THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

AN EXACT SOLUTION ALGORITHM FOR THE COORDINATED CAPACITATED LOT-SIZING PROBLEM

Master Thesis

ZEYNEP SEZER

ISTANBUL, 2013

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
MASTER OF SCIENCE IN INDUSTRIAL ENGINEERING

AN EXACT SOLUTION ALGORITHM FOR THE COORDINATED CAPACITATED LOT-SIZING PROBLEM

Master Thesis

ZEYNEP SEZER

Thesis Supervisor: Asst. Prof. Semra AĞRALI

ISTANBUL, 2013

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
MASTER OF SCIENCE IN INDUSTRIAL ENGINEERING

An Exact Solution Algorithm for the
Coordinated Capacitated Lot-Sizing Problem

Zeynep SEZER
January 24th, 2013

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assoc. Prof. Tunç BOZBURA
Graduate School Director
Signature

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Science.

Asst. Prof. Barış SELÇUK
Program Coordinator
Signature

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Examining Committee Members	Signature
Thesis Supervisor Asst. Prof. Semra AĞRALI	----------------------------------
Member Asst. Prof. İbrahim MUTER	----------------------------------
Member Asst. Prof. Fadime ÜNEY-YÜKSEKTEPE	----------------------------------

ABSTRACT

AN EXACT SOLUTION ALGORITHM FOR THE COORDINATED CAPACITATED LOT-SIZING PROBLEM

Zeynep Sezer

Master of Science in Industrial Engineering

Thesis Supervisor: Asst. Prof. Semra Ağralı

January 2013, 33 pages

In this thesis we study large-scale coordinated capacitated lot sizing problems (CCLSP). CCLSP is the most general type of lot sizing problems, where (1) multiple items are involved in the production; (2) each item requires an individual (minor) setup cost in addition to a production cost; (3) items are grouped into families that share an additional joint (major) setup cost; (4) demand for an item in a period can be satisfied by production in any period; however, early and late productions add inventory holding and backlogging costs, respectively, and (5) production capacity in each period is limited.

The problem is to determine the production schedule over a time horizon consisting of a number of fixed-length production periods that minimizes the total production cost while satisfying a given demand under the capacity constraints.

CCLSP is essentially a mixed integer programming problem. It is known to be NP-hard, and therefore, most of the existing solution procedures are heuristics. CCLSPs considered in the literature include a single product family. In this thesis, we extend CCLSP by considering multiple product families. The goal of this study is to develop an exact solution algorithm for a large-scale multi-family CCLSP. The algorithm is based on Benders decomposition method, and it provides an alternative to existing approaches to solve mixed integer programming problems. The decomposition is based on a natural partitioning of the decision variables into continuous (production variables) and binary (major and minor setup variables) sets. The main contribution of this thesis will be the consideration of multiple product families, their effect on solution times and an exact algorithm to solve multi-family CCLSPs.

The performance of the algorithm is tested with respect to solution times by comparing the results with those obtained by solving the standard mixed integer programming problem without decomposition. Data sets used in comparison are generated to comply with the benchmark examples available in the literature.

Keywords: Coordinated capacitated lot sizing problem, capacitated lot sizing problem, joint setup, multiple product families, backlogging.

ÖZET

KOORDİNELİ KAPASİTELİ ÖBEK BÜYÜKLÜĞÜ BELİRLEME PROBLEMİ İÇİN KESİN SONUÇLU BİR ÇÖZÜM ALGORİTMASI

Zeynep Sezer

Endüstri Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Yrd. Doç. Dr. Semra Ağralı

Ocak 2013, 33 sayfa

Bu tezde büyük ölçekli koordineli kapasiteli öbek büyüklüğü belirleme problemleri (KKÖBP) incelenmiştir. KKÖBP (1) birden çok ürün ailesini içeren; (2) her ürünün üretim maliyetine ek olarak küçük kurulum maliyeti gerektirdiği; (3) ürünlerin büyük kurulum maliyetini paylaştıkları ürün ailelerine gruplandıkları; (4) ürünlerin dönemlik taleplerinin herhangi bir dönemde karşılanabildiği; ancak daha önceki ya da sonraki üretim dönemleriyle karşılanan taleplerin sırasıyla envanter tutma maliyeti ve geriye dönük tedarik maliyeti eklediği; ve (5) dönemlik üretim kapasitesinin sınırlı olduğu en genel öbek büyüklüğü belirleme problemidir.

Problem sabit-süreli üretim dönemlerinden oluşan bir zaman dilimi boyunca kapasite kısıtlarını aşmadan ve bilinen talepleri karşılayacak şekilde bütün üretim maliyetlerini en aza indirgeyen üretim planını belirlemektir.

Esasen KKÖBP karma tamsayılı programlama problemidir. Bu problemler NP-Zor olduklarından mevcut çözüm yöntemlerinin çoğu sezgiseldir. Literatürde dikkate alınmış KKÖBP tek bir ürün ailesini kapsar. Bu tezde KKÖBP'nin kapsamı birden çok ürün ailesini dikkate alarak genişletilmiştir. Bu tezin amacı büyük ölçekli, birden çok ürün ailesini içeren KKÖBP için kesin sonuçlu bir çözüm algoritması geliştirmektir. Önerilen çözüm algoritması Benders ayrıştırma yöntemine dayanmaktadır ve karma tamsayılı programlama problemlerini çözmede kullanılan mevcut yöntemlere alternatif oluşturmaktadır. Ayrıştırma, karar değişkenlerinin doğal olarak sürekli (üretim, envanter tutma ve geriye dönük tedarik maliyetleri) ve ikili (küçük ve büyük kurulum maliyetleri) setlere paylaştırılması temeline dayanmaktadır. Bu tezin başlıca katkısı, birden çok ürün ailesini içeren KKÖBP için kesin sonuçlu bir çözüm algoritması geliştirilmesi ve ürün ailelerinin çözüm sürelerine etkilerinin araştırılmasıdır.

Algoritmanın performansı çözüm sürelerinin ayrıştırılma yapılmamış karma tamsayılı programlama problemlerinin sonuçlarıyla karşılaştırılmasıyla test edilmiştir. Kullanılan veri setleri literatürde mevcut örneklere uygun olarak oluşturulmuştur.

Anahtar Kelimeler: Koordineli kapasiteli öbek büyüklüğü belirleme problemi, **k**apasiteli öbek büyüklüğü belirleme problemi, ortak kurulum, birden çok ürün ailesi, geriye dönük tedarik.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CCLSP	:	Coordinated capacitated lot sizing problem
CLSP	:	Single-item capacitated lot sizing problem
CULSP	:	Coordinated uncapacitated lot sizing problem
LB	:	Lower bound
LSP	:	Single-level, single-resource, big bucket lot-sizing problem with deterministic, dynamic demand
MCCLSP	:	Multiple product family, coordinated capacitated lot sizing problem
MCLSP	:	Multi-item capacitated lot sizing problem
MCULSP	:	Multiple product family, coordinated uncapacitated lot sizing problem
MULSP	:	Multi-item uncapacitated lot sizing problem
UB	:	Upper bound
ULSP	:	Single-item uncapacitated lot sizing problem
VI	:	Valid inequality

LIST OF SYMBOLS

T	:	Number of time periods in the planning horizon, $t \in T$, $t' \in T$.
J	:	Number of product families, $j \in J$.
K_j	:	Number of products in product family j, $k \in K_j$.
$d_{kt'}$	:	Demand for item $k \in K_j$ in time period $t \in T$.
S_{jt}	:	Major setup cost of product family $j \in J$ in time period $t \in T$.
s_{kt}	:	Minor setup cost of item $k \in K_j$ in time period $t \in T$.
c_{kt}	:	Per unit production cost of item $k \in K_j$ in time period $t \in T$.
$h_{ktt'}$	:	Per unit inventory holding cost of item $k \in K_j$ produced in time period $t \in T$ to supply demand in time period $t' \in T$, $t < t'$.
$b_{ktt'}$	:	Per unit backlogging cost of item $k \in K_j$ produced in time period $t \in T$ to supply demand in time period $t' \in T$, $t' < t$.
$C_{ktt'}$	:	The total unit variable cost for item $k \in K_j$ in time period $t \in T$ to serve demand in time period $t' \in T$.
P_t	:	Available production capacity in time period $t \in T$.
$X_{ktt'}$	:	The fraction of demand for product $k \in K_j$ in time period $t' \in T$ that is supplied from a production in time period $t \in T$.
Y_{kt}	:	binary decision variable that take on the value of 1, if a minor setup is scheduled for product $k \in K_j$ in time period $t \in T$ and 0 otherwise.
Z_{jt}	:	binary decision variable that take on the value of 1 if a major setup is scheduled for any of the products that belong to product family $j \in J$ in time period $t \in T$ and 0 otherwise.

1. INTRODUCTION

Lot sizing problems basically deal with determining an optimal production plan over a predetermined time horizon. Their objective is to minimize the sum of production, setup and inventory holding costs while satisfying demand. However, they differ substantially in terms of the assumptions made on the nature of variables and parameters considered. In the presence of so many, deciding on "how much to produce" and "when to produce" becomes a complicated task that has been challenging industry practitioners and research for the last fifty years.

Due to their wide spectrum of features, lot-sizing problems have been classified, modeled and solved in different ways throughout the literature. Classification is usually based on the assumptions made about the following criteria:

i) Planning horizon: Lot sizing problems are modeled by dividing a finite planning horizon into equal-length time intervals and defining all variables and parameters in terms of these non-overlapping time periods. When time periods are long enough to produce more than one item, the problem is referred to as a big bucket problem, otherwise, a small bucket problem. Big bucket problems, as opposed to small bucket problems, do not take into account the sequencing decisions of production lots. In other words, each time period is evaluated independently without taking into account the savings that can be earned by discarding setups between consecutive periods.

ii) Nature of demand: One of the most significant features in characterizing lot sizing problems is the assumptions made on demand. In practice, demand can either be constant (static) or time varying (dynamic) over a planning horizon. Static lot sizing problems triggered the evolution of EOQ models which established the basis for dynamic models. In addition, if demand is assumed to be known in advance, it is considered to be deterministic, otherwise, it is considered stochastic.

iii) Number of stages: The production systems can be characterized as having single-level or multi-level structures. In single-level systems, demand is met through finished products that are manufactured in a lump directly from raw materials. In multi-level systems, manufacturing process includes sequential operations where demand at each stage is dependent on subsequent stages.

iv) Number of resources: It is very common in manufacturing systems to use multiple machines to perform the same task in parallel. Such systems are structurally more complex, because production lots have to be assigned to machines.

Distinction made on aforementioned features provided lot sizing literature to branch off into different problem categories that are studied exclusively. Different model formulations and solution approaches have been proposed for each category in an effort to find optimal or near optimal solutions. However, as the size of problems become larger, computation time necessary to solve them increase tremendously. Existing optimization software, fall short in coming with satisfactory results. That is the reason why lot-sizing problems keep on receiving considerable attention. It is crucial to explore new formulations and methods that will ease the computational burden of solving lot sizing problems.

1.1. SCOPE

Due to the diversity inherent in lot-sizing problems the context of this study will be restricted to single-level, single-resource, big bucket lot sizing problems with deterministic, dynamic demand and will be referred to as LSP. For a comprehensive review of lot-sizing literature, readers may refer to Rizk and Martel (2001).

LSP's are essentially discrete-time optimization problems where the planning horizon is divided into equal length time periods, over each of which all variables and parameters assume fixed values. The objective is to determine the quantity to be produced at each time period that will simultaneously satisfy demand at each period and minimize overall costs incurred throughout the planning horizon.

The mathematical structure and complexity of models depend on the specific characteristics of the production system being modeled. For example, when there is a restriction on maximum number of items that can be produced in a period, the problem becomes "*capacitated*" and when different types of products are grouped into a product family where they can share a joint (major) setup cost, the problem becomes "*coordinated*". Problem classes and how they relate to each other will be discussed in more detail in the literature review. However, it is important to acknowledge that with every additional variable or constraint the problem becomes more complex and thereby the computation time necessary to solve these problems increase rapidly.

The coordinated capacitated lot sizing problem (CCLSP) is the most complex problem class of LSP dealing with multiple items that are subject to individual (minor) setups, joint (major) setups and capacity restrictions. The CCLSP's are intractable, in the sense that no polynomial-time solution algorithm is known to exist. Therefore, much of the research in this field has been devoted to heuristic solution approaches. Although there is not much research done on the CCLSP class, we are confident that a thorough investigation of different model formulations and solution approaches proposed for more specific variations of the LSP will contribute in our attempt to find an exact solution for the CCLSP. In this study, we will follow an analytical approach by considering CCLSP within the broader context of its subclasses.

1.2. OBJECTIVES AND CONTRIBUTIONS

The objective of this thesis is to investigate the computational challenges encountered in solving CCLSP with multiple product families and present an exact solution algorithm by exploiting Benders decomposition technique. The problem will be modeled using an arborescent-network-based mixed integer programming (MIP) formulation proposed by Robinson and Gao (1996). Our contributions lie in two directions: i) Inclusion of multiple product families in the problem formulation as opposed to previous studies which considered problems with single product family. This would allow to investigate the effect of different major setup cost structures on the complexity of the problem. ii) Developing an exact solution algorithm that can be used as an alternative to mixed integer programming in solving large-scale CCLSP's. Our goal is to obtain at least a feasible solution before the program terminates due to memory requirements.

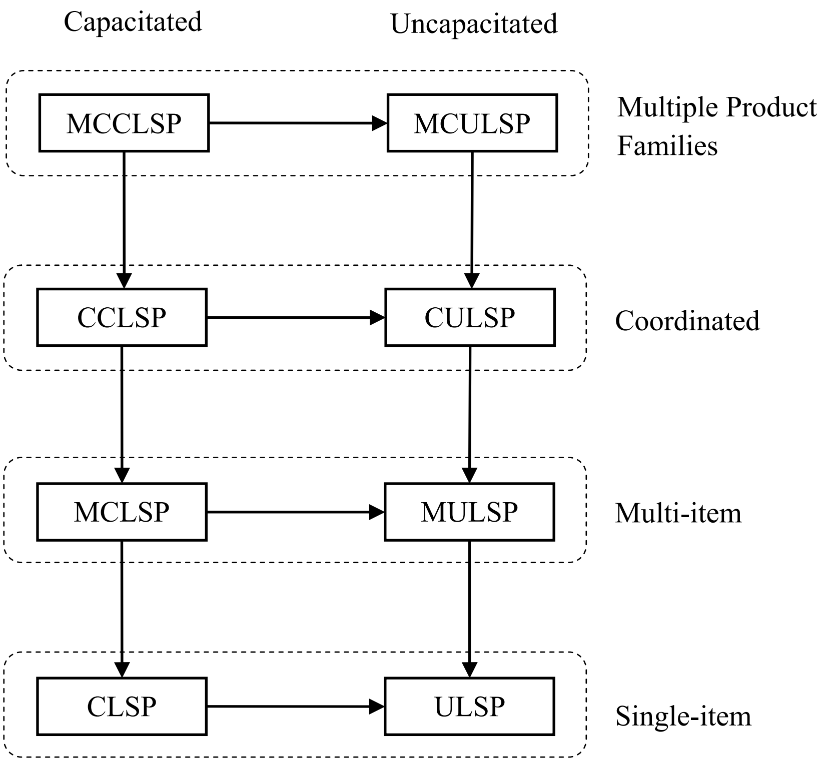
1.3. ORGANIZATION

This thesis is composed of five chapters including the current introductory chapter. In Chapter 2, related literature on dynamic lot-sizing problems will be presented together with the drawbacks of various models commonly used and solution methodologies proposed in the past. In Chapter 3, the mathematical formulation of the problem and the implementation of Benders partitioning procedure will be explained. The solution algorithm will also be given in this chapter. The numerical results based on different data sets and discussions will be given in Chapter 4. Finally, in Chapter 5, conclusions of the work done will be summarized and projections for future research will be specified.

2. LITERATURE REVIEW

The taxonomy of most frequently studied LSP's are presented in Figure 1. Problems are classified according to the number of items produced, capacity restrictions and the setup structure considered. These characteristics affect the number of variables and constraints within the model, and hence the mathematical complexity of the model. Problem classes with more complex structures are generalized versions of their subclasses. The arcs in the figure denote this structural relationship between classes from more general problems to more specific ones. Specific problem classes are in fact relaxed versions of the general problems.

Figure 2.1 - Taxonomy of deterministic dynamic lot-sizing problems.



Numerous studies have been done in the field of LSP to explore better model formulations and to develop faster solution algorithms specific to each problem class. Different cost structures have been analyzed along with problem extensions such as backlogging, remanufacturing, perishable inventory, time windows etc. Having an indepth knowledge on the subclasses of CCLSP can be very advantageous and time saving, especially when the problem can be reduced down to one of its subclasses for which well established approaches are available.

In the next section, a condensed version of the literature on each problem class will be given. Since the main concern of this research is to develop an exact solution algorithm for CCLSP, the heuristic approaches (approximation algorithms) will only be mentioned briefly. For detailed reviews on uncoordinated LSP's, readers may refer to Brahimi et al. (2006) and Karimi, Fatemi Ghomi and Wilson (2003) for the single-item and capacitated problems, respectively.

2.1. SINGLE-ITEM UNCAPACITATED LOT-SIZING PROBLEM (ULSP)

LSP under the dynamic demand assumption was first introduced by Wagner and Whitin (1958), in which they proposed a dynamic programming algorithm to solve a single-item LSP without capacity restrictions and demand backlogging. They used a backward recursive solution procedure that solved the problem in $\mathcal{O}(T^2)$ time, where T indicates the number of time periods in the planning horizon. Later, Zangwill (1966) extended the model to allow backlogging and Evans (1985) provided an efficient computer implementation of the W-W model. Several other dynamic programming algorithms were proposed in an effort to improve upon the computation time. Federgruen and Tzur (1991), Wagelmans, Van Hoesel and Kolen (1992) and Aggarwal and Park (1993) presented improved algorithms that were able to solve the problem in $\mathcal{O}(T)$ and $\mathcal{O}(T \log T)$ time.

The heuristic approaches studied in this field lost their appeal with the emergence of linear time algorithms just mentioned. Nevertheless, for completeness and future reference, it is worth mentioning the Part Period Balancing (De Matteis and Mendoze, 1968), the Least Unit Cost heuristic (Gorham, 1968), and the Silver-Meal heuristic (Silver and Meal, 1973) as the most significant heuristic approaches.

2.2. MULTI-ITEM UNCAPACITATED LOT-SIZING PROBLEM (MULSP)

Due to lack of joint resources (joint setups and capacity constraint), problems in this class are solved for each item separately, as independent ULSP's. Any technique used for ULSP is applicable to these types of problems.

2.3. SINGLE-ITEM CAPACITATED LOT-SIZING PROBLEM (CLSP)

Adding a capacity constraint to ULSP extends the model into a CLSP. Capacity limitations indicate a more realistic representation of production systems because they are caused by scarce resources such as labor, machine capability, storage space etc. They restrict production levels and complicate the mathematical structure of the problem, often resulting in NP-hard problems. However, polynomial-time solutions have been reported for problems with special cost structures and capacity assumptions. The computational complexity of these problems have been studied by Florian, Lenstra and Rinnooy Kan (1980) and Bitran and Yanasse (1982).

The constant capacity version of the CLSP has been mostly tackled with dynamic programming and few polynomial-time algorithms have been developed for their solutions. For example, Florian and Klein (1971) proposed an $\mathcal{O}(T^4)$ time algorithm to solve CLSP's with concave cost functions. The algorithm has been based on the shortest path method and allowed backlogging. Van Hoesel and Wagelmans (1996) improved upon Florian and Klein's algorithm and solved the problem in $\mathcal{O}(T^3)$ time, under linear holding cost assumption.

On the other hand, solution approaches proposed to solve non-polynomial problems under time-varying capacity assumptions, have been usually based on dynamic programming or branch-and-bound method. Baker et al. (1978) studied the problem under the assumption of constant costs and suggested a tree search algorithm that runs in $\mathcal{O}(2^T)$ time. They concluded that their algorithm is practical for reasonably sized problems but less efficient on highly constrained problems. Lambert and Luss (1982) suggested to define capacity constraints as integer multiples of a common divisor and solved the problem in $O(N^2T^4)$ time, where N is the maximum multiplier. Efficient results were reported for problems with relatively small N's. Chung, Flynn and Lin (1994) suggested an approach that combined dynamic programming with branch-and-bound method.

The most general case of CLSP has been solved by a dynamic programming algorithm developed by Chen, Hearn and Lee (1994). The problem assumed piecewise linear cost functions that are neither convex nor concave and has been solved in pseudo-polynomial time. Their algorithm has been the first to solve problems with more than 24 periods. In a more recent study, the same problem has been studied under general holding costs assumption. The dynamic programming algorithm developed by Shaw and Wagelmans (1998) has run in $O(T^2\bar{q}\bar{d})$ time, where T is the number of periods, $\bar{d}$ is the average demand and $\bar{q}$ is the average number of linear pieces required to represent the production cost function.

The heuristic approaches to solve the capacitated LSP's have been mostly suggested for multi-item problems. Although some of these procedures can be applied to single-item problems, they will be mentioned in the following section.

2.4. MULTI-ITEM CAPACITATED LOT-SIZING PROBLEM (MCLSP)

Most solution approaches that have been proposed to solve CLSP employed heuristic methods since the problem has proved to be NP-hard by Florian, Lenstra and Rinnooy Kan (1980). Even though the mixed integer programming formulation of these problems can be solved to optimality using a branch-and-bound method, the computation time increases significantly with the size of the problem. Therefore, few studies have been concentrated on the polyhedral structure of the problem in search for stronger formulations that reduces the solution time. Barany, Van Roy and Wolsey (1984) and Leung, Magnanti and Vachani (1989) employed cutting plane methods to identify valid inequalities by which the reformulation of the model has given a good approximation of the convex hull of feasible solutions. Eppen and Martin (1987) used a variable redefinition technique to obtain tighter linear relaxation for the MCLSP. Nevertheless, solving the MCLSP optimally has not been sufficient to accelerate computational time, which led the researchers to seek heuristic approaches.

Heuristic approaches have been classified into two different categories: i) single-resource (common-sense) heuristics; and ii) mathematical programming based heuristics. The former category is of greedy type and as suggested by Maes and Van Wassenhove (1988) mainly include period-by-period heuristics (Eisenhut, 1975; Lambrecht and Vanderveken, 1979; Dixon and Silver, 1981; Maes and Van Wassenhove, 1986) and improvement heuristics (Dogramaci, Panayiotopoulos and Adam, 1981; Karni and Roll, 1982; Gunther, 1987; Selen and Heuts, 1989; Trigeiro, 1989). The latter category use mathematical programming procedures such as Lagrangian relaxation (Thizy and Van Wassenhove, 1985; Billington, McClain, and Thomas 1983; Trigeiro, 1987; Trigeiro, Thomas and McClain, 1989; Diaby et al., 1992; Millar and Yang, 1994) and column generation techniques based on set-covering and set-partitioning approaches (Chen and Thizy, 1990; Cattrysse et al., 1993). Heuristics in this category produce better quality solutions compared to the heuristics in single-resource category and provide lower bound on the optimal solution. On the other hand, single-resource heuristics are faster and much easier to comprehend.

2.5. COORDINATED UNCAPACITATED LOT-SIZING PROBLEM (CULSP)

The CULSP has been shown to be NP-complete by Arkin, Joneja and Roundy (1989). Earlier solution approaches suggested for these problems are based on dynamic programming (Zangwill, 1966; Veinott, 1969; Kalymon, 1972; Kao, 1979). However, solution times of these algorithms increase significantly with the number of products and with the number of time periods within the problem. Silver (1976) showed that pure dynamic programming approaches were only suitable for small sized problems and that heuristic approaches should be considered for larger problems. Ter Haseborg (1982) studied the optimality conditions of joint ordering policies in an effort to reduce the number of time periods considered by the algorithm.

Another research stream has been built on solving the problem as a series of independent single-item W-W type problems that are coupled by major setup costs. Erenguc (1988) proposed a combined branch-and-bound and dynamic programming approach that solved problems with 12 time periods and 20 items to optimality. It has been reported that the solution times were sensitive to major/minor setup cost ratio and to the length of the planning horizon. Kirca (1995) considered the MIP formulation of the CULSP introduced by Joneja (1990) and proposed to solve the dual of the LP relaxation (a primal-dual heuristic) in order to obtain a strong lower bound on the original problem. The problem is then solved to optimality using branch-and-bound. Solutions to problems with 24 time periods and 50 items have been reported with this procedure.

More recently, Robinson and Gao (1996) formulated the problem as an arborescent fixed charge network programming and proposed a B&B procedure based on the dual ascent, dual adjustment and primal construction concepts introduced by Erlenkotter (1978). They have reported optimal solutions to problems with 12 (36) time periods and 40 (20) items. Computational results have shown the superiority of this procedure over existing approaches.

Heuristic solution approaches that have been proposed to solve CULSP include: Fogarty and Barringer (1987) which is based on the Silver and Kelle (1988) improvement procedure; Atkins and Iyogun (1988) which utilized the Silver and Meal (1973) heuristic; Joneja's (1990) "cost-covering" heuristic; Iyogun (1991) that extends the part period balancing method proposed by De Matteis and Mendoza (1968); Federgruen and Tzur's (1994) where a new greedy-add heuristic and a partitioning heuristic were proposed to obtain UB and LB, respectively. The performance of heuristics mentioned above has been evaluated by Boctor, Laporte and Renaud (2004). The models and algorithms of CCLSP and CULSP, readers may refer to Robinson et al. (2009).

2.6. COORDINATED CAPACITATED LOT-SIZING PROBLEM (CCLSP)

The CCLSP is a generalization of CULSP and MULSP and therefore NP-Hard. No exact solution procedure has yet been proposed to solve this type of problems. However, few heuristic methods have been suggested for different model formulations of CCLSP.

Erenguc and Mercan (1990) have considered a variant of CCLSP in which multiple families were involved and no backlogging was allowed. They assumed capacity in terms of time and hence, removed setup costs from the objective function. The proposed method uses B&B together with a shifting heuristic that shifts the production to earlier periods to avoid capacity violations.

In his dissertation, Lawrence (1999) have extended Robinson and Gao's (1996) arborescent fixed charge network programming formulation and has compared the performance of two Lagrangian relaxation methods that were used with a B&B procedure. He has concluded that relaxing the assignment constraint alone gives tighter lower bounds compared to relaxing the assignment and capacity constraint together. Altay (2001) proposed a cross decomposition procedure which unifies Benders decomposition with Lagrangian relaxation. The algorithm works iteratively between the primal sub-problem, which is generated by fixing the binary variables in the original problem, and a dual sub-problem, which is obtained by relaxing the demand constraint of the primal sub-problem, to attain UB and LB on the original problem. He has concluded that the problem becomes substantially difficult to solve when the ratio of joint setup cost to total cost increase. Robinson and Lawrence (2004) have also suggested a Lagrangian heuristic for a single product family CCLSP with backlogging, however, they were not able to provide satisfactory results. Gao and Robinson (2004) proposed a Lagrangian dual-ascent based heuristic whose performance dropped with an increase in capacity utilization and the joint setup cost. Federgruen, Meissner and Tzur (2004) suggested a strict partitioning (SP) heuristic and a progressive interval/expanding horizon (EH) heuristic. They have found that the performance of heuristic solutions is sensitive to capacity utilization and time-between-orders (TBO) of both items and product families.

3. PROBLEM DESCRIPTION AND SOLUTION METHODOLOGY

The CCLSP is highly encountered in production, procurement and transportation problems in which a family of items shares a common resource. Effective coordination of these resources presents opportunities for cost savings which make the CCLSP an attractive research area. As stated in previous sections, the CCLSP is a generalization of CULSP and MCLSP, and therefore, has a more complex structure entitled both to capacity restrictions and to joint setups. More specifically, the CCLSP demonstrates a production system in which a major (joint) setup cost is incurred when one or more items in a product family are produced in addition to minor setup costs incurred for each item produced. Our research extends the CCLSP by including multiple product families as opposed to earlier research that considered a single product family.

3.1. PROBLEM DESCRIPTION

A close examination of existing model formulations indicate that the MIP formulation and the arborescent network structure proposed by Robinson and Gao (1996) provide more flexibility in including extensions to the problem. By taking advantage of this property, Lawrence (1999) presented the capacitated version of the model. Since the CCLSP in our study includes multiple product families and allows backlogging of demand, we consider the arborescent network formulation to be the most suited to our case.

The CCLSP in this research includes a set of product families each of which exclusively consists of a number of items. The quantity and timing decisions for the production of these items are to be determined over a planning horizon, consisting of a number of equal-length time periods. Demand for an item belonging to a certain product family is dynamic and assumed to be known. The production capacity in each time period is assumed to be limited to a given value. For the sake of feasibility it is further assumed that the total capacity over all periods is greater than or equal to the total demand. A major setup cost is incurred when one or more items of a given product family are produced, a minor setup cost is incurred for every item that is produced and a per unit production cost are incurred individually for each item produced. In addition, when a demand for an item in a certain period is satisfied from an earlier production period, a per unit inventory holding cost is incurred and when a demand for an item in some period is satisfied from a later production period, a per unit backlogging cost is incurred. The following decision variables and parameters are defined for the problem:

Parameters:

T : Number of time periods in the planning horizon, $t \in T$, $t' \in T$.

J : Number of product families, $j \in J$.

K_j : Number of products in product family j, $k \in K_j$.

$d_{kt'}$: Demand for item $k \in K_j$ in time period $t \in T$.

S_{jt} : Major setup cost of product family $j \in J$ in time period $t \in T$.

s_{kt} : Minor setup cost of item $k \in K_j$ in time period $t \in T$.

c_{kt} : Per unit production cost of item $k \in K_j$ in time period $t \in T$.

$h_{ktt'}$: Per unit inventory holding cost of item $k \in K_j$ produced in time period $t \in T$ to supply demand in time period $t' \in T$, $t < t'$.

$b_{ktt'}$: Per unit backlogging cost of item $k \in K_j$ produced in time period $t \in T$ to supply demand in time period $t' \in T$, $t' < t$.

$C_{ktt'}$: The total unit variable cost for item $k \in K_j$ in time period $t \in T$ to serve demand in time period $t' \in T$.

P_t : Available production capacity in time period $t \in T$.

Decision Variables:

$X_{ktt'}$: The fraction of demand for product $k \in K_j$ in time period $t' \in T$ that is supplied from a production in time period $t \in T$.

Y_{kt} : 1, if a minor setup is scheduled for product $k \in K_j$ in time period $t \in T$ and 0 otherwise.

Z_{jt} : 1 if a major setup is scheduled for any of the products that belong to product family $j \in J$ in time period $t \in T$ and 0 otherwise.

With these definitions, the total unit variable cost can be stated as

$$C_{ktt'} = \begin{cases} c_{kt} + h_{ktt'}(t' - t) & if\ t < t' \\ c_{kt} + b_{ktt'}(t - t') & if\ t > t' \end{cases}$$

3.2. MATHEMATICAL FORMULATION

The problem is to determine which items to produce in each period so that the objective function, which is the sum of production, setup and inventory holding costs, is minimized while the demand for each item is satisfied. Thus, the mixed integer programming formulation of the problem can be given as follows:

(P) Minimize $\sum_{j\in J}\sum_{t\in T} S_{jt}Z_{jt} + \sum_{j\in J}\sum_{k\in K_j}\sum_{t\in T} s_{kt}Y_{kt} + \sum_{j\in J}\sum_{k\in K_j}\sum_{t\in T}\sum_{t'\in T} C_{ktt'}d_{kt'}X_{ktt'}$

Subject to $\sum_{t\in T} X_{ktt'} \geq 1 \quad \forall j \in J;\ k \in K_j; t' \in T$ (P-1)

$\sum_{j\in J}\sum_{k\in Kj}\sum_{t'\in T} d_{kt'}X_{ktt'} \leq P_t \quad \forall t \in T$ (P-2)

$X_{ktt'} \leq Y_{kt} \quad \forall j \in J;\ k \in K_j;\ t \in T; t' \in T$ (P-3)

$Y_{kt} \leq Z_{jt} \quad \forall j \in J;\ k \in K_j;\ t \in T$ (P-4)

$X_{ktt'} \geq 0 \quad \forall j \in J;\ k \in K_j;\ t \in T; t' \in T$ (P-5)

$Y_{kt} \in \{0, 1\} \quad \forall j \in J;\ k \in K_j;\ t \in T$ (P-6)

$Z_{jt} \in \{0, 1\} \quad \forall j \in J;\ t \in T$ (P-7)

Constraint set (P-1) guarantees that demand for each product in each time period will be satisfied. Constraint set (P-2) insures total realized production does not exceed the production capacity in each time period. Constraint set (P-3) prevents a minor setup from occurring unless a major setup has taken place. Similarly, constraint set (P-4) prohibits the production of an item unless a minor setup has been made. Constraint sets (P-5), (P-6), and (P-7) include non-negativity and binary requirements.

We will use Benders partitioning procedure to develop an exact algorithm to solve (P).

3.3. SOLUTION METHODOLOGY

Benders (1962) proposed an alternative approach to solve mixed integer problems by decomposing a given problem into smaller sub-problems by exploiting the natural partitioning of variables. The basic idea is to solve relatively easier sub-problems iteratively instead of solving a single large problem. In a mixed integer problem such as CCLSP, the master problem (MP) and the sub-problem (SP) are readily formulated by the natural partitioning of the variables into disjoint subsets of integer variables and continuous variables.

In the following we first provide a brief review of Benders partitioning procedure as interpreted by Taskin (2010), and then apply it to our problem.

3.3.1. Benders Partitioning Procedure

Consider the linear problem

(P) Minimize $f^T x + g^T y$

Subject to $Ax + By = b$

$x \geq 0$

where x is a p-dimensional vector of continuous variables, and y is a q-dimensional vector of binary variables. (P) can be rewritten as

(MP) Minimize $g^T y + q(y)$

where $q(y)$ is the optimal value of

(SP) Minimize $f^T x$

Subject to $Ax = b - By$

$x \geq 0$

which defines a decomposition of (P) into a master problem (MP) and a sub-problem (SP). If (SP) is unbounded for some y, then so are (MP), and hence, (P). We therefore assume (SP) is bounded for all y. Then $q(y)$ can be found by solving the dual of (SP):

(D-SP) Maximize $(b - By)^T \delta$

Subject to $A^T \delta \leq f$

We observe that feasibility of (D-SP) is independent of y, which means that if it is infeasible for some y, then it is infeasible for all y, and hence (P) is infeasible. Thus we assume (P), and hence (D-SP) is feasible, and characterize the feasible region of (D-SP) by its extreme rays $(\rho_1,\ldots,\rho_R)$, and extreme points $(\pi_1,\ldots,\pi_P)$. Clearly, if $(b - By)^T \rho_j > 0$ for an extreme ray ρ_j, then (D-SP) is unbounded for that particular y; and if an extreme point π_i maximizes (D-SP), then (SP) has a finite optimal value. It follows that (SP) is equivalent to

(SPE) Minimize q

Subject to $(b - By)^{\mathrm{T}} \rho_j \leq 0 \quad j = 1, \ldots, R$

$(b - By)^{\mathrm{T}} \pi_i \leq q \quad i = 1, \ldots, P$

and hence, the original problem (P) is equivalent to

(PE) Minimize $g^T y + q$

Subject to $(b - By)^{\mathrm{T}} \rho_j \leq 0 \quad j = 1, \ldots, R$

$(b - By)^{\mathrm{T}} \pi_i \leq q \quad i = 1, \ldots, P$

Benders procedure starts with a small subset of the constraints in (PE), and solves the corresponding relaxed problem to obtain a candidate solution (q^*, y^*). It then solves (D-SP) with y replaced by y^*. Three cases are possible:

i) (D-SP) is unbounded. In this case, a new constraint of the type $(b - By)^T \rho_j \leq 0$ is generated, added to the constraint set of the relaxed problem, and the process is repeated with the augmented constraint set.

ii) (D-SP) has optimal solution $q(y^*)$, and $q(y^*) \geq q^*$. In this case, a constraint of the type $(b - By)^{\mathrm{T}} \pi_i \leq q$ is generated, added to the constraint set of the relaxed problem, and the process is repeated with the augmented constraint set.

iii) (D-SP) has optimal solution $q(y^*)$ with $q(y^*) = q^*$. In this case, (q^*, x^*) is the optimal solution of (SP), and the process is terminated.

Constraints added in cases i) and ii) are known as Benders feasibility cuts and Benders optimality cuts, respectively. Since there are a finite number of cuts, Benders procedure converges to an optimal solution in a finite number of steps.

3.3.2. Application of Benders Procedure to CCLSP

When the original problem (P) is decomposed according to Benders partitioning procedure, the following master problem, containing the integer variables Y_{kt} and Z_{jt}, is obtained:

(MP) Minimize $\sum_{j \in J} \sum_{t \in T} S_{jt} Z_{jt} + \sum_{j \in J} \sum_{k \in Kj} \sum_{t \in T} s_{kt} Y_{kt} + \theta(Y)$

Subject to $Y_{kt} \leq Z_{jt} \quad \forall j \in J;\ k \in K_j;\ t \in T$ (MP-1)

$Y_{kt} \in \{0, 1\} \quad \forall j \in J;\ k \in K_j;\ t \in T$ (MP-2)

$Z_{jt} \in \{0, 1\} \quad \forall j \in J;\ t \in T$ (MP-3)

$\theta(Y) \geq 0$ (MP-4)

{Bender's optimality cut} ∀ dual extreme point (MP-5)*

{Bender's feasibility cut} ∀ dual extreme ray (MP-6)*

where $\theta(Y)$ is a decision variable that is a function of Y. The constraint sets (MP-5) and (MP-6) are generated during iterations and are added later on to the (MP).

The Y_{kt} values obtained from the (MP) are fixed to $\overline{Y}_{kt}$ and given as input to (SP). For a given set of $Y_{kt} = \overline{Y}_{kt}$, the sub-problem reduces to a linear program as shown below.

(SP) Minimize $\sum_{j\in J}\sum_{k\in Kj}\sum_{t\in T}\sum_{t'\in T} C_{ktt'}d_{kt'}X_{ktt'}$

Subject to $\sum_{t\in T} X_{ktt'} \geq 1 \quad \forall\, j \in J;\ k \in K_j;\ t' \in T$ (SP-1)

$\sum_{j\in J}\sum_{k\in Kj}\sum_{t'\in T} d_{kt'}X_{ktt'} \leq P_t \quad \forall\, t \in T$ (SP-2)

$X_{ktt'} \leq \overline{Y}_{kt} \quad \forall\, j \in J;\ k \in K_j;\ t \in T; t' \in T$ (SP-3)

$X_{ktt'} \geq 0 \quad \forall\, j \in J;\ k \in K_j;\ t \in T; t' \in T$ (SP-4)

The essence of Benders decomposition is the observation that the feasibility region of the dual of (SP), denoted as (D-SP), is independent of the $\overline{Y}_{kt}$ passed from the (MP), and hence, it is much more advantageous to solve the (D-SP) rather than (SP) itself.

Let $\upsilon_{kt'}$, μ_t, and $\omega_{ktt'}$ be dual variables corresponding to constraint sets (SP-1), (SP-2), and (SP-3). Then the (D-SP) can be written as follows:

(D-SP) Maximize $\sum_{j\in J}\sum_{k\in Kj}\sum_{t'\in T} \upsilon_{kt'} - \sum_{t\in T} P_t\mu_t - \sum_{j\in J}\sum_{k\in Kj}\sum_{t\in T}\sum_{t'\in T} \overline{Y}_{kt}\omega_{ktt'}$

Subject to $\upsilon_{kt'} - d_{kt'}\mu_t - \omega_{ktt'} \leq C_{ktt'}d_{kt'} \quad \forall\, j \in J;\ k \in K_j;\ t \in T;\ t' \in T$

$\upsilon_{kt'} \geq 0 \quad \forall\, j \in J;\ k \in K_j;\ t' \in T$

$\mu_t \geq 0 \quad \forall\, t \in T$

$\omega_{ktt'} \geq 0 \quad \forall\, j \in J;\ k \in K_j;\ t \in T;\ t' \in T$

Observe that for any given set of $\overline{Y}_{kt}$ the (D-SP) always has at least one feasible solution ($\upsilon_{kt'} = 0, \mu_t = 0, \omega_{ktt'} = 0$), therefore the solution to a (D-SP) can either be optimal or unbounded. An optimal solution refers to one of the extreme points $(\upsilon^p_{kt'}, \mu^p_t, \omega^p_{ktt'})$ defined in constraint set (MP-5) and an unbounded solution refers to one of the extreme rays $(\upsilon^r_{kt'}, \mu^r_t, \omega^r_{ktt'})$ defined in constraint set (MP-6).

When (D-SP) is unbounded for a given set of $\overline{Y}_{kt}$, (SP) is infeasible, thus (P) have no feasible solution for the assigned set of $\overline{Y}_{kt}$. To avoid progressing in the unbounded direction in the next iteration the extreme ray obtained from the unbounded (D-SP) is used to generate the following Benders feasibility cut:

$$\sum_{j\in J}\sum_{k\in Kj}\sum_{t\in T}\sum_{t'\in T} Y_{kt}\overline{\omega}_{ktt'} \geq \sum_{j\in J}\sum_{k\in Kj}\sum_{t'\in T} \overline{\upsilon}_{kt'} - \sum_{t\in T} P_t\overline{\mu}_t$$

When (D-SP) has an optimal solution for a given set of $\overline{Y}_{kt}$, the objective function value of (D-SP), provides an upper bound for the $\theta(Y)$ value in (P), thus the extreme point obtained from (D-SP) is used to generate the following Benders optimality cut:

$$\theta(Y) + \sum_{j\in J}\sum_{k\in Kj}\sum_{t\in T}\sum_{t'\in T} Y_{kt}\overline{\omega}_{ktt'} \geq \sum_{j\in J}\sum_{k\in Kj}\sum_{t'\in T} \overline{\upsilon}_{kt'} - \sum_{t\in T} P_t\overline{\mu}_t$$

Therefore, depending on the solution of the (D-SP) either an optimality cut or a feasibility cut is generated and inserted to the (MP) iteratively to solve for a new set of $\overline{Y}_{kt}$. The restricted (MP), called the (R-MP) which includes all the cuts that have been generated up to that point can be written as follows:

(**R-MP**) Minimize $\sum_{j\in J}\sum_{t\in T} S_{jt}Z_{jt} + \sum_{j\in J}\sum_{k\in Kj}\sum_{t\in T} s_{kt}Y_{kt} + \theta(Y)$

Subject to $Y_{kt} \leq Z_{jt} \quad \forall\, j \in J;\ k \in K_j;\ t \in T$

$Y_{kt} \in \{0,1\} \quad \forall\, j \in J;\ k \in K_j;\ t \in T$

$Z_{jt} \in \{0,1\} \quad \forall\, j \in J;\ t \in T$

$\theta(Y) \geq 0$

$\theta(Y) + \sum_{j\in J}\sum_{k\in Kj}\sum_{t\in T}\sum_{t'\in T} Y_{kt}\overline{\omega}_{ktt'} \geq \sum_{j\in J}\sum_{k\in Kj}\sum_{t'\in T} \overline{\upsilon}_{kt'} - \sum_{t\in T} P_t\overline{\mu}_t \quad \forall\, p \in \overline{P}$

$\sum_{j\in J}\sum_{k\in Kj}\sum_{t\in T}\sum_{t'\in T} Y_{kt}\overline{\omega}_{ktt'} \geq \sum_{j\in J}\sum_{k\in Kj}\sum_{t'\in T} \overline{\upsilon}_{kt'} - \sum_{t\in T} P_t\overline{\mu}_t \quad \forall\, r \in \overline{R}$

where $\overline{P}$ and $\overline{R}$ are a subset of all extreme points and all extreme rays of (D-SP), respectively. The solution obtained from (R-MP) provides a lower bound for the optimal solution of (P). For a given set of $\overline{Y}_{kt}$, lower bound $(P)_{LB}$ and upper bound $(P)_{UB}$ values for the optimal solution of (P) can be calculated as follows:

$$(\mathrm{P})_{\mathrm{LB}} = (\text{R-MP})_{\mathrm{z}}^{\mathrm{k}}$$

$$(\mathrm{P})_{\mathrm{UB}} = (\text{R-MP})_{\mathrm{z}}^{\mathrm{k}} - \theta\left(\mathrm{Y}^{\mathrm{k}}\right) + (\text{D-SP})_{\mathrm{z}}^{\mathrm{k}}$$

where for a specific problem A, $(A)_z^k$ denotes the optimal objective function value with a given set of $\overline{Y}_{kt}$. Therefore, when the optimal solution of (D-SP) equals to the value of $\theta(Y^k)$ for a given set of $\overline{Y}_{kt}$, the upper bound equals to lower bound, proving that we have reached the optimal solution of (P).

The following algorithm represents the Benders approach:

Step 1: *Initialization*

- Select an optimality tolerance ε
- Set iteration counter k ← 0
- Set $(\mathrm{P})_{\mathrm{LB}}$ ← 0 and $(\mathrm{P})_{\mathrm{UB}}$ ← ∞

Step 2: Choose an initial pair of vectors Z^k and Y^k and solve (R-MP) to obtain $\theta(Y^k)$

Step 3: Solve (D-SP) at $\overline{Y} = Y^k$

- If (D-SP) is feasible at $\overline{Y}$, get $(\mathrm{SP})_{\mathrm{z}}^{\mathrm{k}}$
  - If $((\mathrm{P})_{\mathrm{UB}} - (\mathrm{P})_{\mathrm{LB}})/(\mathrm{P})_{\mathrm{LB}} < \varepsilon$: **Stop**
  - If $((\mathrm{P})_{\mathrm{UB}} - (\mathrm{P})_{\mathrm{LB}})/(\mathrm{P})_{\mathrm{LB}} > \varepsilon$: Generate Benders optimality cut and insert it to (R-MP), k ← k + 1, go to Step 2.
- If (D-SP) is unbounded at $\overline{Y}$: Generate Benders feasibility cut and add it to (R-MP), k ← k + 1, go to Step 2.

3.4. EXPERIMENTAL DESIGN

The Bender's algorithm given in the previous section has been programmed in Java and run on a PC with Intel Pentium 2.00 GHz processor, 0.99 GB RAM and Windows XP operating system.

To test the performance of our algorithm random data sets are generated with the following design variables:

Planning horizon	: 12 periods
Capacity utilization levels	: 5%, 45%, 85%
Number of product families	: $J \in \{1, 2, 3, 4, 6\}$
Number of items in each family	: $K_j = K^J, \forall j \in J$
	$K^1 \in \{2, 4, 6, 8, 12, 16, 24\}$
	$K^2 \in \{2, 3, 4, 6, 8, 12\}$
	$K^3 \in \{2, 4, 8\}$
	$K^4 \in \{2, 3, 4, 6\}$
	$K^6 \in \{2, 4\}$

K^j's are chosen such that some total number of items are common to several different $\{J, K^J\}$ combinations to provide a comparison of these combinations. For example, $J = 2$, $K_1 = K_2 = K^2 = 8$ and $J = 4$, $K_1 = K_2 = K_3 = K_4 = K^4 = 4$, both correspond to the same total number of items ($J * K^J = 16$).

For each combination of design parameters, 12 sets of data are generated with parameters chosen from normal distributions except inventory holding costs and backlogging costs which are set to 1 and 3, respectively. Mean and standard deviation of minor setup costs, total unit variable costs and demands, as well as the standard deviation of major setup costs are the same for each data set, and are given in Table 3.1.

Table 3.41- Parameters of data sets

	Mean	Standard Deviation
S_{jt}	No. of item-dependent	36
s_{jkt}	60	18
$C_{jktt'}$	4	1
$d_{jkt'}$	100	20

Means of major setup costs are number-of-items-dependent, and are adjusted so as to have the same set of major setup cost to minor setup cost ratios as the data used in Erenguc (1988), Robinson and Gao (1996), and Nezih (2001). This is achieved by using the linear fit

$$S = 35K^J + 50$$

The solution times of problems solved with Benders algorithm are then compared to those solved with the standard mixed integer programming.

4. COMPUTATIONAL RESULTS & DISCUSSIONS

All problems are run using IBM's commercial optimization package called ILOG CPLEX Optimization Studio V12.4. The algorithm is coded using the lazy constraint callback method of CPLEX which requires the pre-solve option to be turned off to obtain accurate results. This is the only adjustment made on the default settings.

Table 4.1 - Computational results for 5% capacity utilization

Number of product families	Number of items in J	Total number of items	Mean Major Setup Cost	Solution Time (seconds)			
				CPLEX	Our Algorithm	Min	Max
1	2	2	120	0.1928	0.0703	0.0470	0.1100
1	4	4	190	0.2162	0.1199	0.0940	0.1560
1	6	6	260	0.2512	0.1615	0.1100	0.2500
1	8	8	330	0.2759	0.2175	0.1410	0.3130
1	12	12	470	0.3413	0.4206	0.2810	0.7030
1	16	16	610	0.3686	0.6978	0.4680	1.0940
1	24	24	890	0.5013	1.1263	0.7500	1.4070
2	2	4	120	0.2121	0.1367	0.0930	0.2030
2	3	6	155	0.2383	0.2083	0.1400	0.3120
2	4	8	190	0.2743	0.3149	0.2030	0.4690
2	6	12	260	0.3398	0.5821	0.3750	0.8900
2	8	16	330	0.3803	0.9103	0.5000	1.1710
2	12	24	470	0.4714	1.7108	1.0310	2.3130
3	2	6	120	0.2434	0.2241	0.1720	0.2970
3	4	12	190	0.3280	0.6274	0.3750	0.8600
3	8	24	330	0.4689	1.9099	1.3430	2.6710
4	2	8	120	0.2683	0.3423	1.1870	0.4680
4	3	12	155	0.3048	0.6095	0.3600	0.9220
4	4	16	190	0.3686	1.0664	0.7350	1.3120
4	6	24	260	0.4714	2.0871	0.9210	2.7180
6	2	12	120	0.3020	0.7005	0.4530	0.9840
6	4	24	190	0.4727	2.5323	1.8280	3.5620

CPLEX: Results obtained by solving the standard mixed integer programming problem in CPLEX without decomposition.

Results of computations are given in Tables 4.1, 4.2 and 4.3 for capacity utilization levels of 5, 45 and 85 percent, respectively. Comparison of the solution times with respect to total number of items are provided by Figures 4.1, 4.2 and 4.3, respectively.

Table 4.2 - Computational results for 45% capacity utilization

Number of product families	Number of items in J	Total number of items	Mean Major Setup Cost	Solution Time (seconds)			
				CPLEX	Our Algorithm	Min	Max
1	2	2	120	0.2134	0.6602	0.3280	1.1870
1	4	4	190	0.2540	1.5703	0.5780	2.6410
1	6	6	260	0.2458	1.9465	1.2030	2.4990
1	8	8	330	0.2710	14.1984	5.0310	25.2170
1	12	12	470	0.3241	31.3943	10.9840	75.8400
1	16	16	610	0.3894	96.8075	23.4990	205.3960
1	24	24	890	0.4715	73.8791	12.9680	282.6580
2	2	4	120	0.2266	1.1067	0.5000	1.8750
2	3	6	155	0.2462	1.3086	0.7660	2.0780
2	4	8	190	0.2970	6.3891	3.2500	14.4680
2	6	12	260	0.3166	8.6480	3.4210	30.1390
2	8	16	330	0.3814	172.8830	14.2960	741.1490
2	12	24	470	0.4740	110.2091	21.8270	399.6510
3	2	6	120	0.2500	0.9778	0.7030	1.6250
3	4	12	190	0.3243	4.8214	2.0780	7.8590
3	8	24	330	0.4817	23.0703	15.9520	35.9360
4	2	8	120	0.2732	2.7523	1.1250	5.1560
4	3	12	155	0.3243	4.3629	2.9840	8.6870
4	4	16	190	0.3724	21.7281	4.8590	69.7780
4	6	24	260	0.4726	32.1363	13.5460	92.2610
6	2	12	120	0.3333	1.7485	1.3280	2.3280
6	4	24	190	0.4829	7.1884	5.4210	12.0000

CPLEX: Results obtained by solving the standard mixed integer programming problem in CPLEX without decomposition.

Results indicate that for each fixed level of capacity utilization and fixed number of product families solution time increases with increasing number of items with only few exceptions. This result was anticipated since the problem size increases with increasing number of items, which in turn, increases the time required to solve the problem.

A second observation is that for every data set, the solution time increases with increasing levels of capacity utilization. This is consistent with existing results in the literature. Low levels of capacity utilization is equivalent to larger production capacities, as a result the problem approximates an uncapacitated problem. However, higher levels of capacity utilization put tighter constraints, making the problem more difficult to solve.

Table 4.3 - Computational results for 85% capacity utilization

Number of product families	Number of items in J	Total number of items	Mean Major Setup Cost	Solution Time (seconds)			
				CPLEX	Our Algorithm	Min	Max
1	2	2	120	0.3709	2.9959	2.1870	3.9370
1	4	4	190	0.4284	9.8835	4.2180	16.3900
1	6	6	260	0.4220	18.6149	7.7340	75.1210
1	8	8	330	0.5454	30.3213	17.8890	59.1690
1	12	12	470	0.6563	56.0182	15.0000	135.9620
1	16	16	610	0.7617	67.4901	38.9050	92.8860
1	24	24	890	1.0028	97.8205	60.5280	172.0540
2	2	4	120	0.4612	38.5812	4.8120	210.5980
2	3	6	155	0.5220	50.1459	16.7180	138.4620
2	4	8	190	0.6586	137.513	19.3740	717.4000
2	6	12	260	0.7474	165.977	15.1550	543.4410
2	8	16	330	1.1263	233.484	78.6680	447.3830
2	12	24	470	0.8593	869.450	125.4150	2,357.5040
3	2	6	120	0.4946	56.1978	17.5770	125.9940
3	4	12	190	0.7735	433.270	27.5920	1,802.1580
3	8	24	330	1.5441	4,034.47	91.7450	18,985.1830
4	2	8	120	0.8826	219.107	24.2960	1,210.5470
4	3	12	155	0.9298	751.556	41.9510	4,876.5780
4	4	16	190	1.0727	2,554.17	175.8500	6,461.1850
4	6	24	260	1.3034	16,554.8	961.4040	41,790.1440
6	2	12	120	0.8841	769.571	118.2280	2,566.3530
6	4	24	190	1.2083	4,464.27	65.2930	37,339.3740

CPLEX: Results obtained by solving the standard mixed integer programming problem in CPLEX without decomposition.

Figure 4.1 - Solution times for problems with J = 1 and CU = 5%, 45% and 85%

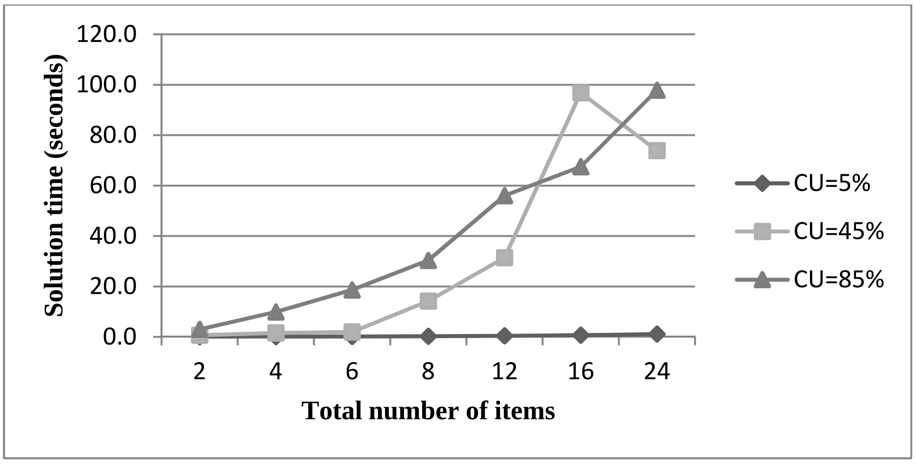


Figure 4.2 - Solution times for problems with J = 2 and CU = 5%, 45% and 85%

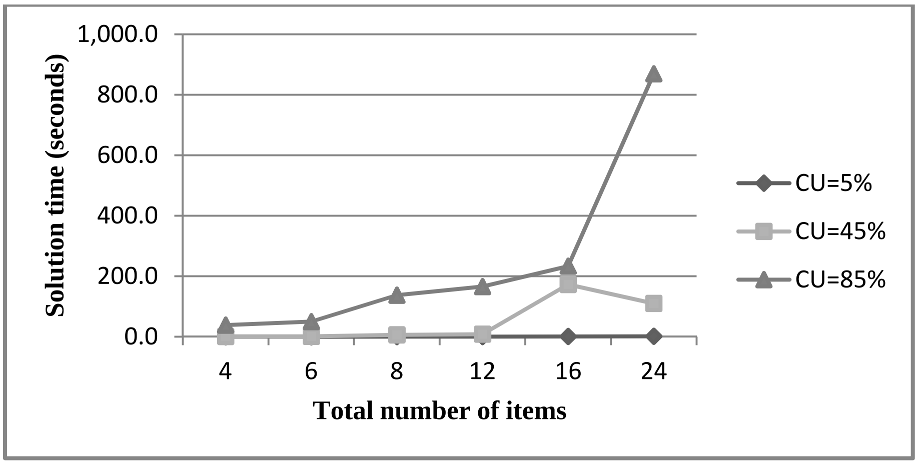
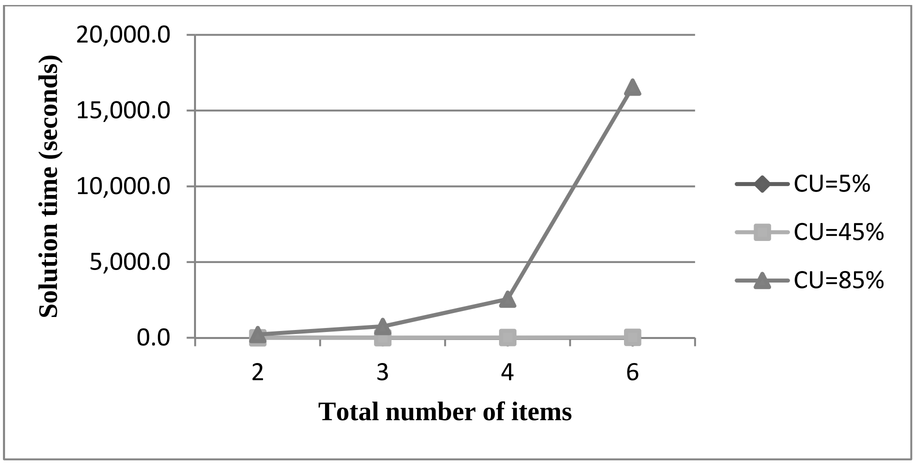


Figure 4.3 - Solution times for problems with J = 4 and CU = 5%, 45% and 85%



A third observation is concerned with the number of product families, which is the main feature that distinguishes this study from earlier work. The comparisons are made on problem sets with 45 and 85 percent capacity utilizations and for a total of 12 and 24 items. For 45 percent capacity utilization, solution times for problems with a total of 12 items decrease considerably with the division of items into multiple families (see Figure 4.4). A similar result is obtained for problems with a total of 24 items (see Figure 4.5).

Figure 4.4 - Solution times for problems with 12 items and CU = 45%

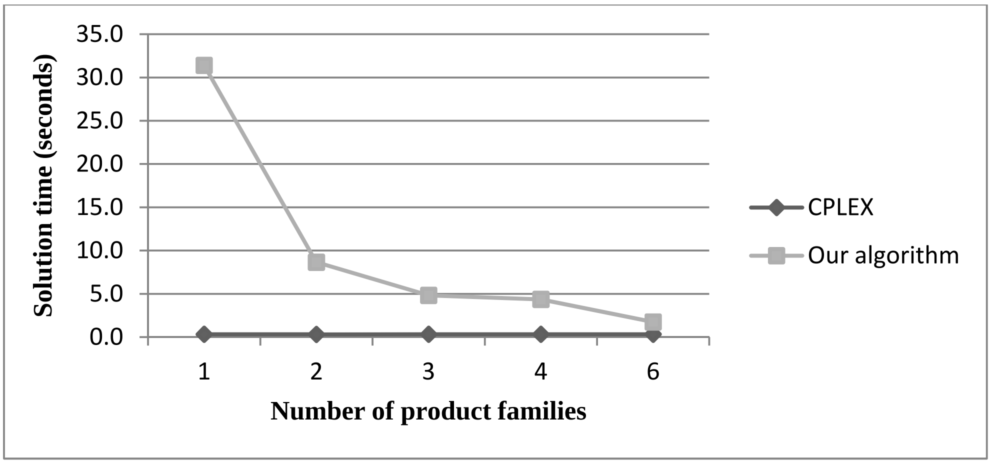
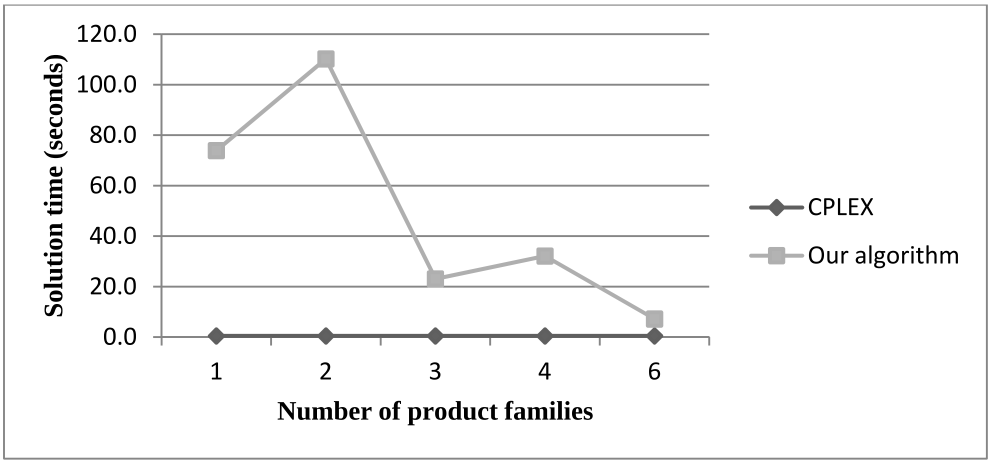


Figure 4.5 - Solution times for problems with 24 items and CU = 45%



For 45 percent capacity utilization, distributing the same number of items into multiple families seems to be greatly beneficial. However, the benefit of distributing items into product families seems to be lost for problems with 85 percent capacity utilization. For these problems, the solution times increase as the number of product families increases (see Figures 4.6 and 4.7). To draw a conclusion, a more detailed analysis on larger data sets is required.

Figure 4.6- Solution times for problems with 12 items and CU = 85%

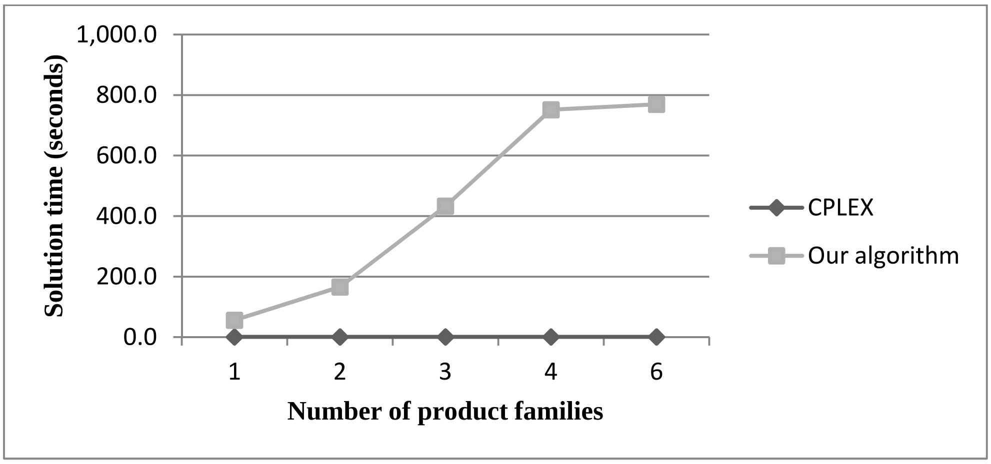
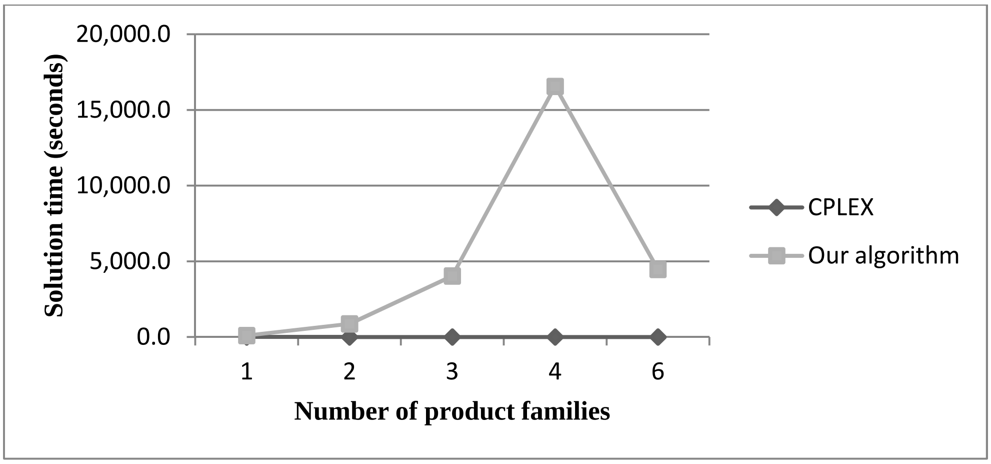
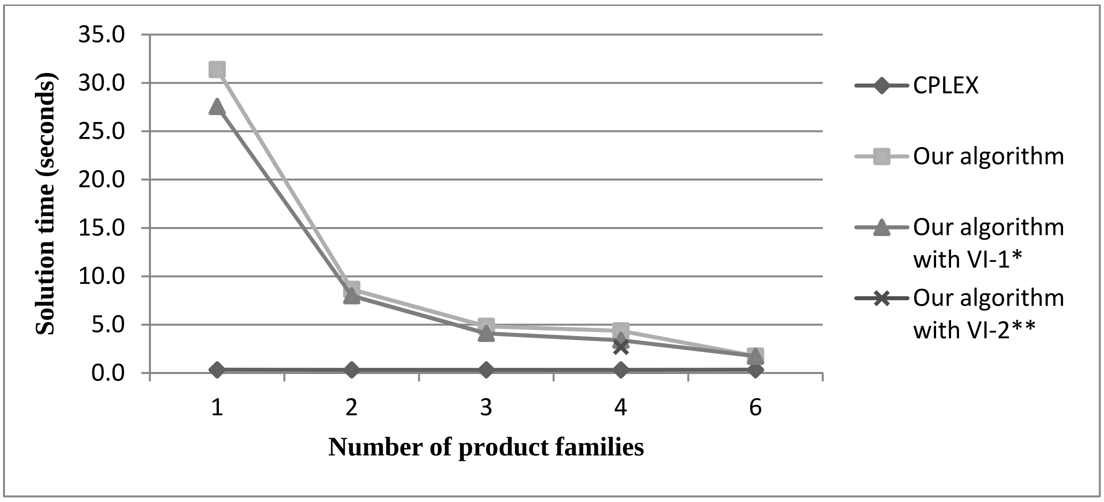


Figure 4.7 - Solution times for problems with 24 items and CU = 85%



Although the preliminary results obtained are not up to our expectations, solution times can be improved by making use of valid inequalities that reduces the solution space. One such inequality is obtained by defining the minimum number of time periods for which production must take place for each item. It is calculated by dividing the total demand for an item by the capacity of each period. Following the same logic, this valid inequality can also be used to define the minimum number of production periods of each product family. The effect of the valid inequality defined for each item (VI-1) was tested on problems with a total of 12 items with 45 and 85 percent capacity utilizations. As can be seen from Figures 4.8 and 4.9, the VI-1 shows improvement with respect to solution time for all data sets with 45 percent capacity utilization, but for only two data sets with 85 percent capacity utilization.

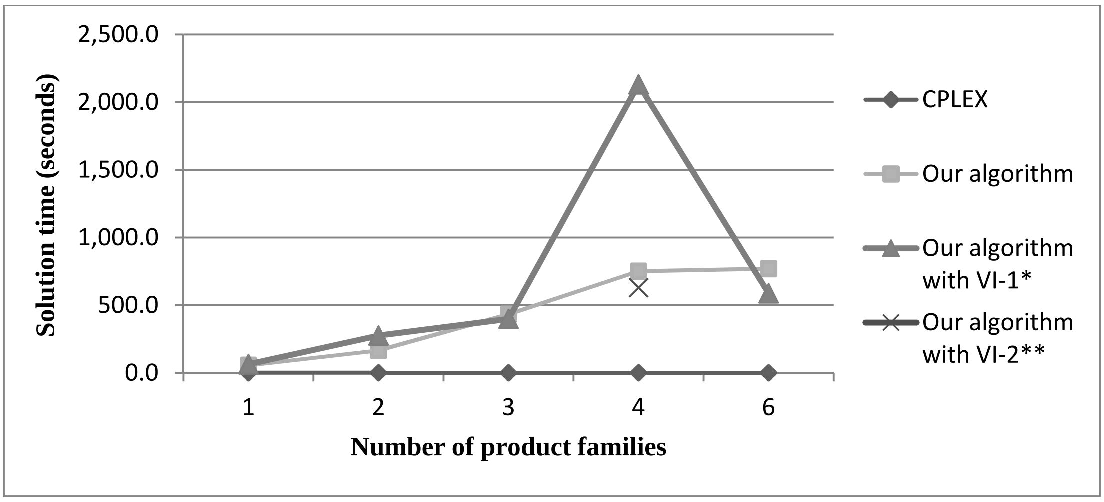
Figure 4.8 - Solution times with V.I.'s for problems with 12 items and CU=45%



*VI-1: Minimum number of production periods defined for each item.

**VI-2: Minimum number of production periods defined for each product family.

Figure 4.9 - Solution times with V.I.'s for problems with 12 items and CU=85%



*VI-1: Minimum number of production periods defined for each item.

**VI-2: Minimum number of production periods defined for each product family.

Table 4.4 - Improvements with the V.I.'s for problems with J=4, K^J=3 and CU=45%

Problem Number	Solution Time (seconds)			Percent Improvement	
	without VI	with VI-1	With VI-2	with VI-1*	With VI-2**
1	3.1090	3.3750	2.5000	-8.56%	19.59%
2	5.7350	3.6720	2.0940	35.97%	63.49%
3	2.9840	2.4840	5.6090	16.76%	-87.97%
4	4.2500	2.5310	1.8120	40.45%	57.36%
5	3.2340	3.5940	3.0310	-11.13%	6.28%
6	8.6870	4.7970	1.5930	44.78%	81.66%
7	3.3900	3.5310	3.1400	-4.16%	7.37%
8	3.8900	3.9690	2.2500	-2.03%	42.16%
9	6.6400	3.1410	2.5310	52.70%	61.88%
10	3.3440	3.1710	1.7810	5.17%	46.74%
11	4.0460	3.2820	2.3900	18.88%	40.93%
12	3.0460	3.1090	3.4690	-2.07%	-13.89%
Average	**4.3629**	**3.3880**	**2.6833**	**22.35%**	**38.50%**

*VI-1: Minimum number of production periods defined for each item.

**VI-2: Minimum number of production periods defined for each product family.

Table 4.5 - Improvements with the V.I.'s for problems with J=4, K^J=3 and CU=85%

Problem Number	Solution Time (seconds)			Percent Improvement	
	without VI	with VI-1	With VI-2	with VI-1*	With VI-2**
1	250.8310	1,378.2100	427.2900	-449.46%	-70.35%
2	45.1070	103.2450	101.7450	-128.89%	-125.56%
3	779.4750	5,542.8100	1252.8110	-611.10%	-60.72%
4	320.0770	408.7760	37.3260	-27.71%	88.34%
5	1,538.6240	601.4540	101.0410	60.91%	93.43%
6	82.6360	95.4950	91.3390	-15.56%	-10.53%
7	503.4430	381.2150	607.2810	24.28%	-20.63%
8	146.5700	265.7680	60.5290	-81.32%	58.70%
9	4,876.5780	15,227.2630	3273.4260	-212.25%	32.87%
10	41.9510	113.0570	128.3220	-169.50%	-205.89%
11	244.1120	274.1270	804.5060	-12.30%	-229.56%
12	189.2710	1,189.5950	665.8100	-528.51%	-251.78%
Average	**751.5563**	**2,131.7513**	**629.2855**	**-183.64%**	**16.27%**

*VI-1: Minimum number of production periods defined for each item.

**VI-2: Minimum number of production periods defined for each product family.

On the other hand, the effect of the valid inequality defined for each product family (VI-2) was tested on problems with 4 product families and a total of 12 items with 45 and 85 percent capacity utilizations. It is observed that the valid inequality defined for each product family provides greater reductions in solution times than does the valid inequality defined for each item. The percentage improvements achieved with both valid inequalities are given in Tables 4.4 and 4.5.

5. CONCLUSION AND FUTURE WORK

We have developed an exact solution algorithm for solving CCLSP with multiple product families. The algorithm utilizes Benders decomposition applied to a mixed integer programming problem, and can be used as an alternative to the classical one-shot solution method. In this aspect, work completed in this thesis constitutes a unique study as all previous studies with respect to CCLSP considered heuristic approaches.

Preliminary results obtained show that our algorithm is not as effective as commercial optimization software with regard to the solution time. This is partly due to incorporation of special techniques developed over the years to improve the performance of the commercial optimization software, namely CPLEX. Additional valid inequalities may help reduce the solution time by shrinking the feasibility set, hence cutting off the number of candidate solutions. One such inequality is provided as an example to show how solution times can be reduced drastically. Further study to obtain new valid inequalities is warranted. In addition, Benders decomposition acceleration techniques, such as generating pareto optimal cuts, increasing the density of the cuts, etc. may be analyzed.

Some practical-size problems prove challenging for CPLEX, mainly in terms of memory requirements, in that CPLEX terminates without even finding a feasible solution. By exchanging the burden on memory requirement with the time of going through several iterations, the solution method proposed in this study may be useful in obtaining at least a feasible solution, if not the optimal one.

In conclusion, with several possibilities for improvement in sight, our algorithm remains a promising approach to solving large-scale CCLSP.

REFERENCES

Books

Taskin, Z.C., 2010. Benders decomposition. *Wiley Encyclopedia of Operations Research and Management Science*. Wiley.

Periodicals

Aggarwal, A. and Park, J.K., 1993. Improved algorithms for economic lot sizing. *Operations Research*. **41** (3), pp. 549-571.

Arkin, E., Joneja, D. and Roundy, R., 1989. Computational complexity of uncapacitated multi-echelon production planning problems, *Operations Research Letters*. **8** (2), pp. 61-66.

Atkins, D. and Iyogun, P., 1988. A heuristic with lower bound performance guarantee for the multi-product lot-size problem. *IIE Transactions*. **20** (4), pp. 369–373.

Baker, K.R., Dixon, P., Magazine, M.J. and Silver, E.A., 1978. An algorithm for the dynamic lot-size problem with time-varying production capacity constraints. *Management Science*. **24**, pp. 1710-1720.

Barany, I., Van Roy, T.J. and Wolsey, L.A., 1984. Strong formulations for multi-item capacitated lot sizing. *Management Science*. **30** (10), pp. 1255-1261.

Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*. **4**, pp.238-252.

Billington, P.J., McClain, J.O. and Thomas, L.J. 1983. Mathematical programming approaches to capacity-constrained MRP systems: review, formulation and problem reduction. *Management Science*. **29** (10), pp. 1126-1141.

Bitran, G.R. and Yanasse, H.H., 1982. Computational complexity of the capacitated lot-size problem. *Management Science*. **28** (10), pp. 1174-1185.

Boctor, F. F., Laporte, G. and Renaud, J., 2004. Models and algorithms for the dynamic-demand joint replenishment problem. *International Journal of Production Research*. **42** (13), pp. 2667-2678.

Brahimi, N., Dauzere-Peres, S., Najid, N.M. and Nordli, A., 2006. Single item lot sizing problems. *European Journal of Operational Research*. **168** (1), pp. 1-16.

Cattrysse, D., Salomon, M., Kuik, R. and Van Wassenhove, L.N., 1993. A dual ascent and column generation heuristic for the discrete lotsizing and scheduling problem with setup times. *Management Science*. **39** (4), pp. 477-486.

Chen, H.D., Hearn, D. and Lee, C.Y., 1994. A new dynamic programming algorithm for the single item capacitated dynamic lot size model. *Journal of Global Optimization*. **4**, pp. 285-300.

Chen, W.H. and Thizy, J.M., 1990. Analysis of relaxations for the multi-item capacitated lot-sizing problem. *Annals of Operations Research*. **26** (1), pp. 29-72.

Chung, C., Flynn, J. and Lin, C.M., 1994. An effective algorithm for the capacitated single item lot size problem. *European Journal of Operational Research*. **75**, pp. 427-440.

De Matteis, J.J. and Mendoze, A.G., 1968. An economic lot sizing technique. *IBM System Journal*. **7**, pp. 30-46.

Diaby, M., Bahl, H.C., Karwan, M.H. and Zionts, S., 1992. A Lagrangian relaxation approach for very-large-scale capacitated lot-sizing. *Management Science*. **38** (9), pp. 1329-1340.

Dixon, P.S. and Silver, E.A., 1981. A heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing problem. *Journal of Operations Management*. **2** (1), pp. 23-39.

Dogramaci, A., Panayiotopoulos, J.C. and Adam, N.R., 1981. The dynamic lot-sizing problem for multiple items under limited capacity. *AIIE Transactions*. **13** (4), pp. 294-303.

Eisenhut, P.S., 1975. A dynamic lot sizing algorithm with capacity constraints. *AIIE Transactions*. **7** (2), pp. 170-176.

Eppen, G.D. and Martin, R.K., 1987. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research*. **35** (6), pp. 832-848.

Erenguc, S.S., 1988. Multi-product dynamic lot-sizing model with coordinated replenishments. *Naval Research Logistics*. **35** (1), pp. 1-22.

Erenguc, S.S. and Mercan, H.M., 1990. A multifamily dynamic lot-sizing model with coordinated replenishments. *Naval Research Logistics*. **37** (4), pp. 539-558.

Erlenkotter, D., 1978. A dual-based procedure for uncapacitated facility location. *Operations Research*. **26** (6), pp. 992-1009.

Evans, J.R., 1985. An efficient implementation of the Wagner-Whitin algorithm for dynamic lot-sizing. *Journal of Operations Management*. **5** (2), pp. 229-235.

Federgruen, A. and Tzur, M., 1991. A simple forward algorithm to solve general dynamic lot sizing models with n periods in O(n log n) or O(n) time. *Management Science*. **37** (8), pp. 909-925.

Federgruen, A. and Tzur, M., 1994. The joint replenishment problem with the time-varying costs and demands: Efficient, asymptotic and ε-optimal solutions. *Operations Research*. **42** (6), pp. 1067–1086.

Federgruen A., Meissner, J. and Tzur, M., 2007. Progressive interval heuristics for multi-item capacitated lot-sizing problems. *Operations Research*. **55** (3), pp. 490-502.

Florian, M. and Klein, M., 1971. Deterministic production planning with concave costs and capacity constraints. *Management Science*. **18** (1), pp. 12-20.

Florian, M., Lenstra, J.K., Rinnooy Kan, A.H.G., 1980. Deterministic production planning: algorithms and complexity. *Management Science*. **26**, pp. 669-679.

Fogarty, D. and Barringer, R., 1987. Joint order release decisions under dependent demand. *Production and Inventory Management*. **28** (1), pp. 55-61.

Gorham, T., 1968. Dynamic order quantities. *Production and Inventory Management*. **20**, pp. 75-81.

Gunther, H.O., 1987. Planning lot sizes and capacity requirements in a single stage production system. *European Journal of Operational Research*. **31** (2), pp. 223-231.

Iyogun, P., 1991. Heuristic methods for the multi-product dynamic lot size problem. *Journal of the Operational Research Society*. **42** (10), pp. 889-894.

Joneja, D., 1990. The joint replenishment problem: new heuristics and worst case performance bounds. *Operations Research*. **38** (4), pp. 711-723.

Kalymon, B.A., 1972. A decomposition algorithm for arborescence inventory systems. *Operations Research*. **20** (4), pp. 860-874.

Kao, E.P.C., 1979. A multi-product dynamic lot-size model with individual and joint setup costs. *Operations Research*. **27** (2), pp. 279-289.

Karimi, B., Fatemi Ghomi, S.M.T. and Wilson, J.M., 2003. The capacitated lot sizing problem: a review of models and algorithms. *Omega*. **31** (5), pp. 365-378.

Karni, R. and Roll, Y., 1982. A heuristic algorithm for the multi-item lot-sizing problem with capacity constraints. *AIIE Transactions*. **14** (4), pp. 249-256.

Kirca, O., 1995. A primal-dual algorithm for the dynamic lotsizing problem with joint set-up costs. *Naval Research Logistics*. **42** (5), pp. 791-806.

Lambert, A. and Luss, H., 1982. Production planning with time-dependent capacity bounds. *European Journal of Operational Research*. **9**, pp. 275-280.

Lambrecht, M.R. and Vanderveken, H., 1979. Heuristic procedures for the single operation, multi-item loading problem. *AIIE Transactions*. **11** (4), pp. 319-326.

Leung, J.M.Y., Magnanti, T.L. and Vachani, R., 1989. Facets and algorithms for capacitated lot sizing. *Mathematical Programming*. **45**, pp. 331-359.

Maes, J. and Van Wassenhove, L.N., 1986. A simple heuristic for the multi item single level capacitated lotsizing problem. *Operations Research Letters*. **4** (6), pp. 265-273.

Maes, J. and Van Wassenhove, L.N., 1988. Multi-item single-level capacitated dynamic lot-sizing heuristics: a general review. *The Journal of the Operational Research Society*. **39** (11), pp. 991-1004.

Millar, H.H. and Yang, M., 1994. Lagrangian heuristics for the capacitated multi-item lot-sizing problem with backordering. *International Journal of Production Economics*. **34** (1), pp. 1-15.

Robinson, E.P. and Gao, L.L., 1996. A dual ascent procedure for multiproduct dynamic demand coordinated replenishment with backlogging. *Management Science*. **42** (11), pp. 1556-1564.

Robinson, P., Narayanan, A. and Sahin, F., 2009. Coordinated deterministic dynamic demand lot-sizing problem: a review of models and algorithms. *Omega*. **37** (1), pp. 3-15.

Robinson, E.P.Jr. and Lawrence, F.B., 2004. Coordinated capacitated lot-sizing problem with dynamic demand: a Lagrangian heuristic. Decision Sciences. **35** (1), pp. 25-54.

Shaw, D.X. and Wagelmans, A.P., 1998. An algorithm for single-item capacitated economic lot sizing with piecewise linear production costs and general holding costs. *Management Science*. **44** (6), pp. 831-838.

Selen, W.J. and Heuts, R.M., 1989. A modified priority index for Gunther's lot-sizing heuristic under capacitated single stage production. *European Journal of Operational Research*. **41** (2), pp. 181-185.

Silver, E.A. and Meal, H.C., 1973. A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. *Production and Inventory Management*. **14**, pp. 64-74.

Silver, E.A., 1979. Coordinated replenishment of items under time-varying demand: dynamic programming formulation. *Naval Research Logistics Quarterly*. **26** (1), pp. 141-151.

Silver, E.A. and Kelle, P., 1988. More on 'Joint order release decisions under dependent demand'. *Production and Inventory Management*. **29** (2), pp. 71–72.

Ter Haseborg, F., 1982. On the optimality of joint ordering policies in a multi-product dynamic lot size model with individual and joint set-up costs. *European Journal of Operational Research*. **9** (1), pp. 47-55.

Thizy, J.M. and Van Wassenhove, L.N., 1985. Lagrangian relaxation for the multi-item capacitated lot-sizing problem: a heuristic implementation. *IIE Transactions*. **17** (4), pp. 308-313.

Trigeiro, W.W., 1987. A dual-cost heuristic for the capacitated lot sizing problem. *IIE Transactions*. **19** (1), pp. 67-72.

Trigeiro, W.W., 1989. A simple heuristic for lot sizing with setup times. *Decision Sciences*. **20** (2), pp. 294-303.

Trigeiro, W.W., Thomas, L.J. and McClain, J.O., 1989. Capacitated lot sizing with setup times. *Management Science*. **35** (3), pp. 353-366.

Van Hoesel, C.P.M. and Wagelmans, A.P.M., 1996. An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities. *Management Science*. **42**, pp. 142-150.

Veinott, A.F.Jr., 1969. Minimum concave-cost solution of Leontief substitution models of multi-facility inventory systems. *Operations Research*. **17** (2), pp. 262-291.

Wagelmans, A., Van Hoesel, S. and Kolen, A., 1992. Economic lot sizing: an O(n log n) algorithm that runs in linear time in the Wagner-Whitin case. *Operations Research*. **4** (S1), pp. S145-S156.

Wagner, H.M. and Whitin, T.M., 1958. Dynamic version of the economic lot size model. *Management Science*. **5** (1), pp. 89-96.

Zangwill, W.I., 1966. A deterministic multiproduct, multi-facility production and inventory model. *Operations Research*. **14** (3), pp. 486-507.

Other Publications

Altay, N., 2001. A cross decomposition algorithm for the capacitated dynamic demand coordinated replenishment problem. *Ph.D. dissertation*. Texas: Texas A&M University.

Lawrence, F.B., (1999). A dual solution procedure for the capacitated dynamic demand coordinated replenishment problem. *Ph.D. dissertation*, Texas: Texas A&M University.

Rizk, N. and Martel, A., 2001. Supply chain flow planning methods: a review of the lot-sizing literature. *Working paper*, DT-2001-AM-1, Université Laval (Canada).

Gao, L.L. and Robinson, E.P., 2004. A Lagrangian/dual-ascent based heuristic for the capacitated coordinated replenishment with dynamic demand and backlogging. *Working paper*, Hofstra University, Long Island, NY.