production site and client c in period t,
- s_t^{1ic}: amount of product i stored at client c in the end of period t.

The problem can be formulated as follows:

$$(JRP-B-ST) \quad \min \textstyle\sum_{c,t} f^{1c}Y_t^c + \sum_{i,c,t}(h_t^{1ic}s_t^{1ic} + e_t^{1ic}b_t^{1ic}) \tag{2.41}$$
$$x_t^{0i} = \textstyle\sum_c x_t^{1ic} \quad \forall\ i,t, \tag{2.42}$$
$$s_{t-1}^{1ic} + x_t^{1ic} - b_{t-1}^{ic} + b_t^{ic} = d_t^{ic} + s_t^{1ic} \quad \forall\ i,c,t, \tag{2.43}$$
$$x_t^{0i} \leq My_t^i \quad \forall\ i,t, \tag{2.44}$$
$$\textstyle\sum_i x_t^{1ic} \leq KY_t^c \quad \forall c,t, \tag{2.45}$$
$$\textstyle\sum_{i\in G}(o^i x_t^{0i} + g^i y_t^i) \leq LK_t^G \quad \forall\ G,t, \tag{2.46}$$
$$x^0 \in \mathbb{R}_+^{NI\times NT},\ x^1, s^1, b \in \mathbb{R}_+^{NI\times NC\times NT}, \tag{2.47}$$
$$y \in \{0,1\}^{NI\times NT},\ Y \in \mathbb{Z}_+^{NC\times NT}. \tag{2.48}$$

The objective function minimizes the total fixed transportation, storage and backlogging costs. Constraints (2.42) are the balance constraints at the production site. Constraints (2.43) are the balance constraints at the client sites taking into consideration the backlogging. Constraints (2.44) are setup constraints at the production site while constraints (2.45) determine the number of vehicles to be used for transportation. Constraints (2.46) are resource constraints (big bucket) limiting the total amount of time that can be used to produce each group of items in each time period. Constraints (2.47) and (2.48) are bound and integrality constraints on the variables. An illustrative example of a network representing the problem can be viewed in Figure 2.9. We have the production site represented by square nodes and the clients by the circular nodes.

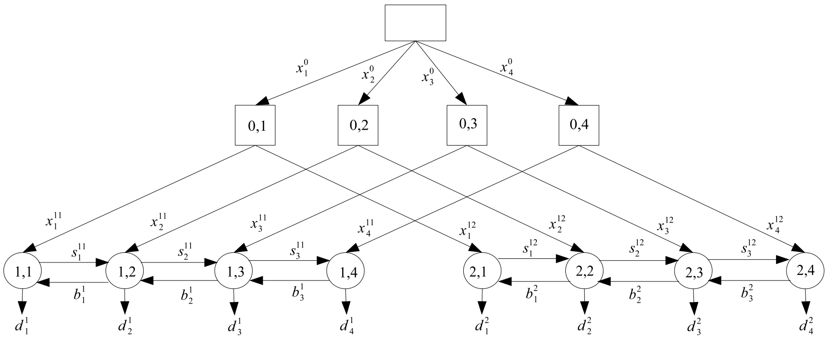


Figure 2.9: Example of the multi-item joint-replenishment problem with backlogging and production/setup times with two clients and four periods

Özdamar and Yazgaç [35] proposed an hierarchical approach. In their approach two models are solved, a reduced aggregated and a larger disaggregated. The aggregated model is solved first and its solution is used as data for the disaggregated model. The aggregated problem was composed of $NI = 2$ items, $NC = 5$ clients and $NT = 6$ periods while the disaggregated one had $NI = 6$ items, $NC = 5$ clients and $NT = 48$ periods.

2.8 Multiple Production Site Supply Chain Problems

In contrast to what happens with single production site problems, multiple production site supply chain problems have not received great attention in the literature. In this section we present some of the limited number of problems studied and highlight the approaches used.

2.8.1 Multi-warehouse Multi-retailer Problem without Client Storage

Production setup
No transportation setup
No storage at the clients
Joint production (big bucket) capacity at the production site

In Eksioglu et al. [12], the authors treat a NP-Hard problem in which there are multiple (NI) items, multiple (NP) production sites and multiple (NC) clients with deterministic demand (d_t^{ic}) over a discrete time horizon of NT periods. Items share a common production resource with item-specific setup costs. No storage is allowed at the client areas. The costs are:

- fixed setup costs (f_t^{0ip}),
- variable production costs (p_t^{0ip}),
- transportation cost is determined only by the variable transportation costs (p_t^{1ipc}),
- there are also storage costs (h_t^{0ip}) at each production site.

We define the variables:

- x_t^{0ip}: amount manufactured of product i at production site p in period t.
- x_t^{1ipc}: amount transported of product i between production site p and client c in period t.
- y_t^{ip}: is equal to 1 if there is manufacture of product i at production site p in period t.
- s_t^{0ip}: amount of product i stored at production site p in the end of period t.

The problem can be formulated as follows:

$$(MWMR - PC - SC0)$$

$$\min \sum_{i,p,t}(f_t^{0ip}y_t^{ip} + p_t^{0ip}x_t^{0ip} + h_t^{0ip}s_t^{0ip}) + \sum_{i,p,c,t} p_t^{1ipc}x_t^{1ipc} \tag{2.49}$$

$$s_{t-1}^{0ip} + x_t^{0ip} = \sum_c x_t^{1ipc} + s_t^{0ip} \quad \forall\ i,p,t, \tag{2.50}$$

$$\sum_p x_t^{1ipc} = d_t^{ic} \quad \forall\ i,c,t, \tag{2.51}$$

$$x_t^{0ip} \leq My_t^{ip} \quad \forall\ i,p,t, \tag{2.52}$$

$$\sum_i x_t^{0ip} \leq LK_t^p \quad \forall\ p,t, \tag{2.53}$$

$$s^0, x^0 \in \mathbb{R}_+^{NI\times NP\times NT},\ x^1 \in \mathbb{R}_+^{NI\times NP\times NC\times NT}, \tag{2.54}$$

$$y \in \{0,1\}^{NI\times NP\times NT}. \tag{2.55}$$

The objective function minimizes the total production, storage and transportation costs. Constraints (2.50) are balance constraints at the production sites. Constraints (2.51) ensure the amount transported satisfies the demand. Constraints (2.52) are setup constraints at the production sites. Constraints (2.53) are resource constraints (big bucket) for each of the production sites at each period. Constraints (2.54) and (2.55) are bound and integrality constraints on the variables. An illustrative example of a network representing the problem can be viewed in Figure 2.10. We have the production site represented by square nodes and the clients by circular nodes.

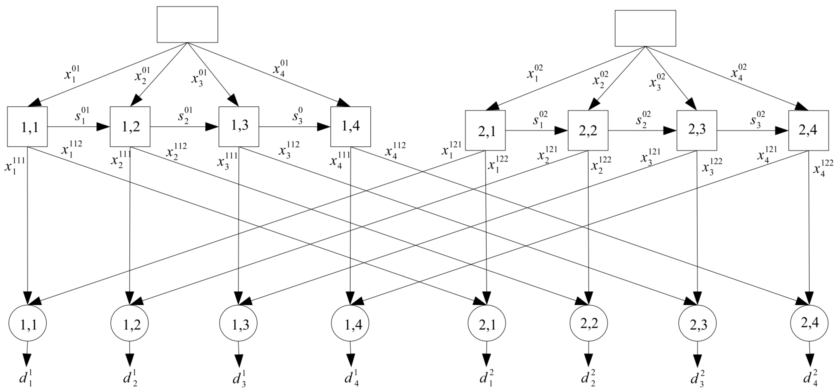


Figure 2.10: Example of the multi-warehouse multi-retailer problem without client storage problem with two production sites, two clients and four periods

Eksioglu et al. [13] propose a primal-dual heuristic for the uncapacitated single-item version of the problem. In Eksioglu et al. [12] an extended formulation and a lagrangean heuristic based on a lagrangean decomposition using this formulation are proposed. They generated heuristic solutions for instances with up to $NI = 50$ items, $NP = 15$ production sites, $NC = 60$ clients and $NT = 35$ periods.

2.8.2 Multi-plant Lot-sizing Problem with Inter-plant Transfers

Production setup
No transportation setup
Big bucket capacity at the production sites
Each production site has its own demands

The Multi-plant lot-sizing problem with inter-plant transfers, studied in Sambasivan [43], is a NP-Hard production/transportation problem with strong similarities with two-level problems although it is not exactly one of them. There are multiple (NP) production sites, each one with deterministic demands (d_t^{ip}) for multiple (NI) items. The costs are:

- fixed setup costs (f_t^{0ip}),

- variable production costs (p_t^{0ip}),
- storage costs at the production sites (h_t^{ip}),
- transfers between two different production sites p and q can be done with a variable transportation cost (p_t^{1pq}).

We define the variables:

- x_t^{0ip}: amount manufactured of product i at production site p in period t,
- x_t^{1ipq}: amount transported of product i from production site p to production site q in period t,
- y_t^{ip}: is equal to 1 if there is manufacture of product i at production site p in period t,
- s_t^{ip}: amount of product i stored at production site p in the end of period t.

The problem can be formulated as follows:

$$(MPLS)$$

$$\min \sum_{i,p,t}(f_t^{0ip} y_t^{ip} + p_t^{0ip} x_t^{0ip} + h_t^{ip} s_t^{ip}) + \sum_{\substack{i,p,q,t\\ q\neq p}} p_t^{1pq} x_t^{1ipq} \tag{2.56}$$

$$s_{t-1}^{ip} + x_t^{0ip} - \sum_{q\neq p} x_t^{1ipq} + \sum_{q\neq p} x_t^{1iqp} = d_t^{ip} + s_t^{ip} \quad \forall i,p,t, \tag{2.57}$$

$$x_t^{ip} \leq M y_t^{ip} \quad \forall i,p,t, \tag{2.58}$$

$$\sum_i (o_t^{ip} x_t^{ip} + g_t^{ip} y_t^{ip}) \leq LK_t^p \quad \forall p,t, \tag{2.59}$$

$$s, x^0 \in \mathbb{R}_+^{NI\times NP\times NT},\ x^1 \in \mathbb{R}_+^{NI\times NP\times NP\times NT}, \tag{2.60}$$

$$y \in \{0,1\}^{NI\times NP\times NT}. \tag{2.61}$$

The objective function minimizes the total production, transportation and storage costs. Constraints (2.57) are balance constraints at each production site which take into consideration the transportation between different production sites. Constraints (2.58) are setup constraints for each production site. Constraints (2.59) are resource constraints (big bucket) for each production site. Constraints (2.60) and (2.61) are bound and integrality constraints on the variables. An illustrative example of a network representing the problem can be viewed in Figure 2.11.

Sambasivan [43] showed that the uncapacitated version of the problem is already NP-Hard. They make an assumption that production and inventory at any production site for any production site satisfy demand for contiguous periods.

Most of the work on the capacitated version of the problem has been concentrated on the development of heuristics. Sambasivan and Schmidt [44] propose

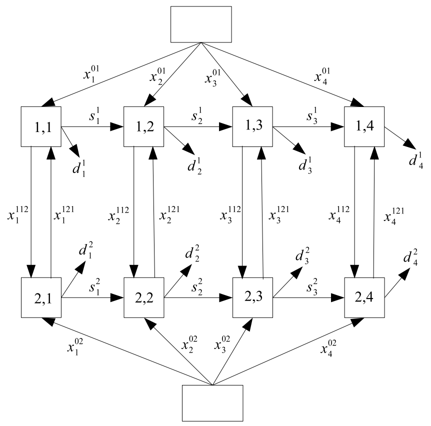


Figure 2.11: Example of the multi-plant lot-sizing problem with inter-plant transfers problem with two production sites

a heuristic in which they first solve the uncapacitated problem and then apply some shift procedures in order to get a feasible solution. Sambasivan and Yahya [45] develop a lagrangean based heuristic. Nascimento et al. [34] developed a Greedy randomized adaptive search procedure (GRASP) metaheuristic for the problem. The authors considered instances with up to $NI = 15$ items, $NP = 4$ production sites and $NT = 6$ periods.

2.8.3 General Problem

Production setup
Transportation setup
Capacitated vehicles for transportation
Big bucket capacity at the production sites
Limited storage at the clients
Core and forecasted demands

Park [36] studies a NP-Hard problem that consists of determining the manufacturing, transportation and storage schedule for the products so as to maximize the revenue minus the total cost. Both production and transportation costs are composed of a fixed cost plus a variable cost that depends on the amount produced/transported. There are NI items, NP production sites, NC clients and NT periods. Each client has his/her core demands (d_t^{ic}) to be satisfied and also his/her forecasted demands ($\bar{d}_t^{ic}$) that do not necessarily have to be totally satisfied, but there is a penalty in case they are not. Vehicles of a given capacity transport items between the production sites and the clients.

There is a resulting variable transportation cost depending on the item and a fixed cost per vehicle incurred whenever it transports a positive amount of one or several items. Finally there are variable storage costs at both production sites and client areas. There is a resource constraint on the total amount produced at the production sites and also upper bound on stocks at the clients. The input data is:

- vehicles capacities (K_1),
- setup times (o^{ip}),
- production times (g^{ip}),
- joint production capacities at the production sites (LK^p),
- storage capacities at the clients (SK^c).

The costs are:

- per unit production costs at the production sites (p^{0ip}),
- per unit transportation costs from the production sites to the clients (p^{1ipc}),
- fixed setup costs at the production sites (f^{0ip}),
- fixed costs per vehicle to transport items between the production sites and the clients (f^{1pc}),
- storage costs at the production sites (h^{0ip}),
- storage costs at the clients (h^{1ic}),
- products selling prices (l^{ic}),
- costs for not satisfying the forecasted demand (e^{ic}).

We define the variables:

- x_t^{0ip}: amount manufactured of product i at production site p in period t,
- x_t^{1ipc}: amount transported of product i between production site p and client c in period t,
- y_t^{ip}: is equal to 1 if there is manufacture of product i at production site p in period t,
- Y_t^{pc}: number of vehicles used for transportation between production site p and client c in period t,

- s_t^{0ip}: amount of product i stored at production site p in the end of period t,
- s_t^{1ic}: amount of product i stored at client c in the end of period t.

The problem can be formulated as follows:

$$(GENP)$$

$$\max \sum_{i,c,t}(l^{ic}s_{t-1}^{1ic} + \sum_p l^{ic}x_t^{1ipc} - l^{ic}s_t^{1ic}) -$$
$$(\sum_{i,p,t} p^{0ip}x_t^{0ip} + \sum_{i,p,t} f^{0ip}y_t^{ip} + \sum_{i,p,t} h^{0ip}s_t^{0ip} + \sum_{i,c,t} h^{1ic}s_t^{1ic} + \quad (2.62)$$
$$\sum_{i,c,t} e^{ic}(\bar{d}_t^{ic} - s_{t-1}^{1ic} - \sum_p x_t^{1ipc} + s_t^{1ic}) + \sum_{p,c,t} f^{1pc}Y_t^{pc} + \sum_{i,p,c,t} p^{1ipc}x_t^{1ipc})$$

$$s_{t-1}^{0ip} + x_t^{0ip} = \sum_c x_t^{1ipc} + s_t^{0ip} \quad \forall\ i, p, t, \quad (2.63)$$

$$s_{t-1}^{1ic} + \sum_p x_t^{1ipc} \geq d_t^{ic} + s_t^{1ic} \quad \forall\ i, c, t, \quad (2.64)$$

$$s_{t-1}^{1ic} + \sum_p x_t^{1ipc} \leq \bar{d}_t^{ic} + s_t^{1ic} \quad \forall\ i, c, t, \quad (2.65)$$

$$x_t^{0ip} \leq My_t^{ip} \quad \forall\ i, p, t, \quad (2.66)$$

$$\sum_i x_t^{1ipc} \leq K_1 Y_t^{pc} \quad \forall\ p, c, t, \quad (2.67)$$

$$\sum_i (o^{ip}x_t^{ip} + g^{ip}y_t^{ip}) \leq LK^p \quad \forall\ p, t, \quad (2.68)$$

$$\sum_i s_t^{1ic} \leq SK^c \quad \forall\ c, t, \quad (2.69)$$

$$s^0, x^0 \in \mathbb{R}_+^{NI \times NP \times NT},\ s^1 \in \mathbb{R}^{NI \times NC \times NT},\ x^1 \in \mathbb{R}^{NI \times NP \times NC \times NT}, \quad (2.70)$$

$$y \in \{0,1\}^{NI \times NP \times NT},\ Y \in \mathbb{Z}_+^{NP \times NC \times NT}. \quad (2.71)$$

The objective function maximizes the revenue minus the costs. Constraints (2.63) are balance constraints for the production sites. Constraints (2.64) ensure all the demands are satisfied while constraints (2.65) guarantee the amount produced is not larger than the forecasted demand. Constraints (2.66) fix the set-up variables to 1 in case production occurs at the production sites. Constraints (2.67) determine the number of vehicles needed to transport the items to the clients. Constraints (2.68) are resource constraints (big bucket) for each production site in each period. Constraints (2.69) limit the total stock at each of the clients. Constraints (2.70) and (2.71) are bound and integrality constraints on the variables. An illustrative example of a network representing the problem can be viewed in Figure 2.12.

Park [36] elaborates a heuristic solution with a local improvement procedure in order to generate feasible solutions. Instances with up to $NI = 5$, $NP = 5$, $NC = 70$ and $NT = 10$ were considered. His results based on heuristic solutions suggest that an integrated model for this two-level problem significantly improves on a two stage hierarchical approach.

In Chapter 5 we will consider a problem equivalent to the General problem.

The problems presented are now summarized in Table 2.1. The first column in the table identifies the problems. The other columns indicate the presence or

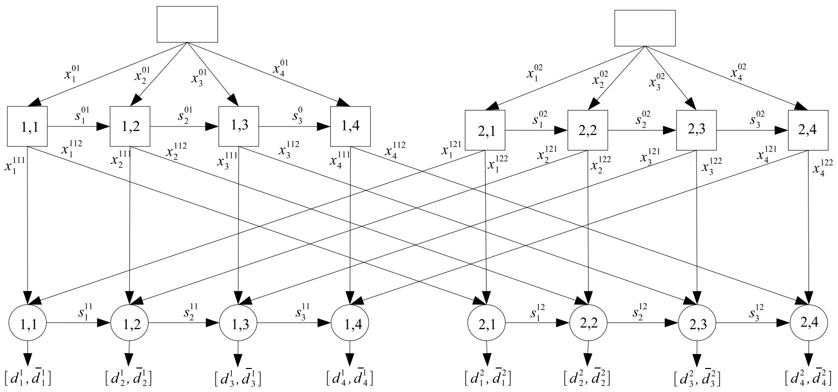


Figure 2.12: Example of the general problem with two production sites and two clients

not of some characteristics, that appear in this order: multiple items, multiple production sites, multiple clients, capacitated transportation, storage in the production sites, storage in the clients, production set-up, transportation set-up, big bucket capacity and backlogging.

Table 2.1: Overview of the problems from Sections 2.7 and 2.8

Problem	MI	MP	MC	Cap. Tr.	Stock Pr.	Stock Cl.	Prod. S-U	Tr. S-U	Big Buc.	Back.
2L-S/LS-U	-	-	-	-	Yes	Yes	Yes	Yes	-	-
JRP	-	-	Yes	-	-	Yes	Yes	Yes	-	-
OWMR	-	-	Yes	-	Yes	Yes	Yes	Yes	-	-
JRP-B-ST	Yes	-	Yes	Yes	-	Yes	Yes	Yes	Yes	Yes
MWMR-PC-SC0	Yes	Yes	Yes	-	Yes	-	Yes	-	Yes	-
MPLS	Yes	Yes	-	-	Yes	-	-	-	Yes	-
GENP	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	-

2.9 Other Problem Variations

In de Matta and Miller [9] the authors study a problem with two production sites in which intermediate products are produced in one production site and shipped to the other in order to produce final products. There are different product families and two different transportation mode categories are available. The authors propose a coordinated approach using a MIP formulation and valid inequalities. They conclude that by using this coordinated approach one can reduce the costs when comparing to a hierarchical approach. Timpe and Kallrath [50] consider a multi-production site problem with different sales points. Different time scales are used for production sites and sales points and transportation can occur between production sites and sales points or between two different sales points. The authors propose a MIP model that can be used to generate near-optimal solutions. Dhaenens-Flipo and Finke [11] study a multi-production site in which each production site has multiple production lines. Production sites are not in the same location of the warehouses and each product for each client constitutes a distribution point. They use a MIP formulation that allows to solve small real-life industrial problems. Wu and Golbasi [55] treat a multi-production site problem in which a set of end products, composed by other intermediate products, should be produced over a time horizon. They propose a lagrangean decomposition scheme for the problem.

Some authors study the integration of production and transportation routing. In these types of problem one has to worry not only about the manufacturing at the production sites but also with the planning of the routes performed by the transportation vehicles to send the products to the clients, see Archetti et al. [2], Bard and Nananukul [5], Chandra [6], Chandra and Fisher [7], Fumero and Vercellis [20], and Strack and Pochet [48].

Chapter 3

Two-level production-in-series lot-sizing

In this chapter we study the uncapacitated two-level production-in-series lot-sizing problem. The problem is formally introduced in Section 3.1 (see also Section 2.7.1). In Section 3.2 we propose a new forward dynamic programming algorithm for the two-level problem whose running time is $O(NT^2 \log NT)$, which can be reduced to $O(NT^2)$ under certain assumptions on the cost structure. The corresponding tight extended formulation has $O(NT^3)$ non-negative variables and $O(NT^2)$ equality constraints and its linear description contains $O(NT^3)$ non-zero coefficients, all ± 1.

In Section 3.3 we treat an extension of the basic problem with sales. We give an $O(NT^4)$ dynamic programming algorithm and a resulting tight extended formulation with $O(NT^4)$ variables and $O(NT^3)$ constraints.

In Section 3.4 we give some reformulations for an extension with multiple items and multiple clients and compare them computationally. Specifically we compare a reformulation based on the tight extended formulation of Section 3.2, an echelon stock reformulation and a multi-commodity reformulation.

We finish the chapter with some concluding remarks in Section 3.5.

3.1 Problem Definition

We consider the uncapacitated two-level production-in-series lot-sizing problem (2L-S/LS-U). This can be thought of either as a two-level production problem in which level zero corresponds to the production of an intermediate product and level one to the final product, or as a production/transportation problem in which level zero ($l = 0$) is at a production site and level one ($l = 1$) is at a client site.

With $l \in \{0, 1\}$ and $1 \leq t \leq NT$, the input data for the problem is:

- d_t: the demand (at level one) for each period,
- f_t^l: the set-up cost at level l in period t,
- $\tilde{p}_t^l$: the production cost at level l in period t,
- $\tilde{h}_t^l$: the stock cost at level l in period t.

We define the variables:

- x_t^l: amount produced at level l in period t,
- y_t: is equal to 1 if production occurs at level 0 in period t and equal to 0 otherwise,
- Y_t: is equal to 1 if production occurs at level 1 in period t and equal to 0 otherwise,
- s_t^l: amount stocked at level l in the end of period t.

The resulting formulation is:

$$(2L-S/LS-U)\ \min \sum_{l=0}^{1}\sum_{t=1}^{NT} \tilde{p}_t^l x_t^l + \sum_{l=0}^{1}\sum_{t=1}^{NT} \tilde{h}_t^l s_t^l + \sum_{t=1}^{NT} f_t^0 y_t + \sum_{t=1}^{NT} f_t^1 Y_t$$

$$s_{t-1}^0 + x_t^0 = x_t^1 + s_t^0 \quad \forall t, \tag{3.1}$$

$$s_{t-1}^1 + x_t^1 = d_t + s_t^1 \quad \forall t, \tag{3.2}$$

$$x_t^0 \leq M y_t \quad \forall t, \tag{3.3}$$

$$x_t^1 \leq M Y_t \quad \forall t, \tag{3.4}$$

$$x, s \in \mathbb{R}_+^{2\times NT},\ y, Y \in \{0,1\}^{NT}. \tag{3.5}$$

In this formulation, constraints (3.1) are balance constraints at level zero in which the demand is the amount produced at level one. Constraints (3.2) are the balance constraints at level one. Constraints (3.3) and (3.4) force the set-up variables to take value 1 when there is positive production ($x_t^l > 0$).

Observation 3.1. *Assuming the initial stocks to be zero, we can eliminate the storage costs in the objective function using* $s_t^0 = \sum_{k=1}^{t}(x_k^0 - x_k^1)$ *and* $s_t^1 = \sum_{k=1}^{t} x_k^1 - d_{1t}$*, and we obtain*

$$\sum_{l=0}^{1}\sum_{t=1}^{NT}(\tilde{p}_t^l x_t^l + \tilde{h}_t^l s_t^l) = \sum_{t=1}^{NT}(\tilde{p}_t^0 + \sum_{k=t}^{NT}\tilde{h}_k^0)x_t^0 + \sum_{t=1}^{NT}[\tilde{p}_t^1 + \sum_{k=t}^{NT}(\tilde{h}_k^1 - \tilde{h}_k^0)]x_t^1 + K, \tag{3.6}$$

where $K = -\sum_{t=1}^{NT} \tilde{h}_t^1 d_{1t}$ *is a constant. This allows us to assume that the storage costs are equal to zero and to use the equivalent objective function*

$$\min \sum_{l=0}^{1} \sum_{t=1}^{NT} p_t^l x_t^l + \sum_{t=1}^{NT} f_t^0 y_t + \sum_{t=1}^{NT} f_t^1 Y_t + \sum_{l=0}^{1} \sum_{t=1}^{NT} 0 s_t^l. \tag{3.7}$$

3.2 Dynamic Programming Recursion and Extended Formulation

We assume that there are no initial and end stocks and that all the data (d, f, p) is nonnegative.

Zangwill [56] studied the multi-level uncapacitated production-in-series lot-sizing problem, a generalization of the problem studied in this chapter in which there are L levels. The author characterized extreme feasible solutions.

Proposition 3.1 (Zangwill [56])**.** *For the multi-level uncapacitated production-in-series lot-sizing problem, there exists an optimal solution with* $s_{t-1}^l x_t^l = 0$ *for all* l, t, *and if* $x_t^l > 0$, *then* $x_t^l = d_{ab}^l$ *for some* $t \leq a \leq b$. *For the last level,* $l = NT - 1$, $x_t^l = d_{tb}^l$, *i.e.* $a = t$.

Proposition 3.1 allow us to make the following observation.

Observation 3.2. *There exists an optimal solution that if a production batch* d_{lj} *is produced at level zero in* k ($x_k^0 = d_{lj}$) *the production sub-batches at level one are a refinement of those at level zero* ($x_a^1 = d_{ab}$ *with* $l \leq a \leq b \leq j$).

This observation is illustrated in Figure 3.1.

Consider the values

- $H(u,t)$: the minimum cost of satisfying demands from u to t at level one, calculated as

$$H(u,t) = \min_{u \leq j \leq t} \{H(u, j-1) + f_j^1 + p_j^1 d_{jt}\}, \tag{3.8}$$

  with $H(u, u-1) = 0$ for every $1 \leq u \leq NT$. In the solution depicted in Figure 3.2 we have $H(j,t) = H(j,j) + f_{j+1}^1 + p_{j+1}^1 d_{j+1,t}$.

- $G(t)$: the minimum cost of the two-level problem restricted to the periods 1 up to t, calculated as

$$G(t) = \min_{1 \leq j \leq t} \{G(j-1) + \min_{1 \leq i \leq j} (f_i^0 + p_i^0 d_{jt}) + H(j,t)\}, \tag{3.9}$$

  with $G(0) = 0$. In the solution of Figure 3.2, we have $G(t) = G(j-1) + (f_i^0 + p_i^0 d_{jt}) + H(j,t)$.

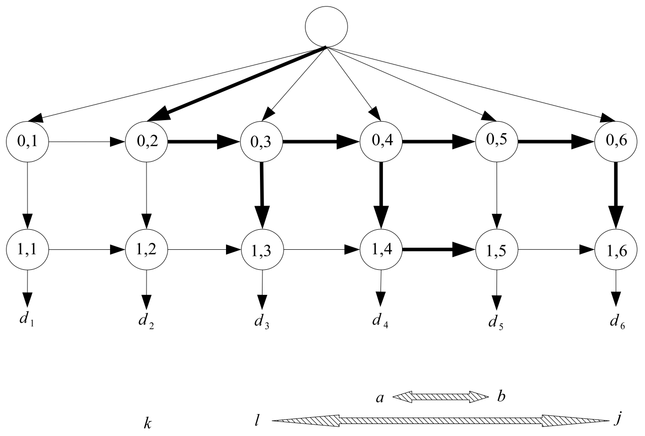


Figure 3.1: Illustrative example of Observation 3.2 with $k = 2$, $l = 3$, $a = 4$, $b = 5$ and $j = 6$.

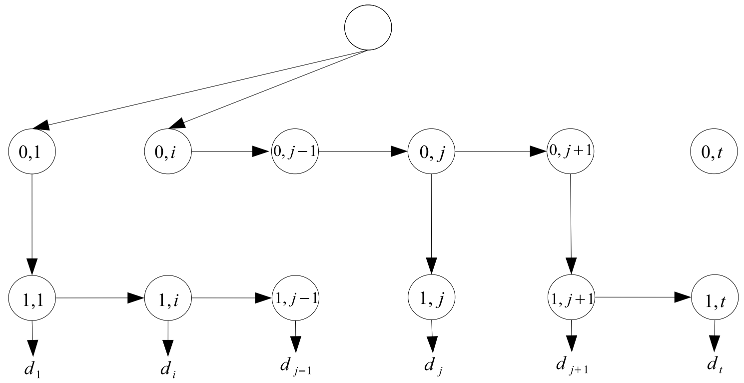


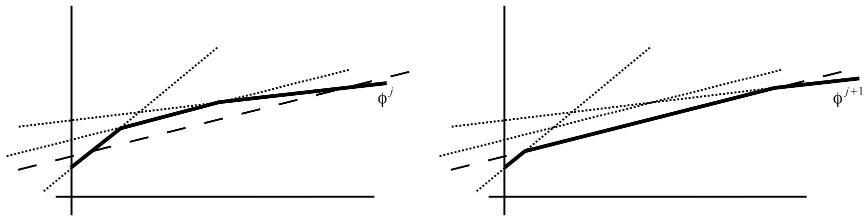
Figure 3.2: Example of a solution

Theorem 3.2. *There is an $O(NT^2 \log NT)$ algorithm for the uncapacitated two-level production-in-series lot-sizing problem.*

Proof: We show that $G(NT)$ can be calculated in $O(NT^2 \log NT)$. Starting at level one, $H(u,t)$, $u \leq t \leq NT$ can be calculated by solving an uncapacitated lot-sizing problem in $O(NT \log NT)$ for a fixed value of u, see Aggarwal and Park [1], Federgruen and Tzur [15] and Wagelmans et al. [53]. Therefore all values of H can be obtained in $O(NT^2 \log NT)$.

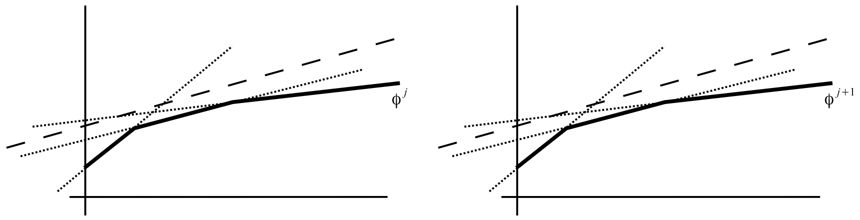
Next, as observed in Stratila [49], we calculate the values $\min_{1 \leq i \leq j}(f_i^0 + p_i^0 d_{jt})$ for all $1 \leq j \leq t \leq NT$ in $O(NT^2)$. Consider the piecewise linear concave functions: $\phi^j(b) = \min_{1 \leq i \leq j}(f_i^0 + p_i^0 b)$ for $1 \leq j \leq NT$. Each of them will be described by a set of at most $q_j \leq j$ triples $(\alpha_k^j, \beta_k^j, \gamma_k^j)$ where α_k^j are the breakpoints with $0 = \alpha_1^j < \alpha_2^j < \cdots < \alpha_{q_j}^j$, β_k^j are the slopes with $\beta_1^j > \beta_2^j > \cdots > \beta_{q_j}^j$ and $\gamma_k^j = \phi^j(\alpha_k^j)$. Now given any b, once one knows between which breakpoints it lies, $\phi^j(b)$ can be calculated in constant time. Clearly for $j = 1$, one just has the triple $\alpha_1^1 = 0, \beta_1^1 = p_1^0$ and $\gamma_1^1 = f_1^0$.

Now we indicate how ϕ^{j+1} can be obtained from ϕ^j in $O(NT)$. Specifically $\phi^{j+1}(b) = \min\{\phi^j(b), f_{j+1}^0 + p_{j+1}^0 b\}$. Observe that as k is increased from 1 to q_j, $\phi^j(\alpha_k^j) - [f_{j+1}^0 + p_{j+1}^0(\alpha_k^j)]$ changes sign at most twice. When there are exactly two changes, this leads to two new breakpoints and the removal of the nonempty set of old breakpoints lying between them as illustrated in Figure 3.3. The situation with less breakpoints is similar, as illustrated in Figures 3.4 and 3.5.



(a) Current function ϕ^j (b) Next function ϕ^{j+1}

Figure 3.3: Example with two new breakpoints

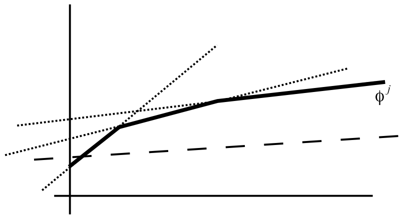


(a) Current function ϕ^j (b) Next function ϕ^{j+1}

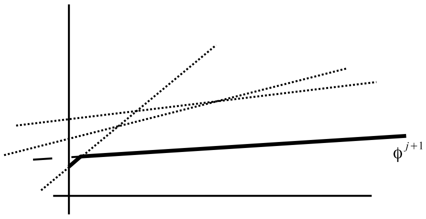
Figure 3.4: Example without new breakpoints

Next all the values of $\phi^j(d_{jt})$ need to be calculated. For fixed j this is carried out by merging the two increasing sequences $\{\alpha_k^j\}_{k=1}^{q_j}$ and $\{d_{jt}\}_{t=j}^{NT}$ in $O(NT)$. Thus again this step is $O(NT^2)$.

Finally, given the values $\phi^j(d_{jt})$, the calculation of $G(t)$ for all t based on (3.9) is $O(NT^2)$, and the claim follows. $\square$



(a) Current function ϕ^j



(b) Next function ϕ^{j+1}

Figure 3.5: Example with one new breakpoint

Example: Consider a case in which our current piecewise concave linear function ϕ^j is formed by the following linear functions: $g^a(x) = 5 + 1.2x$, $g^b(x) = 7 + 0.8x$ and $g^c(x) = 13 + 0.5x$. The new linear function to detemine ϕ^{j+1} is $g^d(x) = f^0_{j+1} + p^0_{j+1}x$ where $f^0_{j+1} = 4$ and $p^0_{j+1} = 0.8$. The example is illustrated in Figure 3.6. The demands to be considered are $d_{j+1,j+1} = 9$, $d_{j+1,j+2} = 25$ and $d_{j+1,j+3} = 35$.
The values corresponding to ϕ^j are:

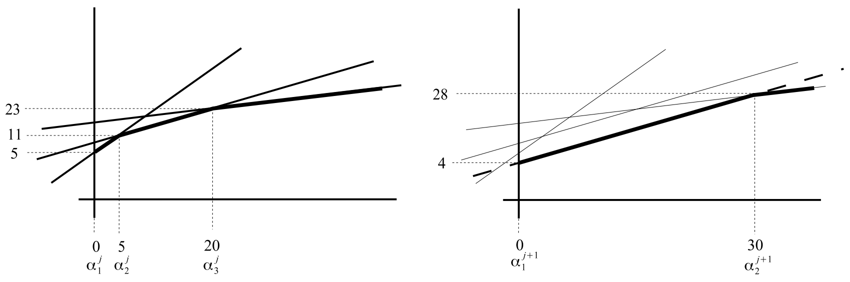
- $\alpha^j_1 = 0$, $\beta^j_1 = 1.2$, $\gamma^j_1 = 5$,
- $\alpha^j_2 = 5$, $\beta^j_2 = 0.8$, $\gamma^j_2 = 11$,
- $\alpha^j_3 = 20$, $\beta^j_3 = 0.5$, $\gamma^j_3 = 23$.

As $g^d(0) < \phi^j(0)$, $g^d(5) < \phi^j(5)$, $g^d(20) < \phi^j(20)$, the new function ϕ^{j+1} is then obtained while removing the breakpoints α^j_2 and α^j_3:

- $\alpha^{j+1}_1 = 0$, $\beta^{j+1}_1 = 0.8$, $\gamma^{j+1}_1 = 4$,
- $\alpha^{j+1}_2 = 30$, $\beta^{j+1}_2 = 0.5$, $\gamma^{j+1}_2 = 28$.

To calculate $\phi^{j+1}(d_{j+1,k})$ for $j + 1 \leq k \leq NT$, we merge the sequences $d = \{9, 25, 35\}$ and $\alpha^{j+1} = \{0, 30\}$. Whenever a demand value $d_{j+1,k}$ lies between two breakpoints α^{j+1}_a and α^{j+1}_{a+1}, the value of $\phi^{j+1}(d_{j+1,k})$ can be easily calculated as $\phi^{j+1}(d_{j+1,k}) = \gamma^{j+1}_a + \beta^{j+1}_a(d_{j+1,k} - \alpha^{j+1}_a)$. We start with the breakpoint $\alpha^{j+1}_1 = 0$, then we calculate $\phi^{j+1}(d_{j+1,j+1} = 9) = 4 + 0.8(9 - 0) = 11.2$ followed by $\phi^{j+1}(d_{j+1,j+2} = 25) = 4 + 0.8(25 - 0) = 24$, then we hit the breakpoint $\alpha^{j+1}_2 = 30$ and we finally calculate $\phi^{j+1}(d_{j+1,j+2} = 35) = 28 + 0.5(35 - 30) = 30.5$ for the last demand.

Note that when there are Wagner-Whitin (non-speculative) costs at level one, namely $p^1_t \geq p^1_{t+1}$ for all t, $H(u, t)$ can be calculated for all t in $O(NT)$, see again Aggarwal and Park [1], Federgruen and Tzur [15] and Wagelmans et al. [53], so all values of H can be obtained in $O(NT^2)$.



(a) Current function ϕ^j (b) Next function ϕ^{j+1}

Figure 3.6: Numerical example

Corollary 3.3. *When $p_t^1 - p_{t+1}^1 = (\tilde{p}_t^1 - \tilde{p}_{t+1}^1) + (\tilde{h}_t^1 - \tilde{h}_t^0) \geq 0$ for all t, there is an $O(NT^2)$ algorithm.*

Given the dynamic programming recursion (3.9)-(3.8), we can rewrite it as a linear programming formulation:

$$(DP)$$

$$z_{DP} = \max G(NT)$$

$$G(t) \leq G(j-1) + f_u^0 + p_u^0 d_{jt} + H(j,t) \quad \text{for } 1 \leq u \leq j \leq t \leq NT, \tag{3.10}$$

$$H(u,t) \leq H(u,j-1) + f_j^1 + p_j^1 d_{jt} \quad \text{for } 1 \leq u \leq j \leq t \leq NT, \tag{3.11}$$

$$G(t) \in \mathbb{R}_+ \quad \text{for } 1 \leq t \leq NT, \tag{3.12}$$

$$H(u,t) \in \mathbb{R}_+ \quad \text{for } 1 \leq u \leq t \leq NT. \tag{3.13}$$

We now present the dual formulation of DP. In this formulation, we have variables υ_{ujt} associated with constraints (3.10) and variables ω_{kjt} associated with constraints (3.11). The variables can be defined as follows:

- υ_{ujt}: is equal to one if production takes place at level zero in period u and the amount produced is d_{jt},
- ω_{ujt}: is equal to one if production of d_{jt} takes place at level one in period j using items from a production batch d_{uq} at level zero in period k with $[j,t]$ a subinterval of the interval $[u,q]$ and $k \leq u \leq j \leq t \leq q$.

Figure 3.7 gives an example of the interpretation of the variables of an optimal solution for a 5-period instance with data: $p = \begin{pmatrix} 3 & 2 & 4 & 3 & 2 \\ 4 & 5 & 5 & 2 & 1 \end{pmatrix}$, $f = \begin{pmatrix} 10 & 15 & 80 & 100 & 100 \\ 10 & 50 & 60 & 10 & 10 \end{pmatrix}$ and $d = \begin{pmatrix} 10 & 15 & 20 & 25 & 30 \end{pmatrix}$.

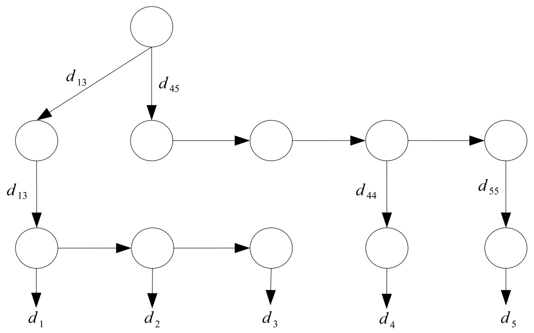


Figure 3.7: Solution with $\upsilon_{113} = \upsilon_{245} = 1$ and $\omega_{113} = \omega_{444} = \omega_{455} = 1$

$$(DDP)$$

$$z_{DDP} = \min \sum_{u=1}^{NT}\sum_{j=u}^{NT}\sum_{t=j}^{NT} \upsilon_{ujt}(f_u^0 + p_u^0 d_{jt}) + \sum_{u=1}^{NT}\sum_{j=u}^{NT}\sum_{t=j}^{NT} \omega_{ujt}(f_j^1 + p_j^1 d_{jt})$$

$$\sum_{u=1}^{NT}\sum_{j=u}^{NT} \upsilon_{uj,NT} = 1, \tag{3.14}$$

$$\sum_{u=1}^{t}\sum_{j=u}^{t} \upsilon_{ujt} - \sum_{u=1}^{t+1}\sum_{j=t+1}^{NT} \upsilon_{u,t+1,j} = 0 \quad \text{for } 1 \le t \le NT-1, \tag{3.15}$$

$$\sum_{j=u}^{t} \omega_{ujt} - \sum_{j=t+1}^{NT} \omega_{u,t+1,j} - \sum_{j=1}^{u} \upsilon_{jut} = 0 \quad \text{for } 1 \le u \le t \le NT, \tag{3.16}$$

$$\upsilon_{ujk},\ \omega_{ujk} \in \mathbb{R}_+ \quad \text{for } 1 \le u \le j \le k \le NT, \tag{3.17}$$

Constraints (3.14)-(3.15) are the "shortest path" constraints for level zero ensuring that the production batches d_{jt} exactly cover the total demand $d_{1,NT}$. Constraint (3.14) implies that there is one production batch finishing in period NT while constraints (3.15) say that if a production interval finishes in period t, then another one starts in period $t+1$. Constraints (3.16) indicate that if there is a subbatch d_{uk} produced at level 1 as part of a batch of d_{tq} at level 0 for some u between t and k ($\sum_{u=t}^{k} \omega_{tuk} = 1$), then either $k = q$ and d_{uk} was the last subbatch of d_{tq} ($\sum_{u=1}^{t} \upsilon_{utk} = 1$), or there is a following subbatch $d_{k+1,u}$ of d_{tq} ($\sum_{u=k+1}^{NT} \omega_{t,k+1,u} = 1$).

Linking the new variables of the formulation DDP to the original (x, y, Y) variables leads to formulation:

$$(DDP')\ z_{DDP'} = \min \sum_{l=0}^{1} \sum_{t=1}^{NT} p_t^l x_t^l + \sum_{t=1}^{NT} f_t^0 y_t + \sum_{t=1}^{NT} f_t^1 Y_t$$

$$(3.14) - (3.17)$$

$$\sum_{j=u}^{NT} \sum_{t=j}^{NT} \upsilon_{ujt} \leq y_u \quad \text{for } 1 \leq u \leq NT, \tag{3.18}$$

$$\sum_{j=u}^{NT} \sum_{t=j}^{NT} \upsilon_{ujt} d_{jt} = x_u^0 \quad \text{for } 1 \leq u \leq NT, \tag{3.19}$$

$$\sum_{u=1}^{j} \sum_{t=j}^{NT} \omega_{ujt} \leq Y_j \quad \text{for } 1 \leq j \leq NT, \tag{3.20}$$

$$\sum_{u=1}^{j} \sum_{t=j}^{NT} \omega_{ujt} d_{jt} = x_j^1 \quad \text{for } 1 \leq j \leq NT, \tag{3.21}$$

$$x \in \mathbb{R}_+^{2\times NT},\ y, Y \in [0,1]^{NT}. \tag{3.22}$$

Let Q be the polyhedron described by the constraints (3.14)-(3.22).

Theorem 3.4. *The linear program* $\min\{px + f^0y + f^1Y : (x, y, Y, \upsilon, \omega) \in Q\}$ *solves the two-level problem.*
$Proj_{x,y,Y}(Q)$ *is the convex hull of the set of points* (x, y, Y) *for which there exists an* s *with* (x, y, Y, s) *satisfying (3.1)-(3.5).*

3.3 An Algorithm for the Case with Sales

The uncapacitated two-level production-in-series lot-sizing problem with sales is an extension of the uncapacitated two-level production-in-series lot-sizing problem. Instead of producing only the demand d_t for each period t, an additional limited amount v_t can be produced in order to get some additional revenue, where e_t is the per unit revenue. A standard formulation for the problem is as follows.

$$(2L-S/LS-U-SL)$$

$$\max \sum_{t=1}^{NT} e_t v_t - \sum_{l=0}^{1}\sum_{t=1}^{NT} p_t^l x_t^l - \sum_{t=1}^{NT} f_t^0 y_t - \sum_{t=1}^{NT} f_t^1 Y_t$$

$$s_{t-1}^0 + x_t^0 = x_t^1 + s_t^0 \quad \forall t, \tag{3.23}$$

$$s_{t-1}^1 + x_t^1 = v_t + d_t + s_t^1 \quad \forall t, \tag{3.24}$$

$$0 \le v_t \le u_t \quad \forall t, \tag{3.25}$$

$$x_t^0 \le M y_t \quad \forall t, \tag{3.26}$$

$$x_t^1 \le M Y_t \quad \forall t, \tag{3.27}$$

$$x_t^l \in \mathbb{R}_+^1 \quad \forall l, t, \tag{3.28}$$

$$y, Y \in \{0,1\}^{NT}. \tag{3.29}$$

With the sales extension we have a fixed charge network flow problem of the form shown in Figure 3.8.

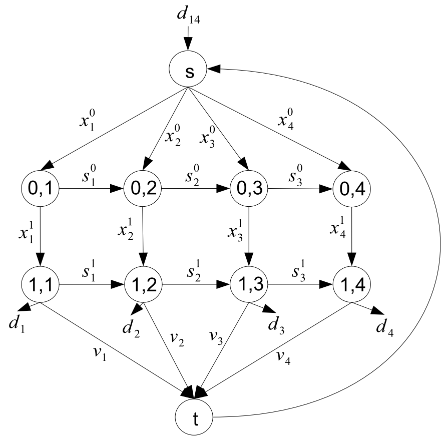


Figure 3.8: Network representation of the two-level lot-sizing with sales problem

The next two observations, namely Observations 3.3 and 3.4, are based on the acyclic property of the basic variables in a minimum cost network flow solution.

Observation 3.3. *In an extreme optimal solution every sales variable v_k assumes one of its bounds, i.e., either $v_k = 0$ or $v_k = u_k$.*

Proof. Note that all the variables in the network are uncapacitated with exception of the sales variables, therefore there is a path from the source to node

$(1, k)$ formed by basic variables whenever $d_k + v_k > 0$ for a period k. Assume by contradiction that we have an extreme optimal solution with $0 < v_k < u_k$ for some k. This implies that the arc corresponding to v_k forms a cycle with the basic variables on the path from the source to node $(1, k)$, see Figure 3.9. Therefore the solution is not extreme optimal and we have a contradiction. □

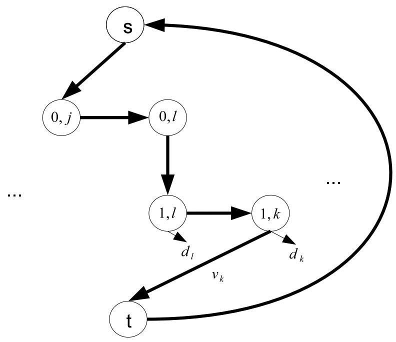


Figure 3.9: Cycle formed by a v_k variable that does not assume one of its bounds

Observation 3.4. *In an extreme optimal solution if production occurs in a period k, then the amount produced is used to completely satisfy demand and possible additional sales for an interval of consecutive periods $\{l, \ldots, j\}$ with $l \geq k$.*

Proof. This is a direct implication of Observations 3.2 and 3.3. □

Let the following values be defined as:

- $V(l, a, w)$: value of the optional revenue for producing the amount u_w at level 0 in period l and at level 1 in period a, calculated as

$$V(l, a, w) = \max\{0, (e_w - p_l^0 - p_a^1)u_w\}. \tag{3.30}$$

  In the solution of Figure 3.10, $V(1, 1, 1)$ and $V(u, k, t)$ take nonzero values.

- $B(l, a, b)$: revenue minus cost from satisfying the demands (plus possible additional sales) from periods a to b when the quantity $\sum_{j=a}^{b}(d_j + v_j)$ is produced at level 0 in period l and at level 1 in period a, calculated as

$$B(l, a, b) \geq \sum_{w=a}^{b} V(l, a, w) - (p_l^0 + p_a^1)d_{ab}. \tag{3.31}$$

  In the solution of Figure 3.10, $B(1, 1, j - 1)$, $B(u, j, j)$ and $B(u, k, t)$ contribute to the revenue minus cost.

- $H(l, j, k)$: maximum revenue minus cost from satisfying the demands (plus possible additional sales) from periods j to k when $\sum_{w=j}^{k}(d_w + v_w)$ units are produced at level 0 in period l, calculated as

$$H(l, j, k) = \max_{j \le w \le k} \{H(l, j, w-1) + B(l, w, k) - f_w^1\}, \tag{3.32}$$

  with $H(l, j, j-1) = 0$. In the solution depicted in Figure 3.10, we have $H(u, j, t) = H(u, j, j) + B(u, k, t) - f_k^1\}$.

- $G(k)$: optimal revenue for periods 1 to k, calculated as

$$G(k) = \max_{1 \le l \le j \le k} \{G(j-1) + H(l, j, k) - f_l^0\}, \tag{3.33}$$

  with $G(0) = 0$. In the solution of Figure 3.10, $G(t) = G(j-1) + H(u, j, t) - f_u^0$.

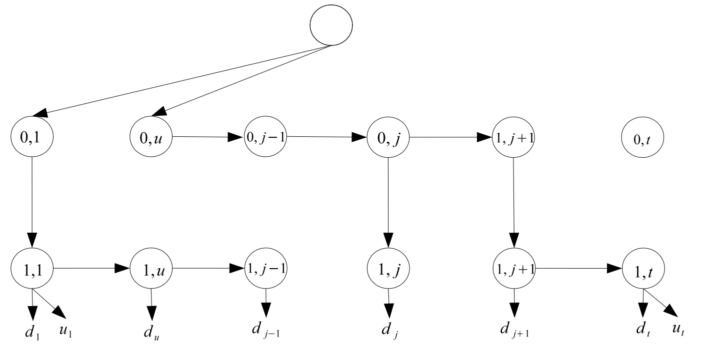


Figure 3.10: Part of a solution to illustrate the values $G(k)$, $H(l, j, k)$, $B(l, a, b)$ and $V(l, a, w)$.

Proposition 3.5. *There is an $O(NT^4)$ algorithm for solving the uncapacitated two-level lot-sizing with sales problem.*

Proof. All the $V(l, a, w)$ values can be calculated in $O(NT^3)$. All the $B(l, a, b)$ values can be calculated in $O(NT^3)$. The calculation of $H(l, j, k)$ for all l, j and k can be performed in $O(NT^4)$. The values of $G(k)$ for all k can be calculated in $O(NT^3)$. Therefore, the overall running time of the algorithm is $O(NT^4)$. □

3.3.1 An Extended Formulation

Using the DP recursion given by (3.33), (3.32), (3.31) and (3.30), we can write the following DP formulation.

$$z_{DPS} = \min G(NT)$$
$$V(l,a,w) \geq (e_w - p_l^0 - p_a^1)u_w \quad \forall 1 \leq l \leq a \leq w \leq NT, \tag{3.34}$$
$$B(l,a,b) \geq \textstyle\sum_{w=a}^{b} V(l,a,w) - (p_l^0 + p_a^1)d_{ab} \text{ for } 1 \leq l \leq a \leq b \leq NT, \tag{3.35}$$
$$H(l,j,k) \geq H(l,j,w-1) + B(l,w,k) - f_w^1 \text{ for } 1 \leq l \leq j \leq w \leq k \leq NT, \tag{3.36}$$
$$G(k) \geq G(j-1) + H(l,j,k) - f_l^0 \quad \text{for } 1 \leq l \leq j \leq k \leq NT, \tag{3.37}$$
$$V \in \mathbb{R}_+^{NT^3}, B \in \mathbb{R}^{NT^3}, H \in \mathbb{R}^{NT^3}, G \in \mathbb{R}^{NT}. \tag{3.38}$$

We present the dual formulation of DP followed by the interpretation of its variables. Associate variables α, β, γ and θ respectively to constraints (3.34), (3.35), (3.36) and (3.37).

$$z_{DDPS} = \max \textstyle\sum_{l,j,k} \alpha_{ljk}(e_k - p_l^0 - p_j^1)u_k - \sum_{l,j,k} \beta_{ljk}(p_l^0 + p_j^1)d_{jk} -$$
$$\textstyle\sum_{l,j,w,k} \gamma_{l,j,w,k} f_w^1 - \sum_{l,j,k} \theta_{ljk} f_l^0$$
$$\alpha_{law} - \textstyle\sum_{j=w}^{NT} \beta_{laj} \leq 0 \quad \forall 1 \leq l \leq a \leq w \leq NT \tag{3.39}$$
$$\beta_{ljk} - \textstyle\sum_{w=l}^{j} \gamma_{lwjk} = 0 \quad \text{for } 1 \leq l \leq j \leq k \leq NT, \tag{3.40}$$
$$\textstyle\sum_{w=j}^{k} \gamma_{ljwk} - \sum_{w=k+1}^{NT} \gamma_{lj,k+1,w} - \theta_{ljk} = 0 \text{ for } 1 \leq l \leq j \leq k \leq NT, \tag{3.41}$$
$$\textstyle\sum_{l=1}^{k} \sum_{j=l}^{k} \theta_{ljk} - \sum_{l=1}^{k+1} \sum_{j=k+1}^{NT} \theta_{l,k+1,j} = 0 \quad \text{for } 1 \leq k \leq NT, \tag{3.42}$$
$$\textstyle\sum_{l=1}^{NT} \sum_{j=l}^{NT} \theta_{lj,NT} = 1, \tag{3.43}$$
$$\alpha \in \mathbb{R}_+^{NT^3}, \beta \in \mathbb{R}_+^{NT^3}, \gamma \in \mathbb{R}_+^{NT^4}, \theta \in \mathbb{R}_+^{NT^3}. \tag{3.44}$$

We use an abuse of notation and denote "total demand" of an interval $[j,k]$ as $\sum_{l=j}^{k}(d_l + v_l)$. The variables in the formulation can be interpreted as follows (and are illustrated in Figure 3.11):

- α_{law}: is equal to one (if the variable takes a positive value, then constraint (3.39) will be tight and therefore this positive value will be 1) if $v_w = u_w$ with production in period l at level 0 and in period a at level 1 (associated with constraints (3.34)),
- β_{lab}: is equal to one if the "total demand" for the interval $[a,b]$ is produced in period l at level 0 and in period a at level 1 (associated with constraints (3.35)),
- γ_{ljwk}: is equal to one if the "total demand" for the interval $[w,k]$ is produced in l at level 0 as part of the "total demand" of an interval starting in period j (associated with constraints (3.36)),
- θ_{ljk}: is equal to one if the "total demand" of interval $[j,k]$ is produced in period l at level 0 (associated with constraints (3.37)).

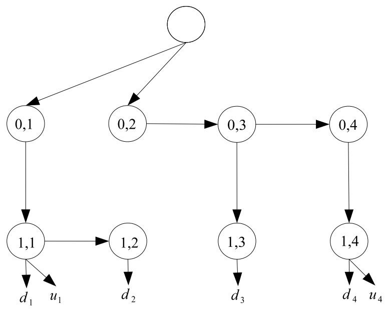


Figure 3.11: Solution with $\alpha_{111} = \alpha_{244} = \beta_{112} = \beta_{233} = \beta_{244} = \gamma_{1112} = \gamma_{2333} = \gamma_{2344} = \theta_{112} = \theta_{234} = 1$

Constraints (3.39) link the sales with the production. Constraints (3.40) state that if a batch $[j, k]$ was produced in period l at level 0 then it is part of a production batch at level 1 starting no later than period j. Constraints (3.41) indicate that if there is a subbatch $[w, k]$ produced at level one as part of a batch $[j, q]$ at level zero for some w between j and k ($\sum_{w=j}^{k} \gamma_{ljwk} = 1$), then either $k = q$ and $[w, k]$ was the last subbatch of $[j, q]$ ($\theta_{ljk} = 1$), or there is a following subbatch $[k+1, w]$ of $[j, q]$ ($\sum_{w=k+1}^{NT} \gamma_{l,j,k+1,w} = 1$). Constraints (3.42) and (3.43) are shortest path constraints for level zero.

We can link the variables of the formulation DDPS to the original (x, y, Y, v) variables to get the following formulation:

$$(DDPS') \ \max \sum_{t=1}^{NT} e_t v_t - \sum_{l=0}^{1} \sum_{t=1}^{NT} p_t^l x_t^l - \sum_{t=1}^{NT} f_t^0 y_t - \sum_{t=1}^{NT} f_t^1 Y_t$$

$$(3.39) - (3.44)$$

$$\sum_{j=t}^{NT} \sum_{k=j}^{NT} \theta_{tjk} \le y_t \quad \text{for } 1 \le t \le NT, \tag{3.45}$$

$$\sum_{j=t}^{NT} \sum_{k=j}^{NT} \beta_{tjk} d_{jk} = x_t^0 \quad \text{for } 1 \le t \le NT, \tag{3.46}$$

$$\sum_{l=1}^{k} \sum_{j=l}^{k} \sum_{w=k}^{NT} \gamma_{ljkw} \le Y_k \quad \text{for } 1 \le k \le NT, \tag{3.47}$$

$$\sum_{j=1}^{k} \sum_{t=k}^{NT} \beta_{jkt} d_{kt} = x_k^1 \quad \text{for } 1 \le k \le NT, \tag{3.48}$$

$$x \in \mathbb{R}_+^{2\times NT}, \ v \in \mathbb{R}^{NT}, \ y, Y \in [0,1]^{NT}. \tag{3.49}$$

Let Q^S be the polyhedron described by the constraints (3.39)-(3.49).

Theorem 3.6. *The linear program*

$$\max\{ev - px - f^0 y - f^1 Y : (x, y, Y, v, \alpha, \beta, \gamma, \theta) \in Q^S\}$$

solves the two-level problem with sales.
$Proj_{x,y,Y,v}(Q^S)$ is the convex hull of the set of points (x, y, Y, v) for which there exists an s with (x, y, Y, v, s) satisfying (3.23)-(3.29).

3.4 Some computations for an extension with multiple items and multiple clients

In this section we consider a two-level uncapacitated problem without sales with a single production site, but multiple items and multiple clients. The goal is to test the "tight" formulation developed in Section 3.2 with other effective (close to tight) formulations. Rather than work on the uncapacitated two-level production-in-series lot-sizing problem (that can be solved by dynamic programming), we test on an NP-hard problem that will enable us to tackle it by MIP which has the uncapacitated two-level production-in-series lot-sizing problem as a subproblem. We assume that vehicles of very large capacity transport items between the production site and the clients for which there is a variable transportation cost depending on the item and a fixed cost for the vehicle (whether it transports a positive amount of one or several items). The formulation is a straightforward generalization of that for the simple case analyzed in the previous sections. Now $\{1, \ldots, NI\}$ is the set of items with index i, $c = 0$ represents the production site and there is a set $\{1, \ldots, NC\}$ of client sites.

The variables are:

- x_t^{0i}: the amount of item i manufactured in the production site in period t,
- y_t^i: indicates that $x_t^{0i} > 0$ so that production takes place at the production site in period t,
- x_t^{1ic}: the amount of item i shipped from the production site to the client area c in period t,
- $Y_t^c = 1$: indicates that $\sum_i x_t^{1ic} > 0$ so that transport takes place between the production site and client c in period t.

A basic MIP formulation is

$$\min \sum_{i,t} p_t^{0i} x_t^{0i} + \sum_{i,t} f_t^{0i} y_t^i + \sum_{i,c,t} p_t^{1ic} x_t^{1ic} + \sum_{c,t} f_t^{1c} Y_t^c \tag{3.50}$$

$$s_{t-1}^{0i} + x_t^{0i} = \sum_{c=1}^{NC} x_t^{1ic} + s_t^{0i} \quad \text{for } 1 \leq i \leq NI, \ \ 1 \leq t \leq NT, \tag{3.51}$$

$$s_{t-1}^{1ic} + x_t^{1ic} = d_t^{ic} + s_t^{1ic} \quad \text{for } 1 \leq i \leq NI, \ 1 \leq c \leq NC, 1 \leq t \leq NT, \tag{3.52}$$

$$x_t^{0i} \leq M y_t^i \quad \text{for } 1 \leq i \leq NI, \ 1 \leq t \leq NT, \tag{3.53}$$

$$\sum_{i=1}^{NI} x_t^{1ic} \leq M Y_t^c \quad \text{for } \ 1 \leq c \leq NC, 1 \leq t \leq NT, \tag{3.54}$$

$$s^0, x^0 \in \mathbb{R}_+^{NI \times NT}, \ s^1 \in \mathbb{R}^{NI \times NC \times NT}, \ x^1 \in \mathbb{R}^{NI \times NC \times NT}, \tag{3.55}$$

$$y \in \{0,1\}^{NI \times NT}, \ Y \in \{0,1\}^{NC \times NT}. \tag{3.56}$$

3.4.1 Extended Formulations

In this section we present the reformulations that are going to be compared later in the computational results. An extended formulation based on the new tight formulation for the uncapacitated two-level production-in-series lot-sizing problem, an echelon stock reformulation and a multi-commodity extended formulation.

3.4.1.1 An Extended Formulation

Based on the extended formulation (3.14)-(3.22), one obtains the following extended formulation for the multi-item multi-client case.

With fixed values of the 0-1 variables, y_t^i and Y_t^c, the problem reduces to a single source network flow problem for each item, and the structure of the solution at each client is precisely the same as that for a single client, so that the variables v_{ujk}^{ic}, ω_{ujk}^{ic} have the same meaning as above with now i denoting the item and $c \in \{1, \ldots, NC\}$ the client.

$$(mDDP)$$

$$z_{mDDP} = \min \sum_{i,t} p_t^{0i} x_t^{0i} + \sum_{i,t} f_t^{0i} y_t^{0i} + \sum_{i,c,t} p_t^{1ic} x_t^{1ic} + \sum_{c,t} f_t^{1c} Y_t^{1c} \quad (3.57)$$

$$\textstyle\sum_{k=1}^{NT} \sum_{j=k}^{NT} \upsilon_{kj,NT}^{ic} = 1 \ \forall\ i, c, \quad (3.58)$$

$$\textstyle\sum_{k=1}^{t} \sum_{j=k}^{t} \upsilon_{kjt}^{ic} - \sum_{k=1}^{t+1} \sum_{j=t+1}^{NT} \upsilon_{k,t+1,j}^{ic} = 0 \ \forall i, c, t, \quad (3.59)$$

$$\textstyle\sum_{k=t}^{l} \omega_{tkl}^{ic} - \sum_{k=l+1}^{NT} \omega_{t,l+1,k}^{ic} - \sum_{k=1}^{t} \upsilon_{ktl}^{ic} = 0 \ \forall\ i, c,\ 1 \leq t \leq l \leq NT, \quad (3.60)$$

$$\textstyle\sum_{k=t}^{NT} \sum_{j=k}^{NT} \upsilon_{tkj}^{ic} \leq y_t^i \ \forall\ i, c, t, \quad (3.61)$$

$$\textstyle\sum_{c=1}^{NC} \sum_{k=t}^{NT} \sum_{j=k}^{NT} \upsilon_{tkj}^{ic} d_{kj}^{ic} = x_t^{0i} \ \forall\ i, t, \quad (3.62)$$

$$\textstyle\sum_{\gamma=1}^{k} \sum_{t=k}^{NT} \omega_{\gamma kt}^{ic} \leq Y_k^c \ \forall\ i, c, k, \quad (3.63)$$

$$\textstyle\sum_{\gamma=1}^{k} \sum_{t=k}^{NT} \omega_{\gamma kt}^{ic} d_{kt}^{ic} = x_k^{1ic} \ \forall\ i, c, k, \quad (3.64)$$

$$\upsilon_{ljk}^{ic},\ \omega_{ljk}^{ic} \in \mathbb{R}_+ \ \forall\ i, c,\ 1 \leq l \leq j \leq k \leq NT, \quad (3.65)$$

$$x^0 \in \mathbb{R}_+^{NI \times NT},\ x^1 \in \mathbb{R}_+^{NC \times NT},\ y \in \{0,1\}^{NI \times NT},\ Y \in \{0,1\}^{NC \times NT}. \quad (3.66)$$

Note however that it is not tight even in the single item case because this multi-client problem is the one-warehouse multi-retailer problem which is already NP-hard, see Arkin et al. [3].

3.4.1.2 Echelon stock reformulation

Our echelon stock reformulation is achieved by using reformulation results for some relaxations of the problem and use them together with the standard formulation (3.51)-(3.56).

For the simple uncapacitated lot-sizing problem

$$\begin{aligned} S_{t-1} + X_t = D_t + S_t, &\quad \text{for } 1 \leq t \leq NT, \\ X_t \leq MY_t, &\quad \text{for } 1 \leq t \leq NT, \\ X, S \in \mathbb{R}^{NT},\ Y \in \{0,1\}^{NT}, & \end{aligned}$$

the simple (l, S) inequalities, sufficient when the costs are *Wagner-Whitin*, are the inequalities

$$S_{k-1} \geq \sum_{j=k}^{l} D_j (1 - Y_k - \ldots - Y_j) \text{ for } 1 \leq k \leq l \leq NT.$$

We first consider the relaxation obtained by aggregating all the balance equations. This involves the so-called level zero echelon stock $e_t^{0i} \equiv \sum_{c=1}^{NC} s_t^{1ic}+$

s_t^{0i} consisting of the total stock of item i anywhere in the system.

$$e_{t-1}^{0i} + x_t^{0i} = \sum_{c=1}^{NC} d_t^{ic} + e_t^{0i} \quad \text{for } 1 \le i \le NI, \quad 1 \le t \le NT,$$
$$x_t^{0i} \le My_t^i \quad \text{for } 1 \le i \le NI, \ 1 \le t \le NT.$$

With $S_t = e_t^{0i}$, $X_t = x_t^{0i}$, $Y_t = y_t^i$ and $D_t = \sum_{c=1}^{NC} d_t^{ic}$, we derive the simple (l, S) inequalities for every product i.

Next we relax the set-up constraints for level one giving

$$s_{t-1}^{1ic} + x_t^{1ic} = d_t^{ic} + s_t^{1ic} \quad \text{for } 1 \le t \le NT,$$
$$x_t^{1ic} \le MY_t^c \quad \text{for } 1 \le t \le NT.$$

We derive the simple (l, S) inequalities using $S_t = s_t^{1ic}$, $X_t = x_t^{1ic}$, $Y_t = Y_t^c$ and $D_t = d_t^{ic}$ for every product i and client c.

Finally with multiple items, we sum over all the items giving

$$\sum_{i=1}^{NI} s_{t-1}^{1ic} + \sum_{i=1}^{NI} x_t^{1ic} = \sum_{i=1}^{NI} d_t^{ic} + \sum_{i=1}^{NI} s_t^{1ic} \quad \text{for } \ 1 \le t \le NT,$$
$$\sum_{i=1}^{NI} x_t^{1ic} \le MY_t^c \quad \text{for } 1 \le t \le NT.$$

and generate simple (l, S) inequalities using $S_t = \sum_{i=1}^{NI} s_t^{1ic}$, $X_t = \sum_{i=1}^{NI} x_t^{1ic}$, $Y_t = Y_t^c$ and $D_t = \sum_{i=1}^{NI} d_t^{ic}$ for every client area c.

3.4.1.3 Multicommodity extended formulation

In the multi-commodity formulation each demand d_t^{ic} for the triple i, c, t is viewed as a distinct product. Consider the variables

- w_{tu}^{0ic}: amount produced, at level zero, of product i in period t to satisfy demand of period u for client c.

- w_{tu}^{1ic}: amount produced, at level one, of product i in period t to satisfy demand of period u for client c.

- s_{tu}^{0ic}: amount stocked, at level zero, of product i to satisfy demand of period u for client c at the end of period t.

- s_{tu}^{1ic}: amount stocked, at level one, of product i to satisfy demand of period u for client c at the end of period t.

The multicommodity extended formulation is then given below.

$$
\begin{aligned}
(MC)\ z_{MC} = \min \textstyle\sum_{i,t} p_t^{0i} x_t^{0i} + \sum_{i,t} q_t^{0i} y_t^i + \sum_{i,c,t} p_t^{1ic} x_t^{1ic} + \sum_{c,t} f_t^{1c} Y_t^c \\
s_{t-1,u}^{0ic} + w_{tu}^{0ic} = w_{tu}^{1ic} + s_{tu}^{0ic} \ \text{ for } \forall\ i, c,\ 1 \le t \le u \le NT \\
s_{t-1,u}^{1ic} + w_{tu}^{1ic} = d_u^{ic} \delta_{tu}^{ic} + s_{tu}^{1ic} \ \text{ for } \forall\ i, c,\ 1 \le t \le u \le NT \\
w_{tu}^{0ic} \le d_u^{ic} y_t^i \ \text{ for } \forall\ i, c,\ 1 \le t \le u \le NT \\
w_{tu}^{1ic} \le d_u^{ic} Y_t^c \ \text{ for } \forall\ i, c,\ 1 \le t \le u \le NT \\
w_{tu}^{0ic}, w_{tu}^{1ic}, s_{tu}^{1ic}, s_{tu}^{0ic} \in \mathbb{R}_+^1 \ \text{ for } \forall\ i, c,\ 1 \le t \le u \le NT \\
y_t^i \in \{0, 1\} \ \text{ for } \forall i,\ 1 \le t \le NT \\
Y_t^c \in \{0, 1\} \ \text{ for } \forall c,\ 1 \le t \le NT \\
x_t^{0i} = \textstyle\sum_{c=1}^{C} \sum_{u=t}^{NT} w_{tu}^{0ic} \ \text{ for } \forall i,\ 1 \le t \le NT \\
x_t^{1ic} = \textstyle\sum_{u=t}^{NT} w_{tu}^{1ic} \ \text{ for } \forall i, c,\ 1 \le t \le NT,
\end{aligned}
$$

where δ_{tu}^{ic} equals one if $t = u$ and zero otherwise.

3.4.2 Computational results

To briefly test the effectiveness of the formulation (3.57)-(3.66), we have generated groups of five random instances each with the dimensions described in Table 3.1. All cost data is time independent and all random data is selected uniformly within the specified range:
- for groups G1, G2 and G3 of single item instances, $\tilde{h}^{0i} \in [0, 0.2]$, $\tilde{h}^{1ic} - \tilde{h}^{0i} \in [0, 0.25]$, $\tilde{p}^{0i} = 0$, $\tilde{p}^{1ic} = 0$, $f^{0i} \in [0, 500]$, $f^{1c} \in [0, 40]$ and $d_t^{ic} \in [0, 10]$,
- for groups G4, G5 and G6 with multiple items, $\tilde{h}^{0i} \in [0, 0.2]$, $\tilde{h}^{1ic} - \tilde{h}^{0i} \in [0, 0.05]$, $\tilde{p}^{0i} = 0$, $\tilde{p}^{1ic} = 0$, $f^{0i} \in [100, 400]$, $f^{1c} \in [20, 50]$ and $d_t^{ic} \in [0, 20]$.
The objective function is considered without the stock variables in the format of (3.6).

Group	Dimensions
G1	$NI = 1,\ NC = 20,\ NT = 30$
G2	$NI = 1,\ NC = 40,\ NT = 15$
G3	$NI = 1,\ NC = 50,\ NT = 20$
G4	$NI = 5,\ NC = 10,\ NT = 15$
G5	$NI = 5,\ NC = 20,\ NT = 18$
G6	$NI = 20,\ NC = 10,\ NT = 15$

Table 3.1: Parameters of the instances

All instances have been run using Xpress-MP version 2.4.1, on a Toshiba notebook with a 1.66 GHz Intel processor and 1 Gb of RAM memory. Each instance was run four times with a time limit of 300 seconds:
i) in default mode;
ii) adding (l, S) inequalities based on the echelon stock reformulation;

iii) using the multicommodity extended formulation;
iv) using the extended formulation (3.57)-(3.66).

Let vLP^j denote the value of the linear programming relaxation for instance j, $vXLP^j$ the value after the addition of system cuts, OPT^j the optimal value, BIP^j the value of the best solution found and BLB^j the value of the best lower bound. In Table 3.2, $LP = \frac{1}{5}(\sum_{j=1}^{5} 100 \sum \frac{vLP^j}{OPT^j})$ and $XLP = \frac{1}{5}(\sum_{j=1}^{5} 100 \frac{vXLP^j}{OPT^j})$ are the average percentage of the optimal value of the linear programming solution before and after the addition of system cuts, *Gap* is the average gap on termination for the instances not solved to optimality given by $100\frac{(BIP^j - BLB^j)}{BIP^j}$, *Time* and *Nodes* are respectively the average time and average number of nodes to prove optimality for the instances solved within the time limit.

Rows G1-G3 of Table 3.2 summarize the results for the single-item, multi-client instances. Using the default formulation, Xpress-MP could only solve two of the 15 instances to optimality. The two instances solved from G2 required an average time of 110.5 seconds. The echelon stock reformulation considerably improved the lower bounds, allowing one to solve all instances but one. For the unsolved instance from group G3 the final gap was 0.2%. The lower bounds given by the multicommodity extended formulation and by the new extended reformulation were very tight for these instances ($vLP^j < OPT^j$ for only one instance)^a and all the instances were solved at the top node. Time-wise the multicommodity extended formulation is clearly faster than the new reformulation.

	Standard			Echelon stock					MC		New	
Group	LP	XLP	Gap	LP	XLP	Gap	Time	Nodes	LP	Time	LP	Time
G1	72.4	91.5	10.2	97.6	98.7	–	76.6	374.6	100.0	4.0	100.0	39.2
G2	68.8	96.6	0.6	98.8	99.3	–	9.0	29.4	100.0	0.2	100.0	3.6
G3	69.8	91.7	10.1	98.1	98.9	0.2	59.5	154.5	100.0^a	5.6	100.0^a	21.2
G4	67.6	83.5	20.6	99.3	99.8	–	41.0	41.8	100.0	2.0	100.0	12.6
G5	71.6	85.6	19.6	99.3	99.6	3.5	–	–	100.0	22.0	100.0	128.6
G6	57.7	80.2	22.1	99.4	99.7	3.7	–	–	100.0^a	75.2	100.0^b	247

Table 3.2: Results for instances with one/multiple items and multiple clients

Rows G4-G6 of Table 3.2 summarize the results for the multi-item, multi-client instances. None of the instances in these groups could be solved to optimality with the standard formulation. Using the echelon stock reformulation, all the instances of group G4 were solved to optimality but all instances of groups G5 and G6 were unsolved after 300 seconds. Again the multicommodity formulation was very tight ($vLP^j < OPT^j$ in only two instances of group G6)^a. All were solved within the time limit. In addition, the instances in groups G4 and G5 were solved at the top node. The new formulation is also tight for these instances, and those from groups G4 and G5 were solved at the top node within the time limit. However for group G6 the LP relaxation was not solved within 300 seconds^b for 4 of the 5 instances.

The assumption made in the test instances that $\tilde{h}^{1ic} \geq h^{0i}$ is normal in a two-level production system, but perhaps less so in a production/transportation setting. Results for a few instances in which this condition did not hold gave similar results to those shown in Table 3.2.

3.5 Concluding Remarks

We proposed a new dynamic programming algorithm for the uncapacitated two-level production-in-series lot-sizing problem. The new $O(NT^2 \log NT)$ algorithm improves on the complexity of the best known algorithm in the literature. As a consequence, we also presented a new compact formulation for the problem with $O(NT^3)$ variables and $O(NT^2)$ constraints. Under some special conditions on the cost there is an $O(NT^2)$ algorithm for the problem.

Pătraşcu and Stratila [37] presented a new $O(NT \log\log NT)$ algorithm for the uncapacitated lot-sizing problem. Given this result we could reduce the complexity to solve our recursion to $O(NT^2 \log\log NT)$.

We also presented an $O(NT^4)$ dynamic programming algorithm for the uncapacitated two-level production-in-series lot-sizing problem with sales as well as a tight extended formulation with $O(NT^4)$ variables and $O(NT^3)$ constraints.

In addition we presented some computational results comparing three reformulation approaches. In our results, we observed that even if the new reformulation is theoretically stronger than the multi-commodity reformulation, it was outperformed in practice when applied to an extension of the uncapacitated two-level production-in-series lot-sizing problem.

Some of the results of this chapter appeared in [33].

In the next chapter we will be analyzing an NP-Hard extension of the uncapacitated two-level production-in-series lot-sizing problem in which level one consists of multiple clients, known as the one-warehouse multi-retailer problem.

Chapter 4

One-Warehouse Multi-Retailer Problem

In this chapter we consider the one-warehouse multi-retailer problem (OWMR), a special case of the multiple production site (or warehouse) problem to be studied in Chapter 5. We treat the OWMR separately because it has been treated in some detail in the literature and some reformulations that are not easily applied to multiple production site problems can be used to tackle it. In Section 4.1 we describe the problem and present a standard MIP formulation (see also Section 2.7.3). In Section 4.2 we present some properties of optimal solutions of the problem. There are different possible approaches to deal with the OWMR. In Section 4.3 we present three available reformulations for the problem and give a theoretical comparison of two of them. Using the (l, S) inequalities available for the uncapacitated two-level production-in-series lot-sizing problem is a first natural option to tackle the OWMR in a reduced space of variables. An extension is to consider the dicut collection inequalities of which the (l, S) inequalities are a very special case that may give much stronger results. The problem with the dicut collection inequalities, however, is that there are too many of them. In order to try to characterize which of these dicut inequalities are needed, we analyze in Section 4.4 the projection of the multi-commodity formulation in the space of original variables for two simple cases, namely the joint-replenishment problem and the uncapacitated two-level production-in-series lot-sizing problem. In Section 4.5 we characterize explicitly some of the dicut inequalities in the original space of variables of the uncapacitated two-level production-in-series lot-sizing problem and give a separation algorithm. In Section 4.6 we perform some limited computational experiments. First we compare the extended formulations of Section 4.3 and later we compare the use of a cutting plane algorithm using the characterized dicut inequalities with the multi-commodity and echelon stock reformulations. We make some final remarks in Section 4.7.

4.1 Problem Description and Formulation

There is one production site which replenishes multiple (NC) clients over a finite time horizon of NT periods. Each client has a time varying deterministic demand d_t^c for each period t in the time horizon. The amount produced by the production site and the amount transported from the production site to the clients are unrestricted. Storage is allowed at the production site and at the clients. Figure 4.1 shows an example with three clients.

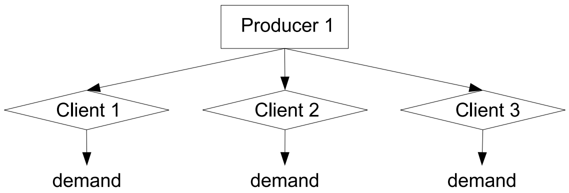


Figure 4.1: Example with one production site and three clients

Consider the variables

- x_t^0: amount produced at the production site in period t,
- x_t^{1c}: amount transported from the production site to client c in period t,
- s_t^0: amount in stock at the production site at the end of period t,
- s_t^{1c}: amount in stock at client c at the end of period t,
- y_t: is equal to 1 if production occurs at the production site in period t and 0 otherwise,
- Y_t^c: is equal to 1 if transportation occurs between the production site and client c in period t and 0 otherwise.

A standard formulation for the problem, as a two-level problem, is given by

$$(OWMR)$$

$$\min \sum_t (h_t^0 s_t^0 + p_t^0 x_t^0 + f_t^0 y_t) + \sum_{c,t} (h_t^{1c} s_t^{1c} + p_t^{1c} x_t^{1c} + f_t^{1c} Y_t^c) \tag{4.1}$$

$$s_{t-1}^0 + x_t^0 = \sum_{c=1}^{NC} x_t^{1c} + s_t^0, \quad \forall t, \tag{4.2}$$

$$s_{t-1}^{1c} + x_t^{1c} = d_t^c + s_t^{1c}, \quad \forall c, t, \tag{4.3}$$

$$x_t^0 \le M y_t, \quad \forall t, \tag{4.4}$$

$$x_t^{1c} \le M Y_t^c, \quad \forall c, t, \tag{4.5}$$

$$s^0, x^0 \in \mathbb{R}_+^{NT}, s^1, x^1 \in \mathbb{R}_+^{NC \times NT}, \tag{4.6}$$

$$y \in \{0,1\}^{NT}, \ Y \in \{0,1\}^{NC \times NT}. \tag{4.7}$$

4.2 Properties of Optimal Solutions

The one-warehouse multi-retailer problem can be represented as a fixed charge network flow problem, as illustrated in Figure 4.2. According to the property

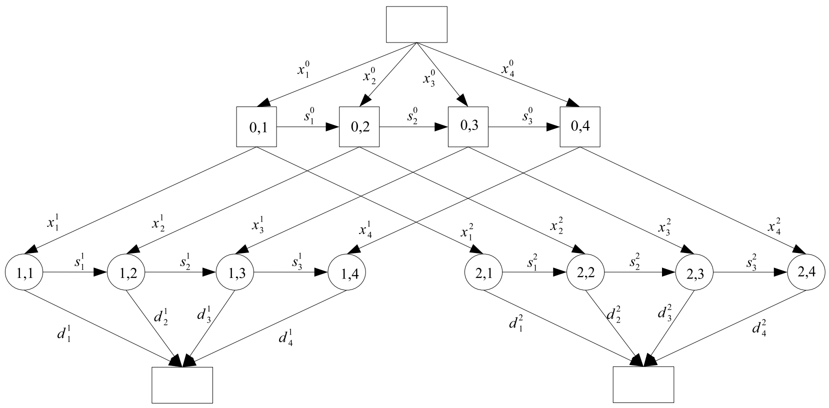


Figure 4.2: Network flow representation of the OWMR

of extreme flows in a network, there exists an optimal solution such that:

- if there is a positive entering stock at the production site/client in the beginning of period t, the flow arriving as production/transportation at t is equal to zero,
- if there is a positive transportation from the production site to a client c in period s it is used to satisfy demand from period s to period t, with $s \le t$.

4.3 Reformulations

One of the approaches in the literature to tackle the OWMR is the use of extended formulations. In this section we present reformulations available for the one-warehouse multi-retailer problem, namely the so-called multi-commodity formulation having $O(NC \times NT^2)$ variables, and the transportation and the shortest-path reformulations having $O(NC \times NT^3)$ variables. After presenting the formulations we compare theoretically the multi-commodity formulation with the transportation formulation, which was effective in some computational results performed in Solyali and Sural [46]. Specifically we show that the multi-commodity formulation is as strong as the transportation formulation.

4.3.1 Multi-commodity Formulation

In the multi-commodity formulation each demand d_t^c for the pair c, t is viewed as a distinct product. We consider the variables

- w_{kt}^{0c}: amount produced in the production site in period k to satisfy demand of client c in period t,
- w_{kt}^{1c}: amount transported from the production site to client c in period k to satisfy demand of period t,
- σ_{kt}^{0c}: amount stocked in the production site at the end of period k to satisfy demand of client c in period t,
- σ_{kt}^{1c}: amount stocked in client c at the end of period k to satisfy demand of period t.

The multi-commodity formulation is as follows.

$$(OWMR - MC)$$

$$\sigma_{k-1,t}^{0c} + w_{kt}^{0c} = w_{kt}^{1c} + \sigma_{kt}^{0c}, \quad \forall c, k, t \text{ with } k \le t, \tag{4.8}$$

$$\sigma_{k-1,t}^{1c} + w_{kt}^{1c} = \delta_{kt} d_t^c + (1 - \delta_{kt})\sigma_{k,t}^{1c}, \quad \forall c, k, t \text{ with } k \le t, \tag{4.9}$$

$$w_{kt}^{0c} \le y_k d_t^c, \quad \forall c, k, t \text{ with } k \le t, \tag{4.10}$$

$$w_{kt}^{1c} \le Y_k^c d_t^c, \quad \forall c, k, t \text{ with } k \le t, \tag{4.11}$$

$$\textstyle\sum_{c=1}^{NC} \sum_{t=k}^{NT} w_{kt}^{0c} = x_k^0, \quad \forall k, \tag{4.12}$$

$$\textstyle\sum_{t=k}^{NT} w_{kt}^{1c} = x_k^{1c}, \quad \forall c, k, \tag{4.13}$$

$$w^0, w^1 \in \mathbb{R}_+^{NC \times NT \times NT}, \tag{4.14}$$

$$y \in \{0,1\}^{NT},\ Y \in \{0,1\}^{NC \times NT}, \tag{4.15}$$

where δ_{kt} is equal to 1 if $k = t$ and 0 otherwise.

Constraints (4.8) are the balance constraints for each commodity at the production site. Constraints (4.9) are the balance constraints for each commodity at the clients. Constraints (4.10) and (4.11) set the binary variables to 1 in case production/transportation occurs. Constraints (4.12) and (4.13) link the multi-commodity variables to the original variables.

4.3.2 Transportation Formulation

Consider the variables

- λ^c_{skt} is the amount produced in the production site in period s, transported to client c in period k to satisfy its demand of period t.

Levi et al. [26] formulated the OWMR problem with a transportation formulation as follows.

$$(OWMR-TR) \qquad \textstyle\sum_{s=1}^{t}\sum_{k=s}^{t}\lambda^c_{skt} = d^c_t, \quad \forall c,t, \tag{4.16}$$
$$\textstyle\sum_{k=s}^{t}\lambda^c_{skt} \le y_s d^c_t, \quad \forall c,s,t \text{ with } s \le t, \tag{4.17}$$
$$\textstyle\sum_{s=1}^{k}\lambda^c_{skt} \le Y^c_k d^c_t, \quad \forall c,k,t \text{ with } k \le t, \tag{4.18}$$
$$\textstyle\sum_{c=1}^{NC}\sum_{k=s}^{NT}\sum_{t=k}^{NT}\lambda^c_{skt} = x^0_s, \quad \forall s, \tag{4.19}$$
$$\textstyle\sum_{s=1}^{k}\sum_{t=k}^{NT}\lambda^c_{skt} = x^{1c}_k, \quad \forall c,k, \tag{4.20}$$
$$\lambda \in \mathbb{R}_+^{NC\times NT\times NT\times NT}, \tag{4.21}$$
$$y \in \{0,1\}^{NT},\ Y \in \{0,1\}^{NC\times NT}. \tag{4.22}$$

Constraints (4.16) ensure each of the demands is satisfied. Constraints (4.17) and (4.18) set the binary variables to 1 in case production/transportation occurs. Constraints (4.19) and (4.20) link the transportation variables to the original variables.

4.3.3 Shortest Path Formulation

The shortest path formulation, used in Solyali and Sural [46], is valid due to the properties of optimal solutions presented in Section 4.2. Consider the variables

- ϕ^c_{srt} is the fraction of d^c_{rt} ($\equiv \sum_{j=r}^{t} d^c_j$) manufactured by the production site in s and sent to client c in period r to satisfy its demands from period r to t.

The formulation is as follows.

$$(OWMR - SP)$$

$$\textstyle\sum_{t=1}^{NT} \phi_{11t}^c = 1, \quad \forall c, \tag{4.23}$$

$$\textstyle\sum_{r=1}^{t-1} \sum_{s=r}^{t-1} \phi_{rs,t-1}^c - \sum_{r=1}^{t} \sum_{s=t}^{NT} \phi_{rts}^c = 0, \quad \forall c, t \geq 2, \tag{4.24}$$

$$\textstyle\sum_{r=s:d_{rt}^c>0}^{t} \phi_{srt}^c \leq y_s, \quad \forall c, s, t \text{ with } s \leq t, \tag{4.25}$$

$$\textstyle\sum_{s=1:d_{rt}^c>0}^{r} \phi_{srt}^c \leq Y_r^c, \quad \forall c, r, t \text{ with } r \leq t, \tag{4.26}$$

$$\textstyle\sum_{c=1}^{NC} \sum_{r=s}^{NT} \sum_{t=r}^{NT} d_{rt}^c \phi_{srt}^c = x_s^0, \quad \forall s, \tag{4.27}$$

$$\textstyle\sum_{s=1}^{r} \sum_{t=r}^{NT} d_{rt}^c \phi_{srt}^c = x_r^{1c}, \quad \forall c, k, \tag{4.28}$$

$$\phi \in [0,1]^{NC \times NT \times NT \times NT}, \tag{4.29}$$

$$y \in \{0,1\}^{NT}, \ Y \in \{0,1\}^{NC \times NT}. \tag{4.30}$$

Constraints (4.23) and (4.24) are shortest path constraints. Constraints (4.25) and (4.26) set the binary variables to 1 in case production/transportation occurs. Constraints (4.27) and (4.28) link the transportation variables to the original variables.

4.3.4 Comparing the Multi-commodity and the Transportation Formulations

In an attempt to analyze the strength of formulations having different sizes, we now compare the linear relaxation of the multi-commodity formulation with that of the transportation formulation. We show that the linear relaxation of the transportation formulation is an extended formulation for the linear relaxation of the multi-commodity formulation and this will result in the corollary that both formulations have the same linear relaxation bound.

Proposition 4.1. *The linear relaxation of $OWMR - TR$ is an extended formulation for the linear relaxation of $OWMR - MC$.*

Proof. Let $P^{OWMR-MC}$ and $P^{OWMR-TR}$ be respectively the sets of feasible solutions of the linear relaxations of $OWMR - MC$ and $OWMR - TR$.

Let $(x^0, x^1, y, Y, w^0, w^1) \in P^{OWMR-MC}$ and $(x^0, x^1, y, Y, w^0, w^1, \lambda) \in P^{OWMR-TR}$, where the link between λ and w in $OWMR - TR$ is given by

$$w_{st}^{0c} = \sum_{k=s}^{t} \lambda_{skt}^c, \tag{4.31}$$

$$w_{kt}^{1c} = \sum_{s=1}^{k} \lambda_{skt}^c. \tag{4.32}$$

First we want to demonstrate that if the solution $(\hat{x}^0, \hat{x}^1, \hat{y}, \hat{Y}, \hat{w}^0, \hat{w}^1, \hat{\lambda}) \in P^{OWMR-TR}$ then $(\hat{x}^0, \hat{x}^1, \hat{y}, \hat{Y}, \hat{w}^0, \hat{w}^1) \in P^{OWMR-MC}$. Using (4.31) and (4.32), nonnegativity of λ imply nonnegativity of w^0 and w^1 and constraints (4.10), (4.11), (4.12) and (4.13) follow by substitution. It suffices to show that constraints (4.8) and (4.9) are satisfied. Note that we can eliminate the stock variables σ^0 and σ^1 and rewrite (4.8) and (4.9) respectively as

$$\textstyle\sum_{j=1}^{k} w_{jt}^{0c} \geq \sum_{j=1}^{k} w_{jt}^{1c}, \quad \forall c, k, t \text{ with } k \leq t, \tag{4.33}$$

$$\textstyle\sum_{k=1}^{t} w_{kt}^{1c} = d_t^c, \quad \forall c, t. \tag{4.34}$$

Constraints (4.16) imply

$$\sum_{s=1}^{k} \hat{w}_{st}^{0c} = \sum_{s=1}^{k} \sum_{j=s}^{t} \hat{\lambda}_{sjt}^{c} \geq \sum_{m=1}^{k} \sum_{l=m}^{k} \hat{\lambda}_{mlt}^{c} = \sum_{l=1}^{k} \sum_{m=1}^{l} \hat{\lambda}_{mlt}^{c} = \sum_{l=1}^{k} \hat{w}_{lt}^{1c}, \tag{4.35}$$

where the inequality holds due to the nonnegativity of the variables, and

$$\sum_{k=1}^{t} \hat{w}_{kt}^{1c} = \sum_{k=1}^{t} \sum_{s=1}^{k} \hat{\lambda}_{skt}^{c} = \sum_{s=1}^{t} \sum_{k=s}^{t} \hat{\lambda}_{skt}^{c} = d_t^c. \tag{4.36}$$

Thus, we have that (4.35) and (4.36) respectively imply (4.33) and (4.34).

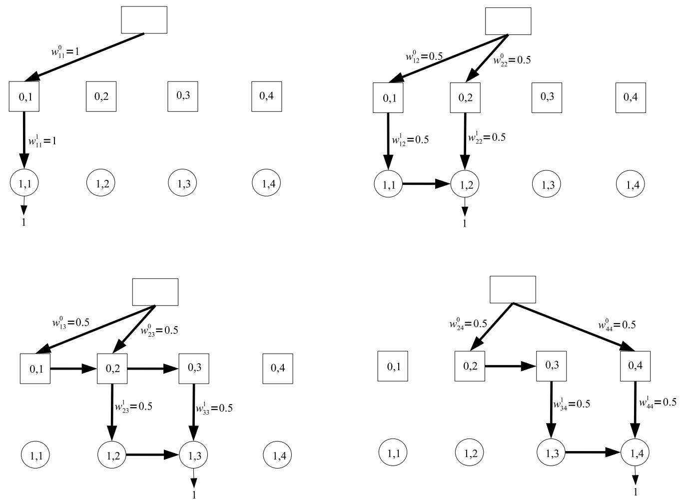
We now show that in case $(x^0, x^1, y, Y, w^0, w^1) \in P^{OWMR-MC}$, there exists λ such that $(x^0, x^1, y, Y, w^0, w^1, \lambda) \in P^{OWMR-TR}$. Observe that for each demand d_t^c, variables w_{kt}^{0c}, w_{kt}^{1c}, σ_{kt}^{0c} and σ_{kt}^{1c} related to commodity (c, t) describe a feasible flow of d_t^c units arriving in node (c, t). Note also that $\sum_{k=1}^{t} w_{kt}^{1c} = d_t^c$. In what follows we restrain our attention to the variables w^0, w^1, since σ^0 and σ^1 can be determined using the values of the former. According to the flow decomposition theorem (Theorem 2.8) any feasible flow on a network can be decomposed into paths and cycles. In our specific directed network structure there are no directed cycles, what implies the feasible flow determined by w^{0c}, w^{1c} can be decomposed into paths λ^c, as exemplified in Figure 4.3. Such a decomposition can be done by using a flow decomposition algorithm in a way that (4.31) and (4.32) are satisfied. By direct substitution using (4.31) and (4.32), the constraints (4.17), (4.18), (4.19) and (4.20) are satisfied. It suffices to show that (4.16) is also satisfied. We have

$$\sum_{k=1}^{t} \hat{w}_{kt}^{1c} = d_t^c,$$

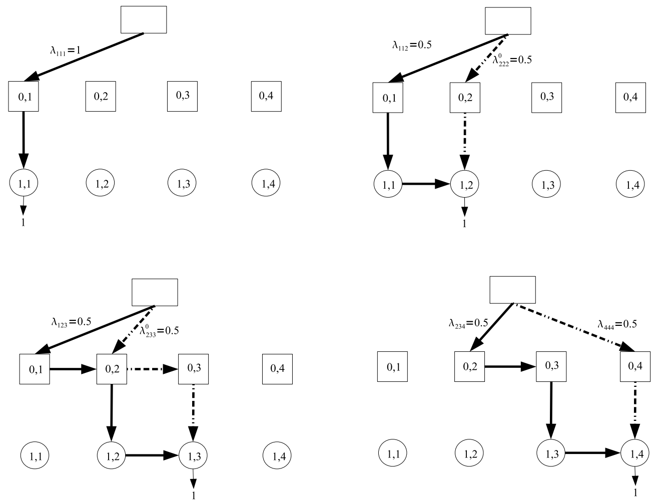
and by substitution

$$\sum_{s=1}^{t} \sum_{k=s}^{t} \hat{\lambda}_{skt}^{c} = \sum_{k=1}^{t} \hat{w}_{kt}^{1c} = d_t^c.$$

$\square$



(a) Flow represented by w^0 and w^1



(b) Flow decomposed in λ paths

Figure 4.3: Flow decomposition example

Let $z_{LP(OWMR-MC)}$, $z_{LP(OWMR-TR)}$ and $z_{LP(OWMR-SP)}$ denote respectively the linear relaxation bounds of $OWMR-MC$, $OWMR-TR$ and $OWMR-SP$. Proposition 4.1 implies the following corollary.

Corollary 4.2. $z_{LP(OWMR-MC)} = z_{LP(OWMR-TR)}$.

This result shows that it is possible to use a formulation that is one order of magnitude smaller and obtain a linear relaxation bound that is as strong as that of the larger formulation.

Solyali and Sural [46] showed that $z_{LP(OWMR-TR)} \leq z_{LP(OWMR-SP)}$, and therefore the next result follows immediately.

Corollary 4.3. $z_{LP(OWMR-MC)} \leq z_{LP(OWMR-SP)}$.

So even though the multi-commodity gives good bounds, there is still a formulation with $O(NC \times NT^3)$ variables that is stronger than it.

4.4 The Projection of the Multi-commodity Formulation

In this section we analyze the projection of the multi-commodity formulation into the original space for two particular cases of the one-warehouse multi-retailer problem. We first show that its projection for the joint-replenishment problem is composed of just simple dicut inequalities and then show that this property is not true for the uncapacitated two-level production-in-series lot-sizing problem.

4.4.1 On the Projection of the Multi-commodity Formulation for the Joint-replenishment Problem

We treat here the Joint-replenishment problem which is a special case of the one-warehouse multi-retailer problem in which storage is not allowed in the production site (see Section 2.7.2).

We consider the joint-replenishment problem formulated as a single-level lot-sizing with joint-setup costs.

$$s_{t-1}^c + x_t^c = d_t^c + s_t^c \quad \forall t, \tag{4.37}$$

$$x_t^c \leq My_t \quad \forall c, t, \tag{4.38}$$

$$x_t^c \leq MY_t^c \quad \forall c, t, \tag{4.39}$$

$$x_t^c, s_t^c \in \mathbb{R}_+^1,\ y_t \in \{0,1\},\ Y_t^c \in \{0,1\} \quad \forall c, t. \tag{4.40}$$

Consider the multi-commodity formulation as a separation problem. We are given a point $(\bar{x}^0, \bar{x}^1, \bar{y}, \bar{Y})$ and we want to determine whether or not it is feasible in the linear program:

$$
\begin{aligned}
(JRP-MC) \qquad & \max 0w \\
& \textstyle\sum_{t=1}^{k} w_{tk}^c = d_k^c \quad \forall c,k, && (4.41)\\
& w_{tk}^c \le d_k^c \bar{y}_t \quad \forall c,t,k, && (4.42)\\
& w_{tk}^c \le d_k^c \bar{Y}_t^c \quad \forall c,t,k, && (4.43)\\
& \textstyle\sum_{k=t}^{NT} w_{tk}^c \le \bar{x}_t^c \quad \forall c,t, && (4.44)\\
& w_{tk}^c \in \mathbb{R}_+^{NC\times NT\times NT}. && (4.45)
\end{aligned}
$$

Consider the dual variables ϵ, γ, θ and ϕ corresponding respectively to constraints (4.41), (4.42), (4.43) and (4.44). The dual linear program is

$$
\begin{aligned}
\min \textstyle\sum_{c,t} \phi_t^c \bar{x}_t^c + \sum_{c,t} \theta_{tk}^c d_k^c \bar{Y}_t^c + \sum_{c,t,k} \gamma_{tk}^c d_k^c \bar{y}_t + \sum_{c,k} \epsilon_k^c d_k^c && (4.46)\\
\gamma_{tk}^c + \theta_{tk}^c + \phi_t^c + \epsilon_k^c \ge 0 \quad \forall c,t,k, && (4.47)\\
\gamma, \theta, \phi \ge 0. && (4.48)
\end{aligned}
$$

A solution $(\bar{x}^0, \bar{x}^1, \bar{y}, \bar{Y})$ satisfies the constraints of the multi-commodity formulation if

$$\sum_{c,t} \phi_t^c \bar{x}_t^c + \sum_{c,t} \theta_{tk}^c d_k^c \bar{Y}_t^c + \sum_{c,t,k} \gamma_{tk}^c d_k^c \bar{y}_t + \sum_{c,k} \epsilon_k^c d_k^c \ge 0,$$

for all extreme rays (i.e. solutions to (4.47)-(4.48)).

Observe that the ϵ variables are the only free variables, and we have $\epsilon_k^c \le 0$ in an extreme ray with negative cost. We can, therefore, make a normalization by assuming $\epsilon_k^c \ge -1$.

Claim 4.4. *The matrix corresponding to constraints (4.47) is TU.*

Proof. Denote by A the matrix corresponding to (4.47). Observe that we have one identity submatrix corresponding to variables γ and another one corresponding to variables θ, such that $A = (I, I, B)$ and B is composed of a transportation matrix for each client c. Take as an example the matrix B with $c = 2$ and $NT = 3$ and consider the columns of B in the following order:

$\phi_1^1, \phi_2^1, \phi_3^1, \epsilon_1^1, \epsilon_2^1, \epsilon_3^1, \phi_1^2, \phi_2^2, \phi_3^2, \epsilon_1^2, \epsilon_2^2, \epsilon_3^2,$

$$B = \begin{pmatrix}
1 & & & 1 & & & & & & & & \\
1 & & & & 1 & & & & & & & \\
1 & & & & & 1 & & & & & & \\
 & 1 & & & 1 & & & & & & & \\
 & 1 & & & & 1 & & & & & & \\
 & & 1 & & & 1 & & & & & & \\
 & & & & & & 1 & & & 1 & & \\
 & & & & & & 1 & & & & 1 & \\
 & & & & & & 1 & & & & & 1 \\
 & & & & & & & 1 & & & 1 & \\
 & & & & & & & 1 & & & & 1 \\
 & & & & & & & & 1 & & & 1
\end{pmatrix}.$$

We want to show that the conditions in Observation 2.3 are satisfied by B^T. Condition (a) is satisfied since $b_{ij} \in \{0, 1\}$. In order to satisfy (b), we build M_1 and M_2 as follows.
For every line $i \in M$ related to an ϵ variable, add i to M_1.
For every line $i \in M$ related to an ϕ variable, add i to M_2.
B is totally unimodular since B^T is totally unimodular.

□

Proposition 4.5. *$proj_{(x,y,Y)} JRP - MC$ is composed of just simple dicut inequalities.*

Proof. Claim 4.4 implies that the solution to the following problem is integer.

$$z'_D = \min \textstyle\sum_{c,t} \phi_t^c \bar{x}_t^c + \sum_{c,t,k} \theta_{tk}^c d_k^c \bar{Y}_t^c + \sum_{c,t,k} \gamma_{tk}^c d_k^c \bar{y}_t + \sum_{c,k} \epsilon_k^c d_k^c \quad (4.49)$$
$$\gamma_{tk}^c + \theta_{tk}^c + \phi_t^c + \epsilon_k^c \geq 0 \quad \forall c, t, k \quad (4.50)$$
$$\gamma, \theta, \phi \geq 0, \quad (4.51)$$
$$-1 \leq \epsilon \leq 0. \quad (4.52)$$

The variables can be iterpreted as follows:

- ϵ_k^c: is equal to -1 in case there is a $k-$dicut for period k in the inequality,
- γ_{tk}^c: is equal to 1 if y_t is contained in the $k-$dicut for period k,
- θ_{tk}^c: is equal to 1 if Y_t^c is contained in the $k-$dicut for period k,
- ϕ_t^c: is equal to 1 if x_t^c is contained in at least one $k-$dicut.

Constraints (4.50) imply that for each period k for which a dicut is considered ($\epsilon_k^c = -1$), there exists at least one variable for period $t \leq k$ cutting off the flow to period k.

$\square$

Observation 4.1. *There exists a most violated inequality in which $\phi_t^c + \gamma_{tk}^c + \theta_{tk}^c \leq 1$ for every c, t, k.*

Proof. We can simply rewrite (4.50) as $\gamma_{tk}^c + \theta_{tk}^c + \phi_t^c \geq -\epsilon_k^c$. The result follows for the fact that $\epsilon_k^c \in \{-1, 0\}$ and $\bar{x}, \bar{y}, \bar{Y}, \phi, \gamma, \theta \geq 0$. $\square$

4.4.2 On the Projection of the Multi-commodity Formulation for the $2L - S/LS - U$

We now show that contrary to what happens with the joint-replenishment problem, for the $2L - S/LS - U$ that cannot be reduced to a single level problem, the projection of the multi-commodity formulation into the space of the original variables is not composed of just simple dicut inequalities .

Consider again the multi-commodity formulation as a separation problem for the uncapacitated two-level production-in-series lot-sizing problem. The point $(\bar{x}^0, \bar{x}^1, \bar{y}, \bar{Y})$ is an input and we want to generate one valid inequality cutting off this point.

$$(2L - S/LS - U - MCp)$$

$$\max 0w$$

$$\sum_{k=1}^{s} w_{kt}^1 \leq \sum_{k=1}^{s} w_{kt}^0 \quad \forall\, 1 \leq s \leq t \leq NT, \tag{4.53}$$

$$\sum_{k=1}^{t} w_{kt}^1 = d_t \quad \forall\, 1 \leq t \leq NT, \tag{4.54}$$

$$w_{kt}^0 \leq \bar{y}_k d_t \quad \forall\, 1 \leq k \leq t \leq NT, \tag{4.55}$$

$$w_{kt}^1 \leq \bar{Y}_k d_t \quad \forall\, 1 \leq k \leq t \leq NT, \tag{4.56}$$

$$\sum_{t=k}^{NT} w_{kt}^0 \leq \bar{x}_k^0 \quad \forall\, 1 \leq k \leq NT, \tag{4.57}$$

$$\sum_{t=k}^{NT} w_{kt}^1 \leq \bar{x}_k^1 \quad \forall\, 1 \leq k \leq NT, \tag{4.58}$$

$$w^0, w^1 \in \mathbb{R}_+^{NT \times NT}. \tag{4.59}$$

Associate dual variables α, ϵ, γ, θ, ϕ, δ to constraints (4.53), (4.54), (4.55), (4.56), (4.57) and (4.58), respectively. The dual separation of (OWMR-MCp) is given by

$$(D2L - S/LS - U - MCp)$$

$$\min \sum_t \delta_t \bar{x}_t^1 + \sum_t \phi_t \bar{x}_t^0 + \sum_{k,t} \theta_{kt} \bar{y}_k d_t + \sum_{k,t} \gamma_{kt} \bar{Y}_k d_t + \sum_t \epsilon_t d_t$$

$$\sum_{s=k}^{t} \alpha_{st} + \epsilon_t + \gamma_{kt} + \delta_k \geq 0 \quad \forall\, 1 \leq k \leq t \leq NT, \tag{4.60}$$

$$-\sum_{s=k}^{t} \alpha_{st} + \theta_{kt} + \phi_k \geq 0 \quad \forall\, 1 \leq k \leq t \leq NT, \tag{4.61}$$

$$\alpha, \gamma, \theta, \phi, \delta \geq 0. \tag{4.62}$$

Our next result shows that contrary to what happens with the joint-replenishment problem, the projection of the multi-commodity into the space of the original variables is not composed of only simple dicut inequalities for the uncapacitated two-level production-in-series lot-sizing problem.

Proposition 4.6. *$proj_{x^0,x^1,y,Y} 2L-S/LS-U-MCp$ is not composed of just simple dicut inequalities.*

Proof. We show an example where the dual separation problem D2L-S/LS-U-MCp gives rise to a facet-defining dicut inequality that is not a simple dicut inequality. Consider the example with demands $d = (8, 3, 6, 12)$.

The standard formulation $STD - ex$ for the example is

$$x_1^0 \geq x_1^1, \tag{4.63}$$
$$x_1^0 + x_2^0 \geq x_1^1 + x_2^1, \tag{4.64}$$
$$x_1^0 + x_2^0 + x_3^0 \geq x_1^1 + x_2^1 + x_3^1, \tag{4.65}$$
$$x_1^0 + x_2^0 + x_3^0 + x_4^0 = x_1^1 + x_2^1 + x_3^1 + x_4^1, \tag{4.66}$$
$$x_1^1 \geq 8, \tag{4.67}$$
$$x_1^1 + x_2^1 \geq 8 + 3, \tag{4.68}$$
$$x_1^1 + x_2^1 + x_3^1 \geq 8 + 3 + 6, \tag{4.69}$$
$$x_1^1 + x_2^1 + x_3^1 + x_4^1 = 8 + 3 + 6 + 12, \tag{4.70}$$
$$x_t^0 \leq M y_t \quad \forall\, t, \tag{4.71}$$
$$x_t^1 \leq M Y_t \quad \forall\, t. \tag{4.72}$$

We now give the dimension of $conv(STD - ex)$. Observe that we have two equality constraints and $d_1 > 0$ implies $y_1 = Y_1 = 1$, therefore we have $dim_{conv(STD-ex)} \leq 16 - 4 = 12$. In the matrix that follows, we give 13 affinely independent points what implies that $dim_{conv(STD-ex)} = 12$. Each line represents one point and the columns are organized in the following order $(x_1^0, x_2^0, x_3^0, x_4^0, x_1^1, x_2^1, x_3^1, x_4^1, y_1, y_2, y_3, y_4, Y_1, Y_2, Y_3, Y_4)$.

$$Points^{STD-ex} = \begin{pmatrix} 29 & 0 & 0 & 0 & 29 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 29 & 0 & 0 & 0 & 8 & 21 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 29 & 0 & 0 & 0 & 8 & 3 & 18 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 29 & 0 & 0 & 0 & 8 & 3 & 6 & 12 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 8 & 21 & 0 & 0 & 8 & 21 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 8 & 3 & 18 & 0 & 8 & 3 & 18 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 8 & 3 & 6 & 12 & 8 & 3 & 6 & 12 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 29 & 0 & 0 & 0 & 29 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 29 & 0 & 0 & 0 & 29 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 29 & 0 & 0 & 0 & 29 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 29 & 0 & 0 & 0 & 29 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 29 & 0 & 0 & 0 & 29 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 29 & 0 & 0 & 0 & 29 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Consider now the inequality

$$x_1^0 + x_2^0 + 6y_3 + x_1^1 + 9Y_2 + x_3^1 + x_4^1 \geq 46, \tag{4.73}$$

which can be obtained from the following $t-$dicut collections

- $\Gamma^1 = \{\{x_1^0\}, \{x_1^1\}\}$,
- $\Gamma^2 = \{\{x_1^0, x_2^0\}, \{x_1^1, Y_2\}\}$,
- $\Gamma^3 = \{\{x_1^0, x_2^0, y_3\}, \{x_1^1, Y_2, x_3^1\}\}$,
- $\Gamma^4 = \{\{x_1^0, x_2^0, x_3^1, x_4^1\}\}$.

Observe that (4.73) is a dicut inequality and therefore it is valid. Note that it is not a simple dicut inequality though. If we take the simple dicut inequalitiy $x_1^0 + x_2^0 + 6y_3 \geq 17$ which constitutes part of (4.73), we would need $x_1^1 + 9Y_2 + x_3^1 + x_4^1 \geq 29$ to complete (4.73) what cannot be obtained as a combination of simple dicut inequalities (we would need at least $x_1^1 + 21Y_2 + x_3^1 + x_4^1 \geq 29$ which would give the inequality $x_1^0 + x_2^0 + 6y_3 + x_1^1 + 21Y_2 + x_3^1 + x_4^1 \geq 46$).

We now show that (4.73) can be obtained from the projection of the multicommodity formulation. We give a feasible solution to D2L-S/LS-U-MCp with nonzero variables $\delta_1 = 1$, $\delta_3 = 1$, $\delta_4 = 1$, $\phi_1 = 1$, $\phi_2 = 1$, $\theta_{33} = 1$, $\gamma_{22} = 1$, $\gamma_{23} = 1$, $\epsilon_1 = -2$, $\epsilon_2 = -2$, $\epsilon_3 = -2$, $\epsilon_4 = -1$, $\alpha_{11} = 1$, $\alpha_{22} = 1$, $\alpha_{24} = 1$, $\alpha_{33} = 1$. All the other variables are zero valued. This solution gives inequality (4.73).

It can be checked by simple substitution that the solution proposed is feasible in D2L-S/LS-U-MCp.

We show that the dimension of the inequality is $dim_{IN} = dim_{conv(STD-ex)} - 1 = 11$ by giving 12 linear independent points satisfying (4.73) at equality. This implies that the inequality is a facet.

$$A^I = \begin{pmatrix}
29 & 0 & 0 & 0 & 8 & 21 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
8 & 21 & 0 & 0 & 8 & 21 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
8 & 9 & 0 & 12 & 8 & 9 & 0 & 12 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
11 & 0 & 18 & 0 & 11 & 0 & 18 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
11 & 0 & 18 & 0 & 11 & 0 & 6 & 12 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
11 & 0 & 6 & 12 & 11 & 0 & 6 & 12 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
17 & 0 & 0 & 12 & 17 & 0 & 0 & 12 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
17 & 0 & 0 & 12 & 8 & 9 & 0 & 12 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
29 & 0 & 0 & 0 & 8 & 21 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
29 & 0 & 0 & 0 & 8 & 21 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
29 & 0 & 0 & 0 & 8 & 21 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
29 & 0 & 0 & 0 & 8 & 21 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0
\end{pmatrix}.$$

Therefore we have a facet-defining dicut inequality that is not simple.

□

4.5 Valid Inequalities in the Space of the Original Variables for the $2L - S/LS - U$

In this section we consider the uncapacitated two-level production-in-series lot-sizing problem and try to analyze valid inequalities in the space of the original variables. Observe that valid inequalities could be derived directly for the OWMR, but analyzing the inequalities for the $2L - S/LS - U$ appears to be more manageable. Thus we now consider a formulation that is intermediate between the original formulation and the MC formulation described in Section 4.3.1.

Note that the one-warehouse multi-retailer problem is equivalent to a multi-item uncapacitated two-level production-in-series lot-sizing problem with joint setup costs on the upper level. Consider the variables

- x_t^{0c}: the amount produced for client c at the production site in period t, and
- s_t^{0c}: the stock for client c at the production site at the end of period t,

this equivalent problem can be formulated as

$$(OWMR')$$

$$\min \sum_{c,t}(h_t^0 s_t^{0c} + p_t^0 x_t^{0c}) + \sum_t f_t^0 y_t + \sum_{c,t}(h_t^{1c} s_t^{1c} + p_t^{1c} x_t^{1c} + f_t^{1c} Y_t^c) \tag{4.74}$$

$$s_{t-1}^{0c} + x_t^{0c} = x_t^{1c} + s_t^{0c}, \quad \forall c, t, \tag{4.75}$$

$$s_{t-1}^{1c} + x_t^{1c} = d_t^c + s_t^{1c}, \quad \forall c, t, \tag{4.76}$$

$$x_t^{0c} \leq M y_t, \quad \forall c, t, \tag{4.77}$$

$$x_t^{1c} \leq M Y_t^c, \quad \forall c, t, \tag{4.78}$$

$$s^0, s^1, x^0, x^1 \in \mathbb{R}_+^{NC \times NT}, \tag{4.79}$$

$$y \in \{0,1\}^{NT}, \ Y \in \{0,1\}^{NC \times NT}. \tag{4.80}$$

We can write the feasible region

$$X^{OWMR'} = \bigcap_{c=1}^{NC} X^{2L-S/LS-U_c},$$

where $X^{2L-S/LS-U_c}$ is the set of feasible solutions of the uncapacitated two-level production-in-series lot-sizing problem $(2L-S/LS-U_c)$ studied in Chapter 3 and the sets are only linked by variables y.

$$(2L-S/LS-U_c) \quad s^{0c}_{t-1} + x^{0c}_t = x^{1c}_t + s^{0c}_t, \quad \forall t, \tag{4.81}$$

$$s^{1c}_{t-1} + x^{1c}_t = d^c_t + s^{1c}_t, \quad \forall t, \tag{4.82}$$

$$x^{0c}_t \leq M y_t, \quad \forall t, \tag{4.83}$$

$$x^{1c}_t \leq M Y^c_t \quad \forall t, \tag{4.84}$$

$$s^0, s^1, x^0, x^1 \in \mathbb{R}^{NT}_+, \tag{4.85}$$

$$y \in \{0,1\}^{NT}, \; Y^c \in \{0,1\}^{NT}. \tag{4.86}$$

Therefore, one approach to try to improve results for the OWMR would be to work with the formulation OWMR' and then add valid inequalities for $(2L-S/LS-U)$ for each client.

4.5.1 The $G-(l,S)$ inequalities

We introduce the $G-(l,S)$ inequalities, that are generalizations of the (l,S) inequalities for the uncapacitated lot-sizing problem. Consider the indices $1 \leq j \leq k \leq l \leq NT$ and the following sets of variables

- $\overline{\Delta}$ and $\overline{\Pi}$ associated to variables at level zero, where $\overline{\Delta} \cup \overline{\Pi} = \{1, \ldots, k\}$ with $\overline{\Delta} \cap \overline{\Pi} = \emptyset$,
- $\underline{\Delta}$ and $\underline{\Pi}$ associated to variables at level one where $\underline{\Delta} \cup \underline{\Pi} = \{k+1, \ldots, l\}$ with $\underline{\Delta} \cap \underline{\Pi} = \emptyset$,
- $\underline{\Lambda}$ and $\underline{\Theta}$ associated to variables at level one, where $\underline{\Lambda} \cup \underline{\Theta} = \{1, \ldots, j\}$ with $\underline{\Lambda} \cap \underline{\Theta} = \emptyset$.

Proposition 4.7. *The $G-(l,S)$ inequalities*

$$\sum_{\substack{t=1 \\ k \in \underline{\Delta}}}^{j} Y_t d_{tj} + \sum_{\substack{t=1 \\ t \in \underline{\Theta}}}^{j} x^1_t + \sum_{\substack{t=1 \\ t \in \overline{\Delta}}}^{k} y_t d_{\max(t,j+1),l} + \sum_{\substack{t=1 \\ t \in \overline{\Pi}}}^{k} x^0_t + \sum_{\substack{t=k+1 \\ t \in \underline{\Delta}}}^{l} Y_t d_{tl} + \sum_{\substack{t=k+1 \\ t \in \underline{\Pi}}}^{l} x^1_t \geq d_{1l} \tag{4.87}$$

are valid for $(2L-S/LS-U)$.

Proof. The proof consists in showing that the inequalities are simple dicut inequalities and are therefore valid. For $t \leq j$, get as $t-$dicuts $\Gamma^t = \{Y_b : 1 \leq b \leq t, b \in \underline{\Lambda}\} \cup \{x^1_b : 1 \leq b \leq t, b \in \underline{\Theta}\}$.
For $t \geq j+1$, get $\Gamma^t = \{y_b : 1 \leq b \leq k, b \in \overline{\Delta}\} \cup \{x^0_b : 1 \leq b \leq k, b \in \overline{\Pi}\} \cup \{Y_b : k+1 \leq b \leq t, b \in \underline{\Delta}\} \cup \{x^1_b : k+1 \leq b \leq t, b \in \underline{\Pi}\}$. These $t-$dicuts give the

simple dicut inequality

$$\sum_{\substack{b=1\\b\in\underline{\Theta}}}^{j} x_b^1+\sum_{\substack{b=1\\b\in\overline{\Pi}}}^{k} x_b^0+\sum_{\substack{b=k+1\\b\in\underline{\Pi}}}^{l} x_b^1+\sum_{\substack{b=1\\b\in\underline{\Lambda}}}^{j}\sum_{t=b}^{j} Y_b d_t+\sum_{\substack{b=1\\b\in\overline{\Delta}}}^{k}\sum_{t=\max(b,j+1)}^{l} y_b d_t+\sum_{\substack{b=k+1\\b\in\underline{\Delta}}}^{l}\sum_{t=b}^{l} d_t Y_b \geq \sum_{t=1}^{l} d_t,$$

that is what (4.87) implies. Therefore the inequalities can be obtained as simple dicut inequalities.

□

Example: The inequality represented in Figure 4.4 can be obtained as follows. Take $\underline{\Lambda}=\emptyset$, $\underline{\Theta}=\{1\}$, $\overline{\Delta}=\{1,2\}$, $\overline{\Pi}=\{3\}$, $\underline{\Delta}=\{4\}$ and $\underline{\Pi}=\{5\}$, and we have $j=1$, $k=3$ and $l=5$. This gives the inequality

$$(x_1^1)+(d_{25}y_1+d_{25}y_2+x_3^0)+(d_{45}Y_4+x_5^1)\geq d_{15}.$$

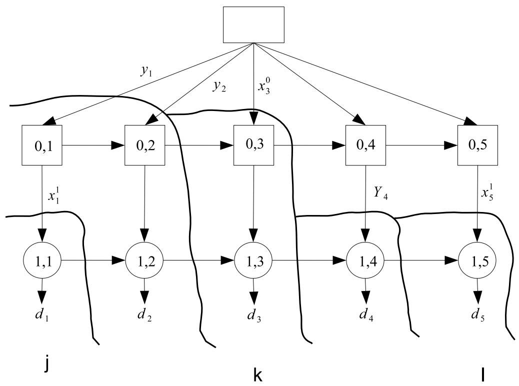


Figure 4.4: An example of $G-(l,S)$ inequality with $j=1$, $k=3$ and $l=5$

Proposition 4.8. *There exists an $O(NT^3)$ algorithm to separate the $G-(l,S)$ inequalities.*

Proof. Let $(\bar{x}^0,\bar{x}^1,\bar{y},\bar{Y})$ be a solution for which we want to test whether it violates any $G-(l,S)$ inequality.

The separation problem can be posed as minimizing

$$\min_{\substack{l\in\{1,\ldots,NT\},j\in\{0,\ldots,l\},\\k\in\{j+1,\ldots,l+1\}}}\left[\sum_{t=1}^{j}\min(\bar{Y}_t d_{tj},\bar{x}_t^1)+\sum_{t=1}^{k}\min(\bar{y}_t d_{\max(t,j+1),l},\bar{x}_t^0)+\sum_{t=k+1}^{l}\min(\bar{Y}_t d_{tl},\bar{x}_t^1)-d_{1l}\right], \quad (4.88)$$

where the first sum is related to the sets $\underline{\Lambda}$ and $\underline{\Theta}$, the second sum to $\overline{\Delta}$ and $\overline{\Pi}$ and the third sum to $\underline{\Delta}$ and $\underline{\Pi}$. Observe that a violated inequality is found in case the above expression is strictly negative. Let

$$\Sigma_j^1 = \sum_{t=1}^{j} \min(\bar{Y}_t d_{tj}, \bar{x}_t^1)$$

and

$$\Sigma_{jl}^{2,3} = \min_{k=j,\ldots,l} \left[\sum_{t=1}^{k} \min(\bar{y}_t d_{\max(t,j),l}, \bar{x}_t^0) + \sum_{t=k+1}^{l} \min(\bar{Y}_t d_{tl}, \bar{x}_t^1) \right].$$

The values Σ_j^1 are related to the determination of $\underline{\Lambda}$ and $\underline{\Theta}$ while the values $\Sigma_{jl}^{2,3}$ to the determination of $\overline{\Delta}$, $\overline{\Pi}$, $\underline{\Delta}$ and $\underline{\Pi}$. The parts of the calculations are illustrated in Figure 4.5.

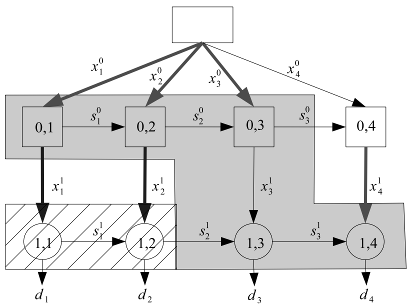


Figure 4.5: The lined region represents the Σ_j^1 values and the shadowed region the Σ_{jl}^{23}

Using the values just described, we can determine (4.88) as

$$\min_{j=0,\ldots,l} \left(\Sigma_j^1 + \Sigma_{j+1,l}^{2,3} \right). \tag{4.89}$$

The calculation of (4.89) can be done by simply enumerating the values involved in its calculation.

Algorithm for $G-(l,S)$ Separation
Step 1: Calculate all Σ^1 values.
The next steps are repeated for each period $l \in \{1,\ldots,NT\}$.
Step 2: Calculate the $\Sigma^{2,3}$ values.
Step 3: Determine (4.89).
Step 4: Output a most violated inequality determined by Step 3 in case there

is a violation.

We now analyze the run-time complexity. Step 1 is performed in $O(NT^2)$. Steps 2-4 are performed in $O(NT^2)$. Therefore the inequalities can be separated in $O(NT^3)$ for every $l \in \{1, \ldots, NT\}$.

□

4.6 Some Limited Computational Results

4.6.1 One-warehouse Multi-retailer Problem (OWMR)

In this section we report on some limited computational experiments to analyze the behavior of the different reformulations presented in Section 4.3. All experiments were performed on a machine running under Ubuntu 9.10 with a Intel Xeon 2.93GHz processor, 8 Gb of RAM memory using Xpress-Optimizer version 21.01.00.

Costs are generated as in Solyali and Sural [46]. Fixed production cost is generated in the interval $q^0 \in [1500, 4500]$, fixed transportation cost is generated in the interval $q^{1c} \in [5, 100]$, storage cost at the production site is $h^0 = 0.5$, storage cost at client c is generated in the interval $h^{1c} \in [0.5, 1.0]$. Production cost is $p^0 = 0$ and transportation costs are $p^{1c} = 0$ for every c. Demands are generated in the interval $d_t^c \in [5, 100]$. With exception of the storage costs all the parameters are integer valued. The number of clients is generated as $NC \in \{50, 100, 150, 200\}$, while the number of periods as $NT \in \{30, 45, 60\}$.

Some limited results are presented in Table 4.1. The first two columns give the dimensions of the instances, where the first column gives the number of clients and the second column the number of time periods. The third column gives the value of the optimal integer solution for the instance. The other columns present the results for the different reformulations. For each reformulation we present the linear relaxation, the time and the number of nodes needed to solve the instances to optimality.

We can see that the approaches provided lower bounds that are close to the optimal integer value. In general, the multi-commodity formulation performs better with the exception of some of the smaller instances. All instances could be solved to optimality when using the multi-commodity reformulation while none of the bigger instances could be solved to optimality while using the O($NC \times NT^3$) reformulations.

We also generate a second set of instances varying the transportation costs. The transportation costs can assume values $p^{1c} \in [1.5, 2.5]$. The results are presented in Table 4.2 whose columns are the same as in the previous table.

We can see that the results are not much different from the results without transportation costs in Table 4.1. Again the multi-commodity formulation

Table 4.1: Results using the multi-commodity, transportation and shortest path reformulations

			MC			TR			SP		
NC	NT	IP	lp	time	nodes	lp	time	nodes	lp	time	nodes
50	30	107835.3	107756.3	10	3	107756.3	17	1	107756.3	8	1
100	30	198656.7	198476.5	44	3	198476.5	38	3	198476.5	50	3
150	30	228847.4	228485.5	67	15	228485.5	71	5	228485.5	108	7
200	30	304979.7	303807.4	85	19	303807.4	97	11	303807.4	108	11
50	45	130767.9	130153.4	25	7	130153.4	52	7	130155.4	106	11
100	45	243744.5	242845.1	86	25	242845.1	105	7	242845.1	–	–
150	45	332919.7	332919.7	16	1	332919.7	24	1	332919.7	36	1
200	45	478165.8	476640.8	285	27	476640.8	463	17	–	–	–
50	60	190846.21	190570.2	63	21	190570.2	–	–	190570.6	–	–
100	60	388174.9	387797.5	187	5	–	–	–	–	–	–
150	60	505739.8	504863.7	617	7	–	–	–	–	–	–
200	60	615703.2	612315.0	1665	313	–	–	–	–	–	–

Table 4.2: Results using the multi-commodity, transportation and shortest path reformulations varying transportation costs

			MC			TR			SP		
NC	NT	IP	lp	time	nodes	lp	time	nodes	lp	time	nodes
50	30	251624.1	251561.5	11	3	251561.5	14	1	251561.5	9	1
100	30	473081.8	472407.3	33	11	472407.3	39	11	472407.3	56	9
150	30	691925.2	691925.2	5	1	691925.2	6	1	691925.2	10	1
200	30	895258.7	895258.7	7	1	895258.7	10	1	895258.7	13	1
50	45	420176.0	420061.3	43	9	420061.3	83	5	420176.0	18	1
100	45	696313.9	695530.4	77	23	695530.4	124	11	695530.4	–	–
150	45	1106748.6	1105093.2	245	7	1105093.2	–	–	1105093.2	–	–
200	45	1366608.3	1366608.3	21	1	1366608.3	25	1	–	–	–
50	60	509558.3	509153.8	58	15	509153.8	158	13	509155.7	443	11
100	60	999048.3	997068.9	152	25	–	–	–	–	–	–
150	60	1428489.0	1425981.8	480	55	–	–	–	–	–	–
200	60	1917273.5	1912982.7	611	49	–	–	–	–	–	–

performs better in most of the instances and all instances could be solved to optimality when using it, but the bigger instances could not be solved to optimality while using the reformulations with $O(NC \times NT^3)$ reformulations.

4.6.2 The Multi-item One-warehouse Multi-retailer Problem (OWMR-MI)

One natural extension of the one-warehouse multi-retailer problem is the problem in which multiple (NI) items are considered. The goal of this section is not to provide extensive computations but mainly to make some limited computational experiments to see how the use of an implementation of the cutting plane algorithm described in the proof of Proposition 4.8 performs compared to the multi-commodity formulation and to the results available for an echelon stock reformulation which will be defined later in this section.

The multi-item model considered here is an extension of OWMR' described previous in this section. It considers set-ups that are dependent on the items produced at the production site.

$$
\begin{gathered}
(OWMR' - MI) \\
\min \textstyle\sum_{c,t}(h_t^0 s_t^{0ic} + p_t^{0ic} x_t^{0ic}) + \sum_t f_t^0 y_t^i + \sum_{c,t}(h_t^{1c} s_t^{1ic} + p_t^{1c} x_t^{1ic} + f_t^{1c} Y_t^c) \\
s_{t-1}^{0ic} + x_t^{0ic} = x_t^{1ic} + s_t^{0ic} \quad \forall\ i,c,t, \\
s_{t-1}^{1ic} + x_t^{1ic} = d_t^{ic} + s_t^{1ic} \quad \forall\ i,c,t, \\
x_t^{0ic} \le M y_t^i \quad \forall\ i,c,t, \\
x_t^{1ic} \le M Y_t^c \quad \forall\ i,c,t, \\
s^0, s^1, x^0, x^1 \in \mathbb{R}_+^{NC \times NT}, \\
y \in \{0,1\}^{NI \times NT},\ Y \in \{0,1\}^{NC \times NT}.
\end{gathered}
$$

The valid inequalities are separated for each item i and client c.

The echelon stock reformulation is obtained by adding the facility location reformulation for each of the three uncapacitated lot-sizing sets (that are obtained as relaxations of $OWMR' - MI$) that follow.

For every item i and client c we consider the two lot-sizing sets

$$
\begin{gathered}
(s_{t-1}^{0ic} + s_{t-1}^{1ic}) + x_t^{0ic} = d_t^{ic} + (s_t^{0ic} + s_t^{1ic}) \quad \forall\ t, \\
x_t^{0ic} \le M y_t^i \quad \forall\ t, \\
s^{0ic}, s^{1ic}, x^{0ic} \in \mathbb{R}_+^{NT}, \\
y^i \in \{0,1\}^{NT},
\end{gathered}
$$

and

$$\begin{aligned} s_{t-1}^{1ic} + x_t^{1ic} = d_t^{ic} + s_t^{1ic} \quad \forall\, t, \\ x_t^{1ic} \leq MY_t^c \quad \forall\, t, \\ s^{1ic}, x^{1ic} \in \mathbb{R}_+^{NT}, \\ Y^c \in \{0,1\}^{NT}. \end{aligned}$$

For every client c we consider the set

$$\begin{aligned} (\textstyle\sum_{i=1}^{NI} s_{t-1}^{1ic}) + (\sum_{i=1}^{NI} x_t^{1ic}) = (\sum_{i=1}^{NI} d_t^{ic}) + (\sum_{i=1}^{NI} s_t^{1ic}) \quad \forall\, t, \\ (\textstyle\sum_{i=1}^{NI} x_t^{1ic}) \leq MY_t^c \quad \forall t, \\ s^{1ic}, x^{1ic} \in \mathbb{R}_+^{NI \times NT}, \\ Y^c \in \{0,1\}^{NT}. \end{aligned}$$

We created a set of five instances with $NI = 30$ items, $NC = 40$ clients and $NT = 14$ periods based on the recent instances for the one-warehouse multi-retailer problem with a single item as created in Solyali and Sural [46]. The random data generation was described in Section 4.6.1.

The results are summarized in Table 4.3. Columns 1 and 2 show the instance identification followed by its optimal integer solution. Columns 3 and 4 show, for the multi-commodity formulation, the linear relaxation and the time to solve the instance to optimality. Columns 5 to 8 show respectively the bound obtained by the solver after adding its automatic cuts to the standard formulation, the bound obtained after adding our cuts, the number of cuts added and the time to solve the instance to optimality. In Columns 9 and 10 we show, for the echelon stock reformulation, the linear realaxation bound and the time to solve the instance to optimality.

Table 4.3: Results for the instances with $NI = 30$, $NC = 40$ and $NT = 14$

		MC		CP				ES	
Instance	IP	LP	time	XLP	XLP+CP	#cuts	time	LP	time
a1	866337.5	866337.5	305	252102.8	866337.5	103873	112	866337.5	219
a2	862104.0	862104.0	307	248751.9	862104.0	105289	103	862104.0	224
a3	856562.5	856562.5	312	247091.8	856562.5	104955	106	856562.5	237
a4	806697.0	806697.0	208	224812.4	806697.0	101080	94	806697.0	221
a5	845235.0	845235.0	350	240937.0	845235.0	102737	100	845235.0	213

We can see from the table that differently from what happened with the single item instances from the previous section, the linear relaxation bound obtained using the three approaches is surprisingly equal to the optimal integer value for all instances. This is probably due to the fact that the instance generator was set-up to provide solutions that are not too easy for the single item problem.

The cutting plane approach performed best, followed by the echelon stock and then the multi-commodity reformulations. We can see that the cutting plane gives a great contribution to the bound obtained by the solver when using the standard formulation. Also, although a big number of cuts was added during the procedure the cutting plane approach finished at least two times faster than the other approaches for almost all instances, and the number of cuts added did not change considerably among the different instances.

Similar to Section 4.6.1, we generate an additional set of instances with the same characteristics of the ones just analyzed with the difference that we vary the transportation costs in the interval $p^{1c} \in [1.5, 2.5]$. The results are summarized in Table 4.4 in which the columns are the same as in Table 4.3.

Table 4.4: Results for the instances with $NI = 30$, $NC = 40$ and $NT = 14$ and varying transportation costs

		MC		CP				ES	
Instance	IP	LP	time	XLP	XLP+CP	#cuts	time	LP	time
b1	2629069.4	2629069.4	144	2014786.9	2629069.4	111549	109	2629069.4	3224
b2	2564147.5	2564147.5	196	1996884.1	2564147.5	107823	94	2564147.5	1716
b3	2695326.1	2695326.1	319	2062644.3	2695326.1	112106	117	2695326.1	2010
b4	2636644.8	2636644.8	279	2023596.4	2636644.8	112090	111	2636644.8	2983
b5	2587004.6	2587004.6	254	2036343.1	2587004.6	104191	93	2587004.6	1807

As observed for the previous instances, in this set of instances the linear relaxation bound obtained using the three approaches is also equal to the optimal integer value for all instances. Again, the cutting plane approach could solve considerably faster all the instances but the number of cuts added is still quite large. In contrast to what happened to the previous instances, we see that the echelon stock reformulation faced problems to solve these instances with varying transportation costs where for almost all the instances the time spent was more than half an hour.

These very limited computational results suggest that the use of cutting planes in formulations with a reduced number of variables may in some cases be a viable alternative to the use of larger reformulations.

4.7 Concluding Remarks

In this chapter we studied the one-warehouse multi-retailer problem and also considered briefly the particular case known as the joint-replenishment problem.

We compared the multi-commodity formulation with the transportation formulation showing that the multi-commodity and the transportation formulations have the same linear relaxation bound.

For the joint-replenishment problem we showed that the projection of the multi-commodity formulation into the original space of variables is formed by just simple dicut inequalities. We showed that the same does not follow for the OWMR, since the projection of the multi-commodity formulation into the original space of variables has facet-defining inequalities that are not simple dicut inequalities.

We also analyzed valid inequalities for the two-level lot-sizing problem that are not simple (l, S) inequalities and that can be used for the one-warehouse multi-retailer problem. Based on the results we can suggest that in some situations the use of a cutting plane using valid inequalities for the uncapacitated two-level production-in-series lot-sizing problem may be used effectively but it requires further study.

In the next chapter we will consider more general production and transportation problems with more than one production site and other additional characteristics.

Chapter 5

Two-level Production-Transportation Problems

In this chapter we study a two-level production-transportation problem that is sufficiently general to cover a variety of deterministic demand problems arising in the literature and in practice.

As example, it covers one-warehouse multi-retailer problems and their generalizations to treat multiple products and production sites. It also covers two-level production/transportation supply chain models as the ones discussed in Chapter 2. This sort of model was also a major topic of the EU financed LISCOS project [28], but at that time little progress was made in effectively tackling any but small-sized instances. We also consider a profit-maximizing model in which sales can lie between predetermined bounds, a type of variant that has been developed by BASF among others.

The problem treated involves multiple items, multiple production sites and multiple client areas over a discrete time horizon of NT periods. Some considered costs are fixed set-up and variable costs per item of production, joint transportation fixed costs as well as variable per item transportation costs, and variable storage costs at both the production sites and the client areas. The objective of the problem is to satisfy the demands of each client for each item in each period at minimum total cost. Typical additional aspects such as limited transportation capacities, (big bucket) limits on the total production at each site, limits on the total storage capacity at each client and potential additional sales are also considered.

Our goal is to develop formulations and heuristics that allow one to tackle medium-sized instances using a MIP solver. In particular the heuristics have the property of providing "realistic" a posteriori performance guarantees within a reasonable time in contrast to simply primal heuristics that do not give any information on how good the solutions obtained are.

Specifically we show that uncapacitated instances of the basic problem with on the order of 5-10 items, 2-5 production sites, 10-40 clients and 12-24 periods can often be solved to optimality. On the other hand we show that a hybrid MIP heuristic based on two different MIP formulations permits us to find solutions guaranteed to be within 10% of optimality for instances with limited transportation capacity and/or with additional sales. For instances with big bucket production or aggregate storage capacity constraints the gaps can be larger (up to 40%), but in all cases the heuristic significantly improves on the direct use of an MIP solver.

The remaining of this chapter is organized as follows. In Section 5.1 we give a formal description of the problem and formulate it as a mixed integer program. We also make precise the special cases that will be treated separately, the *basic uncapacitated version* in which both production quantities and transportation are uncapacitated, the *basic capacitated version* in which the transportation is capacitated (the vehicles used for transportation between producers and clients have a fixed size), and the *general version* in which we allow joint production and storage capacities as well as additional sales.

In Section 5.2 we present the reformulations that we use in our computational tests. First we show that two natural relaxations lead to single item lot-sizing sets for which tight formulations are known. We then present the multi-commodity reformulation of the uncapacitated two-level single-item problem with arbitrary values of NP, NC and NT, as well as a facility location reformulation for the variant with sales.

Given that the multi-commodity/facility location formulations for each item provide good lower bounds, but are too large to be used in branch-and-cut, the challenge is then how to find good feasible solutions. One option would be to use the heuristic approach of Eppen and Martin [14]. In this approach they solve the linear relaxation of the extended formulation, fix the integer variables that take integer solutions in the linear relaxation and then solve the mixed integer programming on the remaining variables. Unfortunatelly this approach can still lead to large constrained problems. Assuming that the heuristic approach of Eppen and Martin does not work because the restricted extended formulations are still too large for branch and bound, we modify their approach by fixing the values of certain variables of a second weaker MIP formulation and thereby we obtain an MIP of manageable size which then provides the feasible solutions and upper bounds. The resulting hybrid MIP heuristic combining formulations is presented in Section 5.3. Computational results for the problem without sales are presented in Section 5.4. First we show that the uncapacitated problem can be solved to optimality using the multi-commodity formulation. For the more difficult capacitated problem we compare direct use of the MIP solver with hybrid heuristics that make use of both the multi-commodity and the original weaker but smaller formulations. Results for the general model with sales are presented in Section 5.4.5. We end with some conclusions and a discussion of potential directions for research.

5.1 Two Level Supply Chain: Description and Formulation

The two-level supply chain problem is the problem of determining the manufacturing, transportation and storage schedule for the products so as to minimize the total cost. NP production sites produce NI different items in order to satisfy the demands of NC clients in a time horizon of NT periods. Each client has his/her time varying demands for each item to be satisfied. Vehicles of a given capacity transport items between the production sites and the clients such that different types of items can be transported in the same cargo. Production costs are composed by a fixed cost whenever production occurs plus variable costs depending on the amount produced. As transportation costs, each vehicle has a fixed transportation cost whenever it transports a positive amount of items plus a variable cost that depends on the amount of items carried in the transportation. There are also variable storage costs at the production sites and at the client areas.

We will define three different variants of the problem which will be considered in this chapter, namely the *basic uncapacitated version*, the *basic capacitated version* and the *general version*.

5.1.1 The Basic Version

In the basic version, besides the characteristics presented in the beginning of this section we have:

- the amount produced per item in each period is unrestricted (uncapacitated production),

- the capacities of the vehicles can be either unrestricted (uncapacitated transportation) in the basic uncapacitated version or limited in the basic capacitated version.

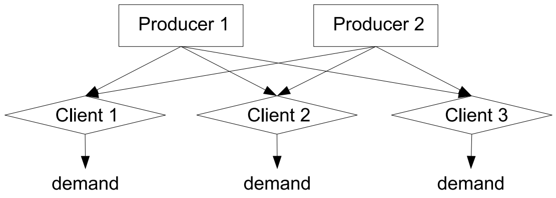
Figure 5.1(a) depicts an example with two production sites and three client areas.

5.1.2 The General Version

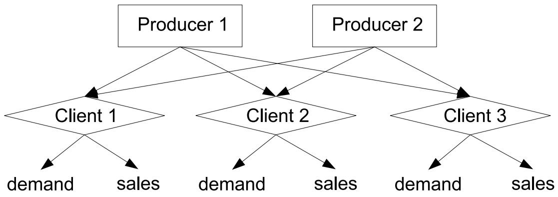
The general version possesses more characteristics of practical problems. In addition to the characteristics presented in the beginning of this section, in the general version there are:

- machine capacity (big bucket) constraints: the total production at each production site in each period is limited,
- stock upper bounds constraints: the total amount of stock held by a client in each period is limited,
- sales: production can exceed the demand up to a certain limit and the excess of production can be sold in order to get some extra profit.

The general version of the problem is illustrated in Figure 5.1(b).



(a) Example of the basic problem



(b) Example of the general problem

5.1.3 Formulation

First we introduce the input data for the problem, and then we give a standard formulation for the general version as a mixed integer program.

The constraint data is as follows:

- d_t^{ic}: demand of item i at client c in period t,
- K_1: capacity of each vehicle (unlimited or constant),
- a^{ip}: time required to produce an item i at site p,
- b^{ip}: setup time required to produce item i at site p,

- LK^p: total time available for production at site p,
- SK^c: total amount of storage available at client c,
- U_t^{ic}: maximum additional amount of item i that can be sold to client c in period t,
- M: very large number.

The cost data is:

- p^{0ip}: per unit cost to produce item i in production site p,
- p^{1ipc}: per unit transportation cost for item i from production site p to client c,
- q^{0ip}: fixed setup cost of item i in production site p,
- q^{1pc}: fixed cost per vehicle traveling from production site p to client c,
- h^{0ip}: unit storage cost of item i in production site p,
- h^{1ic}: unit storage cost for item i at client c,
- l^{ic}: unit sales price for additional units of item i sold to client c.

We define the variables:

- x_t^{0ip}: amount of item i produced at production site p in period t,
- x_t^{1ipc}: amount of item i transported from production site p to client c in period t,
- s_t^{0ip}: amount of item i in stock at production site p at the end of period t,
- s_t^{1ic}: amount of item i in stock at client c at the end of period t,
- v_t^{ic}: additional amount of item i sold to client c in period t,
- y_t^{ip}: production setup variable equal to 1 if production of item i occurs at production site p in period t and 0 otherwise,
- Y_t^{pc}: number of vehicles used for transportation between production site p and client c in period t.

We now give a standard formulation for the general problem $2L - PT$:

$$(STD)$$

$$\min \sum_{i,p,t} (p^{0ip} x_t^{0ip} + h^{0ip} s_t^{0ip} + q^{0ip} y_t^{ip}) + \sum_{i,c,t} h^{1ic} s_t^{1ic} +$$

$$\textstyle\sum_{i,p,c,t} p^{1ipc} x_t^{1ipc} + \sum_{p,c,t} q^{1pc} Y_t^{pc} - \sum_{i,c,t} l^{ic} v_t^{ic}$$

$$s_{t-1}^{0ip} + x_t^{0ip} = \sum_{c=1}^{NC} x_t^{1ipc} + s_t^{0ip} \quad \forall\, i, p, t, \tag{5.1}$$

$$s_{t-1}^{1ic} + \sum_{p=1}^{NP} x_t^{1ipc} = d_t^{ic} + v_t^{ic} + s_t^{1ic} \quad \forall\, i, c, t, \tag{5.2}$$

$$x_t^{0ip} \le M y_t^{ip} \quad \forall\, i, p, t, \tag{5.3}$$

$$\sum_{i=1}^{NI} x_t^{1ipc} \le K_1 Y_t^{pc} \quad \forall\, p, c, t, \tag{5.4}$$

$$\sum_{i=1}^{NI} (a^{ip} x_t^{0ip} + b^{ip} y_t^{ip}) \le LK^p \quad \forall\, p, t, \tag{5.5}$$

$$\sum_{i=1}^{NI} s_t^{1ic} \le SK^c \quad \forall\, c, t, \tag{5.6}$$

$$0 \le v_t^{ic} \le U_t^{ic} \quad \forall\, i, c, t, \tag{5.7}$$

$$s^0, x^0 \in \mathbb{R}_+^{NI\times NP\times NT},\ s^1 \in \mathbb{R}^{NI\times NC\times NT},\ x^1 \in \mathbb{R}^{NI\times NP\times NC\times NT}, \tag{5.8}$$

$$y \in \{0,1\}^{NI\times NP\times NT},\ Y \in \mathbb{Z}^{NP\times NC\times NT}. \tag{5.9}$$

In this formulation, constraints (5.1) are balance constraints at the production sites in which the demand is the amount transported to the clients. They imply for each item i and production site p that the amount arriving as stock from period $t-1$ plus the amount produced in period t is equal to the amount transported to the clients in period t plus the stock that will be available at the end of period t. Constraints (5.2) are the balance constraints at each client. They imply for each item i and client c that the amount arriving as stock from period $t-1$ plus the amount arriving from the production sites is equal to the demand plus the amount to be sold plus the stock that will be available at the end of period t. Constraints (5.3) force the production set-up variables to take value one when there is positive production while (5.4) determine the number of trucks to be used for transportation. Constraints (5.5) limit the production time available at a production site during period t to LK^p. Constraints (5.6) give an upper limit of SK^c to the total stock in client c at period t. Constraints (5.7) limit the amount of additional sales to U_t^{ic}.

Observation 5.1. *The basic problem is the case in which $LK^p = M$ for all p*

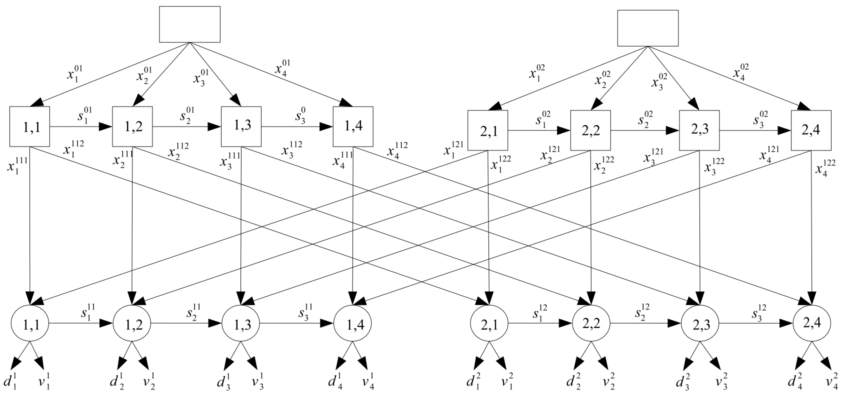


Figure 5.1: Example of the general problem with two production sites and two clients

(unrestricted production), $SK^c = M$ for all c (unrestricted stocks at the clients) and $U_t^{ic} = 0$ for all i, c, t (no sales).

5.2 Relaxations and Reformulations

Deriving reformulations and valid inequalities for problem relaxations has been shown to be effective for numerous production planning problems. Examples can be seen in Pochet and Wolsey [40]. We will first review some reformulation results available for single item lot-sizing problems and then we will show how these single item problems can be obtained as relaxations of $2L - PT$.

The single item lot-sizing set is given by

$$(LS-C)\quad S_{t-1} + X_t = D_t + S_t \quad \forall\, t, \tag{5.10}$$

$$X_t \leq K_t Y_t \quad \forall\, t, \tag{5.11}$$

$$X, S \in \mathbb{R}^{NT},\ Y \in \{0,1\}^{NT}. \tag{5.12}$$

Constraints (5.10) are balance constraints. Constraints (5.11) determine the values of the Y variables. Constraints (5.12) are constraints on the variables.

The uncapacitated lot-sizing $(LS - U)$ is the case where $K_t = M$ for all t. We now present a reformulation result available for the $LS - U$. Consider the variables:

- W_{ut}: amount produced in period u to satisfy demand in period t.

The facility location formulation Q^{FL-U} is described as:

$$\sum_{u=1}^{t} W_{ut} = D_t \quad \forall t, \tag{5.13}$$
$$W_{ut} \leq D_t Y_u \quad \forall u, t \text{ with } u \leq t, \tag{5.14}$$
$$X_u = \sum_{t=u}^{NT} W_{ut} \quad \forall u, \tag{5.15}$$
$$Y \in [0,1]^{NT}, W_{ut} \in \mathbb{R}^1_+, \tag{5.16}$$
$$Y \in \mathbb{Z}^{NT}. \tag{5.17}$$

Constraints (5.13) guarantee the demand of each period is satisfied. Constraints (5.14) fix the setup variables to 1 in case production occurs. Constraints (5.15) link the facility location variables with the original production variables. Constraints (5.16) and (5.17) are constraints on the variables. Denote $Q^{FL-U} = \{(X,Y,W) : (5.13) - (5.16)\}$.

Theorem 5.1. *(Krarup and Bilde [24]) $proj_{X,Y} Q^{FL-U} = proj_{X,Y} conv(X^{LS-U})$.*

The Wagner-Whitin relaxation of the single item lot-sizing set is given by

$$(WW-C)\ S_{k-1} + \sum_{u=k}^{t} K_u Y_u \geq D_{kt} \quad \forall\ k,t, \text{ with } k \leq t,$$
$$S \in \mathbb{R}^{NT},\ Y \in \{0,1\}^{NT}.$$

Consider the following formulation for the uncapacitated version of the Wagner-Whitin relaxation:

$$S_{k-1} \geq \sum_{j=k}^{l} D_j(1 - Y_k - \ldots - Y_j) \quad \forall\ k,l \text{ with } k \leq l, \tag{5.18}$$
$$S \in \mathbb{R}^{NT+1}_+, Y \in [0,1]^{NT}, \tag{5.19}$$

where (5.18) are (l,S) inequalities for the Wagner-Whitin relaxation. This relaxation is also also valid when dealing with integer variables, which means $Y \in \mathbb{R}^{NT}_+$ would be the relaxation of $Y \in \mathbb{Z}^{NT}_+$. Define $R^{WW-U} = \{S, Y : (5.18) - (5.19)\}$.

Theorem 5.2. *(Pochet and Wolsey [39]) $R^{WW-U} = conv(X^{WW-U})$*

Consider now the constant capacity Wagner-Whitin problem $(WW-CC)$ where $K_t = CC$ for all t.

Proposition 5.3. *(Pochet and Wolsey [39]) A tight extended formulation for $conv(X^{WW-CC})$ is*

$$S_{k-1} = C \sum_{t\in[k,NT]} \zeta^k_t \delta^k_t + C\mu^k \quad \forall k, \tag{5.20}$$
$$\sum_{u=k}^{t} Y_u \geq \sum_{\tau\in\{0\}\cup[k,NT]} \left\lceil \frac{D_{kt}}{C} - \zeta^k_\tau \right\rceil \delta^k_\tau - \mu^k \quad \forall k,t, \text{ with } k \leq t, \tag{5.21}$$
$$\sum_{t\in\{0\}\cup[k,NT]} \delta^k_t = 1 \quad \forall k, \tag{5.22}$$
$$\mu^k \geq 0, \delta^k_t \geq 0 \quad \forall t \in \{0\}\cup[k,NT],\ \forall k, \tag{5.23}$$
$$Y \in [0,1]^{NT}, \tag{5.24}$$

where $\zeta_0^k = 0$, $[k, NT] = \{k, \ldots, NT\}$ *and* $\zeta_\tau^k = \frac{D_{k\tau}}{C} - \left\lfloor \frac{D_{k\tau}}{C} \right\rfloor$. *The additional variables* δ_t^k *indicate that* $S_{k-1} = C\zeta_t^k \mod C$.

We now show that several relaxations of $2L-PT$ give rise to such single item sets.

5.2.1 An Echelon Stock Lot-sizing Relaxation - No Sales

Definition 5.1. *The echelon stock of item* i *in period* t *is the sum of the stock of* i *in every production site and client area in period* t.

By summing (5.1) over p,

$$\sum_{p=1}^{NP} s_{t-1}^{0ip} + \sum_{p=1}^{NP} x_t^{0ip} = \sum_{p=1}^{NP}\sum_{c=1}^{NC} x_t^{1ipc} + \sum_{p=1}^{NP} s_t^{0ip} \quad \forall\, i, t. \tag{5.25}$$

By summing (5.2) over c,

$$\sum_{c=1}^{NC} s_{t-1}^{1ic} + \sum_{c=1}^{NC}\sum_{p=1}^{NP} x_t^{1ipc} = \sum_{c=1}^{NC} d_t^{ic} + \sum_{c=1}^{NC} s_t^{1ic} \quad \forall\, i, t. \tag{5.26}$$

We can add (5.25) and (5.26), and together with a relaxation of (5.4) we obtain the following echelon stock relaxation:

$$\sum_{c=1}^{NC} s_{t-1}^{1ic} + \sum_{p=1}^{NP} s_{t-1}^{0ip} + \sum_{p=1}^{NP} x_t^{0ip} = \sum_{c=1}^{NC} d_t^{ic} + \sum_{p=1}^{NP} s_t^{0ip} + \sum_{c=1}^{NC} s_t^{1ic} \quad \forall\, t,$$
$$\sum_{p=1}^{NP} x_t^{0ip} \le M \sum_{p=1}^{NP} y_t^{ip} \quad \forall\, t.$$

We can take, for each $t = 1, \ldots, NT$

- $S_t = \sum_{c=1}^{NC} s_{t-1}^{1ic} + \sum_{p=1}^{NP} s_{t-1}^{0i,p}$,
- $X_t = \sum_{p=1}^{NP} x_t^{0ip}$,
- $Y_t = \sum_{p=1}^{NP} y_t^{ip}$ and $D_t = \sum_{c=1}^{NC} d_t^{ic}$,

and we obtain a set of the form $LS-U$. We will later use the reformulation available for the $WW-U$ relaxation, namely (5.18)-(5.19).

5.2.2 A Constant Capacity Transportation Relaxation - No Sales

Here we consider a set formed by using constraints (5.2) and (5.4).

Consider a subset $I \subseteq \{1, \ldots, NI\}$ of the items. Summing (5.2) over $i \in I$, we get

$$\sum_{i \in I} s_{t-1}^{1ic} + \sum_{i \in I} \sum_{p=1}^{NP} x_t^{1ipc} = \sum_{i \in I} d_t^{ic} + \sum_{i \in I} s_t^{1ic} \quad \forall\ c, t. \tag{5.27}$$

In addition, by summing (5.4) over p,

$$\sum_{p=1}^{NP} \sum_{i=1}^{NI} x_t^{1ipc} \leq \sum_{p=1}^{NP} K_1 Y_t^{pc} \quad \forall\ c, t. \tag{5.28}$$

Therefore, by using (5.27) and (5.28) we obtain the relaxation

$$\begin{aligned} \sum_{i \in I} s_{t-1}^{1ic} + \sum_{i \in I} \sum_{p=1}^{NP} x_t^{1ipc} &= \sum_{i \in I} d_t^{ic} + \sum_{i \in I} s_t^{1ic} \quad \forall\ c, t, \\ \sum_{p=1}^{NP} \sum_{i \in I} x_t^{1ipc} &\leq K_1 \sum_{p=1}^{NP} Y_t^{pc} \quad \forall\ c, t. \end{aligned}$$

Taking for $t = 1, \ldots, NT$

- $S_t = \sum_{i \in I} s_t^{1ic}$,
- $X_t = \sum_{i \in I} \sum_{p=1}^{NP} x_t^{1ipc}$,
- $Y_t = \sum_{p=1}^{NP} Y_t^{pc}$ and
- $D_t = \sum_{i \in I} d_t^{ic}$,

we have a set of the form $LS - CC$.

We will only consider the cases in which we take individual items $I = \{i\}$ or all items together $I = \{1, \ldots, NI\}$. We will use either the reformulation for the $WW-U$ relaxation, (5.18)-(5.19), or the one for the $WW-CC$ relaxation, (5.20)-(5.24).

5.2.3 Extended Formulations

Consider the basic uncapacitated problem, i.e. $K_1 = M$. In this case one can replace the constraints fixing the transportation setup variables (5.4) by the constraints $x_t^{1ipc} \leq M Y_t^{pc}$ for each i.

Now we can write the feasible region

$$X^{2L-PT} = \bigcap_{i=1}^{NI} X_i^{2L-MP-U},$$

where $X_i^{2L-MP-U}$ is the set

$$(2L - MP - U_i) \quad s_{t-1}^{0ip} + x_t^{0ip} = \sum_{c=1}^{NC} x_t^{1ipc} + s_t^{0ip} \quad \forall\, p, t, \tag{5.29}$$

$$s_{t-1}^{1ic} + \sum_{p=1}^{NP} x_t^{1ipc} = d_t^{ic} + v_t^{ic} + s_t^{1ic} \quad \forall\, c, t, \tag{5.30}$$

$$x_t^{0ip} \leq M y_t^{ip} \quad \forall\, p, t, \tag{5.31}$$

$$x_t^{1ipc} \leq M Y_t^{pc} \quad \forall\, p, c, t, \tag{5.32}$$

$$0 \leq v_t^{ic} \leq U_t^{ic} \quad \forall\, c, t, \tag{5.33}$$

$$y \in \{0,1\}^{NP \times NT}, Y \in \mathbb{Z}_+^{NP \times NC \times NT}. \tag{5.34}$$

Note that the only variables linking the sets $X_i^{2L-MP-U}$ are the transportation set-up variables Y.

5.2.3.1 Multi-commodity Formulation for the case without Sales ($U_t^{ic} = 0$)

We now present a multi-commodity reformulation for the set $X_i^{2L-MP-U}$. Each demand d_t^{ic} for the triple i, c, t is viewed as a distinct product. Thus we introduce the following variables:

- w_{tu}^{0ipc}: amount produced, at production site p, of product i in period t to satisfy demand of period u for client c,
- w_{tu}^{1ipc}: amount transported from production site p of product i in period t to satisfy demand of period u for client c,
- σ_{tu}^{0ipc}: amount stocked of product i at production site p at the end of period t to satisfy demand of period u for client c,
- σ_{tu}^{1ic}: amount stocked of product i at client c at the end of period t to satisfy demand of period u.

The resulting multi-commodity formulation (MC_i) is as follows:

$$
\begin{aligned}
(MC_i)\ \ & \sigma^{0ipc}_{t-1,u} + w^{0ipc}_{tu} = w^{1ipc}_{tu} + \sigma^{0ipc}_{tu} \quad \forall\, p,c,t,u, \text{ with } t \le u,\\
& \sigma^{1ic}_{t-1,u} + \sum_p w^{1ipc}_{tu} = d^{ic}_u \delta_{tu} + \sigma^{1ic}_{tu} \quad \forall\, c,t,u, \text{ with } t \le u,\\
& w^{0ipc}_{tu} \le d^{ic}_u y^{ip}_t \quad \forall\, p,c,t,u, \text{ with } t \le u,\\
& w^{1ipc}_{tu} \le d^{ic}_u Y^{pc}_t \quad \forall\, p,c,t,u, \text{ with } t \le u,\\
& x^{0ip}_t = \sum_{c=1}^{NC}\sum_{u=t}^{NT} w^{0ipc}_{tu} \quad \forall\, p,t,\\
& x^{1ipc}_t = \sum_{u=t}^{NT} w^{1ipc}_{tu} \quad \forall\, p,c,t,\\
& w^{0ipc}_{tu}, w^{1ipc}_{tu}, \sigma^{1ic}_{tu}, \sigma^{0ipc}_{tu} \in \mathbb{R}^1_+ \quad \forall\, p,c,t,u, \text{ with } t \le u,\\
& y \in \{0,1\}^{NP\times NT}, Y \in \mathbb{Z}_+^{NP\times NC\times NT},
\end{aligned}
$$

where $\delta_{tu} = 1$ if $t = u$ and $\delta_{tu} = 0$ otherwise. Here the first two sets of constraints are flow balance constraints at production sites and client areas, the next two are the tightened variable upper bound constraints, and the last two link the original production and transportation variables to the corresponding multi-commodity variables.

5.2.3.2 Facility Location Formulation for the case with Sales

One can produce a multi-commodity formulation for the case with sales by doubling the number of variables, but a slightly more compact variant (without the need of stock variables) can be obtained by generalizing the facility location reformulation of Krarup and Bilde [24] for $LS-U$ that has been extended to treat sales in Loparic et al. [29]. Let $D^{ic}_t = d^{ic}_t + U^{ic}_t$ be the maximum amount of item i that can be delivered to client c in period t.

Define the variables:

- γ^{ipc}_{tu}: fraction of the demand d^{ic}_u produced at production site p in period t,
- ϕ^{ipc}_{tu}: fraction of D^{ic}_u produced at production site p in period t,
- α^{ipc}_{tu}: fraction of d^{ic}_u transported from producer p to client c in period t,
- β^{ipc}_{tu}: fraction of D^{ic}_u transported from producer p to client c in period t.

The resulting facility location formulation for item i is:

$$
\begin{aligned}
(FCS_i) \quad & \sum_{p=1}^{NP}\sum_{t=1}^{u}(d_u^{ic}\alpha_{tu}^{ipc} + D_u^{ic}\beta_{tu}^{ipc}) = d_u^{ic} + v_u^{ic} \quad \forall\, c,u, && (5.35)\\
& \sum_{p=1}^{NP}\sum_{t=1}^{u}(\alpha_{tu}^{ipc} + \beta_{tu}^{ipc}) = 1 \quad \forall\, c,u, && (5.36)\\
& \sum_{p=1}^{NP}\sum_{t=1}^{u}(\gamma_{tu}^{ipc} + \phi_{tu}^{ipc}) = 1 \quad \forall\, c,u, && (5.37)\\
& \sum_{k=1}^{t}\alpha_{ku}^{ipc} \leq \sum_{k=1}^{t}\gamma_{ku}^{ipc} \quad \forall\, p,c,t,u, \text{ with } t \leq u, && (5.38)\\
& \sum_{k=1}^{t}\beta_{ku}^{ipc} \leq \sum_{k=1}^{t}\phi_{ku}^{ipc} \quad \forall\, p,c,t,u, \text{ with } t \leq u, && (5.39)\\
& \gamma_{tu}^{ipc} + \phi_{tu}^{ipc} \leq y_t^{ip} \quad \forall\, p,c,t,u, \text{ with } t \leq u, && (5.40)\\
& \alpha_{tu}^{ipc} + \beta_{tu}^{ipc} \leq Y_t^{pc} \quad \forall\, p,c,t,u, \text{ with } t \leq u, && (5.41)\\
& \sum_{c=1}^{NC}\sum_{u=t}^{NT}(d_u^{ic}\gamma_{tu}^{ipc} + D_u^{ic}\phi_{tu}^{ipc}) = x_t^{0ip} \quad \forall\, p,t, && (5.42)\\
& \sum_{u=t}^{NT}(d_u^{ic}\alpha_{tu}^{ipc} + D_u^{ic}\beta_{tu}^{ipc}) = x_t^{1ipc} \quad \forall\, p,c,t, && (5.43)\\
& \alpha_{tu}^{ipc}, \beta_{tu}^{ipc}, \phi_{tu}^{ipc}, \gamma_{tu}^{ipc} \in [0,1] \quad \forall\, p,c,t,u, \text{ with } t \leq u, && (5.44)\\
& y \in \{0,1\}^{NP\times NT}, Y \in \{0,1\}^{NP\times NC\times NT}. && (5.45)
\end{aligned}
$$

Constraints (5.35) impose that the demand plus the amount to be sold is transported to the clients. Constraints (5.36) and (5.37) state that the fractions of the demand produced and transported sum to 1. Constraints (5.38) and (5.39) guarantee that the amount produced up to a certain period to satisfy a demand in that or a later period is at least as large as the amount transported under the same conditions. Constraints (5.40) and (5.41) ensure that the setup (fixed cost) variables are activated when production/transportation occurs. Constraints (5.42) and (5.43) link the facility location variables to the original production, transportation, sales and stock variables.

5.3 The Hybrid MIP Heuristic

It will be seen below that the application of single item lot-sizing reformulations to $2L-PT$ leads to lower bounds that are still weak, so that the Lagrangean or MIP-based heuristics often used for different production planning problems are unlikely to perform well, and even when they can be adapted for $2L-PT$, they may not be able to find good primal feasible solutions. Using extended formulations that are not restricted to a single level is more promising as these provide reasonably strong lower bounds. Here the heuristic of Eppen and Martin using the MC_i extended formulation for each item i is a natural candidate, but it turns out that even after variable fixing the resulting formulation remains large and it is not possible to find good solutions quickly via branch and bound. This suggests that one might use a second smaller formulation in order to obtain primal feasible solutions quickly. Thus the question is now how to work with two formulations in an effective way.

5.3.1 The Hybrid Heuristic using Two Distinct MIP Formulations

The idea behind the heuristic is to use a strong formulation with a large number of variables and constraints to try to get some information about what would be a good structure for a solution. Then, to use a weaker formulation with much less variables and constraints to perform a local search around the structure provided by the larger formulation in hope to get reasonably good solutions in a limited amount of time.

We first give some definitions and assumptions, followed by a high-level description of the heuristic based on two distinct MIP formulations. After that, we specialize the heuristic to the problem $2L-PT$.

5.3.1.1 Definitions and Assumptions

We are given a (mixed) integer problem

$$z^* = \min\{cx : x \in X\}$$

with initial formulation

$$P = \{x \in \mathbb{R}^n : Ax \geq b\}$$

where $X = P \cap \mathbb{Z}^n$. We assume for simplicity that all variables are integer. There is also a second formulation

$$Q = \{(x, w) \in \mathbb{R}^n \times \mathbb{R}^p : Cx + Dw \geq d\}$$

with $X = \text{proj}_x(Q) \cap \mathbb{Z}^n$ and

$$z^* = \min\{cx + 0w : Cx + Dw \geq d, x \in \mathbb{Z}^n, w \in \mathbb{R}^p\}.$$

We assume two conditions related to the strength of the formulations to be used:

Condition 1: $z_{LP}(P) = \min\{cx : x \in P\}$ is a relatively "weak" lower bound for z^*. The formulation may have been improved by some tightening with cuts, but the number of variables is "small".
Condition 2:

$$\text{conv}(X) \subset \text{proj}_x(Q) \subset P$$

with $\text{proj}_x(Q)$ a "good" approximation of $\text{conv}(X)$ so that

$$z_{LP}(Q) = \min\{cx : x \in Q\}$$

provides a "strong" lower bound on z^*.

5.3.1.2 The Hybrid Heuristic

We now describe the heuristic in general terms. Let $(\bar{x}, \bar{w})$ denote the linear programming solution of LP(Q) and $\hat{x}$ the linear programming solution of LP(P). The heuristic can be described by the following three steps:
Step 1: Solve the linear program LP(Q) and possibly also the linear program LP(P).
Step 2: Use $\bar{x}$ and possibly $\hat{x}$ to determine a neighborhood $N_Q(\bar{x})$ of $\bar{x}$.
Step 3: Run the MIP:

$$\min\{cx : x \in P \cap N_Q(\bar{x}) \cap \mathbb{Z}^n\}$$

for a limited amount of time. The best solution found $\tilde{x}^H$ (if any) is the hybrid heuristic solution with value $z^H = c\tilde{x}^H$ guaranteed to be within $100\frac{z^H - z_{LP}(Q)}{z^H}\%$ of optimal.

5.3.1.3 Choice of the Neighborhood

One important decision when producing local search heuristics is the choice of the neighborhoods in which improvements will be tried to be achieved. In this section we give some examples of neighborhoods that can be used in our heuristic framework.

There are several different ways in which one might select the neighborhood $N_Q(\bar{x})$. One obvious way is to fix or restrict the values of certain integer variables. Let $T \subseteq \{1, \ldots, n\}$ denote a chosen subset of the integer variables, and $S \subseteq \mathbb{Z}_+$ a chosen set of integer values that these integer variables may take. Following are three examples of neighborhood that can be used in the presented framework.
i) Fixing: Fix integer variables in T taking integer values in S:

$$N_Q(\bar{x}) = \{x \in \mathbb{Z}^n : x_j = \bar{x}_j \text{ if } \bar{x}_j \in S \subseteq \mathbb{Z}_+, j \in T\}.$$

ii) RINS: Motivated by Relaxation Induced Neighborhood Search, fix variables in T if they take the same integer value in both linear programming solutions:

$$N_Q(\bar{x}) = \{x \in \mathbb{Z}^n : x_j = \bar{x}_j \text{ if } \bar{x}_j = \hat{x}_j,\ j \in T\}.$$

iii) Local Branching: Another possibility is to impose that the integer variables in T do not differ too much from their values in the linear programming solution, with k a small positive value:

$$N_Q(\bar{x}) = \{x \in \mathbb{Z}^n : \sum_{j \in T} |x_j - \bar{x}_j| \leq k\}.$$

5.3.1.4 The Hybrid Heuristic for 2L-PT

We now describe the main variants used in the experiments of this chapter. P corresponds to the original formulation (STD) and Q is the formulation tightened by using the multi-commodity reformulation approach described in Section 5.2. We describe three different variants of the heuristic. Let $(\bar{w}^0, \bar{w}^1, \bar{x}, \bar{z}, \bar{y}, \bar{Y})$ be the optimal solution of the linear relaxation of Q and $(\hat{x}, \hat{z}, \hat{y}, \hat{Y})$ be the optimal solution of the linear relaxation of P with addition of cuts at the top node by the MIP optimizer:

Heuristic 1 (H1): (Fixing-based with T including all the y and Y variables and $S = \mathbb{Z}_+$) We fix every variable that takes an integer value in the linear relaxation of the multi-commodity formulation. We take

$$N_Q = \{y : y_t^{ip} = \bar{y}_t^{ip} \text{ if } \bar{y}_t^{ip} \in \mathbb{Z}^1\} \cup \{Y : Y_t^{pc} = \bar{Y}_t^{pc} \text{ if } \bar{Y}_t^{pc} \in \mathbb{Z}^1\}.$$

Heuristic 2 (H2): (RINS-based with S and T as in Heuristic 1) We fix every variable that takes the same integer value in both the linear relaxations of the multi-commodity and the standard formulations. We take

$$N_Q = \{y : y_t^{ip} = \bar{y}_t^{ip} \text{ if } \bar{y}_t^{ip} = \hat{y}_t^{ip} \text{ and } \bar{y}_t^{ip} \in \mathbb{Z}^1\} \cup \\ \{Y : Y_t^{pc} = \bar{Y}_t^{pc} \text{ if } \bar{Y}_t^{pc} = \hat{Y}_t^{pc} \text{ and } \bar{Y}_t^{pc} \in \mathbb{Z}^1\}.$$

Heuristic 3 (H3): (Fixing-based with T including all the y and Y variables and $S = \{0\}$) We fix every variable that takes the value 0 in the linear relaxation of the multi-commodity formulation. We take

$$N_Q = \{y : y_t^{ip} = \bar{y}_t^{ip} \text{ if } \bar{y}_t^{ip} = 0\} \cup \{Y : Y_t^{pc} = \bar{Y}_t^{pc} \text{ if } \bar{Y}_t^{pc} = 0\}.$$

The complement of Heuristic 3, namely the fixing of the variables taking nonzero integer values, is a natural extension but it will not be used based on some observations in the computational experiments.

We observe that in more constrained variants (where big bucket constraints and/or bounds on stock constraints are present), the fixing of both y and Y variables can occasionally lead to infeasible subproblems. In such cases one possibility is to limit the candidate set T to the production setup variables y.

5.4 Computational Results

In this section we present results of computational tests using the formulations of Section 5.2 and the hybrid heuristics of Section 5.3.

Taking a first set of seven instances, in a first moment we test the effectiveness of the multi-commodity reformulation for the uncapacitated problem. We compare the direct use of the solver over the standard formulation, direct optimization over the multi-commodity reformulation and optimization over the standard formulation strengthened with reformulations for the single item relaxations.

In a second moment we vary the transportation capacities and the machine/storage capacities. We compare direct optimization over the original model, direct optimization over the multi-commodity reformulation and three versions of the hybrid heuristic. From this we select one version of the heuristic that is used in most of the later runs. This version of the heuristic is then further tested on a second data set consisting of over 400 randomly generated instances.

We also analyze the general problem with sales. We generate some instances for the problem and we compare direct optimization over the standard formulation with the use of hybrid heuristics.

All experiments were performed on a machine running under Ubuntu 9.10 with a Intel Xeon 2.93GHz processor, 8 Gb of RAM memory using Xpress-Optimizer version 21.01.00.

5.4.1 The Test Instances

Our initial test set consists of seven instances whose dimensions are presented in Table 5.1.

The demands and time independent costs are uniformly distributed in the specified intervals: $d_t^{ic} \in [0, 20]$, $h^{0ip} \in [0, 0.2]$, $h^{1ic} \in [0, 0.05]$, $q^{0ip} \in [100, 400]$, $q^{1pc} \in [20, 50]$, $p^{1ipc} = 0$ and $p^{0ip} = 0$. These values were chosen to provide instances with non-trivial solutions, but no special effort was made to create hard instances.

5.4.2 The Basic Uncapacitated Problem - No Sales

For the uncapacitated problem, we compare the behavior of an MIP solver on the initial formulation (STD) as well as using the different formulations shown

Table 5.1: Sizes of instances

Inst	Size
A1	$NI = 10$, $NP = 1$, $NC = 10$, $NT = 12$
A2	$NI = 5$, $NP = 2$, $NC = 12$, $NT = 12$
A3	$NI = 10$, $NP = 3$, $NC = 20$, $NT = 12$
A4	$NI = 5$, $NP = 5$, $NC = 40$, $NT = 12$
A5	$NI = 5$, $NP = 2$, $NC = 20$, $NT = 24$
A6	$NI = 10, NP = 2, NC = 10, NT = 24$
A7	$NI = 10$, $NP = 2$, $NC = 20$, $NT = 24$

in Section 5.2. Specifically we compare three approaches:

1. Take the original formulation (STD) and run the default version of the MIP optimizer;

2. Add the multi-commodity reformulation (MC_i) from Section 5.2.3.1 for each item to (STD) and run the MIP optimizer (with system cuts turned off and the barrier algorithm used to solve the initial linear program). In the what follows we refer to this as the multi-commodity formulation MC;

3. Add the single item lot-sizing reformulations (ES) for each item to (STD), specifically the convex hull of solutions of the echelon stock formulation $WW - U$ from Section 5.2.1 and the convex hull of solutions of $WW - U$ for each client individually and also for the aggregation of all the clients from Section 5.2.2.

In Table 5.2 we see the results obtained for the uncapacitated problem $(K_1 = M)$ with a maximum run time for the branch and bound phase of 3600 seconds. Column 1 indicates the instance, followed by information about the different approaches. z_{LP} is the value of the linear relaxation, z_{XLP} is the lower bound after the addition of the solver cuts in the root node, $\bar{z}^{FORM}$ is the best integer solution found using formulation $FORM$ at the end of the execution and Γ_{FORM} is the remaining gap at the end of the time limit (3600 seconds) when using formulation $FORM$. Since all the instances were solved within the time limit when using (MC) the column *time* indicates the time in seconds to solve the instance to optimality.

The results show that the multi-commodity formulation is very effective and allows us to solve all seven test instances. Typically the linear programming solution is integer, with the exception of instance A7 for which the branch-and-bound tree contains just three nodes.

Table 5.2: Results for the uncapacitated instances

	(STD)				(MC)			(ES)			
Inst	z_{LP}	z_{XLP}	$\bar{z}^{STD}$	Γ_{STD}	z_{LP}	$\bar{z}^{MC}$	$time$	z_{LP}	z_{XLP}	$\bar{z}^{ES}$	Γ_{ES}
A1	2827.0	3018.9	4237.2	0.0	4237.2	4237.2	2	3836.2	4099.1	4237.2	0.0
A2	15.1	1755.8	2431.0	0.0	2431.0	2431.0	2	1144.7	2312.1	2431.0	0.0
A3	45.2	54.2	5618.2	83.5	5618.2	5618.2	20	2635.4	4594.4	5695.8	15.8
A4	41.0	755.8	5930.0	72.3	5270.0	5270.0	24	3292.0	4410.1	6513.0	31.3
A5	48.3	3209.1	9529.4	64.9	8119.7	8119.7	41	2689.8	5870.0	10365.4	43.3
A6	55.4	2788.0	8439.0	60.7	6918.4	6918.4	34	3678.6	5637.6	8872.8	36.4
A7*	124.2	1823.4	17173.4	86.0	13237.1	13239.9	452	5159.1	8298.3	16869.2	50.8

5.4.3 Capacitated Versions - No Sales

In this section we give some results on the performance test of the three different variants of the heuristic against the performance of the MIP solver using the two formulations STD and MC. The time allowed for the heuristics was the time to solve the linear programming relaxation of the multi-commodity reformulation (step 1 of the hybrid heuristic) plus 100 seconds to solve the restricted MIP (step 3 of the hybrid heuristic).

In Table 5.3 we see the results obtained for the basic model with $K_1 \in \{50, 100, 200\}$. The value K_1 is the transportation capacity of each vehicle, $z_{LP(MC)}$ is the value of the linear relaxation of (MC), $time$ is the time in seconds to solve the $LP(MC)$ plus 100 seconds that was used to run (STD) with the values fixed, $\bar{z}^{HEUR}$ is the value found by heuristic $HEUR$, Γ_{H3} is the percentage gap between $z_{LP(MC)}$ and $\bar{z}^{H3}$, z_{XLP} is the lower bound after the addition of the solver cuts in the root node, $\bar{z}^{STD}$ is the best solution obtained by the solver using (STD), Γ_{STD} is the remaining gap at the end of the time limit (3600 seconds) when using formulation STD, $\bar{z}^{MC}$ is the best solution obtained by the solver using the multi-commodity reformulation and Γ_{MC} is the remaining gap at the end of the time limit (3600 seconds) when using the multi-commodity reformulation.

The values in bold indicate when the heuristic H3 produced a solution better than or as good as that arising while using the solver on both MIP formulations STD and MC. We see that in 13 out of the 21 instances, heuristic H3 did best. What is more it is almost always best on the larger instances A5-A7. We also tested the heuristic of Eppen and Martin. Variables are fixed in the extended formulation Q rather than in the smaller formulation P, so that the MIP in Step iii) of the heuristic is: $\min\{cx : x \in Q \cap N_Q\bar{x} \cap \mathbb{Z}^n\}$. This worked well for the smaller instances, but for the larger instances only a small number of nodes could be enumerated in the limited time available, and hence typically good primal solutions were not found.

Next we consider instances involving the different joint capacity constraints. From now on we just compare the heuristic H3 against the two different formulations. In Table 5.4 we summarize the results. The columns represent the same values as in the previous table with the exception that we have added

Table 5.3: Results for constant capacity instances

		$(HEURISTICS)$						(STD)			(MC)	
Inst	K_1	$z_{LP(MC)}$	$time$	$\bar{z}^{H1}$	$\bar{z}^{H2}$	$\bar{z}^{H3}$	Γ_{H3}	z_{XLP}	$\bar{z}^{STD}$	Γ_{STD}	$\bar{z}^{MC}$	Γ_{MC}
A1	50	12101.6	102	12309.3	12331.6	**12309.1**	1.7	10938.0	12313.9	1.6	12309.1	1.6
A1	100	7981.2	102	8164.7	8169.8	8164.7	2.2	6819.0	8179.1	2.1	8164.1	1.8
A1	200	5924.2	102	6106.4	6105.7	6106.4	3.0	4741.3	6126.7	2.8	6105.6	1.8
A2	50	6576.8	104	6908.2	7045.9	6952.2	5.4	5314.3	7002.2	15.8	6889.2	4.1
A2	100	4278.8	103	4579.7	4668.8	**4579.7**	6.6	3322.4	4702.1	20.8	4632.8	7.0
A2	200	3134.8	103	3621.0	3401.1	3621.0	13.4	2388.0	3518.2	18.5	3405.2	5.6
A3	50	21323.3	138	22550.3	23244.4	**22573.8**	5.5	14378.6	22625.6	30.2	22729.5	6.0
A3	100	13002.1	134	14198.6	14857.3	**14164.7**	8.2	7283.8	14422.2	40.0	14391.3	9.5
A3	200	8851.8	138	9725.3	10392.3	**9883.9**	10.4	3752.6	9891.3	48.4	10331.8	14.1
A4	50	17267.3	156	19221.6	20137.5	19269.5	10.4	11766.9	18996.6	34.6	19501.7	11.3
A4	100	10434.5	157	12068.0	13040.4	**12064.4**	13.5	6181.0	12530.2	48.6	12523.2	16.5
A4	200	7003.8	150	8016.2	9145.8	8016.2	12.6	3602.3	8695.4	52.9	7842.9	9.9
A5	50	21700.7	163	22173.8	23316.9	22231.1	2.4	16517.6	23137.3	27.7	22158.1	2.1
A5	100	14270.4	169	14841.9	16184.8	**14845.7**	3.9	9522.2	15646.5	38.9	15014.1	4.9
A5	200	10630.5	172	11187.7	12464.3	**11140.2**	4.6	6065.4	12278.0	50.4	11390.7	6.6
A6	50	19625.5	173	20870.5	21611.0	**20864.0**	5.9	14883.0	21666.2	29.9	21106.2	7.0
A6	100	12540.8	148	13114.9	15193.9	**13106.4**	4.3	8332.5	14806.9	42.4	13356.5	6.0
A6	200	9002.3	152	9545.1	11523.7	**9545.1**	5.7	5373.5	11149.1	50.0	9813.3	8.2
A7	50	39600.2	241	40993.5	43406.8	**40974.0**	3.4	29213.9	41688.5	27.1	41794.8	5.2
A7	100	24492.8	276	26172.7	27991.2	26114.4	6.2	15938.9	26232.7	36.0	26108.4	6.1
A7	200	17064.8	254	18798.1	22186.0	**18646.9**	8.5	9246.5	20579.9	52.5	19204.9	11.1

two columns giving the joint capacities LK and SK.

Table 5.4: Results with joint production/storage constraints

				$(HEURISTIC)$				(STD)			MC	
Inst	K_1	LK	SK	$z_{LP(MC)}$	$time$	$\bar{z}^{H3}$	Γ_{H3}	z_{XLP}	$\bar{z}^{STD}$	Γ_{STD}	$\bar{z}^{MC}$	Γ_{MC}
A1	M	3000	M	6170.5	103	6774.9	8.9	3628.4	6968.7	34.0	6738.7	3.5
A1	M	M	300	6008.1	103	6927.0	13.3	3095.4	7155.9	42.5	6869.1	10.5
A1	200	3000	300	7921.0	104	8496.3	6.8	5366.9	8607.4	28.5	8371.4	3.9
A2	M	3000	M	2731.3	104	3127.4	12.7	2115.8	3287.9	29.2	3099.5	0.0
A2	M	M	300	2875.6	104	**3032.5**	5.2	2189.2	3545.6	24.7	3032.5	0.0
A2	200	3000	300	3255.8	107	3718.5	12.4	2718.3	4094.7	28.9	3646.5	7.3
A3	M	3000	M	6884.5	210	11372.7	39.5	1315.9	12889.5	76.3	10484.6	34.3
A3	M	M	300	6962.7	146	9290.0	25.1	309.9	13655.4	86.0	9150.1	23.9
A3	200	3000	300	9327.6	166	**13623.5**	31.5	5312.8	15602.1	58.8	37161.0	74.9
A4	M	3000	M	5965.6	225	**10187.9**	41.4	3411.5	10836.7	64.0	10460.6	43.0
A4	M	M	300	5804.9	196	**6079.0**	4.5	3104.4	13274.4	75.8	6322.6	7.9
A4	200	3000	300	7191.1	229	**11339.6**	36.6	5491.9	11998.0	54.0	26965.7	73.3
A5	M	3000	M	8988.4	230	10466.6	14.1	4490.6	14522.9	68.4	10277.6	12.4
A5	M	M	300	10333.8	209	**14836.7**	30.3	3244.9	16180.5	77.1	45632.9	77.4
A5	200	3000	300	11455.6	225	**15367.2**	25.5	7316.3	15889.6	53.8	39170.3	70.8
A6	M	3000	M	7577.6	209	**10685.1**	29.1	4208.9	12548.8	64.9	11314.2	33.0
A6	M	M	300	7518.7	190	**10554.3**	28.8	3720.4	14171.9	69.9	11113.7	32.3
A6	200	3000	300	9242.4	191	**12288.8**	24.8	6817.0	14882.1	53.9	27358.0	66.2
A7	M	3000	M	15211.8	611	**23610.5**	35.6	5247.8	25665.6	74.7	59605.8	74.5
A7	M	M	300	14532.1	448	**20670.2**	29.7	3044.8	26605.5	84.7	50832.8	71.4
A7	200	3000	300	18702.8	509	**27449.1**	31.9	12714.3	27757.6	52.0	65603.2	71.5

Here the heuristic takes between 200 and 700 seconds (because of the variable time to solve the linear program over Q), while the optimizer is run on each instance for one hour. The table shows that in 13 out of the 21 instances, heuristic H3 generated better solutions than those generated by the MIP solver using both formulations. We see that as the size of instances increased (A4 and bigger) the effectiveness of the heuristic is more evident as in only one instance was the result worse than those found by the solver. It is well known and this is confirmed in the Table that duality gaps can be very large as soon as there are big bucket or joint storage capacity constraints. Unfortunately, even though the heuristic outperforms the MIP optimizer for most of the instances, the gaps are still very large for certain of the instances.

5.4.4 A Second Test Set

We generated a different set of instances in order to further investigate the quality of the solutions produced by the hybrid heuristic H3. Here after solving the linear program over Q and fixing variables, one runs branch-and-cut on the weaker formulation for 100 seconds.

We consider the following parameter sizes $NI = \{5, 10\}$, $NP = \{2, 4\}$, $NC = \{15, 30\}$ and $NT = \{12, 24\}$. All the random data is uniformly distributed. Integer valued demands are generated every 3 periods, namely if $t \bmod 3 = 0$, then $d_t^{ic} \in [0, 50]$. There is one random value associated to each item $\psi_i \in [0.1, 1.0]$ and one associated with each production site $\xi_p \in [0.9, 1.1]$. Storage costs depend only on the item and are determined as $h^{0ip} = 0.2\psi_i$ and $h^{1ic} = 0.25\psi_i$. Production costs are $p^{0ip} = 0.09\psi_i\xi_p$. Production sites and clients are uniformly distributed in a 1.0×1.0 box and we denote $dist(p, c)$ the euclidean distance between production site p and client c. Transportation costs are $p^{1ipc} = 0.03 \times dist(p, c)$. Fixed costs are $q^{0ip} \in [200, 800]$, $q^{1pc} \in [30, 60]$. Capacities on the vehicles can assume the values $K_1^a = 100$ and $K_1^b = 200$, while joint production capacities at the production sites can take the values $LK^a = \left\lfloor \sum_{i,c,t} d_t^{ic}/NT \right\rfloor$, $LK^b = \left\lfloor 0.7 \times \sum_{i,c,t} d_t^{ic}/NT \right\rfloor$, $LK^c = \left\lfloor 1.3 \times \sum_{i,c,t} d_t^{ic}/(NT \times NP) \right\rfloor$.

For each possible combination of sizes, we generate 5 instances with each of the capacity values, giving a total of 400 instances created.

Tables 5.5 and 5.6 sumarize the results for the instances with 5 and 10 items respectively. The columns show in order the size of the instance, the capacity constraint, the geometric mean time used for the five instances and the geometric mean duality gap for the five instances.

The results show that for this second set of instances the heuristic gives solutions whose quality guarantees are on average a little lower than those of the first test set. As before the gaps for the instances with transportation capacity constraints are considerably lower than those with budget capacities. For the instances with 10 items tightening budget constraints typically leads to larger duality gaps as one might expect, though this does not seem to hold for the instances with 5 items. The results on some of the largest instances appear to be affected by the 100 second time limit. Thus in all the instances

Table 5.5: Results of additional instances with 5 items

Size (NI,NP,NC,NT)	cap	time	Γ_{H3}
5,2,15,12	K_1^a	65.4	3.8
	K_1^b	1.9	2.0
	LK^a	67.5	11.6
	LK^b	91.4	10.7
	LK^c	91.1	10.3
5,2,15,24	K_1^a	110.7	1.7
	K_1^b	36.9	1.4
	LK^a	112.7	10.1
	LK^b	112.9	6.8
	LK^c	112.1	6.6
5,2,30,12	K_1^a	104.7	3.5
	K_1^b	4.9	3.2
	LK^a	106.1	11.1
	LK^b	106.4	12.4
	LK^c	106.7	13.5
5,2,30,24	K_1^a	123.6	2.9
	K_1^b	97.4	2.3
	LK^a	131.3	12.2
	LK^b	129.9	9.3
	LK^c	128.3	9.5
5,4,15,12	K_1^a	35.1	4.2
	K_1^b	6.2	1.3
	LK^a	105.4	12.1
	LK^b	105.5	11.2
	LK^c	105.4	8.2
5,4,15,24	K_1^a	154.2	2.8
	K_1^b	76.6	1.6
	LK^a	153.8	10.6
	LK^b	151.2	9.8
	LK^c	151.2	6.2
5,4,30,12	K_1^a	75.2	4.6
	K_1^b	19.9	3.1
	LK^a	117.9	16.8
	LK^b	117.9	17.2
	LK^c	116.9	11.4
5,4,30,24	K_1^a	188.4	10.5
	K_1^b	169.4	2.6
	LK^a	209.1	14.6
	LK^b	201.0	25.6
	LK^c	191.1	22.2

Table 5.6: Results of additional instances with 10 items

Size (NI,NP,NC,NT)	cap	time	Γ_{H3}
10,2,15,12	K_1^a	105.9	2.6
	K_1^b	31.2	2.1
	LK^a	37.7	7.0
	LK^b	106.6	9.1
	LK^c	107.6	9.2
10,2,15,24	K_1^a	146.3	1.3
	K_1^b	140.8	1.8
	LK^a	151.9	7.1
	LK^b	150.8	8.0
	LK^c	151.6	8.9
10,2,30,12	K_1^a	111.8	2.8
	K_1^b	114.3	2.6
	LK^a	115.4	9.1
	LK^b	115.2	11.0
	LK^c	116.3	11.4
10,2,30,24	K_1^a	187.7	1.9
	K_1^b	197.2	2.5
	LK^a	206.1	7.4
	LK^b	211.2	10.1
	LK^c	206.7	10.1
10,4,15,12	K_1^a	123.8	3.6
	K_1^b	59.3	2.2
	LK^a	128.5	12.7
	LK^b	129.2	11.1
	LK^c	127.3	18.6
10,4,15,24	K_1^a	342.5	3.4
	K_1^b	317.0	3.5
	LK^a	352.7	11.5
	LK^b	421.5	15.3
	LK^c	367.1	25.6
10,4,30,12	K_1^a	144.7	3.9
	K_1^b	151.2	4.8
	LK^a	161.3	14.4
	LK^b	162.4	17.8
	LK^c	156.9	31.8
10,4,30,24	K_1^a	461.1	17.7
	K_1^b	508.7	8.6
	LK^a	574.0	13.6
	LK^b	576.0	16.6
	LK^c	509.5	22.4

$5, 4, 30, 24$ with LK^b and LK^c and $10, 4, 30, 24$ with K_1^a, the top node heuristic solution found by Xpress was not improved on in the branch-and-cut phase, which may explain the relatively high gaps.

In Tables 5.7 and 5.8, we summarize the results for a new set of instances with 5 and 10 items respectively created to see what happens when we take the same instance and vary the capacities. Instead of creating one instance for each size and capacity, 5 instances are created for each size and the capacity is determined as for the previous test.

As in Tables 5.5 and 5.6, the columns in Tables 5.7 and 5.8 show in this order the size of the instances, the capacity constraint, the geometric mean time used for the five instances and the geometric mean gap for the five instances.

Table 5.7: Results of additional instances with 5 items (similar)

Size (NI,NP,NC,NT)	cap	time	Γ_{H3}
5,2,15,12	K_1^a	27.9	2.4
	K_1^b	1.7	2.0
	LK^a	61.7	13.5
	LK^b	102.0	12.6
	LK^c	101.9	12.4
5,2,15,24	K_1^a	110.4	2.0
	K_1^b	34.5	1.5
	LK^a	112.7	9.9
	LK^b	111.9	7.6
	LK^c	111.9	8.1
5,2,30,12	K_1^a	95.5	3.7
	K_1^b	5.2	2.7
	LK^a	106.6	14.0
	LK^b	105.8	12.8
	LK^c	106.0	12.8
5,2,30,24	K_1^a	125.0	2.4
	K_1^b	91.8	2.4
	LK^a	130.2	11.5
	LK^b	130.3	9.8
	LK^c	132.1	9.4
5,4,15,12	K_1^a	30.5	4.0
	K_1^b	7.8	2.1
	LK^a	105.3	14.3
	LK^b	105.2	12.8
	LK^c	104.8	8.4
5,4,15,24	K_1^a	145.6	3.0
	K_1^b	144.2	3.3
	LK^a	148.3	10.2
	LK^b	148.3	7.2
	LK^c	149.1	6.8
5,4,30,12	K_1^a	115.3	5.2
	K_1^b	15.7	3.2
	LK^a	117.8	17.8
	LK^b	118.1	17.7
	LK^c	116.9	10.6
5,4,30,24	K_1^a	207.1	5.8
	K_1^b	178.1	2.6
	LK^a	200.3	14.8
	LK^b	200.0	18.2
	LK^c	196.5	23.9

Table 5.8: Results of additional instances with 10 items (similar)

Size (NI,NP,NC,NT)	cap	time	Γ_{H3}
10,2,15,12	K_1^a	105.5	2.7
	K_1^b	61.4	2.2
	LK^a	83.9	7.9
	LK^b	102.9	8.9
	LK^c	94.3	9.7
10,2,15,24	K_1^a	145.0	1.2
	K_1^b	141.3	1.7
	LK^a	151.6	7.5
	LK^b	152.0	8.7
	LK^c	150.4	9.5
10,2,30,12	K_1^a	112.5	2.6
	K_1^b	111.8	2.7
	LK^a	115.8	8.6
	LK^b	115.2	10.8
	LK^c	115.7	10.8
10,2,30,24	K_1^a	176.5	1.8
	K_1^b	181.5	2.5
	LK^a	210.4	6.9
	LK^b	198.2	7.9
	LK^c	205.3	8.4
10,4,15,12	K_1^a	122.9	3.7
	K_1^b	47.9	2.3
	LK^a	126.4	11.6
	LK^b	127.2	11.9
	LK^c	125.4	16.6
10,4,15,24	K_1^a	369.8	2.8
	K_1^b	360.9	4.2
	LK^a	407.1	11.3
	LK^b	379.6	14.0
	LK^c	383.7	20.6
10,4,30,12	K_1^a	156.8	4.2
	K_1^b	148.8	3.1
	LK^a	162.4	16.5
	LK^b	161.8	18.5
	LK^c	158.7	32.9
10,4,30,24	K_1^a	548.7	12.1
	K_1^b	594.0	3.6
	LK^a	573.7	10.7
	LK^b	594.9	14.7
	LK^c	502.2	18.6

The results in Tables 5.7 and 5.8 are similar to those in Tables 5.5 and 5.6 with lower gaps for the instances with transportation capacity constraints when compared to the instances with budget capacities.

5.4.5 The General Two-Level Problem with Sales

We generated a new set of instances for the general two-level problem in an attempt to reproduce the instances in Park [36]. (This was apparently not fully successful as the joint stock constraints turned out to be inactive in the instances generated). However the data below is generated as described in his paper. Demands are randomly generated in the interval $d_t^{ic} \in [50, 70]$ and upper bounds on sales are given by $U_t^{ic} = 0.5 d_t^{ic}$. Vehicles capacity K_1 is equal to 100. Unit processing time is $a^{ip} \in [5, 10]$ while setup time is calculated as $b^{ip} = 100 a^{ip}$. The joint production capacity (bucket constraint) is set to

$$LK^p = \max_{t \in \{1,\ldots,NT\}} (\sum_{i=1}^{NI} ((\sum_{c=1}^{NC} d_t^{ic}/NP) a^{ip} + b^{ip}))/0.8.$$

Storage capacity for the retailers is set to $SK^c = \max_{t\in\{1,\ldots,NT\}}(\sum_{i=1}^{NI} d_t^{ic})/1.0$. Unit processing costs are randomly generated $p^{0ip} \in [15, 20]$ and the unit selling prices are calculated as $l^{ic} = [2.5, 3.5] \times \sum_{p=1}^{P} p^{0ip}/NP$. The production setup costs are set to $q^{0ip} = 300 p^{0ip}$, the producer holding costs are $h^{0ip} = 0.2 p^{0ip}$ and the client holding costs are $h^{1ic} = 0.1 l^{ic}$. There is also a stockout cost of $e^{ic} = 0.15 l^{ic}$ per unit for the difference between the maximum possible amount $d_t^{ic} + U_t^{ic}$ deliverable to client c and the amount actually delivered $d_t^{ic} + v_t^{ic}$. Fixed transportation cost is $q^{1pc} = 500$. The production sites and client areas are generated randomly in a 100×100 square, and the transportation costs are in the interval $[1, 3]$ where the smallest distance is normalized to 1 and the biggest is normalized to 3.

For two specifications of the dimensions (corresponding to two of the largest instances in [36]) $NI = 5$, $NP = 5$, $NC = 40$, $NT = 12$, denoted Park A, and $NI = 5$, $NP = 5$, $NC = 70$, $NT = 10$, denoted Park B, we have randomly generated five instances each.

The objective function used by Park after translation to our variables is:

$$\max \sum_{i,c,t} l^{ic}(v_t^{ic} + d_t^{ic}) - (\sum_{i,p,t} p^{0ip} x_t^{0ip} + \sum_{i,p,t} q^{0ip} y_t^{ip} + \sum_{i,p,t} h^{0ip} s_t^{0ip} + \sum_{i,c,t} h^{1ic} \sigma_t^{ic} + \sum_{p,c,t} q^{1pc} Y_t^{pc} + \sum_{i,p,c,t} p^{1ipc} x_t^{1i,p,c}) - \sum_{i,c,t} e^{ic}(U_t^{i,c} - v_t^{i,c}).$$

In Tables 5.9 and 5.10 we show results for versions of our two groups of instances with different capacity constraints. The first column gives the instance name, columns 2-6 give the results of the hybrid heuristics H2 and H3 running for 300 seconds in the branch-and-bound phase with $z_{LP(FL)}$ giving the facility location lower bound obtained using the barrier algorithm, *time* denotes the time in seconds to solve $LP(FL)$ plus 300 seconds to run (STD) with the variables fixed, $\bar{z}^{H2}$ and $\bar{z}^{H3}$ giving heuristics H2 and H3 solution values respectively and Γ_{H3} the corresponding duality gap of heuristic H3. Again H3 was chosen for comparison since it performed better in most of the cases.

Columns 7-9 give the results of the MIP solver using the standard formulation with z_{XLP} giving the bound obtained by the linear relaxation followed by the solver automatic cutting plane at the root node, $\bar{z}^{STD}$ the best solution found and Γ_{STD} the duality gap at the end of the time limit (3600 seconds). In Column 10, *imptime* gives the time it took the solver to find a solution better than that found by heuristic H3. We omit in the table the results using the facility location reformulation, since they were worse than those of both the heuristic and the standard formulation for all the instances.

Table 5.9: Park instances (original capacities)

	(HEURISTICS)					*(STD)*			
Inst	$z_{LP(FL)}$	*time*	$\bar{z}^{H2}$	$\bar{z}^{H3}$	Γ_{H3}	z_{XLP}	$\bar{z}^{STD}$	Γ_{STD}	*imptime*
ParkA1	7737929	564	7558187	7563241	2.3	7761714	7562682	2.5	–
ParkA2	7084497	601	6866757	6878938	3.0	7105135	6874702	3.2	–
ParkA3	7098082	624	6864604	6899819	2.9	7105675	6855298	3.5	–
ParkA4	7301276	569	7010793	7048995	3.6	7322657	7039042	3.8	–
ParkA5	7537357	582	7277784	7281459	3.5	7561075	7295911	3.5	1138
ParkB1	10919963	668	10481200	10604564	3.0	10965434	10582904	3.5	–
ParkB2	11064751	673	10729783	10667690	3.7	11066235	10713097	3.1	1412
ParkB3	11274852	980	10859276	10901026	3.4	11253485	10890575	3.2	–
ParkB4	10489566	643	9956718	10081939	4.0	10542077	10100285	4.3	2280
ParkB5	11474370	655	11137665	11169294	2.7	11455687	11149934	2.7	–

In Table 5.10 we take the same ten instances, but tighten the capacity constraints by reducing K_1 by 50% and LK^p by 30%. Since heuristic H3 performed better than H2 in this group of instances as well, we only show the results for H3.

Table 5.10: Park instances (reduced capacities)

	(HEURISTIC)				*(STD)*			
Inst	$z_{LP(FL)}$	*time*	$\bar{z}^{H3}$	Γ_{H3}	z_{XLP}	$\bar{z}^{STD}$	Γ_{STD}	*imptime*
ParkA1B	5050866	555	4731978	6.7	5044377	4800571	4.8	612
ParkA2B	4593853	524	4296798	6.9	4527774	4332555	4.3	2426
ParkA3B	4773648	571	4500360	6.1	4715142	4489886	4.8	–
ParkA4B	4325888	552	3899064	10.9	4305012	3943899	8.9	1310
ParkA5B	4668550	508	4352007	7.3	4642254	4393352	5.2	1127
ParkB1B	6210881	630	5685256	9.2	6208153	5776103	7.1	943
ParkB2B	6366365	575	5823602	9.3	6303221	5866998	7.1	3016
ParkB3B	6344660	583	5879237	7.9	6326506	5896389	6.9	2429
ParkB4B	6664446	592	6260035	6.5	6612825	6230450	5.9	–
ParkB5B	6501884	544	5959616	9.1	6390830	6035755	5.8	3229

Here we observe that in contrast to the results for the first data set, the bound obtained at the top node z_{XLP} is often as good as the bound obtained using the extended formulation $z_{LP(FL)}$. However it appears from some initial testing that the solution of the extended formulation is a better candidate for use in the fixing heuristic than the top node solution after cuts. This is probably because there are many less variables taking value zero after addition of the cuts. Here the heuristic solution does not systematically dominate the solution found by the MIP solver. However the column *imptime* indicates that on the whole the heuristic is faster, and on this set of instances gives solutions with a duality gap of maximum 11% in approximately ten minutes.

5.5 Concluding Remarks

In this chapter, we studied a general production and transportation problem with multiple production sites and multiple clients. We identified some relaxations of the problem for which "strong" reformulations could be obtained. We showed that some medium size uncapacitated instances of a basic problem can often be solved to optimality by a commercial solver when using the multi-commodity reformulation. In addition, we proposed a hybrid heuristic approach that combines different formulations. The approach adopted leads to a simple class of heuristics that give solutions with an *a posteriori* quality guarantee. This hybrid heuristic appears to be applicable to several other problems in which one or more extended formulations are available. One such problem is the parallel machine lot-sizing problem with start-ups, see Gicquel et al. [22].

Chapter 6

Production and Transportation in a Commit-to-delivery Business Mode

In this chapter we consider a very special type of production and transportation optimization problem. There are different orders, each of them composed of a certain quantity of a single type of product, that have to be delivered by their due dates. The transportation to the clients is made by a third company. The goal is to determine an optimal production policy optimizing the total shipping cost.

Stecke and Zhao [47] have considered this problem. In their case they defined it as a problem of optimizing production and transportation integration in a make-to-order manufacturing company with a commit-to-delivery business mode in which the transportation is performed by a third-part logistics company. This type of problem appears in the context of companies such as Dell working in a make-to-order environment, that commit-to-ship or commit-to-deliver by a certain date when an order is placed. They consider two cases, one in which parts of an order can be shipped separately (divisible problem) and one in which an order cannot be divided in different shippings (indivisible problem). They derived integer programming formulations for both cases. When all orders are divisible, the linear programming relaxation of their formulation solves the problem. When all orders are indivisible, their formulation is capable of solving very small instances, and therefore they developed a simple heuristic that gives good solutions for the instances tested.

We develop an integer programming formulation for the indivisible problem with the property that its linear programming relaxation is at least as strong as that provided by the value of the divisible problem. We show that these formulations allow one to solve large instances, including instances of the type tackled by Stecke and Zhao, to optimality within a few seconds.

The remainder of this chapter is organized as follows. In Section 6.1 we formally define the problem. In Section 6.2, we derive formulations for the divisible order (DO) and indivisible order (IO) problems. After considering first the simple DO problem, we present an initial mixed integer programming formulation of IO and then derive valid inequalities and a relaxation that solves problem IO. In addition we analyze the strength of its linear programming relaxation, and suggest some modeling refinements. In Section 6.3 we present some computational results. Based on the results observed, we also propose a simple algorithm with a local search procedure. In Section 6.4 we end up with some final considerations .

6.1 Problem Definition

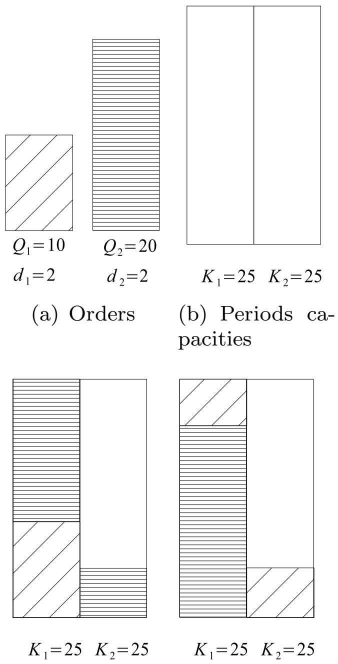
A company produces goods following a make-to-order policy, which means they manufacture products only when they are demanded by specific customers. The company operates in a commit-to-delivery business mode, i.e., it delivers orders to customers on or before committed delivery dates. There is a capacity on the total daily production. Different customers may place orders of varying sizes. There is a unit shipping cost which dependends on the time distance to the delivery date. The delivery of the orders to the clients is made by a third-part logistics company which has different shipping modes. The transportation cost increases the closer one is to the due date since a faster service has to be used.

The problem can be formally described as follows. There are NO orders such that each order i is composed of Q^i units all of which must be produced by the due date d^i. Total production is limited to a capacity of K_t units in period t. There is a unit shipping cost f_t^i which is a nondecreasing function of t. The objective is to produce all the orders while satisfying the production capacity constraints so as to minimize the total shipping costs. Summarizing, the parameters of the problem are

- NO: number of orders,
- NT: number of periods,
- Q^i: size of order i,
- K_t: production capacity in period t and

- f_t^i: unit shipping cost for order i in period t.

Orders can be of two types: *divisible* and *indivisible*. **Divisible orders** occur when partial delivery is allowed which means that orders can be delivered in different shipments, so that part of the order can be shipped as soon as it is available. **Indivisible orders** occur when partial delivery is not allowed, implying that orders can be delivered only when the order is complete.



(a) Orders (b) Periods capacities

(c) Divisible orders optimal solution (d) Indivisible orders optimal solution

Figure 6.1: Example of divisible and indivisible orders

In Figure 6.1 we illustrate a two-period problem with two orders, order 1 with size $Q_1 = 10$ and due date $d_1 = 2$ and order 2 with size $Q_2 = 20$ and due date $d_2 = 2$. There is a constant capacity $K_t = 25$ units per period. The unit shipping costs are $f_1^1 = f_1^2 = 1$, $f_2^1 = 3$ and $f_2^2 = 4$. For the divisible orders problem it is optimal to produce orders 1 and 2 in this order. Then the totality of order 1 and 15 units of order 2 can be shipped in period 1 and the remaining 5 units of order 2 can only be shipped in period 2, giving a total cost of 40 (if the order were the opposite the cost would be 45). For the indivisible orders problem, it is not possible to ship both orders at the same time period.

The optimal solution is then to produce orders 2 and 1 in this order. Order 2 can be shipped in period 1 but order 1 can only be finished in period 2 and therefore can only be shipped in that period, giving a total cost of 60 (in the opposite order we would have a cost of 70).

6.2 Mixed Integer Programming Formulations

Here we present and discuss mixed integer programming formulations for the divisible order and the indivisible order problems, and then establish a result on the linear programming relaxations of the proposed MIP formulations for the indivisible order problem.

6.2.1 The Divisible Order Problem

We first present the basic linear programming formulation for the divisible order programming, given by Stecke and Zhao [47], and then we indicate some of its special features.

We define the variables that will be used for both the divisible order (DO) problem:

- x_t^i: amount of order i produced in period t, defined for $i = 1, \ldots, NO$ and $t = 1, \ldots, d^i$.

For simplicity of notation we assume that x_t^i exist with $x_t^i = 0$ whenever $d_t^i < t \leq NT$.

Stecke and Zhao's basic formulation for the DO problem is the transportation problem:

$$(D)\ \ z_D = \min \textstyle\sum_{i=1}^{NO} \sum_{t=1}^{d^i} f_t^i x_t^i$$

$$\textstyle\sum_{t=1}^{d^i} x_t^i = Q^i \quad \forall\ i, \tag{6.1}$$

$$\textstyle\sum_{i=1}^{NO} x_t^i \leq K_t \quad \forall\ t, \tag{6.2}$$

$$x_t^i \in \mathbb{R}_+^1 \quad \forall i, t,$$

where the constraints (6.1) determine the size of each order and constraints (6.2) model the production capacity constraint.

Stecke and Zhao rightly made the following observation.

Observation 6.1. *Because the shipping costs f_t^i are non-decreasing in t, one produces as early as possible.*

They also stated the following observation which is easy to see.

Proposition 6.1. *(Stecke and Zhao) When the transportation costs are independent of i, a production schedule in which orders are sorted according to earlier due date first and processed non-preemptively and continuously without idle time is optimal for DO.*

We now make some basic observations. Our first observation is related to the feasibility of D. It states that the total cumulative production until a certain period t must be enough to produce all the orders with due date no later than t and that the cumulative capacity until t is enough to produce all these orders.

Observation 6.2. *(i) If x is feasible in D, then*

$$\sum_{i:d^i \leq t} Q^i \leq \sum_{i=1}^{NO} \sum_{u=1}^{t} x_u^i \leq \sum_{u=1}^{t} K_u \quad \forall\ t.$$

(ii) Problem D is feasible if and only if

$$\sum_{i:d^i \leq t} Q^i \leq \sum_{u=1}^{t} K_u \quad \forall\ t.$$

We now give a relaxation that solves problem D. This result will be used later when solving the problem with indivisible orders.

Proposition 6.2. *The relaxation of problem D:*

$$\begin{aligned}
(RD) \qquad & \min \textstyle\sum_{i=1}^{NO} \sum_{t=1}^{d^i} f_t^i x_t^i \\
& \textstyle\sum_{t=1}^{d^i} x_t^i = Q^i \quad \forall\ i, && (6.3)\\
& \textstyle\sum_{i=1}^{NO} \sum_{u=1}^{t} x_u^i \leq \sum_{u=1}^{t} K_u \quad \forall\ t, && (6.4)\\
& x_t^i \in \mathbb{R}_+^1 \quad \forall\ i, t,
\end{aligned}$$

solves D provided $f_t^i \leq f_{t+1}^i$ for all i, t.

Proof. We have to show that constraints (6.2) are satisfied. Consider problem D and let u be the first period in which the total capacity exceeds the size of the orders, i.e., $\sum_{k=1}^{u} K_k > \sum_i Q^i$. We show that $\sum_{i=1}^{NO} \sum_{k=1}^{j} x_k^i = \sum_{k=1}^{j} K_k$ for every $j \leq u$ and that $\sum_{i=1}^{NO} \sum_{k=1}^{u} x_k^i < \sum_{k=1}^{u} K_k$. According to Observation 6.1, production occurs at full capacity for every period $j \leq u-1$, implying $\sum_{i=1}^{NO} \sum_{k=1}^{j} x_k^i = \sum_{k=1}^{j} K_k$ and consequently $\sum_{i=1}^{NO} x_j^i = K_j$. The remaining $\sum_i Q^i - \sum_{k=1}^{u-1} K_k$ is produced at period u implying $\sum_{i=1}^{NO} x_u^i < K_u$. Observe that $\sum_{i=1}^{NO} \sum_{k=u+1}^{NT} x_k^i = 0$. □

Proposition 6.3. *(RD) gives an integer solution whenever Q and K are integer vectors.*

Proof. We show that the solution is integer by showing that the matrix formed by the constraints of RD is totally unimodular. This is done by showing that the conditions in Observation 2.3 are satisfied. The columns are arranged in increasing fashion based on order indices and periods, as $(x_1^1, x_2^1, \ldots, x_{d^1}^1, \ldots, x_1^{NI}, \ldots, x_{d^{NI}}^{NI})$. The following matrix is an example with $NI = 3$ and $NT = 3$ in which $d^i = 3$ for every order i.

$$A = \begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ 1 & & & 1 & & & 1 & & \\ 1 & 1 & & 1 & 1 & & 1 & 1 & \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Let M_a be the lines corresponding to constraints (6.3) belonging to M and M_b be the lines corresponding to constraints (6.4) belonging to M. For every line $i \in M_a$, add i to M_1. This implies $0 \le |\sum_{\substack{i \in M_1 \\ i \in M_a}} a_{ij} - \sum_{\substack{i \in M_2 \\ i \in M_a}} a_{ij}| \le 1$. For the lines $i \in M_b$, starting from the latest (largest t) to the earliest (lowest t), add alternating between M_2 and M_1, starting with M_2 then M_1 and continue with this sequence. This implies that $-1 \le |\sum_{\substack{i \in M_1 \\ i \in M_b}} a_{ij} - \sum_{\substack{i \in M_2 \\ i \in M_b}} a_{ij}| \le 0$. We have $0 \le |\sum_{i \in M_1} a_{ij} - \sum_{i \in M_2} a_{ij}| \le 1$ and therefore A is totally unimodular.

□

Our next observation is based on the fact that in problem D $\sum_{i=1}^{NO} x_t^i \le K_t$, what implies $\sum_{i=1}^{NO} \sum_{k=1}^{t} x_k^i \le \sum_{k=1}^{t} K_k$.

Observation 6.3. *Let $\bar{x}$ be a feasible solution of RD. Then there exists a feasible solution $\tilde{x}$ of D such that*
(i) $\sum_{u=1}^{t} \tilde{x}_u^i \ge \sum_{u=1}^{t} \bar{x}_u^i \quad \forall i,\ t = 1, \ldots, d^i$.
(ii) $\sum_{i=1}^{NO} \sum_{t=1}^{d^i} f_t^i \tilde{x}_t^i \le \sum_{i=1}^{NO} \sum_{t=1}^{d^i} f_t^i \bar{x}_t^i$ *when* $f_t^i \le f_{t+1}^i \quad \forall\ i, t$.

6.2.2 The Indivisible Order Problem

We give two MIP formulations for the indivisible order (IO) problem.

Theorem 6.4. *(Stecke and Zhao) The indivisible problem is strongly NP-hard.*

Theorem 6.4 was proved by a reduction from the 3-Partition problem.

We can use the variables x_t^i as defined for the DO problem. In addition, we define the new variables

- Y_t^i: is equal to 1 if order i is completed and shipped in period t, defined for $i = 1, \ldots, NO$ and $t = 1, \ldots, d^i$.

As for the x_t^i variables, we assume that Y_t^i exist with $Y_t^i = 0$ whenever $d_t^i < t \leq NT$ for simplicity of notation.

$$
\begin{aligned}
(I) \qquad\qquad & z_I = \min \textstyle\sum_{i=1}^{NO} \sum_{t=1}^{d^i} f_t^i Q^i Y_t^i \\
& \textstyle\sum_{t=1}^{d^i} x_t^i = Q^i \quad \forall\, i, && (6.5)\\
& \textstyle\sum_{i=1}^{NO} x_t^i \leq K \quad \forall\, t, \\
& \textstyle\sum_{t=1}^{d^i} Y_t^i = 1 \quad \forall\, i, && (6.6)\\
& \textstyle\sum_{t=u}^{d^i} x_t^i \leq Q^i \sum_{t=u}^{d^i} Y_t^i \quad \forall\, i,\ u = 1, \ldots, d^i, && (6.7)\\
& x_t^i \in \mathbb{R}_+^1,\ Y_t^i \in \{0,1\} \quad \forall\, i, t,
\end{aligned}
$$

where the additional equations (6.6) ensure that each order is shipped on or before its due date, and inequality (6.7) states that if the total amount of order i produced in the interval $[u, d^i]$ is positive, then necessarily this amount is bounded by Q^i and the order must be completed during the interval $[u, d^i]$ – in other words $x_t^i > 0$ implies $\sum_{u=t}^{d^i} Y_u^i = 1$, so one cannot ship an order until its production is completed.

Now we propose a valid inequality for the problem and develop a relaxation of problem I using this valid inequality.

Proposition 6.5. *The inequality*

$$\sum_{i=1}^{NO} \sum_{u=1}^{t} Q^i Y_u^i \leq \sum_{u=1}^{t} K_u \quad \forall\, t$$

is valid for I.

Proof:

$$
\begin{aligned}
\sum_{i=1}^{NO} Q^i \sum_{u=1}^{t} Y_u^i &= \sum_{i=1}^{NO} Q^i (1 - \sum_{u=t+1}^{d^i} Y_u^i) \\
&= (\sum_{i=1}^{NO} Q^i - \sum_{i=1}^{NO} Q^i \sum_{u=t+1}^{d^i} Y_u^i) \\
&\leq \sum_{i=1}^{NO} (Q^i - \sum_{u=t+1}^{d^i} x_u^i) \\
&= \sum_{i=1}^{NO} \sum_{u=1}^{t} x_u^i \\
&\leq \sum_{u=1}^{t} K_u.
\end{aligned}
$$

where the first inequality uses (6.7) and the second follows from Observation 6.2. □

Proposition 6.6. *The relaxation*

$$(RI) \qquad z_{RI} = \min \textstyle\sum_{i=1}^{NO} \sum_{t=1}^{d^i} f_t Q^i Y_t^i \tag{6.8}$$

$$\textstyle\sum_{t=1}^{d^i} Y_t^i = 1 \quad \forall\ i, \tag{6.9}$$

$$\textstyle\sum_{i=1}^{NO} \sum_{u=1}^{t} Q^i Y_u^i \leq \sum_{u=1}^{t} K_u \quad \forall\ t, \tag{6.10}$$

$$Y_t^i \in \{0,1\} \quad \forall\ i,\ t = 1, \ldots, d^i \tag{6.11}$$

solves the problem I.

Proof: Suppose that $\bar{Y}$ is feasible in RI. We have $\sum_{t=1}^{d^i} Y_t^i = 1$, so $\bar{Y}$ satisfies constraints (6.6). Define $\bar{x}$ by setting $\bar{x}_t^i = Q^i \bar{Y}_t^i$ for all i, t.

First we show that the solution $(\bar{x})$ is feasible in RD. Constraints (6.9) imply $\sum_{t=1}^{d^i} \bar{x}_t^i = \sum_{t=1}^{d^i} Q^i \bar{Y}_t^i = Q^i$, therefore $\bar{x}$ satisfies (6.3). Constraints (6.10) imply that $\sum_{i=1}^{NO} \sum_{u=1}^{t} \bar{x}_u^i = \sum_{i=1}^{NO} \sum_{u=1}^{t} Q^i \bar{Y}_u^i \leq \sum_{u=1}^{t} K_u$, so $\bar{x}$ satisfies (6.4). Thus $(\bar{x})$ is feasible in RD.

We now want to show that there exists $\tilde{x}$ such that the solution $(\tilde{x}, \bar{Y})$ is feasible in I. From Observation 6.3, there exists a vector $\tilde{x}$ that is feasible in D with $\sum_{u=1}^{t} \tilde{x}_u^i \geq \sum_{u=1}^{t} \bar{x}_u^i$. Thus $\tilde{x}$ satisfies (6.1), i.e. $\sum_{t=1}^{d^i} \tilde{x}_t^i = Q^i$, and (6.2), i.e. $\sum_{i=1}^{NO} \tilde{x}_t^i \leq K_t$. As $\sum_{u=t}^{d^i} \tilde{x}_u^i \leq \sum_{u=t}^{d^i} \bar{x}_u^i = Q^i \sum_{u=t}^{d^i} \bar{Y}_u^i$, $(\tilde{x}, \bar{Y})$ satisfies (6.7). So $(\tilde{x}, \bar{Y})$ is feasible in I. As the objective functions have the same value, the claim follows. □

6.2.3 Tightness of the Bounds

Our initial motivation was to derive a MIP formulation for the IO problem giving bounds as strong as those provided by the DO problem. Here we show that the optimal value z_D of the transportation problem D and the values z_I^{LP} and z_{RI}^{LP} of the linear programming relaxations of I and RI, respectively, are exactly the same.

Proposition 6.7. $z_D = z_I^{LP} = z_{RI}^{LP}$.

Proof: We show first that $z_D \leq z_I^{LP}$. Let (x, Y) be optimal in $LP(I)$. Clearly x is feasible in D. We show that for each order i, $\sum_{t=1}^{d^i} f_t^i x_t^i \leq \sum_{t=1}^{d^i} Q^i f_t Y_t^i$.

Specifically

$$\begin{aligned}
\sum_{t=1}^{d^i} f_t^i x_t^i &= \sum_{t=1}^{d^i} (f_t^i - f_{t-1}^i) \sum_{u=t}^{d^i} x_u^i \\
&\leq Q^i \sum_{t=1}^{d^i} (f_t^i - f_{t-1}^i) \sum_{u=t}^{d^i} Y_u^i \\
&= \sum_{t=1}^{d^i} Q^i f_t^i Y_t^i
\end{aligned}$$

where the inequality follows using (6.7) and the fact that $f_t^i - f_{t-1}^i \geq 0$ for all t as f_t^i is nondecreasing in t. It follows that $z_D \leq \sum_{i=1}^{NO} \sum_{t=1}^{d^i} f_t x_t^i \leq \sum_{i=1}^{NO} \sum_{t=1}^{d^i} Q^i f_t Y_t^i = z_{RI}^{LP}$.

Next we show that $z_I^{LP} \leq z_{RI}^{LP}$. Let $\bar{Y}$ be an optimal solution of $LP(RI)$. Set $\bar{x}_t^i = Q^i \bar{Y}_t^i$ for all i, t. From (6.9), we have that $\sum_{t=1}^{d_i} \bar{x}_t^i / Q^i = 1$ and thus (6.1) is satisfied. From (6.10), we have that $\sum_{i=1}^{NO} \sum_{u=1}^{t} \bar{x}_u^i \leq \sum_{u=1}^{t} K_u$. Again by Observation 6.3, this implies the existence of $\tilde{x}$ such that $(\tilde{x}, \bar{Y})$ is feasible in $LP(I)$, and the claim follows.

Finally we show that $z_D \geq z_{RI}^{LP}$. Let $\bar{x}$ be an optimal solution of D. Set $\bar{Y}_t^i = \bar{x}_t^i / Q^i$ for all i, t. From (6.1), we have that $\sum_{t=1}^{d^i} \bar{x}_t^i = \sum_{t=1}^{d^i} Q^i \bar{Y}_t^i = Q^i$ and thus dividing by Q^i, (6.9) holds for $\bar{Y}$. Also from (6.2), $\sum_{i=1}^{NO} \sum_{u=1}^{t} Q^i \bar{Y}_u^i \leq \sum_{u=1}^{t} K_u$. Thus $\bar{Y}$ is feasible in $LP(RI)$ with value z_D, and so the optimal value $z_{RI}^{LP} \leq z_D$. □

6.2.4 Change of Variable

Mixed integer programming solvers include their own cutting plane routines. Some of these routines depend on recognizing certain structures, such as knapsack constraints. Because of this, it is possible that, even though two formulations of an MIP have linear programming relaxations taking the same value, an MIP solver performs better on one formulation than on the other. Here we make a change of variable with the idea that certain simple knapsack constraints may be more evident to the solver in the new formulation. We define the variables

- Z_t^i: equal to 1 if order i is shipped within the interval $[1, t]$.

Observe that $Z^i_t = \sum_{u=1}^t Y^i_u$. The formulation RI now becomes

$$(RI')\ z_I = \min \sum_{i=1}^{NO} \sum_{t=1}^{d^i} Q^i (f^i_t - f^i_{t-1})(1 - Z^i_{t-1})$$

$$\sum_{i=1}^{NO} Q^i Z^i_t \le \sum_{u=1}^t K_u \quad \forall\, t, \tag{6.12}$$

$$Z^i_t \le Z^i_{t+1} \quad \forall\, i,\ t = 1, \ldots, d^i - 1 \tag{6.13}$$

$$Z^i_1 \ge 0, Z^i_{d^i} = 1 \quad \forall\, i, \tag{6.14}$$

$$Z^i_t \in \{0,1\} \quad \forall\, i,\ t = 1, \ldots, d^i, \tag{6.15}$$

because $\sum_{t=1}^{d^i} f^i_t Y^i_t = \sum_{t=1}^{d^i} (f^i_t - f^i_{t-1}) \sum_{u=t}^{d^i} Y^i_u = \sum_{t=1}^{d^i} (f^i_t - f^i_{t-1})(1 - Z^i_{t-1})$.

As indicated above, a potential advantage of this formulation is that the constraints (6.12) are simple 0-1 knapsack constraints, and the automatic cut generation algorithms embedded in the MIP solvers are more likely to generate cuts off such constraints than off the constraints (6.10).

6.2.5 An Extension with Production and Storage Costs

Here we consider one possible extension to the indivisible order problem. We assume that there are unit production costs p^i_t and unit storage costs h^i_t. Letting s^i_t denote the stock of order i at the end of period t for $1 \le t \le d^i - 1$, the new objective function becomes

$$\min \sum_{i=1}^{NO} [\sum_{t=1}^{d^i} p^i_t x^i_t + \sum_{t=1}^{d^i-1} h^i_t s^i_t + \sum_{t=1}^{d^i} f^i_t Q^i Y^i_t].$$

Observing that $s^i_t = \sum_{u=1}^t x^i_u - Q^i Z^i_t$ is nothing but the slack variable in (6.7), we can eliminate s^i_t by substitution, and the resulting MIP in the x, Z variables is

$$(RIPS)$$

$$\min \sum_{i=1}^{NO} [\sum_{t=1}^{d^i} (p^i_t + \sum_{u=t}^{d^i-1} h^i_u) x^i_t + \sum_{t=1}^{d^i} Q^i (f^i_t - f^i_{t-1})(1 - Z^i_{t-1}) + p^i Q^i] \tag{6.16}$$

$$\sum_{t=1}^{d^i} x^i_t = Q^i \quad \forall\, i, \tag{6.17}$$

$$\sum_{i=1}^{NO} x^i_t \le K_t \quad \forall\, t, \tag{6.18}$$

$$\sum_{i=1}^{NO} Q^i Z^i_t \le \sum_{u=1}^t K_u \quad \forall\, t, \tag{6.19}$$

$$\sum_{u=1}^t x^i_u \ge Q^i Z^i_t \quad \forall\, i, t = 1, \ldots, d^i - 1, \tag{6.20}$$

$$Z^i_t \le Z^i_{t+1} \quad \forall\, i,\ t = 1, \ldots, d^i - 1, \tag{6.21}$$

$$Z^i_{d^i} = 1 \quad \forall\, i, \tag{6.22}$$

$$Z^i_t \in \{0,1\} \quad \forall\, i,\ t = 1, \ldots, d^i. \tag{6.23}$$

$$\tag{6.24}$$

The objective function minimizes the total cost. Constraints (6.17) guarantee the orders are totally produced. Constraints (6.18) ensure the capacities are satisfied. Constraints (6.19) state that all the orders shipped until period t do not exceed the total capacity until period t. Constraints (6.20) guarantee that an order is only shipped once it is complete. Constraints (6.21) set the Z^i_{t+1} variables to 1 in case shipping of order i occurred before period $t+1$. Constraints (6.22) state that the orders are shipped at most at their due dates. Constraints (6.23) are bounds on the shipping variables.

In practice production and storage costs are typically constant over time in which case the objective function simplifies to

$$\min \sum_{i=1}^{NO} [\sum_{t=1}^{d^i} (d^i - t + 1) h^i x^i_t + \sum_{t=1}^{d^i} Q^i (f^i_t - f^i_{t-1})(1 - Z^i_{t-1})]. \tag{6.25}$$

Now clearly there is a conflict between the normalized production costs that are decreasing with time and the shipping costs that are increasing. Therefore it is no longer possible to argue that one will produce as early as possible.

6.3 Computational Results

Having shown that our MIP formulations of the IO problem provide lower bounds of the quality desired a priori, we now test whether they are effective in generating provably optimal solutions in a reasonable amount of time.

6.3.1 The Indivisible Order Problem

To test our formulations for problem IO, we have generated four sets of instances as described in the next paragraphs. The first set is composed of 100 instances while the other three sets have ten instances each.

- *Class 1* instances were created with $\{100, 200, \ldots, 1000\}$ orders, five time periods, 70% of the orders with size equal to one, 25% of the orders with size two and the remaining 5% with the size randomly generated in the interval $[3, 10]$. For each of the different numbers of orders, ten instances were created.
- *Class 2* instances have 500 orders, eight time periods and the size of each order is randomly generated in the interval $[1, 60]$.
- *Class 3* is composed of instances with 500 orders, eight time periods and order sizes randomly generated in the interval $[21, 60]$.
- *Class 4* instances were specified with 250 orders, 15 time periods and, similar to Class 3 instances, the size of each order was randomly generated in the interval $[21, 60]$.

For all the instances, the constant capacity K is a value randomly selected in the range $[K_{min}, \lceil 1.02 \times K_{min} \rceil - 1]$, where the value $K_{min} = \text{argmax}_{t \in 1..NT} \left\lceil (\sum_{i:d(i) \le t} Q^i)/t \right\rceil$.

Class 1 is designed to replicate the classes of instances with small orders tested by Stecke and Zhao - as the capacities are not described in their work, they are generated to be random and relatively tight. Class 2 is designed to generate a class with larger orders. Classes 3 and 4 are an attempt to create more difficult instances having no very small orders. They way the instances are generated is summarized in Table 6.1.

Table 6.1: Characteristics of the instances

Type	NO	NT	Q^i
Class 1	100..1000	5	1 (70%), 2 (25%), $[3, 10]$ (5%)
Class 2	500	8	$[1, 60]$
Class 3	500	8	$[21, 60]$
Class 4	250	15	$[21, 60]$

The instances have been solved using the mixed integer programming solver Xpress-MP v2.2.0 on a computer with a 3.20GHz Intel Xeon processor and 4 Gb of RAM memory. A time limit of 300 seconds was imposed on the solver to find the optimal solution for each of the instances.

In Table 6.2 we present the results for the four classes using the Y-formulation RI and the Z-formulation RI' running Xpress-MP in default mode. For each we give the average time in seconds and the average number of nodes in the tree for the solved instances, as well as the number of instances unsolved within the time limit. For the first class we only present the results for $NO = 1000$ as results for the smaller instances are similar.

Table 6.2: Results using XPress-MP in default mode

	RI			RI'		
Type	Avg. time(s)	Avg. nodes	not solved	Avg. time(s)	Avg. nodes	not solved
Class 1	0.2	1.0	0	0.1	1.0	0
Class 2	8.6	11002.6	0	0.2	38.8	0
Class 3*	14.9	12106.8	2	0.6	538.3	2
Class 4*	14.4	19463.0	8	0.3	33.0	6

Examination of Table 6.2 and the more detailed results led us to the following observations:

(i) Instances of classes 1 and 2 are easily solved in less than 1 second and typically without enumeration.

(ii) For classes 3 and 4 on rare occasions the values obtained at the top node after addition of cuts differ slightly between the Y and Z formulations. In all the cases in which there is a difference, the Z gives a better bound.

(iii) For all the solved instances of classes 3 and 4 the value obtained at the top node after addition of cuts using the Z formulation is the optimal value of the IO instance.

(iv) For the unsolved instances, the duality gap is tiny, never more that 0.1% after 52 seconds and never more than 0.05% at the end of the time limit.

Point (iii) along with the significant number of nodes required for certain instances indicates that the face of optimal solutions is apparently very large. To try to reduce the number of nodes, we decided to increase the number of rounds of Gomory mixed integer cuts added. Point (iv) indicates that one quickly has a very good solution, and suggests its use as the starting point of a local search heuristic to find the optimal integer solution.

We propose a to use a simple heuristic, which is a variant of the relaxation induced neighborhood search heuristic (RINS) [8]. Essentially whereas RINS optimizes over the variables that take different values in the linear programming solution $\bar{Z}$ and the best feasible solution $\hat{Z}$, we increase the interval of optimization by one period in both directions.

Algorithm with Local Search
Step 1: Solve the linear program at the top node with 40 rounds of Gomory cuts to obtain the solution $\bar{Z}$.
Step 2: Run branch-and-cut to obtain a best feasible solution $\hat{Z}$ obtained after K_1 seconds.
Step 3: For each order i for which $\bar{Z}^i \neq \hat{Z}^i$, let $\sigma^i = \min\{t : \bar{Z}^i_t \neq \hat{Z}^i_t\}$ and $\rho^i = \max\{t : \bar{Z}^i_t \neq \hat{Z}^i_t\}$. Then allow $Z^i_t \in \{0,1\}$ for t in the interval $[\sigma^i - 1, \rho^i + 1]$ and fix $Z^i_t = 0$ for $t < \sigma - 1$ and $Z^i_t = 1$ for $t > \rho + 1$.
For each order i for which $\bar{Z}^i = \hat{Z}^i$, let $\tau^i = \min\{t : \bar{Z}^i_t = 1\}$. Then allow $Z^i_t \in \{0,1\}$ for t in the interval $[\tau^i - 1, \tau^i + 1]$ and fix $Z^i_t = 0$ for $t < \tau - 1$ and $Z^i_t = 1$ for $t > \tau + 1$.
Run the optimizer on this modified problem. If a feasible solution is found with the same value as that of the LP solution $\bar{Z}$, it is optimal. Otherwise stop after $K_2 = 300 - K_1$ seconds with the best feasible solution found.
Step 4: Return to steps 1 and 2 and solve the original complete formulation to optimality using the best incumbent value as a cutoff and with the limit of K_1 seconds removed.

In Table 6.3 we only consider the instances of classes 3 and 4 and use the more effective Z-variable formulation RI'. In columns 2-4 we present results with the Xpress-MP default except that there 40 rounds of Gomory cuts are added and the optimizer's heuristic is set to be more active than in default mode. We see that the cut strategy is effective. Six instances of Class 3 are solved in less than one second and one instance remain unsolved after 300 seconds. Five of the Class 4 instances are solved within less than 10 seconds, but three instances remain unsolved after 300 seconds. In columns 5-7 we present results using the complete algorithm described above with both Gomory cuts and the primal heuristic, and settings $K_1 = 10$ and $K_2 = 290$. Now all the instances are solved to optimality well within the time limit.

Table 6.3: Results with (i) Gomory cuts, (ii) the complete algorithm

	RI' (i)			RI' (ii)		
Type	Avg. time(s)	Avg. nodes	not solved	Avg. time(s)	Avg. nodes	not solved
Class 3	11.9	14663.9	1	6.4	8230.9	0
Class 4	45.5	65500.4	3	10.6	15483.5	0

6.3.2 Extension with Production and Storage Costs

To analyze the extension of the problem with production and storage costs and to test the formulation $RIPS$, four new sets of instances were created with the specifications in Table 6.1 and with constant production and unit storage costs. We use the $RIPS$ formulation with the simplified objective function (6.25). We run the optimizer for 300 seconds using the Xpress-MP defaults except that again up to 40 rounds of Gomory cuts are allowed. The results are summarized in Table 6.4.

In columns 2-5 we have the average time, the average number of nodes, the number of unsolved instances and the average duality gap for the unsolved instances when using only $RIPS$.

Classes 1 and 2 are still relatively easy, but now the Classes 3 and 4 instances are considerably more difficult to solve to optimality. However after just 10 seconds the duality gaps are very small.

Table 6.4: Results for $RIPS$

Type	Avg. time(s)	Avg. nodes	not solved	Gap	Gap_{10s}
Class 1	0.6	1.0	0	0.000	0.000
Class 2	14.1	3452.4	0	0.000	0.001
Class 3*	49.9	11698.7	3	0.004	0.025
Class 4*	163.4	21799.0	9	0.049	0.488

6.4 Concluding Remarks

In this chapter we have studied a production and transportation integration problem with a commit-to-delivery business mode in which the transportation is performed by a third-part logistics company.

We have shown that there is a mixed integer programming formulation for the indivisible order problem that appears to be effective on the instances studied by Stecke and Zhao [47] and also on certain apparently more difficult instances. As shipping of complete orders is a natural requirement in certain models, this formulation will hopefully also be useful in such models. A first such model is the problem RIPS including not just production capacity constraints but also production and/or storage costs that we have examined briefly.

The algorithm with a local search procedure was effective to solve instances that could not be solved by using the formulations within the time limit. One could try to develop a different local search procedure. Some preliminary results show that the results can be improved in some instances when different neighborhoods are used in the variation of RINS used. Some variations of Local branching [18] could also be used.

A version of this chapter appeared in Melo and Wolsey [32].

Chapter 7

Conclusions

Throughout this thesis we have studied certain joint production and transportation problems. Here we will highlight the contributions of this thesis and discuss possible extensions for future research.

In Chapter 3 we studied the uncapacitated two-level production-in-series lot-sizing problem (2L-S/LS-U). We provided a new dynamic programming algorithm with a worst case running time of $O(NT^2 \log NT)$ (which runs in $O(NT^2)$ under some assumptions on the cost) which is lower than that of the best known $O(NT^3)$ algorithm in the literature and presented a new compact extended formulation with $O(NT^3)$ variables and $O(NT^2)$ constraints. We also presented a $O(NT^4)$ dynamic programming algorithm and a compact extended formulation with $O(NT^4)$ variables and $O(NT^3)$ constraints for a generalization in which a limited amount can be produced for each period in addition to the demand in order to obtain some extra revenue. It would be interesting to know whether one can find a compact formulation with less than $O(NT^3)$ variables. Another question is to find out whether one can obtain extended formulations of reasonable size solving two-level problems involving constant capacities.

In Chapter 4 we analyzed the one-warehouse multi-retailer problem (OWMR). During the last years a considerable amount of effort has been put in obtaining approximation algorithms for the OWMR but recently some authors compared different formulations for the problem. The work in this chapter was in the direction of studying mixed integer programming approaches for tackling the problem. We compared the strength of the multi-commodity formulation with the so-called transportation formulation which was used in the comparisons in the literature. We also analyzed the projection of the multi-commodity formulation for the joint-replenishment problem (JRP) and for the 2L-S/LS-U, showing that it is composed of only simple dicut inequalities for the JRP which is not the case for the 2L-S/LS-U. In addition, we presented some valid inequalities for the 2L-S/LS-U that are not just simple (l, S) inequalities and used them to tackle the OWMR computationally in a space of variables intermediate between the original and the multi-commodity spaces.

As far as we know, there is no combinatorial algorithm known to separate the simple dicut inequalities. Therefore it would be interesting trying to develop an efficient algorithm to separate the simple dicut inequalities for the joint-replenishment problem. By observing the computational experiments, we can see that a more efficient separation algorithm could be achieved specially given the large number of inequalities that was generated using our cutting plane. One can see that the $G - (l, S)$ inequalities, which are a special case of the simple dicut inequalities, can still be generalized. It would be interesting if one could find a generalization of the $G - (l, S)$ inequalities that is as strong as the simple dicut inequalities for the OWMR problem and can be "easily" separated.

In Chapter 5 we considered a general two-level production-transportation problem with multiple production sites. We used reformulations for relaxations of the problem in order to get better bounds. Our computational experiments showed that medium size instances could be solved to optimality with a standard solver by using these reformulations. We provided a MIP heuristic framework combining different formulations that provides an a posteriori performance guarantee. In our experiments, solutions within 10% of optimality for were achieved for instances with limited transportation capacity and/or with additional sales but the gaps became larger (up to 40%) with big bucket production or aggregate storage capacity constraints. One comment about the computational results concerns how the solvers behave when using different instance generators and the difficulty of the test instances. Is the fact that the duality gaps provided by the heuristic are larger for the simpler instances without sales than for those with sales is due to the greater flexibility provided by sales? Some initial tests have shown that decreasing the selling price in the instances with sales leads to larger gaps, so this question among others is still open. From our results it seems that medium size uncapacitated instances can be solved to optimality by using a multi-commodity reformulation. For instances with vehicle capacities it seems possible to achieve reasonably good bounds by using our heuristic framework. For the harder general instances, more studies should be done in order to try to reduce the duality gaps obtained.

What can we do when faced with even larger instances so that the extended formulations available, e.g. the multi-commodity formulation, become too large? In certain cases, one possible way to apply this approach could be to find and make efficient use of approximate extended formulations, such as in Van Vyve and Wolsey [52]. Using Benders' algorithm, for instance, one can solve linear programs over the extended formulations to generate cutting planes in the original space of variables. However, at least for the general two-level problem, it appears that an enormous number of cutting planes is needed just to get back close to the multi-commodity LP bound. It would also be interesting to study special purpose heuristics that can provide primal feasible solutions of the same quality, or better, even without quality guarantees. Another possibility is to study other ways to use an MIP solver more effectively by analyzing more carefully the characteristics of the reformulations and trying to identify which algorithms would be more likely to perform better in different situations.

In Chapter 6 we investigated a production and transportation problem in a commit-to-delivery business mode. We developed MIP formulations that perform well in practice and a local search procedure that helped the solver to tackle some more challenging instances. Our approach could solve to optimality instances similar to the those that were treated heuristically in the paper in which the problem was introduced.

Some extensions to the problem could be tackled with our approach. The case in which shipping costs depend on customer locations can be treated directly by the proposed formulation, and the case involving a mixture of divisible and indivisible orders can easily be treated by combining the formulations for divisible and indivisible orders. Another variant with quantity discounts leading to shipping costs that are not nondecreasing is a more challenging problem, and requires further study.

Finally it remains the challenge of treating problems with three or more levels. As far as we know, there is no compact extended formulation of reasonable size for the uncapacitated production-in-series lot-sizing with a general number of levels. It would also be interesting to study effective computational methods for extensions of the general production-transportation problem with more than two levels.

Bibliography

[1] A. Aggarwal and J. Park. Improved algorithms for economic lot-size problems. *Operations Research*, 41(3):549–571, 1993.

[2] C. Archetti, L. Bertazzi, G. Paletta, and M. Grazia Speranza. Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, 38(12):1731 – 1746, 2011.

[3] E. Arkin, D. Joneja, and R. Roundy. Computational complexity of uncapacitated multi-echelon production planning problems. *Operations Research Letters*, 8:61–66, 1989.

[4] I. Barany, T.J. Van Roy, and L.A. Wolsey. Uncapacitated lot-sizing: The convex hull of solutions. *Mathematical Programming*, 22:32–43, 1984.

[5] J.F. Bard and N. Nananukul. A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research*, 37(12):2202 – 2217, 2010.

[6] P. Chandra. A dynamic distribution model with warehouse and customer replenishment requirements. *The Journal of the Operational Research Society*, 44(7):681–692, 1993.

[7] P. Chandra and M.L. Fisher. Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503 – 517, 1994.

[8] E. Danna, E. Rothberg, and C. Le Pape. Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming*, 102:71–90, 2005.

[9] R. de Matta and T. Miller. Production and inter-facility transportation scheduling for a process industry. *European Journal of Operational Research*, 158(1):72 – 88, 2004.

[10] M. Denizel, O. Solyali, and H. Süral. Tight formulations for the two and three level serial lot-sizing problems. In *International Workshop on Lot-Sizing, Gardanne, France, August*, 2010.

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

[12] S. Eksioglu, B. Eksioglu, and H. Romeijn. A lagrangian heuristic for integrated production and transportation planning problems in a dynamic, multi-item, two-layer supply chain. *IIE Transactions*, 39:191–201, 2007.

[13] S.D. Eksioglu, H.E. Romeijn, and P.M. Pardalos. Cross-facility management of production and transportation planning problem. *Computers & Operations Research*, 33(11):3231 – 3251, 2006.

[14] Eppen and R.K. Martin. Solving capacitated multi-item lot-sizing problems using variable definition. *Operations Research*, 35:832–848, 1987.

[15] A. Federgruen and M. Tzur. A simple forward algorithm to solve general dynamic lot sizing models with n periods in $0(n \log n)$ or $0(n)$ time. *Management Science*, 37(8):909–925, 1991.

[16] A. Federgruen and M. Tzur. The joint replenishment problem with time-varying costs and demands: Efficient, asymptotic and ϵ-optimal solutions. *Operations Research*, 42(6):1067–1086, 1994.

[17] A. Federgruen and M. Tzur. Time-partitioning heuristics : Application to one warehouse, multiitem, multiretailer lot-sizing problems. *Naval Research Logistics*, 46(5):463–486, 1999.

[18] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98:23–48, 2003.

[19] L.R. Ford and D.R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, 1962.

[20] F. Fumero and C. Vercellis. Synchronized development of production, inventory, and distribution schedules. *Transportation Science*, 33:330–340, 1999.

[21] A. Ghouila-Houri. Caractérisation des matrices totalement unimodulaires. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences*, 254:1192–1194, 1962.

[22] C. Gicquel, L. Wolsey, and M. Minoux. On discrete lot-sizing and scheduling on identical parallel machines. *Optimization Letters*, 2011. Online first.

[23] A.J. Hoffman and J.B. Kruskal. Integral boundary points of convex polyhedra. In *Linear Inequalities and Related Systems; Annals of Mathematical Study*, volume 38, pages 223–246, Princeton, 1956. Princeton University Press.

[24] J. Krarup and O. Bilde. Plant location, set covering and economic lot sizes: An $O(mn)$ algorithm for structured problems. In L. Collatz et al., editor, *Optimierung bei Graphentheoretischen und Ganzzahligen Probleme*, pages 155–180, Basel, 1977. Birkhauser Verlag.

[25] C-Y. Lee, S. Çetinkaya, and W. Jaruphongsa. A dynamic model for inventory lot sizing and outbound shipment scheduling at a third-party warehouse. *Operations Research*, 51(5):735–747, 2003.

[26] R. Levi, R. Roundy, D. Shmoys, and M. Sviridenko. A constant approximation algorithm for the one-warehouse multi-retailer problem. *Management Science*, 54:763–776, 2008.

[27] R. Levi, R.O. Roundy., and D.B. Shmoys. Primal-dual algorithms for deterministic inventory problems. *Mathematics of Operations Research*, 31:267–284, 2006.

[28] LISCOS. EU GROWTH Project LISCOS, Large scale integrated supply chain optimization software, 1999–2003.

[29] M. Loparic, Y. Pochet, and L.A. Wolsey. The uncapacitated lot-sizing problem with sales and safety stocks. *Mathematical Programming*, 89(3):487–504, 2001.

[30] S.F. Love. A facilities in series inventory model with nested schedules. *Management Science*, 18:327–338, 1972.

[31] R.K. Martin. Generating alternative mixed-integer programming models using variable redefinition. *Operations Research*, 35(6):820–831, 1987.

[32] R.A. Melo and L.A. Wolsey. Optimizing production and transportation in a commit-to-delivery business mode. *European Journal of Operational Research*, 203(3):614 – 618, 2010.

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

[34] M.C.V. Nascimento, M.G.C. Resende, and F.M.B. Toledo. GRASP heuristic with path-relinking for the multi-plant capacitated lot sizing problem. *European Journal of Operational Research*, 200(3):747 – 754, 2010.

[35] L. Özdamar and T. Yazgaç. A hierarchical planning approach for a production-distribution system. *International Journal of Production Research*, 37(16):3759–3772, 1999.

[36] Y.B. Park. An integrated approach for production and distribution planning in supply chain management. *International Journal of Production Research*, 43(6):1205–1224, 2005.

[37] M. Pătraşcu and D. Stratila. Faster primal-dual algorithms for the economic lot-sizing problem. In *20th International Symposium on Mathematical Programming*, 2009.

[38] Y. Pochet. *Lot-sizing problems: Reformulations and cutting plane algorithms*. PhD thesis, Université Catholique de Louvain, Belgium, 1987.

[39] Y. Pochet and L.A. Wolsey. Polyhedra for lot-sizing with Wagner–Whitin costs. *Mathematical Programming*, 67:297–324, 1994.

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

[41] R.L. Rardin and L.A. Wolsey. Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems. *European Journal of Operational Research*, 71:95–109, 1993.

[42] P. Robinson, A. Narayanan, and F. Sahin. Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms. *Omega*, 37:3–15, 2009.

[43] M. Sambasivan. *Uncapacitated and capacitated lot-sizing for multi-plant, multi-item, multi-period problems with inter-plant transfers*. PhD thesis, University of Alabama, United States, 1994.

[44] M. Sambasivan and S. Yahya. A heuristic procedure for solving multi-plant, mutli-item, mutli-period capacitated lot-sizing problems. *Asia - Pacific Journal of Operational Research*, 19(1):87–105, 2002.

[45] M. Sambasivan and S. Yahya. A lagrangean-based heuristic for multi-plant, multi-item, multi-period capacitated lot-sizing problems with inter-plant transfers. *Computers & Operations Research*, 32(3):537–555, 2005.

[46] O. Solyali and H. Sural. The one-warehouse multi-retailer problem: Reformulation, classification, and computational results. Technical Report 08-02, Industrial Engineering Department, METU Ankara, Turkey, 2009.

[47] K.E. Stecke and X. Zhao. Production and transportation integration for a make-to-order manufacturing company with a commit-to-delivery business mode. *Manufacturing and Service Operations Management*, 9:206–224, 2007.

[48] G. Strack and Y. Pochet. An integrated model for warehouse and inventory planning. *European Journal of Operational Research*, 204(1):35 – 50, 2010.

[49] D. Stratila. Private communication.

[50] C.H. Timpe and J. Kallrath. Optimal planning in large multi-site production networks. *European Journal of Operational Research*, 126(2):422 – 435, 2000.

[51] S. van Hoesel, H.E. Romeijn, D. Romero Morales, and A.P.M. Wagelmans. Integrated lot sizing in serial supply chains with production capacities. *Management Science*, 51:1706–1719, 2005.

[52] M. Van Vyve and L.A. Wolsey. Approximate extended formulations. *Mathematical Programming B*, 105:501–522, 2006.

[53] A. Wagelmans, S. van Hoesel, and A. Kolen. Economic lot-sizing: An $O(n \log n)$ algorithm that runs in linear time in the Wagner-Whitin case. *Operations Research*, 40:145–156, 1992.

[54] H.M. Wagner and T.M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5(1):89–96, 1958.

[55] S.D. Wu and H. Golbasi. Multi-item, multi-facility supply chain planning: Models, complexities, and algorithms. *Computational Optimization and Applications*, 28:325–356, 2004.

[56] W.I. Zangwill. A backlogging model and a multi-echelon model of a dynamic economic lot size production system - a network approach. *Management Science*, 15:506–526, 1969.