AN INTEGRATED CUTTING STOCK AND LOT SIZING PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYŞE BURCU ŞANAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

SEPTEMBER 2021

Approval of the thesis:

AN INTEGRATED CUTTING STOCK AND LOT SIZING PROBLEM

submitted by **AYŞE BURCU ŞANAN** in partial fulfillment of the requirements for the degree of **Master of Science** i**n Industrial Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Esra Karasakal
Head of the Department, **Industrial Engineering** ________________

Prof. Dr. Meral Azizoğlu
Supervisor, **Industrial Engineering, METU** ________________

Prof. Dr. Murat Köksalan
Co-Supervisor, **Ross School of Management, University of Michigan** ________________

Examining Committee Members:

Assist. Prof. Dr. Sakine Batun
Industrial Engineering, METU ________________

Prof. Dr. Meral Azizoğlu
Industrial Engineering, METU ________________

Prof. Dr. Murat Köksalan
Ross School of Management, University of Michigan ________________

Prof. Dr. Ferda Can Çetinkaya
Industrial Engineering, Çankaya University ________________

Assist. Prof. Dr. Yasemin Limon
Business Administration, Bilkent University ________________

Date: 06.09.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Ayşe Burcu Şanan

Signature :

ABSTRACT

AN INTEGRATED CUTTING STOCK AND LOT SIZING PROBLEM

Şanan, Ayşe Burcu
Master of Science, Industrial Engineering
Supervisor: Prof. Dr. Meral Azizoğlu
Co-Supervisor: Prof. Dr. Murat Köksalan

September 2021, 61 pages

In this thesis, we consider an integrated two dimensional cutting stock and lot sizing problem arising in an aircraft manufacturing plant. The items are to be cut from steel panels of identical size to satisfy all periodic demands over a specified planning horizon.

Two objectives, minimizing the number of panels cut and total inventory carrying cost of the items, are defined and all nondominated objective vectors with respect to the defined objectives are generated. To generate each nondominated objective vector, we propose a mixed integer linear programming model whose efficiency is improved by optimality properties and bounding mechanisms. We propose a decomposition-based heuristic algorithm to solve the instances having the smallest number of panels.

The results of our experiments based on real data and data taken from the literature have revealed that the instances with few items can be solved for up to 14 periods and the instances with more items can be solved for up to 7 periods, within our termination limit of two hours.

Keywords: Two Dimensional Cutting Stock Problems, Lot Sizing Problems, Integrated Problems, Multiobjective Programming

ÖZ

BÜTÜNLEŞİK STOK KESME VE PARTİ BÜYÜKLÜĞÜ BELİRLEME PROBLEMİ

Şanan, Ayşe Burcu
Yüksek Lisans, Endüstri Mühendisliği
Tez Yöneticisi: Prof. Dr. Meral Azizoğlu
Ortak Tez Yöneticisi: Prof. Dr. Murat Köksalan

Eylül 2021, 61 sayfa

Bu çalışmada, bir hava aracı imalat fabrikasında bütünleşik stok kesme ve parti büyüklüğü belirleme problemi ele alınmıştır. Belirlenen planlama periyodu içinde parçalar çelik panellerden kesilmekte ve talep karşılanmaktadır.

İki amaç, kesilen panel sayısını enazlama ve toplam envanter taşıma maliyetini enazlama, tanımlanmış ve tanımlanan amaçlara göre tüm baskın amaç vektörleri yaratılmıştır. Her bir baskın amaç vektörünün yaratılması için tamsayılı doğrusal programlama modeli önerilmiştir. Modelin verimliliği optimal çözüm özellikleri ve sınırlama mekanizmaları ile artırılmıştır. En az panel sayılı problemleri çözmek için ayrıştırma tabanlı sezgisel algoritma önerilmiştir.

Gerçek veri ve literatürden alınan veri kullanılarak yapılan deneylerin sonuçları iki saatlik durma limitimiz içinde parça sayısı az olan problemlerin 14 periyoda, parça sayısı çok olan problemlerin ise 7 periyoda kadar çözülebildiğini göstermiştir.

Anahtar Kelimeler: İki Boyutlu Stok Kesme Problemleri, Parti Büyüklüğü Belirleme Problemleri, Bütünleşik Problemler, Çok Kriterli Programlama

To my beloved family

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude towards my supervisors Prof. Dr. Meral Azizoğlu and Prof. Dr. Murat Köksalan for their valuable guidance and support in this study. I am so grateful to Meral Azizoğlu as she is always kind and helpful. She always encouraged me to do all the hard work of this study. Whenever I stumbled, she was there to give me a hand. Beyond everything, she has been a great companion in this journey. I am so lucky to have the chance of studying with Murat Köksalan as he enables me to think from a different point of view in every conversation we have. With his wisdom and open-minded thinking, he brought light to our study whenever we needed guidance.

I would like to express my deepest feelings to my family. Each and every member of my dear big family supported me in every way they could. My dear mother Fatma Ergen, father Ahmet Ergen, sister Gözde Nur Ergen and brother Arif Emre Ergen witnessed everything I had been through studying this thesis. My dear mother-in-law Nazife Şanan, father-in-law Ali Şanan and sister-in-law Tuğçe Şanan were there for me when I needed them most. My dear husband Yunus Şanan lived through the difficulty of facing all challenges and the happiness of celebrating all accomplishments together with me. They all expressed their endless love and support in every step of this journey.

I would like to thank all my friends from school and all my colleagues for their support and belief in me to go through all the challenges of this study. I would like to thank my unit chief in my workplace for his valuable support in the moments when I needed it most.

Lastly, I would like to thank TÜBİTAK for financial support in the 2210/A National Graduate Scholarship program.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

The cutting stock problem and the lot sizing problem have been studied by many researchers for several decades owing to their practical importance. The majority of the past research has considered those two problems separately due to their individual computational complexity. Recent researchers have recognized that integrating the problems leads to improved product volumes and reduced production costs. Moreover, the newly developed advanced computing technologies could handle higher complexities, hence offering a challenge to studying the integrated problems.

In this thesis, we study an integrated two dimensional cutting stock and lot sizing problem. We consider a given planning horizon where the periodic demand for the small rectangular items should be obtained from larger rectangular blocks of identical size. We assume two-stage guillotine cuts, where the first stage is horizontal. The problem is to find the number of rectangular items to be cut in each period. Two objectives of our concern are to minimize the number of panels (cutting stock problem related) and to minimize the total inventory holding cost of the small items (lot size problem related).

The two dimensional cutting stock and lot sizing problem has many application areas cited in the literature including but not limited to copper, furniture, glass and fiberglass industries. Our particular interest in the problem is from an application to an aircraft company located in Ankara, Turkey, to manage its steel cutting operations. The steel cutting operations in the plant are performed using a guillotine cutting machine that is shown in Figure 1.1.

Figure 1.1. The Guillotine Cutting Machine

The items having specified demand on each day of the week have to be cut from identical big steel plates. Figure 1.2 and Figure 1.3 depict the rectangular plates that are ready to be cut by the guillotine cutting machine and that are waiting in the panel stock area for future use, respectively.

Figure 1.2. The Panels at the Shop Floor

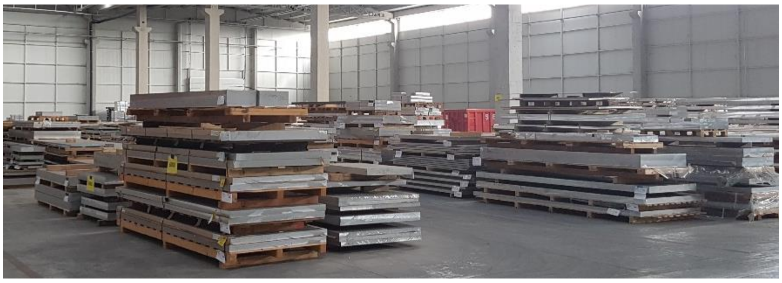

Figure 1.3. The Panels at the Panel Stock Area

The number of plates to be used in each period defines the total raw material cost and raw material inventory holding amounts at the stock area that should be minimized.

The small steel items have irregular shapes as shown in Figure 1.4.

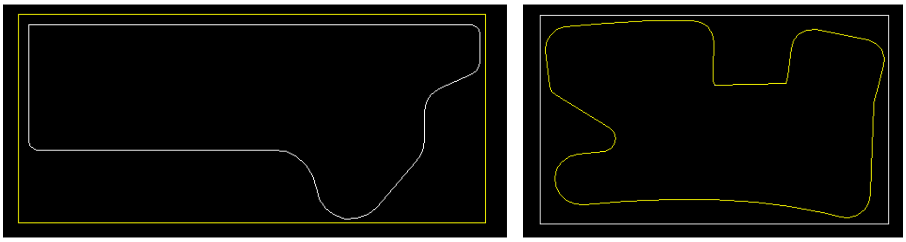

Figure 1.4. The Items to Be Cut

The cutting operations at the plant involve two stages. First, the smallest rectangular shape covering the entire irregular item is cut by the guillotine cutting machine and then a precise shape is obtained using a programmable rotating cutter. The items have rigid orientations such that their widths and their lengths should fit those of the panels.

The irregular-shaped items are used as components in the final products of the company, hence their requirements (number of units in each day) are projected from the target production levels of the final products. As the items are cut for the final products, any delay in their cutting times would delay the promised delivery times of the final products; hence the items should be cut no later than their required times. Moreover, their inventory carryings over long periods are undesirable, as some of them are fragile and some occupy space till their required times.

The production managers of the aircraft company want to see a set of solutions that demonstrates meaningful trade-offs between raw material costs of the panels and inventory carrying costs of the items. To help the production managers, we generate all nondominated objective vectors taking these two objectives into consideration. From this set, they can make a final choice from the presented cutting plans using their preferences. To generate the set of nondominated objective vectors, we develop several mathematical models along with several properties of the best solutions.

To the best of our knowledge, there is no reported research that proposes a multi criteria approach to the integrated two dimensional cutting stock and lot sizing problem. We hope that our study opens a new research avenue in this area.

The rest of this thesis is organized as follows. In Chapter 2, we review the related research. Chapter 3 defines our problem and presents the mathematical models. In Chapter 4, we discuss the details of our solution approach that generates all nondominated objective vectors. We discuss the results of our experiments based on the real instances and instances from the literature in Chapter 5. In Chapter 6, we give the main conclusions and discuss the future research directions.

CHAPTER 2

LITERATURE REVIEW

The literature review focuses on the most closely related studies to our multi criteria two dimensional cutting stock and lot sizing problem. In Section 2.1, we review single criterion two dimensional cutting stock problems. In Section 2.2, we study multi criteria cutting stock problems. Section 2.3 reviews integrated cutting stock and lot sizing problems.

2.1 Single Criterion Two Dimensional Cutting Stock Problems

A comprehensive review of single criterion one-dimensional cutting stock was given in Delorme et al. (2016). They examined the reported mathematical formulations, exact algorithms, heuristic procedures using a new data set. Their results showed that branch-and-price algorithms are the most effective class of exact algorithms and pseudo-polynomial formulations perform superior.

Gilmore and Gomory (1965) introduced the Two Dimensional Cutting Stock Problem (2DCSP) along with *k*-stage and multiple stock size extensions. They proposed a mathematical model and a column generation technique to get the optimal solutions. Their mathematical model and column generation technique became an important basis of the developing research.

Extensive research on two dimensional cutting stock problem exists; some noteworthy of which are due to Cintra et al. (2008), Furini et al. (2012), Silva et al. (2010), Furini and Malaguti (2013), and Ayasandır and Azizoğlu (2021). Cintra et al. (2008) developed heuristic algorithms using the linear programming relaxations of Gilmore and Gomory's (1965) mathematical formulation for the two-stage

2DCSP. Later on, Furini et al. (2012) proposed a heuristic algorithm using the column generation idea for the two-stage 2DCSP and showed that the algorithm is superior to that of Cintra et al. (2008). Pseudo-polynomial mathematical models for the two and three stages 2DCSP were developed by Silva et al. (2010). Furini and Malaguti (2013) extended Silva et al. (2010)'s pseudo-polynomial model and Gilmore and Gomory's (1965) column generation ideas to the two-stage 2DCSP with multiple stock size problem and proposed three mathematical programming models. Their objective was to minimize the total area of the panels. Recently, Ayasandır and Azizoğlu (2021) extended the most efficient formulation of Furini and Malaguti (2013) to the total profit objective. They also proposed a new model along with some dominance properties and showed that the new model outperforms Furini and Malaguti (2013)'s model.

2.2 Multi Criteria Cutting Stock Problems

We are aware of too few published research on the multi-criteria cutting stock problem; all of which are discussed below.

The multi-criteria studies by Cui and Yang (2010) and Filho et al. (2018) were for one dimensional cutting stock problems. Cui and Yang (2010) considered three objectives: the total panel cost, the profit from the leftovers (the unused length of a panel is leftover once it is longer than a threshold) and the profit from the leftovers coming from past cutting operations. They proposed a two-stage heuristic algorithm: the first stage is a linear program that cuts the major part of the item demand whereas the second stage is a sequential heuristic that cuts the remaining item demand. Filho et al. (2018) studied two objectives: the number of times a cutting pattern is used and the number of different cutting patterns. They aimed to generate all nondominated objective vectors and presented four procedures: the weighted sum method, the Chebyshev's metric, the ε-Constraint method and an improved version of Chebyshev's metric.

De Armas et al. (2011) and Mellouli et al. (2019) studied multi-criteria two dimensional cutting stock problems. de Armas et al. (2011) considered two objectives: the total profit and the number of cuts. To generate the set of nondominated objective vectors, they proposed a nondominated sorting genetic algorithm, strength pareto evolutionary algorithm and indicator-based evolutionary algorithm. They showed the superiority of their algorithms over the simple heuristic procedures. Mellouli et al. (2019) considered the material waste and the setup costs as two objectives. They used a genetic algorithm combined with a linear programming approach to find the set of all nondominated objective vectors. The experimental results showed that their genetic algorithm produces a near optimal set of the nondominated objective vectors.

Gonzalez et al. (2016) studied a three-dimensional cutting stock problem with two objectives: the total volume and weight of the items placed. They applied evolutionary algorithms and developed a multiple-level filling heuristic to obtain all nondominated objective vectors. Their experimental results confirmed that the two objectives are highly conflicting.

2.3 Integrated Cutting Stock and Lot Sizing Problems

Melega et al. (2018) reviewed the literature on the integrated cutting stock and lot sizing problems. They classified the problems in two categories using a mathematical model. In the first category, the integration between multiple periods is satisfied with inventory holding whereas in the second category input and output relationships between the production levels are used for the integration.

All studies on the integrated cutting stock and lot sizing problems considered a single criterion, usually a weighted combination of the lot sizing and cutting stock related objectives. The only exception is Campello et al. (2020) who considered a one-dimensional cutting stock problem with two objectives.

We first review the single criterion integrated cutting stock and lot sizing studies in chronological order and then discuss the multi criteria study.

Farley (1988) was the first author who addressed the integrated cutting-stock and lot-sizing problem. He considered the clothing industry and proposed integer and quadratic programming models so as to minimize the total cutting, sewing and storing costs.

Hendry et al. (1996) proposed a two-stage solution approach for the integrated cutting stock and lot sizing problem in the copper industry. In the first stage, the cutting stock problem was solved heuristically and in the second stage, the lot sizing problem was solved using an integer programming model.

Nonås and Thorstenson (2000, 2008) studied the integrated two dimensional cutting stock and lot sizing problem arising in a Norwegian truck company. They assumed irregular shapes and stochastic demand and aimed to minimize the total raw material cost and setup cost. Nonås and Thorstenson (2000) presented a column generation procedure to solve small-sized problem instances. Nonås and Thorstenson (2008) improved the column generation procedure of Nonås and Thorstenson (2000) using the sequential heuristic by Haessler (1971) to solve larger-sized problem instances.

Poltroniere et al. (2008) studied an integrated one dimensional cutting stock and lot-sizing problem arising in paper industry. They aimed to minimize the sum of inventory costs, setup costs, material waste costs and final item inventory costs. They developed an integer programming model and two heuristic procedures. The first heuristic algorithm utilizes a lagrangian relaxation idea and initially solves the lot sizing problem. The second heuristic solves the cutting stock problem first and then uses the solution to find the lot sizes.

Gramani and França (2006) studied an integrated two dimensional cutting stock and lot sizing problem in a wooden industry. They aimed to minimize the total setup costs, number of plates and inventory carrying costs of items, and proposed a network flow-based solution approach. Gramani et al. (2009) extended the model in

Gramani and Franca (2006) so as to include the production and inventory holding costs of the final products. They proposed a lagrangian relaxation-based heuristic approach where the subproblems require an exponential effort.

Silva et al. (2014) studied an integrated two dimensional cutting stock and lot sizing problem in the furniture industry. They aimed to minimize the total material, waste and storage costs. They considered the leftovers for their potential use in the subsequent periods of the planning horizon that might help to reduce the total waste. They proposed two integer programming models by extending the previously reported models in the literature. They showed that the models do not dominate each other, hence can be used together.

Leao et al. (2017) studied the integrated one dimensional cutting stock and lot sizing problem in the paper industry. They developed three mathematical models: pattern-oriented, period decomposition and machine decomposition. To solve linear programming relaxations of the models, a column generation technique is used. To solve the machine decomposition model, they proposed a heuristic that uses a column generation method along with an adaptive neighborhood search. They observed that the decomposed pattern-oriented model results in near optimal solutions for the data set from the literature and the machine decomposition heuristic gives better results for the real data.

Vanzela et al. (2017) studied the integrated two dimensional cutting stock and lot sizing problem arising in the Brazilian furniture industry. They proposed a mathematical model that minimizes the raw material waste, production and inventory costs and includes the safety stock level and saw capacity constraints. They proposed a column generation technique-based heuristic approach to solve large-sized real instances. They showed the superiority of their approaches --in terms of improving planning operations of the company-- over the approaches that make the cutting stock and lot sizing decisions separately.

Campello et al. (2020) presented a multi criteria study for the one dimensional integrated cutting stock and lot sizing problems arising in the paper industry. Their two objectives were minimizing total production costs, inventory costs of paper rolls and setup costs of machines and minimizing total material waste and inventory costs of items. They proposed two solution approaches: the weighting approach that minimizes the weighted sum of the objectives and an ε-constraint method where lot sizing related objective is minimized and cutting stock related objective is added as a constraint. To solve the instances of larger sizes, heuristic approaches were used in place of the exact model solutions. The computational experiments revealed that the lot sizing and cutting stock related objectives are conflicting.

In this study, we consider multi criteria approach to the integrated two dimensional cutting stock and lot sizing problem. The most closely related study to ours is that of Campello et al. (2020) that proposes a multi criteria approach to the integrated one dimensional cutting stock and lot sizing problem.

CHAPTER 3

PROBLEM DEFINITION AND MATHEMATICAL MODELS

We consider a two dimensional cutting stock problem with a single panel type and multiple periods. There are n item types to be cut from several copies of the single panel type that is defined by its length L and width W. Item j, $j \in \{1, \dots, n\}$ is characterized by its length l_j and width w_j. We assume that the items are sorted in their nonincreasing order of widths, i.e., $w_1 \geq w_2 \geq \cdots \geq w_n$.

The cuts are guillotine type and orientations of the items on the panels are important, i.e., the width and length of the item should be consistent with those of the panel.

The following figure depicts the placement of the items on a particular panel.

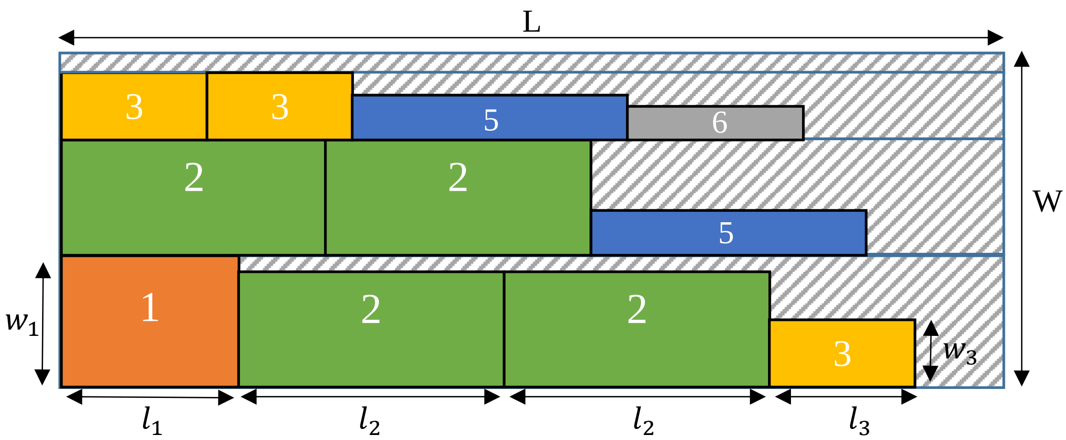

Figure 3.1. Placement of the Items on a Panel

Figure 3.1 shows a two-stage guillotine cut where the first stage is horizontal. Each horizontal cut defines a level and the width of the level is defined by the first item, which is, by construction, the largest-width item assigned to that level. Note that we define horizontal levels because the first stage guillotine cut is horizontal. Vertical levels would be defined if the first stage of the guillotine cuts was vertical.

The shaded area of the panel shows the waste resulting from the placements of items on the panel.

We say an item initializes a level if it is the first assigned item to that level. In Figure 3.1, item 1, item 2 and item 3 initialize levels 1, 2 and 3, respectively. The other items assigned to a level are referred to as additional items. In Figure 3.1, item 2 and item 3 are the additional items assigned to level 1; item 2 and item 5 are the additional items assigned to level 2; item 3, item 5 and item 6 are the additional items assigned to level 3.

There are T time periods whose cutting assignments should be planned together. Item j has a demand of d_{jt} units in period t. All demand of period t should be satisfied either from the cuts (productions) of the previous periods, i.e., from inventory or from the cuts of period t. The cost of holding a single unit inventory of item j is h_j money units. No backlogging or lost sales is allowed.

Our aim is to define a cutting plan for each panel used at each period t with a perspective of minimizing the following two measures over the planning horizon.

1. Total inventory holding cost – TIC
2. Total number of panels used, i.e., total cost of panels – TP

We proposed two mixed integer linear programming models each of which is modified from the model by Ayasandır and Azizoğlu (2021) for the single period, single objective, multiple panel types two dimensional cutting stock problem. We consider additional decisions to capture the inventory carrying amounts between two consecutive periods. We next explain these mixed integer linear programming (MILP) models.

3.1 MILP Model I

We define $D = \sum_{j=1}^{n} \sum_{t=1}^{T} d_{jt}$ as the total demand over all items and periods. At least one item should be put to each level of a panel, hence there are at most D levels over all panels.

In period t, item j can be cut from levels 1 through α_{jt} where $\alpha_{jt} = \sum_{s=1}^{j} \sum_{r=t}^{T} d_{sr}$ and $\alpha_{0t} = 0$.

The item type initializing level i in period t is defined by β_{it} where $\beta_{it} = min\ \{r{:}\, 1 \leq r \leq n, \alpha_{rt} \geq i\}$. The item types in [$\beta_{it}$,n] can be cut in period t, using level k.

The binary decision variables are defined as:

$$q_{kt} \quad = \begin{cases} 1, & if\ panel\ k\ is\ used\ in\ period\ t\ \forall\ k \in \{1, \dots, D\}, t \in \{1, \dots, T\} \\ 0, & otherwise \end{cases}$$

$$y_{it} \quad = \begin{cases} 1, & if\ item\ i\ initializes\ a\ level\ in\ period\ t\ \forall\ i \in \{1, \dots, D\}, \\ & \qquad t \in \{1, \dots, T\} \\ 0, & otherwise \end{cases}$$

$$z_{kit} \quad = \begin{cases} 1, & if\ level\ i\ is\ allocated\ to\ panel\ k\ in\ period\ t\ \forall\ \ i \in \{k, \dots, D\}, \\ & \qquad k \in \{1, \dots, D-1\}, t \in \{1, \dots, T\} \\ 0, & otherwise \end{cases}$$

The quantity-based decision variables are as stated below:

x_{ijt} = Number of additional items of type j put in level i in period t $\forall\ i \in \{1, \dots, D-1\}, j \in \{\beta_{it}, \dots, n\}, t \in \{1, \dots, T\}$

I_{jt} = Amount of item j inventory carried from period t to $t+1$. $\forall\ j\ \in \{1, \dots, n\}, t \in \{1, \dots, T\}$

I_{jt} variables are related to inventory planning decisions while all others are for cutting stock plans.

The constraint set is defined below.

1. Inventory - Cut Amount Balance

The amount that initializes a level together with the amount cut further at that level plus inventory carried from the previous period is equal to the amount demanded in the current period plus the inventory carried to the next period.

$$\sum_{i=1}^{\alpha_{jt}} x_{ijt} + \sum_{i=\alpha_{j-1,t}+1}^{\alpha_{jt}} y_{it} + I_{j,t-1} - I_{jt} = d_{jt} \quad \forall\, j \in \{1, \dots, n\}, t \in \{1, \dots, T\} \tag{1}$$

2. Length Capacity

For each period, the sum of the lengths of the items cut from any level should not exceed the length of the panel.

$$\sum_{j=\beta_{it}}^{n} x_{ijt} * l_j \leq \left(L - l_{\beta_{it}}\right) * y_{it} \quad \forall\, i \in \{1, \dots, D-1\}, t \in \{1, \dots, T\} \tag{2}$$

3. Width Capacity

The sum of the widths of the items that initialize the levels of the panel should not exceed the width of the panel.

$$\sum_{i=k+1}^{D} z_{kit} * \ w_{\beta_{it}} \leq \left(W - w_{\beta_{kt}}\right) * q_{kt} \quad \forall\, k \in \{1, \dots, D-1\}, t \in \{1, \dots, T\} \tag{3}$$

4. Level-Panel Relation

A defined level should either initialize a panel or be assigned to a panel.

$$\sum_{k=1}^{i-1} z_{kit} + q_{it} = y_{it} \qquad \forall\, i \in \{1, \dots, D\}, t \in \{1, \dots, T\} \tag{4}$$

5. Binary Variables

$$z_{kit} \in \{1,0\} \ \forall \ k \in \{1, \dots, D\}, i \in \{1, \dots, D\}, t \in \{1, \dots, T\} \tag{5}$$

$$y_{it} \in \{1,0\} \ \forall \ i \in \{1, \dots, D\}, t \in \{1, \dots, T\} \tag{6}$$

$$q_{kt} \in \{1,0\} \ \forall \ k \in \{1, \dots, D\}, t \in \{1, \dots, T\} \tag{7}$$

6. Nonnegativity and Integrality

$$x_{ijt} \geq 0 \ and \ integer \ \forall \ i \in \{1, \dots, D\}, j \in \{1, \dots, n\}, t \in \{1, \dots, T\} \tag{8}$$

$$I_{jt} \geq 0 \ \forall \ j \in \{1, \dots, n\}, t \in \{1, \dots, T\} \tag{9}$$

The performance criteria are defined as:

1. Minimize Total Inventory Cost, TIC

$$Min \sum_{j=1}^{n} \sum\nolimits_{t=1}^{T} h_j * I_{jt}$$

2. Minimize Total Number of Panels Cut, TP

$$Min \sum_{k=1}^{D} \sum\nolimits_{t=1}^{T} q_{kt}$$

We hereafter refer to the constraint sets (1) through (9) as $x \in X_{M1}$ and refer to Model I as:

$$Min \ TIC$$

$$Min \ TP$$

$$s.t. \ \ x \in X_{M1}$$

3.2 MILP Model II

Ayasandır and Azizoğlu (2021) noted that once the total demand over all items (D) is high, the model with decision variables defined on the total demand value --like Model I-- may become hard to solve. Recognizing this fact, they proposed an alternative model that defines the decision variables on the number of available panels. We extended their alternative model so as to include the inventory-related decisions over multiple periods.

We let P be an upper bound on the number of panels cut and let UBL_{jt} be an upper bound on the number of levels for item j in period t.

Later, we will discuss the way P is defined in our implementation.

Note that $\sum_{r=t}^{T} d_{jr}$ is the maximum amount of item j cut in period t, as no backlogs are allowed and all demand should be met. From each panel, up to $\left\lfloor \frac{W}{w_j} \right\rfloor$ units of item j can be cut and over P panels $P * \left\lfloor \frac{W}{w_j} \right\rfloor$ units can be cut. Using these expressions, we define UBL_{jt} as follows:

$$UBL_{jt} = Min \left\{ \sum_{r=t}^{T} d_{jr} , P * \left\lfloor \frac{W}{w_j} \right\rfloor \right\} \quad j \in \{1, \dots, n\}, \quad t \in \{1, \dots, T\}$$

We define the decision variables as:

$$q_{kt} \quad = \begin{cases} 1, & if\ panel\ k\ is\ used\ in\ period\ t\ \forall\ k \in \{1, \dots, P\}, t \in \{1, \dots, T\} \\ 0, & otherwise \end{cases}$$

$$z_{imkt} \quad = \begin{cases} 1, & if\ item\ i\ initializes\ level\ m\ of\ panel\ k\ in\ period\ t \\ & \forall\ i \in \{1, \dots, n\}, m \in \{1, \dots, UBL_{it}\}, k \in \{1, \dots, P\}, t \in \{1, \dots, T\} \\ 0, & otherwise \end{cases}$$

The quantity-based variables are defined as:

x_{jimkt} = Number of additional items of type j cut at level m initialized by item i in panel k in period t. $\forall\ i, j \in \{1, \dots, n\}\ and\ i \leq j\ and\ \left\lfloor \frac{L-l_i}{l_j} \right\rfloor \geq 1,\ m \in \{1, \dots, UBL_{it}\},\ k \in \{1, \dots, P\},\ t \in \{1, \dots, T\}$

$w_j \geq w_i\ as\ i < j$, hence item j can be assigned to level i. $\left\lfloor \frac{L-l_i}{l_j} \right\rfloor \geq 1$is used to force an integer solution.

I_{jt} = Inventory of item j inventory from period t to $t+1$. $\forall\ t \in \{1, \dots, T\}, j \in \{1, \dots, n\}$

The constraint set is given below.

1. Inventory – Cut Amount Balance

Amount of item j cut in period t plus amount coming from period $t-1$ should be equal to the amount going to period t.

$$\sum_{i=1}^{n} \sum_{m=1}^{UBL_{it}} \sum_{k=1}^{P} x_{jimkt} + \sum_{m=1}^{UBL_{jt}} \sum_{k=1}^{P} z_{jmkt} + I_{j,t-1} - I_{jt} = d_{jt} \qquad (10)$$

$$\forall\ j \in \{1, \dots, n\},\ t \in \{1, \dots, T\}$$

2. Length Capacity

The sum of the lengths of the items assigned to any level cannot exceed the length of the panel for each period.

$$\sum_{j=i}^{n} l_j * x_{jimkt} + l_i * z_{imkt} \leq L * z_{imkt} \qquad (11)$$

$$\forall\ i \in \{1, \dots, n\},\ m \in \{1, \dots, UBL_{it}\},\ k \in \{1, \dots, P\},\ t \in \{1, \dots, T\}$$

3. Width Capacity

The sum of the widths of the items that initialize the levels of the panel cannot exceed the width of the panel.

$$\sum_{i=1}^{n} \sum_{m=1}^{UBL_{it}} w_i * z_{imkt} \le W * q_{kt} \quad \forall\ k \in \{1, \dots, P\},\ t \in \{1, \dots, T\} \tag{12}$$

4. Binary Variables

$$z_{imkt} \in \{1,0\}\ \forall\ i \in \{1, \dots, n\}, m \in \{1, \dots, UBL_{it}\}, k \in \{1, \dots, P\}, t \in \{1, \dots, T\} \tag{13}$$

$$q_{kt} \in \{1,0\}\ \forall\ k \in \{1, \dots, P\}, t \in \{1, \dots, T\} \tag{14}$$

5. Nonnegativity and Integrality

$$x_{jimkt} \ge 0\ and\ integer\ \forall\ j \in \{1, \dots, n\}, i \in \{1, \dots, n\}, m \in \{1, \dots, UBL_{it}\}, k \in \{1, \dots, P\}, t \in \{1, \dots, T\} \tag{15}$$

$$I_{jt} \ge 0\ \forall\ j \in \{1, \dots, n\}, t \in \{1, \dots, T\} \tag{16}$$

Note that the value of P is important in terms of the number of decision variables. The smaller the value of P is, the fewer is the number of decision variables.

The performance criteria are defined as:

1. Minimize Total Inventory Cost, TIC

$$Min \sum_{j=1}^{n} \sum_{t=1}^{T} h_j * I_{jt}$$

2. Minimize Total Number of Panels Cut, TP

$$Min \sum_{k=1}^{P} \sum_{t=1}^{T} q_{kt}$$

We hereafter refer to constraint sets (10) through (16) as $x \in X_{M2}$ and refer to Model II as:

$$Min\ TIC$$
$$Min\ TP$$
$$s.t.\quad x \in X_{M2}$$

CHAPTER 4

SOLUTION APPROACH

Our aim is to generate all nondominated objective vectors and an efficient solution corresponding to each nondominated objective vector. We basically use the constraint set of Model II, i.e., $x \in X_{M2}$, in generating the nondominated objective vectors.

The chapter is organized as follows: In Section 4.1, we define nondominated objective vectors and extreme nondominated objective vectors along with their generations. Section 4.2 discusses the generation of all nondominated objective vectors. In Section 4.3, we develop some mechanisms to reduce the computational burden of the model used to generate all nondominated objective vectors.

4.1 Nondominated Objective Vectors

A solution (cutting and lot sizing plan) S in $x \in X_{M2}$ is said to be efficient if there does not exist a solution S' in $x \in X_{M2}$ such that $TIC(S') \leq TIC(S)$ and $TP(S') \leq TP(S)$ with strict inequality holding at least once. The objective vector $(TIC(S), TP(S))$ corresponding to the efficient solution S is said to be a nondominated objective vector.

An efficient solution S is said to be extreme efficient if it has the best possible value of one objective. The associated nondominated objective vector is said to be an extreme nondominated objective vector.

We now discuss the generation of the two extreme nondominated objective vectors.

i. Extreme Nondominated Objective Vector having smallest TIC Value

Consider the following Mixed Integer Linear Program (MILP):

$$Min\ TIC$$
$$s.t.\ \ x \in X_{M2}$$

TIC^*, i.e., the optimal TIC value, is zero corresponding to the solution that produces all $\sum_{j=1}^{n} d_{jt}$ units in period t. Hence, the problem decomposes into two t single period models with no inventory balance variables and associated constraints.

Now we define problem $P(t)$ for each t, $t \in \{1, \dots, T\}$ as:

$$Min\ P(t)$$
$$s.t.\ \ \ x \in X'_{M2(t)}$$

Se $x \in X'_{M2(t)}$ is defined as follows:

$$\sum_{i=1}^{n}\sum_{m=1}^{UBL_{it}}\sum_{k=1}^{P} x_{jimkt} + \sum_{m=1}^{UBL_{jt}}\sum_{k=1}^{P} z_{jmkt} = d_{jt} \tag{10'}$$
$$\forall\, j \in \{1, \dots, n\},\ t \in \{1, \dots, T\}$$

$$\sum_{j=i}^{n} l_j * x_{jimkt} + l_i * z_{imkt} \le L * z_{imkt} \tag{11}$$
$$\forall\, i \in \{1, \dots, n\},\ m \in \{1, \dots, UBL_{it}\},\ k \in \{1, \dots,\ P^*\},\ t \in \{1, \dots, T\}$$

$$\sum_{i=1}^{n}\sum_{m=1}^{UBL_{it}} w_i * z_{imkt} \le W * q_{kt} \quad \forall\ k \in \{1, \dots,\ P\},\ t \in \{1, \dots, T\} \tag{12}$$

$$z_{imkt} \in \{1,0\}\ \forall\, i \in \{1, \dots, n\}, \qquad m \in \{1, \dots, UBL_{it}\}, \tag{13}$$
$$k \in \{1, \dots, P\}, t \in \{1, \dots, T\}$$

$$q_{kt} \in \{1,0\}\ \forall\, k \in \{1, \dots, P\}, t \in \{1, \dots, T\} \tag{14}$$

$$x_{jimkt} \ge 0\ and\ integer\ \forall\, j \in \{1, \dots, n\}, i \in \{1, \dots, n\}, m \tag{15}$$
$$\in \{1, \dots, UBL_{it}\}, k \in \{1, \dots, P\}, t \in \{1, \dots, T\}$$

P is an upper bound on the number of panels. Let $P^*(t)$ be the optimal solution to the $P(t)$ model.

The minimum number of panels with zero TIC value is $\sum_{t=1}^{T} P^*(t)$.

Hence, $(TIC, TP) = (0, \sum_{t=1}^{T} P^*(t))$ is an extreme nondominated objective vector. We let $P_{max} = \sum_{t=1}^{T} P^*(t)$ and $I_{min} = 0$.

ii. Extreme Nondominated Objective Vector Having Smallest TP Value

Consider the following problem.

$$Min\, TP$$
$$s.t.\ \ x \in X_{M2}$$

Let TP^* is the minimum TP value. TP^* is the smallest TP value of all efficient solutions.

An optimal solution of the above model that delivers TP^* may not be efficient as there might exist another solution with a smaller TIC value. Such a solution can be found through the following model:

$$Min\, TIC$$
$$s.t.\ \ x \in X_{M2(t)}$$
$$\sum_{k=1}^{P} \sum_{t=1}^{T} q_{kt} = TP^*$$

TP^*can be found in a simpler way by aggregating the demand of each item and solving a single period model.

To sum up, we use the following two-step procedure to find the extreme nondominated vector with the smallest TP value.

Procedure 1 - Generating an Extreme Nondominated Objective Vector with the Smallest $\boldsymbol{TP}$ value

Step 1. Find TP^* via the following problem for T = 1 and

$d_{j1} = \sum_{t=1}^{T} d_{jt} \ \forall j \ \in \{1, \dots, n\}$.

$$Min\, TP$$
$$s.t.\ x \in X_{M1}$$

Step 2. Solve the below problem:

$$Min\, TIC$$

$$s.t.\ \ x \in X_{M2}$$

$$\sum_{k=1}^{TP^*} \sum_{t=1}^{T} q_{kt} = TP^*$$

Let TIC^* be the optimal solution, $P_{min} = TP^*$ and $I_{max} = TIC^*$.

We depict the extreme nondominated efficient solutions in Figure 4.1.

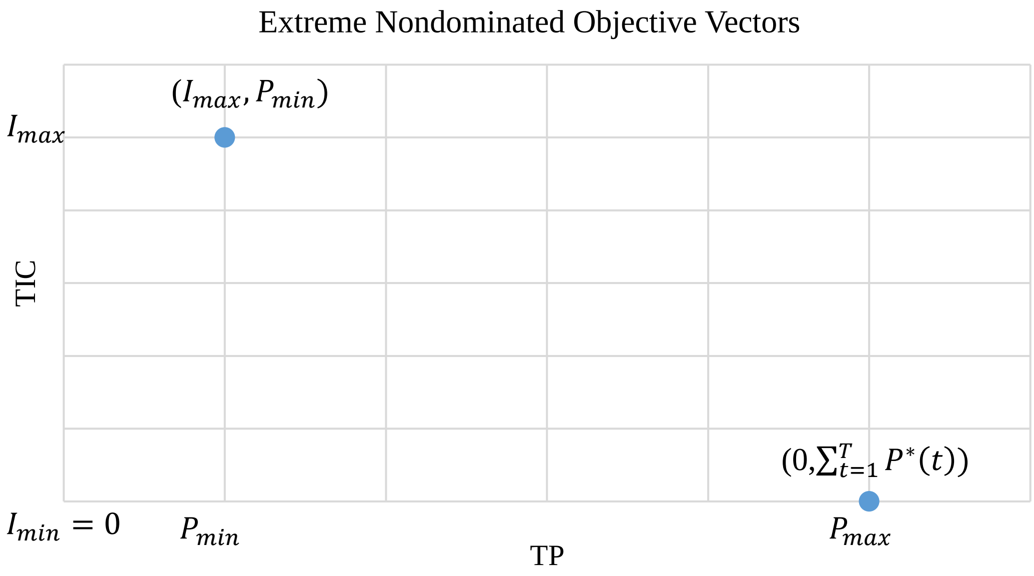

Figure 4.1. Extreme Nondominated Objective Vectors

All other nondominated objective vectors have TIC in $[I_{min} + 1,\ I_{max} - 1]$ and TP in $[P_{min} + 1,\ P_{max} - 1]$ when all parameters, thereby TIC and TP values, are integers.

We illustrate extreme nondominated objective vectors and their generation using a problem instance with 6 items and 7 periods. The panel dimensions are set as $W = 1250$ and $L = 2500$. Item data are given in Table 4.1.

Table 4.1 Item Data for Example Instance

j	1	2	3	4	5	6
w_j	620	590	530	400	130	110
l_j	670	810	2280	1200	440	115
h_j	4	4	10	3	1	3
d_{j1}	1	0	1	0	0	2
d_{j2}	0	0	0	0	4	0
d_{j3}	2	0	0	1	0	0
d_{j4}	0	0	1	3	1	0
d_{j5}	0	2	0	0	0	2
d_{j6}	0	2	0	2	1	0
d_{j7}	0	0	1	0	0	1

Using the procedures discussed so far, we find the extreme nondominated objective vectors using as ($I_{max} = 98, P_{min} = 4$) and ($I_{min} = 0,\ P_{max} = 8$) and show them in Figure 4.2.

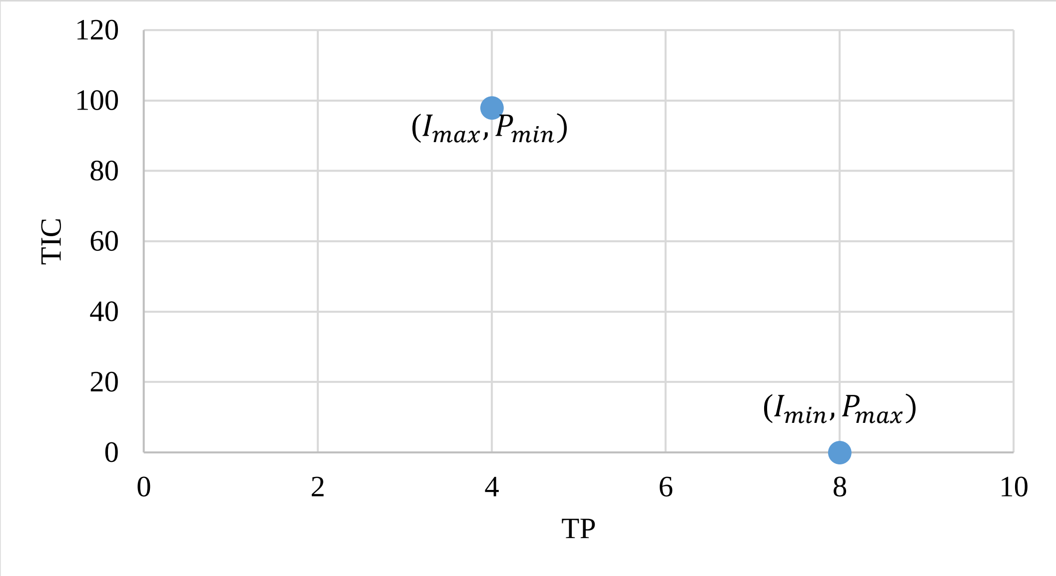

Figure 4.2. Extreme Nondominated Objective Vectors of Example Instance

Note that P_{min} is the number of panels for cutting all items in a single period. Hence, it is an upper bound on the number of panels used per period. Recognizing this fact, we use P_{min} in place of P (like using $k \in \{1, \dots, P_{min}\}, \sum_{k=1}^{P_{min}} \sum_{t=1}^{T} q_{kt}$) in the model that returns the extreme solution with zero total inventory.

The models that deliver the first extreme solution with zero inventory and the model that delivers the P_{min} value of the second extreme solution are two dimensional cutting stock models with guillotine cuts. Two dimensional cutting stock problems with guillotine cuts are strongly NP-hard (see Furini and Malaguti (2013)), so is our problem of generating extreme, thereby all, nondominated objective vectors.

4.2 Generating All Nondominated Objective Vectors

We first present a property of the nondominated objective vectors (Theorem 4.1) and use it to generate their set.

Theorem 4.1. There exists a nondominated objective vector corresponding to each number of panels between P_{min} and P_{max}.

Proof. Among all nondominated objective vectors $P_{max} = \sum_{t=1}^{T} P^*(t)$ is the minimum number of panels while carrying no inventory and P_{min} is the minimum number of panels while carrying maximum inventory.

Assume $P_m^*(t)$ is the number of panels used in period t in the P_{min} solution, i.e., $P_{min} = \sum_{t=1}^{T} P_m^*(t)$. For any period t, for which $P_m^*(t) < P^*(t)$ putting an extra panel to $P_m^*(t)$ reduces the total inventory amount. Extra panels can be put till the number of panels reaches P_{max} and each addition reduces the total inventory amount. Hence, there exists a nondominated objective vector for each value of P between P_{min} and P_{max}. #

Using the result of the above theorem, we define the following problem to generate an efficient solution with P panels:

$$Min\ TIC$$
$$s.t.\ \ x \in X_{M2(t)}$$
$$\sum_{k=1}^{P_{min}} \sum_{t=1}^{T} q_{kt} = P$$

The following procedure is developed for generating all nondominated objective vectors:

Procedure 2: Generating All Nondominated Objective Vectors

Step 1. Solve the following problem $P(t)$ for each single period t:

$$Min\ P(t)$$
$$s.t.\ \ x \in X_{M1(t)}$$

Let $P^*(t)$ be the optimal solution for single period t and $P_{max} = \sum_{t=1}^{T} P^*(t)$. We solve the below problem using P_{max} as an upper bound on the number of panels.

Model II requires an upper bound on the number of the panels. To find such an upper bound, we solve Model I and find P_{max}.

$(0,\ P_{max})$ is the first nondominated objective vector.

Solve below problem for $T = 1$ with $d_{j1} = \sum_{t=1}^{T} d_{jt}\ \ \forall j\ \in \{1, \dots, n\}$:

$$Min\ TP$$
$$s.t.\ \ x \in X_{M2(t)}$$
$$where$$

Let TP^* be the optimal solution.

$P_{min} = TP^*$

$r = 1$

$P = P_{max}$

Step 2. Let $r = r + 1$ and $P = P - 1$.

Solve the below problem:

$$Min\ TIC$$
$$s.t.\ \ x \in X_{M2}$$
$$\sum_{k=1}^{P_{min}} \sum_{t=1}^{T} q_{kt} = P$$

Let TIC^* be the optimal solution and (TIC^*, P) is the r^{th} nondominated objective vector.

Step 3. If $P \geq P_{min}$ then go to Step 2.

Else, all $r = P_{max} - P_{min} + 1$ nondominated objective vectors are generated.

We hereafter refer to the model solved in Step 2 as the $Min\ TIC|P$ model.

We next develop some mechanisms for the efficient solution of the *Min TIC*|*P*.

Recall the example instance given in Section 4.1 in Table 4.1. The instance has $P_{min} = 4$ and $P_{max} = 8$, hence the exact number of nondominated objective vectors is $r = P_{max} - P_{min} + 1 = 8 - 4 + 1 = 5$.

Figure 4.3 shows these 5 nondominated objective vectors of the example instance that are found via Procedure 2.

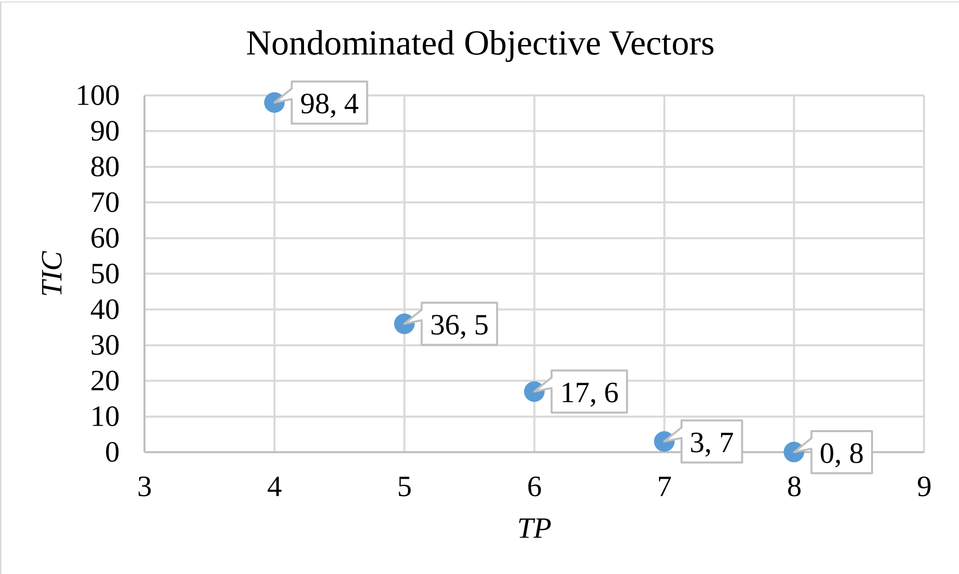

Figure 4.3. All Nondominated Objective Vectors for the Example Instance

Each nondominated objective vector gives an efficient solution, i.e., a cutting plan of each period. In Figure 4.4, we provide an output file explaining a cutting plan of the example instance for $P_{min} = 4$. The placement of the items on Panel 1 at the first period is shown in Figure 4.5.

```
CUTTING PLAN

Number of 1250 X 2500 mm panels used : 4

Panel 1 at t = 1
---------------------------------------
Level 1 : 1 - 1 - 1 - 6
Level 2 : 3 - 6


Panel 2 at t = 2
---------------------------------------
Level 1 : 2 - 4
Level 2 : 3
Level 3 : 5 - 5 - 5 - 5


Panel 3 at t = 4
---------------------------------------
Level 1 : 4 - 4
Level 2 : 4 - 4
Level 3 : 4 - 5


Panel 4 at t = 5
---------------------------------------
Level 1 : 2 - 2 - 2
Level 2 : 3 - 6
Level 3 : 5 - 6 - 6
```

Figure 4.4. Cutting Plan of the Example Instance for $P_{min} = 4$

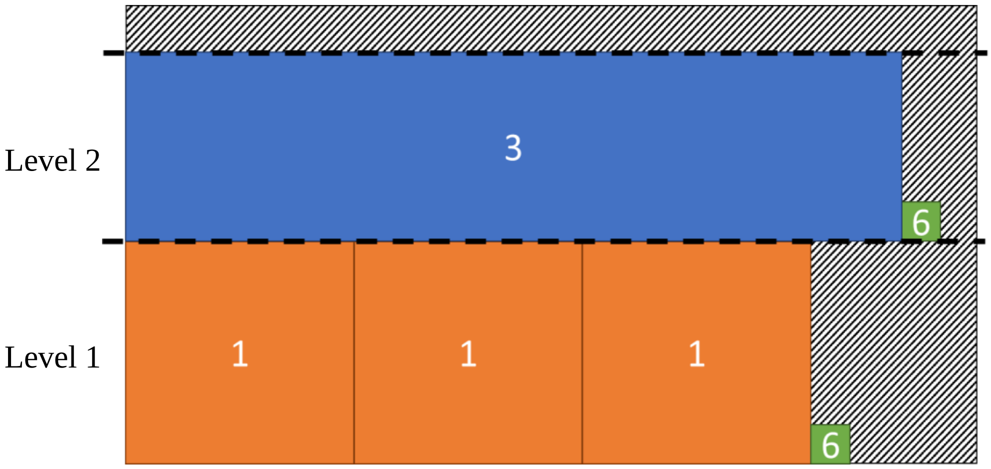

Figure 4.5. Cutting Scheme of Panel 1 at $t = 1$

4.3 Solving the $Min\ TIC|P$ Model

In this section, we develop some mechanisms to reduce the computational burden of the model used to generate all nondominated objective vectors.

In Subsections 4.3.1 and 4.3.2 we present upper and lower bounds on the number of panels of any efficient solution and discuss the use of the bounds. Subsection 4.3.3 presents constraints that use lower bounds on the total inventory cost value. Subsection 4.3.4 discusses some reductions for the panel-related variables and Subsection 4.3.5 introduces some valid inequalities to reduce the feasible solution space. In Subsection 4.3.6 we give the model that incorporates all the presented mechanisms. Finally, in Subsection 4.3.7 we present a decomposition-based algorithm to solve the *Min TIC*|*P*=P_{min} problem.

4.3.1 Upper Bounds on the TP Values

We derive several upper bounds on the number of panels to be used in period *t* for any nondominated objective vector. We use the best of those upper bounds to reduce the number of panel-related decision variables and constraints.

i. Upper Bound 1; $P_{max}(t,1)$

Recall that P_{min} is the maximum number of panels that could be used for any period t. The minimum number of panels that should be used in periods 1 through *t*-1 is $\left\lceil \frac{\sum_{r=1}^{t-1}\sum_{j=1}^{n} d_{jr}*l_j*w_j}{L*W} \right\rceil$.

Hence, $P_{max}(t,1) = P_{min} - \left\lceil \frac{\sum_{r=1}^{t-1}\sum_{j=1}^{n} d_{jr}*l_j*w_j}{L*W} \right\rceil$ is a valid upper bound on the number of panels used in period *t*.

ii. Upper Bound 2; $P_{max}(t,2)$

${P_{max}}_t$ is the minimum number of panels for a single period *t* with $d_{j1} = d_{jt}$.

$P_{max}(t,2) = \sum_{r=t}^{T} P_{max_r}$ is the number of panels with no inventory carrying for periods t through T, hence is a valid upper bound on the number of panels in period t.

iii. Upper Bound 3; $P_{max}(t,3)$

We let $TP^*(t)$ be the minimum number of panels for a single period t with $d_{j1} = \sum_{r=1}^{t-1} d_{jr}$. P is the exact number of panels to be used over T periods. At least $TP^*(t)$ of P panels should be used in the first $(t-1)$ periods. This leaves at most $P_{max}(t,3) = P - TP^*(t)$ panels for period t.

An overall upper bound on the number of panels used in period t is denoted as $P_{max}(t)$ and is found through the following relation:

$$P_{max}(t) = Min\{P_{max}(t,1), P_{max}(t,2), P_{max}(t,3)\}$$

In our models, in place of P, we use $P_{max}(t)$ at all appropriate places.

Moreover, we update the parameter UBL_{jt} as follows:

$$UBL_{jt} = Min\left\{\sum_{r=t}^{T} d_{jr}, P_{max}(\mathrm{t}) * \left\lfloor \frac{W}{w_j} \right\rfloor\right\}$$

4.3.2 Lower Bounds on the TP Values

We derive two lower bounds on the panels used in periods 1 through t for any nondominated objective vector. We define a constraint to enhance the efficiency of the model using the best of the lower bounds.

i. Lower Bound 1; $P_{min}(t,1)$

The minimum number of panels to be used in periods 1 through t is found by considering the total area required by the total demand of the periods and the total area available. The resulting expression is stated below:

$$P_{min}(t,1) = \left\lceil \frac{\sum_{r=1}^{t-1}\sum_{j=1}^{n} d_{jr} * l_j * w_j}{L * W} \right\rceil$$

ii. Lower Bound 2; $P_{min}(t,2)$

Recall that P_{max_t} is the minimum number of panels required to satisfy the demand of period t. In the first t periods, at least $P_{min}(t,2) = Max\{P_{max_r}\}\ where\ r \in \{1, \dots, t\}$ panels should be used.

We introduce the following constraint to Model II that uses the lower bounds.

$$\sum_{k=1}^{P_{max}(t)} \sum_{r=1}^{t} q_{kr} \geq P_{min}(t) \qquad \forall\ t \in \{1, \dots, T\} \tag{17}$$

where $P_{min}(t) = Max\{P_{min}(t,1), P_{min}(t,2)\}$

4.3.3 Bounds on the TIC Values

We included the following two constraints based on the total inventory cost.

$$TIC(P) \geq TIC(P-1) + 1 \tag{18}$$

$$TIC(P) \leq TIC(P_{min}) - P + P_{min} \tag{19}$$

where $TIC(P)$ is the optimal TIC value with P panels.

Constraint (18) uses the definition of the nondominated objective vectors and the integrality of the TIC values. Constraint (19) uses the fact that there are $(P - P_{min})$ nondominated objective vectors having number of panels in $[P_{min} + 1, P]$. Hence, the difference between TIC values of P_{min} and P solutions should be at least $P - P_{min}$.

As $TIC(P_{min})$ is not available, we find an upper bound for the TIC value of a solution having P_{min} panels using the below procedure:

Procedure 3: Finding an Upper Bound for the $TIC(P_{min})$

Step 1. Let t be the earliest period having positive demand.

Step 2. Select the item having the largest demand in period t.

Step 3. Assign the panel having the maximum number of the selected item to period t.

Update the demands of all items considering the contents of the assigned panel.

Step 4. Stop if all updated demand figures are zero (all demand is satisfied).

Else, go to Step 1.

4.3.4 Bounds on the Decision Variables

We derive a theorem that restricts the panel cuts only to the periods having positive demand.

Theorem 4.2. If $\sum_{j=1}^{n} d_{jt} = 0$ for period t, then $\sum_{k=1}^{P_{max}(t)} q_{kt} = 0$ for period t.

Proof. Assume there is a panel cut in period t with $\sum_{j=1}^{n} d_{jt} = 0$. Then the panel can be shifted to period $t+1$ without increasing the number of panels and while reducing the total inventory cost. Hence, a solution in which there is a panel cut cannot be efficient. #

Using the result of the above theorem, we only define decision variable q_{kt}, z_{imkt} and x_{jimkt} for period t, such that $\sum_{j=1}^{n} d_{jt} > 0$, however, keep I_{jt} variables for all t.

4.3.5 Additional Constraints for Panel Assignments

We first fix some q_{kt} values for the first period, $t = 1$, as follows;

$$q_{k1} = 1 \ \forall \ k \in \{1, \dots, P^*(1)\} \tag{20}$$

This is due to the fact that at least *P*(1)* panels should be used in the first period as no backlogging is allowed. For the panels P*(1) through $P_{max}(t)$ we use the following relation:

$$q_{k1} \geq q_{(k+1)1} \ \forall \ k \ \in \{P^*(1), \dots, P_{max}(t)\} \tag{21}$$

$$q_{kt} \geq q_{(k+1)t} \ \forall \ k \ \in \{1, \dots, P_{max}(t)\}, t \ \in \{2, \dots, T\} \tag{22}$$

Constraints sets (21) and (22) are used to eliminate the solutions that skip panel k but use panel $k + 1$. Constraint set (23) uses the same idea for the levels and eliminates the solutions of the same panel that skips level *m* but uses level *m*+1.

$$\begin{gathered} z_{imkt} \geq z_{i(m+1)kt} \\ \forall \ i \in \{1, \dots, n\}, m \in \{1, \dots, UBL_{it}\} \ \ k \in \{1, \dots, P_{max}(t)\}, t \in \{1, \dots, T\}, \\ \text{where} \sum_{j=1}^{n} d_{jt} > 0 \end{gathered} \tag{23}$$

4.3.6 Improved Min $\boldsymbol{TIC|P}$ Model

Our initial experiments have revealed that each elimination mechanism is helpful in reducing the complexity of the solutions. Hence, we use all constraints that support the elimination mechanisms to define our improved model. In this subsection, we give the improved *Min TIC|P* model for the sake of completeness.

$$Min \sum_{j=1}^{n}\sum_{t=1}^{T} h_j * I_{jt} \tag{24}$$

subject to

$$\sum_{i=1}^{n}\sum_{m=1}^{UBL_{it}}\sum_{k=1}^{P_{max}(t)} x_{jimkt} + \sum_{m=1}^{UBL_{jt}}\sum_{k=1}^{P_{max}(t)} z_{jmkt} + I_{j,t-1} - I_{jt} = d_{jt} \tag{25}$$

$$\forall\, j \in \{1, \dots, n\}, t \in \{1, \dots, T\}$$

$$\sum_{j=i}^{n} l_j * x_{jimkt} + l_i * z_{imkt} \le L * z_{imkt} \tag{26}$$

$$\forall\, i \in \{1, \dots, n\}, m \in \{1, \dots, UBL_{it}\}, k \in \{1, \dots, P_{max}(t)\}, t \in \{1, \dots, T\}$$

$$\sum_{i=1}^{n}\sum_{m=1}^{UBL_{it}} w_i * z_{imkt} \le W * q_{kt} \tag{27}$$

$$\forall\, k \in \{1, \dots, P_{max}(t)\}, t \in \{1, \dots, T\}$$

$$\sum_{k=1}^{P_{max}(t)}\sum_{r=1}^{t} q_{kr} \ge P_{min}(t) \qquad \forall\, t \in \{1, \dots, T\} \tag{28}$$

$$\sum_{j=1}^{n}\sum_{t=1}^{T} h_j * I_{jt} \ge TIC(P-1) + 1 \tag{29}$$

$$\sum_{j=1}^{n}\sum_{t=1}^{T} h_j * I_{jt} \le TIC(P_{min}) - P + P_{min} \tag{30}$$

$$q_{k1} = 1 \;\; \forall\, k \in \{1, \dots, P^*(1)\} \tag{31}$$

$$q_{k1} \ge q_{(k+1)1} \; \forall\, k \in \{P^*(1), \dots, P_{max}(t)\} \tag{32}$$

$$q_{kt} \ge q_{(k+1)t} \; \forall\, k \in \{1, \dots, P_{max}(t)\}, t \in \{2, \dots, T\} \tag{33}$$

$$z_{imkt} \ge z_{i(m+1)kt} \tag{34}$$

$$\forall\, i \in \{1, \dots, n\}, m \in \{1, \dots, UBL_{it}\}\;\; k \in \{1, \dots, P_{max}(t)\}, t \in \{1, \dots, T\}$$

$$q_{kt} \in \{1,0\} \;\; \forall\, t \in \{1,\dots,T\}, k \in \{1,\dots,P_{max}(t)\} \; where \sum_{j=1}^{n} d_{jt} > 0 \tag{35}$$

$$z_{imkt} \in \{1,0\} \; \forall\, i \in \{1,\dots,n\}, m \in \{1,\dots,UBL_{it}\}, k \in \{1,\dots,P_{max}(t)\}, \\ t \in \{1,\dots,T\} \; where \sum_{j=1}^{n} d_{jt} > 0 \tag{36}$$

$$x_{jimkt} \geq 0 \; and \; integer \; \forall\, j \in \{1,\dots,n\}, i \in \{1,\dots,n\}, m \in \{1,\dots,UBL_{it}\}, \\ k \in \{1,\dots,P_{max}(t)\}, t \in \{1,\dots,T\} \; where \sum_{j=1}^{n} d_{jt} > 0 \tag{37}$$

$$I_{jt} \geq 0 \; \forall\, j \in \{1,\dots,n\}, t \in \{1,\dots,T\} \tag{38}$$

4.3.7 Heuristic Algorithm for the $\boldsymbol{Min\, TIC|P = P_{min}}$ Problem

The results of our computational experiment have revealed that the most difficult instances are associated with the ones having the minimum number of panels. To solve those instances, we develop a decomposition-based heuristic algorithm. In doing so, we divide the problem into two subproblems each of which is defined as:

Subproblem 1. $Min\, TP + \varepsilon_T * TIC$

$s.t.\;\; x \in X_{M2}$ for the first $\lceil T/2 \rceil$ periods

Subproblem 2. $Min\, TP + \varepsilon_T * TIC$

$s.t.\;\; x \in X_{M2}$ for the last $T - \lceil T/2 \rceil$ periods

ε_T should be small enough that the number of panels should not increase even for the maximum reduction of total inventory holding cost value.

Hence $TP^* + \varepsilon_T * TIC_{max} \leq TP^* + 1 + \varepsilon_T * TIC_{max}$

where TIC_{max} is the maximum possible TIC value, i.e., total inventory found by the optimal solution of the $Min\, TIC|P = P_{min}$.

TIC_{min} is the minimum possible TIC value, i.e., $TIC_{min} = 0$.

This follows, $\varepsilon_T^* \leq \frac{1}{TIC_{max}}$.

We set $\varepsilon_T^* = \frac{1}{TIC_{max}+1}$.

We let (TIC_i^*, TP_i^*) be the optimal solution of subproblem i and take the (TIC, TP) values of the heuristic as $(TIC_1^* + TIC_2^*, TP_1^* + TP_2^*)$.

CHAPTER 5

COMPUTATIONAL EXPERIMENTS

In this chapter, we present the computational experiments designed to test the performances of our solution approaches. To generate the nondominated objective vector set through mathematical model optimization, we integrate CPLEX version 12.8.0 to the Java programming language. The heuristic algorithm for finding an extreme nondominated objective vector is coded in the Java programming language using Netbeans IDE 8.2. The experiments are conducted on Intel® Xeon® E-2246G CPU @ 3.6 GHz, 16.0 GB RAM Windows 10.

In Section 5.1, we present the features of our instances and data generation scheme. In Section 5.2, we discuss the results of our experiments in detail. We give a comparative analysis of the mathematical models and their improved versions.

5.1 The Instance Features and Data Generation

In our experiments, we use both real life data and data from the literature.

We take a total of 56 instances from the aircraft manufacturing plant, 40 of which have 7 periods and 16 of which have 14 periods. We define the features of the instances for 7 and 14 periods in Table 5.1 and Table 5.2, respectively. The tables report the number of item types, n, the total demand, D, $wmax$ ($wmin$) is the maximum (minimum) width over all items of the problem instance, lmax ($lmin$) is the maximum (minimum) length over the items in the problem instance. W and L are the width and length of the panel, respectively. The measurement units of all dimensions are millimeters.

In Table 5.1, instances 1 through 8 have originally 7 periods whereas instances 9 through 40 are formed by splitting one 14-period instance into two parts. The split instances are shown by the ‘*’ sign. Note that 16 T = 14 instances are split so as to obtain 32 T = 7 instances. We refer to the real data as SetR.

Table 5.1 Features of the Problem Instances, T=7, SetR

Instance	n	D	$wmax$	$wmin$	$lmax$	$lmin$	W	L
1	3	45	60	55	2785	1930	1220	3900
2	4	50	165	56	3048	66	1220	3660
3	5	32	508	102	3048	152	1220	3660
4	5	54	170	75	230	75	1220	2500
5	5	300	400	85	1200	110	1250	2500
6	7	61	279	89	2438	140	1220	3660
7	11	264	146	76	965	117	1220	3660
8	14	24	914	70	2235	152	1220	3660
9*	2	24	125	64	215	175	1220	3660
10*	2	36	125	64	215	175	1220	3660
11*	1	10	102	102	3048	3048	1220	3660
12*	2	12	1016	508	2540	1092	1220	3660
13*	2	29	76	51	432	114	1220	3660
14*	1	15	254	254	254	254	1220	3660
15*	1	15	55	55	2300	2300	1220	3900
16*	2	30	60	55	2325	2300	1220	3900
17*	1	94	64	64	175	175	1250	2500
18*	2	144	125	125	215	200	1250	2500
19*	1	72	110	110	120	120	1220	3660
20*	2	216	90	45	150	110	1220	3660
21*	2	24	191	114	241	114	1220	3660
22*	2	18	216	51	330	211	1220	3660
23*	2	16	590	590	810	810	1250	2500
24*	4	40	620	590	810	670	1250	2500
25*	1	9	64	64	89	89	1220	3660
26*	3	48	76	33	279	84	1220	3660
27*	4	56	620	590	810	670	1250	2500
28*	4	72	620	590	810	670	1250	2500
29*	3	120	89	70	311	127	1220	3660
30*	1	24	95	95	622	622	1220	3660
31*	2	14	89	53	1046	104	1220	3660
32*	4	36	254	53	2515	104	1220	3660
33*	5	42	610	89	711	191	1220	3660
34*	2	24	140	89	140	102	1220	3660
35*	7	61	1016	61	3048	2540	1220	3660
36*	2	23	229	97	3048	838	1220	3660
37*	2	51	152	89	330	216	1220	3660
38*	7	66	152	51	229	89	1220	3660
39*	6	186	585	45	2485	60	1250	2500
40*	5	252	410	40	1810	45	1250	2500

*Split instances

Table 5.2 Features of the Problem Instances, T=14, SetR

Instance	n	D	$wmax$	$wmin$	$lmax$	$lmin$	W	L
1	2	60	125	64	215	175	1220	3660
2	3	22	1016	102	3048	1092	1220	3660
3	3	44	254	51	432	114	1220	3660
4	3	45	60	55	2325	2300	1220	3900
5	3	238	125	64	215	175	1250	2500
6	3	288	110	45	150	110	1220	3660
7	4	42	216	51	330	114	1220	3660
8	4	56	620	590	810	670	1250	2500
9	4	57	76	53	279	84	1220	3660
10	4	128	620	590	810	670	1250	2500
11	4	144	95	70	622	127	1220	3660
12	6	50	254	53	2515	104	1220	3660
13	7	66	610	89	711	102	1220	3660
14	9	84	1016	61	3048	838	1220	3660
15	9	117	152	51	330	89	1220	3660
16	10	438	585	40	2485	45	1250	2500

We further conduct experiments on problem instances from the literature. We take 18 problem instances from the data set of Hifi and Roucairol (2001). The set had been used by many researchers including Furini and Malaguti (2013) and Ayasandır and Azizoğlu (2021).

We try each of the 18 instances for T values of 7 and 14 and for n values of 3 and 5, hence solve a total of 72 problem instances. We refer to the literature data set as SetL.

Table 5.3 gives the instance names (as referred to in the literature), number of items, D (total demand), $wmax, wmin, lmax, lmin, W$ and L values.

Table 5.3 Features of the Problem Instances Taken from the Literature, SetL

							$T=7$		$T=14$	
							$n=3$	$n=5$	$n=3$	$n=5$
Instance	$wmax$	$wmin$	$lmax$	$lmin$	W	L	D	D	D	D
A1	43	11	33	9	54	60	59	113	101	173
A2	42	14	33	12	54	72	62	116	103	191
A3	43	14	35	15	72	84	42	104	82	190
A4	43	11	33	9	63	108	65	99	95	175
A5	63	12	69	13	90	158	60	109	98	172
CHL2	31	9	31	11	49	74	52	106	89	192
CHL5	14	2	20	1	18	24	55	90	85	151
Hchl3s	65	13	54	15	88	152	52	120	91	182
Hchl4s	65	13	54	15	88	152	50	107	91	182
Hchl6s	101	38	109	35	219	303	53	110	89	179
Hchl8s	14	2	20	1	18	58	65	107	98	174
HH	65	13	54	18	88	152	55	103	93	176
Of1	36	4	55	9	36	84	49	116	78	183
Of2	27	4	47	13	36	84	41	101	76	148
Sts4	49	16	44	14	89	118	44	120	83	192
W	33	9	43	11	36	84	65	121	88	193
2s	35	7	31	9	63	48	45	112	79	176
3s	43	11	33	9	63	48	55	89	96	155

We generate the l_j values from a discrete uniform distribution (DU) between $lmin$ and $lmax$, w_j values from DU between $wmin$ and $wmax$. The demand of item j for period t, i.e., d_{jt} is generated from DU between 0 and 5. h_j values are taken from DU between 1 and 3 to be compatible with the values of our practical application.

5.2 Results of the Experiments

In this subsection, we first compare the performance of Model I and Model II (without any improvement mechanism) on SetR instances. Table 5.4 and Table 5.5 report the total CPU time of generating all nondominated objective vectors and their average and maximum CPU times for $T = 7$ and $T = 14$, respectively.

Table 5.4 Model I and Model II Performances, $T = 7$, SetR

		Model I			Model II		
Instance	**Number of Nondom. Vectors**	**Total CPU Time (Sec)**	**Avg. CPU Time (sec)**	**Max. CPU Time (sec)**	**Total CPU Time (Sec)**	**Avg. CPU Time (sec)**	**Max. CPU Time (sec)**
1	2	7.77	3.88	7.42	1.17	0.59	1.13
2	2	15.23	7.62	14.81	4.77	2.38	4.23
3	2	1.72	0.86	1.31	3.09	1.55	3.05
4	3	91.03	30.34	85.25	1.42	0.47	1.41
5	3	14572.25	4857.42	7200 (2)[1]	7583.06	2527.69	7200 (1)
6	3	887.95	295.98	502.31	13.02	4.34	7.09
7	5	28825.86	5765.17	7200 (4)	14617.72	2923.54	7200 (2)
8	2	4.06	2.03	3.83	2.92	1.46	2.63
9*	2	3.55	1.77	3.38	0.08	0.04	0.06
10*	3	12.81	4.27	8.86	0.78	0.26	0.59
11*	1	0	0	0	0.02	0.02	0.02
12*	2	0.03	0.02	0.02	0.19	0.09	0.14
13*	1	0.44	0.44	0.44	0.06	0.06	0.06
14*	1	0.03	0.03	0.03	0.02	0.02	0.02
15*	1	0	0	0	0	0	0
16*	1	0.20	0.20	0.20	0	0	0
17*	1	0.20	0.20	0.20	0	0	0
18*	1	3.03	3.03	3.03	0.11	0.11	0.11
19*	1	2.09	2.09	2.09	0.03	0.03	0.03
20*	2	1838.53	919.27	1834.83	0.16	0.08	0.14
21*	2	0.94	0.47	0.84	0.03	0.02	0.03
22*	2	2.00	1.00	1.83	0.16	0.08	0.09
23*	2	3.95	1.98	3.44	4.56	2.28	4.17
24*	2	18.45	9.23	17.17	23.56	11.78	21.25
25*	1	0	0	0	0	0	0
26*	3	67.91	22.64	48.67	1.94	0.65	1.77
27*	1	2.52	2.52	2.52	3.48	3.48	3.48
28*	3	355.23	118.41	303.80	326.94	108.98	192.14
29*	2	28.72	14.36	27.63	0.06	0.03	0.03
30*	1	0.42	0.42	0.42	0	0	0
31*	2	0.67	0.34	0.58	0.05	0.02	0.03
32*	4	13978.61	3494.65	7200 (1)	3.19	0.80	1.80
33*	1	0.83	0.83	0.83	0.63	0.63	0.63
34*	2	5.55	2.77	5.20	0.17	0.09	0.09
35*	2	116.11	58.05	114.53	6.16	3.08	5.67
36*	2	5.05	2.52	4.77	1.23	0.62	1.11
37*	2	10.89	5.45	9.45	0.05	0.02	0.03
38*	3	58.67	19.56	57.02	1.58	0.53	1.44
39*	4	21740.28	5435.07	7200 (2)	15105.20	3776.30	7200 (2)
40*	2	7212.41	3606.20	7200 (1)	176.02	88.01	162.69
Average	2	2247	617	977	947	237	550

[1]Numbers in parentheses is the number of nondominated solutions that remain unsolved in 2 hours.

The tables also include the number of nondominated objective vectors that could not be found in 7200 seconds. In finding the total, average and maximum CPU times, the CPU times of the unsolved instances are taken as 7200 seconds.

Table 5.5 Model I and Model II Performances, $T = 14$, SetR

		Model I			Model II		
Instance	**Number of Nondom. Vectors**	**Total CPU Time (Sec)**	**Avg. CPU Time (sec)**	**Max. CPU Time (sec)**	**Total CPU Time (Sec)**	**Avg. CPU Time (sec)**	**Max. CPU Time (sec)**
1	5	22312.14	4462.43	7200 (3)[1]	18.28	3.66	7.03
2	3	1.13	0.38	0.56	4.94	1.65	2.75
3	2	178.22	89.11	176.03	0.28	0.14	0.16
4	1	1.89	1.89	1.89	0.03	0.03	0.03
5	2	7231.25	3615.63	7200 (1)	25.64	12.82	23.77
6	3	8022.14	2674.05	7200 (1)	2.38	0.79	2.17
7	4	386.75	96.69	232.34	4.25	1.06	2.28
8	3	251.42	83.81	138.41	450.30	150.10	282.14
9	4	2924.16	731.04	1406.63	6.16	1.54	3.17
10	3	8721.84	2907.28	7200 (1)	5506.23	1835.41	4111.95
11	3	960.56	320.19	662.36	2.03	0.68	1.88
12	6	28875.61	4812.60	7200 (4)	40.16	6.69	13.48
13	3	10854.89	3618.30	7200 (1)	37.06	12.35	19.30
14	3	10833.23	3611.08	7200 (1)	7253.53	2417.84	7200 (1)
15	5	15903.08	3180.62	7200 (2)	43.56	8.71	20.88
16	6	41701.50	6950.25	7200 (5)	38354.44	6392.41	7200 (5)
Average	4	9947	2322	4215	3234	678	1181

[1]Numbers in parentheses is the number of nondominated solutions that remain unsolved in 2 hours.

As can be observed from the above tables, the number of dominated vectors increases as the number of periods increases. Note that on average there are 2 and 4 nondominated objective vectors when there are 7 and 14 periods, respectively. Moreover, the average CPU time, i.e., the average time to reach a nondominated objective vector, increases from 617 seconds to 2322 seconds for Model I and from 237 seconds to 678 seconds for Model II, as T increases from 7 to 14.

From Tables 5.4 and 5.5, we observe the significantly better performance of Model II over Model I. Note that for $T = 7$ when Model I is used, the average total and average CPU times are 2247 and 617 seconds, respectively. The respective average total and average CPU times reduce to 947 and 237 seconds when Model II is used.

Moreover, when $T = 7$ and $T = 14$, Model I cannot find 10 out of 82, and, 19 out of 56 nondominated objective vectors, respectively. On the other hand, Model II cannot find 5 and 6 nondominated objective vectors, for $T = 7$ and $T = 14$, respectively. Those results are due to the fact the decision variables of Model I are a function of the total demand, whereas Model II generates the decision variables based on the total number of panels used. In many practical applications and in our real case, the number of panels is significantly smaller than the total demand, hence, giving a nice challenge to the use of Model II.

After observing the superiority of Model II over Model I, we developed improvement mechanisms to further enhance the efficiency of Model II. Tables 5.6 and 5.7 report the performances of Model II and Improved Model II (Model II that uses all reduction mechanisms and bounding procedures) using SetR instances for 7 and 14 periods, respectively.

Table 5.6 Model II and Improved Model II Performances, $T = 7$, SetR

		Model II			Improved Model II		
Instance	**Number of Nondom. Vectors**	**Total CPU Time (Sec)**	**Avg. CPU Time (sec)**	**Max. CPU Time (sec)**	**Total CPU Time (Sec)**	**Avg. CPU Time (sec)**	**Max. CPU Time (sec)**
1	2	1.17	0.59	1.13	0.06	0.03	0.05
2	2	4.77	2.38	4.23	0.13	0.06	0.08
3	2	3.09	1.55	3.05	0.14	0.07	0.09
4	3	1.42	0.47	1.41	0.13	0.04	0.11
5	3	7583.06	2527.69	7200 (1)[1]	180.83	60.28	173.58
6	3	13.02	4.34	7.09	2.70	0.90	1.20
7	5	14617.72	2923.54	7200 (2)	1856.50	371.30	1808.53
8	2	2.92	1.46	2.63	0.20	0.10	0.19
9*	2	0.08	0.04	0.06	0.03	0.02	0.03
10*	3	0.78	0.26	0.59	0.09	0.03	0.05
11*	1	0.02	0.02	0.02	0	0	0
12*	2	0.19	0.09	0.14	0.06	0.03	0.03
13*	1	0.06	0.06	0.06	0	0	0
14*	1	0.02	0.02	0.02	0.05	0.05	0.05
15*	1	0	0	0	0	0	0
16*	1	0	0	0	0.03	0.03	0.03
17*	1	0	0	0	0	0	0
18*	1	0.11	0.11	0.11	0	0	0
19*	1	0.03	0.03	0.03	0	0	0
20*	2	0.16	0.08	0.14	0.02	0.01	0.02
21*	2	0.03	0.02	0.03	0.02	0.01	0.02
22*	2	0.16	0.08	0.09	0.03	0.02	0.03
23*	2	4.56	2.28	4.17	0.13	0.06	0.11
24*	2	23.56	11.78	21.25	0.14	0.07	0.08
25*	1	0	0	0	0	0	0
26*	3	1.94	0.65	1.77	0.08	0.03	0.05
27*	1	3.48	3.48	3.48	0.11	0.11	0.11
28*	3	326.94	108.98	192.14	9.89	3.30	4.72
29*	2	0.06	0.03	0.03	0	0	0
30*	1	0	0	0	0	0	0
31*	2	0.05	0.02	0.03	0.03	0.02	0.03
32*	4	3.19	0.80	1.80	0.14	0.04	0.05
33*	1	0.63	0.63	0.63	0.03	0.03	0.03
34*	2	0.17	0.09	0.09	0.03	0.02	0.02
35*	2	6.16	3.08	5.67	3.53	1.77	2.97
36*	2	1.23	0.62	1.11	0.02	0.01	0.02
37*	2	0.05	0.02	0.03	0.02	0.01	0.02
38*	3	1.58	0.53	1.44	0.11	0.04	0.09
39*	4	15105.20	3776.30	7200 (2)	118	29.5	96.59
40*	2	176.02	88.01	162.69	2.25	1.125	1.47
Average	2	947	237	550	54	12	52

[1]Numbers in parentheses is the number of nondominated solutions that remain unsolved in 2 hours.

Table 5.7 Model II and Improved Model II Performances, $T = 14$, SetR

		Model II			Improved Model II		
Instance	**Number of Nondom. Vectors**	**Total CPU Time (Sec)**	**Avg. CPU Time (sec)**	**Max. CPU Time (sec)**	**Total CPU Time (Sec)**	**Avg. CPU Time (sec)**	**Max. CPU Time (sec)**
1	5	18.28	3.66	7.03	0.77	0.15	0.34
2	3	4.94	1.65	2.75	0.13	0.04	0.08
3	2	0.28	0.14	0.16	0.02	0.01	0.02
4	1	0.03	0.03	0.03	0.09	0.09	0.09
5	2	25.64	12.82	23.77	0.98	0.49	0.92
6	3	2.38	0.79	2.17	0.08	0.03	0.06
7	4	4.25	1.06	2.28	0.48	0.12	0.44
8	3	450.30	150.10	282.14	1.86	0.62	0.78
9	4	6.16	1.54	3.17	0.14	0.04	0.06
10	3	5506.23	1835.41	4111.95	34.09	11.36	14.92
11	3	2.03	0.68	1.88	0.06	0.02	0.06
12	6	40.16	6.69	13.48	1.50	0.25	0.66
13	3	37.06	12.35	19.30	0.48	0.16	0.30
14	3	7253.53	2417.84	7200 (1)[1]	26.13	8.71	24.66
15	5	43.56	8.71	20.88	1.53	0.31	1.02
16	6	38354.44	6392.41	7200 (5)	16436.77	2739.46	7200 (2)
Average	4	3234	678	1181	1032	173	453

[1]Numbers in parentheses is the number of nondominated solutions that remain unsolved in 2 hours.

As can be observed from Tables 5.6 and 5.7, the performance of Model II improves significantly once the improvement mechanisms are used. This implies the effort spent to generate the mechanisms is much less than the CPU time reductions obtained through their use by the model. For almost all instances -with few exceptions having too small differences- the CPU time required by Improved Model II is less than that of required by Model II. The average CPU times for $T = 7$ and $T = 14$ reduce from 237 to 12 seconds and from 678 to 173 seconds, respectively. The number of unsolved nondominated objective vectors is 5 for Model II while Improved Model II finds all nondominated objective vectors when $T = 7$. When $T = 14$, the unsolved instances are 6 and 2 for Model II and Improved Model II, respectively.

We continue our experiments using the data from the literature, i.e., SetL. We solve the problem instances with Improved Model II, attributing to its superiority over

Model I and Model II. Tables 5.8 and 5.9 report the results of the experiments with $T = 7$ and $T = 14$, respectively.

Table 5.8 Improved Model II Performance, $T = 7$, SetL

	$n = 3$				$n = 5$			
Instance	**Number of Nondom. Vectors**	**Total CPU Time (Sec)**	**Avg. CPU Time (sec)**	**Max. CPU Time (sec)**	**Number of Nondom. Vectors**	**Total CPU Time (Sec)**	**Avg. CPU Time (sec)**	**Max. CPU Time (sec)**
A1	5	13.64	2.73	6.13	5	230.67	46.13	99.34
A2	4	35.77	8.94	25.64	5	12736.02	2547.20	7200 (1)[1]
A3	3	0.81	0.27	0.44	6	251.94	41.99	78.03
A4	5	28.73	5.75	9.00	5	74.28	14.86	40.95
A5	4	10.44	2.61	5.88	5	158.33	31.67	74.31
CHL2	5	8.86	1.77	3.55	5	167.72	33.54	119.34
CHL5	5	3.16	0.63	1.02	5	20.47	4.09	6.22
Hchl3s	3	13.56	4.52	12.94	4	79.42	19.86	62.23
Hchl4s	4	13.70	3.43	7.34	5	183.64	36.73	94.31
Hchl6s	4	3.41	0.85	1.58	4	34.28	8.57	21.11
Hchl8s	7	4.64	0.66	1.56	4	2.64	0.66	1.13
HH	3	3.42	1.14	1.81	5	229.95	45.99	170.81
Of1	3	7.92	2.64	3.77	7	108.00	15.43	36.36
Of2	5	3.17	0.63	1.39	6	179.61	29.93	100.09
Sts4	5	11.00	2.20	6.34	4	59.92	14.98	38.78
W	5	36.66	7.33	13.31	6	258.28	43.05	78.17
2s	2	6.38	3.19	5.91	4	183.13	45.78	117.75
3s	4	16.36	4.09	10.34	10	242.58	24.26	97.28
Average	4	12	3	7	5	844	167	469

[1]Numbers in parentheses is the number of nondominated solutions that remain unsolved in 2 hours.

Table 5.9 Improved Model II Performance, $T = 14$, SetL

Instance	$n = 3$ Number of Nondom. Vectors	Total CPU Time (Sec)	Avg. CPU Time (sec)	Max. CPU Time (sec)	$n = 5$ Number of Nondom. Vectors	Total CPU Time (Sec)	Avg. CPU Time (sec)	Max. CPU Time (sec)
A1	12	10658.95	888.25	7200 (1)	11	38566.27	3506.02	7200 (5)[1]
A2	11	5011.80	455.62	1116.56	9	44942.13	4993.57	7200 (6)
A3	8	219.25	27.41	81.66	5	1510.39	302.08	1040.80
A4	9	160.67	17.85	48.08	9	18345.39	2038.38	7200 (2)
A5	6	741.17	123.53	611.86	7	6299.38	899.91	5320.84
CHL2	8	1762.98	220.37	656.50	7	10109.05	1444.15	3933.86
CHL5	9	48.53	5.39	12.09	11	26229.19	2384.47	7200 (1)
Hchl3s	9	371.14	41.24	135.19	9	12255.89	1361.77	6195.52
Hchl4s	8	109.28	13.66	68.09	12	40627.78	3385.65	7200 (4)
Hchl6s	8	93.33	11.67	21.66	7	6187.81	883.97	5175.73
Hchl8s	8	33.81	4.23	10.47	7	669.61	95.66	142.89
HH	6	45.52	7.59	13.42	11	47924.64	4356.79	7200 (5)
Of1	9	407.42	45.27	153.73	8	31202.17	3900.27	7200 (4)
Of2	6	39.44	6.57	14.22	9	15035.69	1670.63	7200 (1)
Sts4	10	107.59	10.76	34.31	9	13493.09	1499.23	7200 (1)
W	9	2727.08	303.01	1197.28	14	32849.84	2346.42	7200 (4)
2s	10	234.06	23.41	87.45	10	41454.72	4145.47	7200 (5)
3s	11	418.70	38.06	72.44	11	32449.22	2949.93	7200 (2)
Average	9	1288	125	641	8	23342	2342	6012

[1]Numbers in parentheses is the number of nondominated solutions that remain unsolved in 2 hours.

As the number of items or number of periods increases the minimum and the maximum number of panel values in the nondominated objective vectors set increases; however the range of the minimum and the maximum number of panel values, thereby the number of nondominated objective vectors, may not be affected. So we do not expect any remarkable increases in the number of nondominated objective vectors with increases in n or T. The results in Tables 5.8 and 5.9 are in line with our expectations. The number of nondominated objective vectors increases slightly as the number of periods increases. Note that the respective average number of nondominated objective vectors are 4 and 5 for $n = 3$ and $n = 5$ when $T = 7$ and the respective average number of nondominated objective vectors are 9 and 8 for $n = 3$ and $n = 5$ when $T = 14$. On the other hand, we could not observe any relationship between the number of items and the number of the nondominated objective vectors. For example for instances A1, A2, A3, the number of nondominated objective vectors increases as the number of the items increases (from 5, 4, 3 to 5, 5, 6) when $T = 7$ and decreases (from 12, 11, 8 to 11, 9, 5) when $T = 14$.

Tables 5.8 and 5.9 indicate that as the number of items increases, the CPU times increase significantly. When $T = 7$, the respective average CPU times are 3 and 167 seconds for $n = 3$ and $n = 5$ and when $T = 14$, the respective average CPU times are 125 and 2342 seconds for $n = 3$ and $n = 5$. This is due to the fact that as n increases the number of binary variables, thereby the complexity of the mixed integer programs, increases exponentially. An increase in the complexity of the solutions also affects the number of unsolved instances. Note from Table 5.8 that for the problem with $n = 3$ and $T = 7$, all 76 nondominated objective vectors are solved very quickly with an average maximum CPU time of 7 seconds. When the number of items increases to 5, in only 1 out of 95 nondominated objective vectors, the optimal solution cannot be found in 2 hours. Once we exclude the unsolved instance, the average CPU time decreases from 167 to 102 seconds. That shows the dominance of a single instance in defining the average CPU time.

We also observe an effect of the total item demand on the complexity of the solutions. For fixed n and T, an increase in the total demand values increases the CPU times. Note from Table 5.9 that when $n = 3$ and $T = 14$, A2 and Of2 instances have a total demand value of 103 and 76, respectively and their respective average CPU times are 455.62 and 6.57 seconds. When $n = 5$ those instances have total demand values of 191 and 148 leaving 6 instances and a single instance unsolved in two hours.

We next discuss the relative performances of SetR and SetL over the instances for $T = 7$; when $n = 3$ and $n = 5$ and $T = 14$; when $n = 3$. We could not compare the sets for $n = 5$ and $T = 14$ combination as SetR does not include such instances.

When $T = 7$, SetR resides 3 instances with $n = 3$, having an average of 7 nondominated objective vectors and 4 instances with $n = 5$ having an average of 11 nondominated objective vectors. SetL has 4 and 5 nondominated objective vectors when $n = 3$ and $n = 5$, respectively. The total demand of SetR (71 and 136 for $n = 3$ and $n = 5$, respectively) is higher than the total demand of SetL (54 and 90 for $n = 3$ and $n = 5$, respectively) hence SetL has fewer average nondominated objective vectors.

The results for $T = 7$ show that SetL has higher CPU times than SetR which is due to the distribution of the total demand over the planning horizon. The total demand of SetR (71 and 136 for $n = 3$ and $n = 5$, respectively) is heavily on a couple of periods while the total demand of SetL (54 and 90 for $n = 3$ and $n = 5$, respectively) is distributed evenly to all periods. When Improved Model II is used in both sets, the average CPU times are 0.02 and 3 seconds for SetR and SetL, respectively when $n = 3$, while the respective average CPU times are 12 and 167 seconds for SetR and SetL, when $n = 5$.

A similar observation can be done for $n = 3$ and $T = 14$ combination. There are 5 instances with 11 nondominated objective vectors with an average total demand of 127 in SetR. In SetL, the average total demand over 18 instances is 108. The average CPU times are 0.13 and 125 seconds for SetR and SetL, respectively.

We observe that over all problem set the hardest instances have the smallest number of panels. That is the nondominated objective vectors with $P = P_{min}$ are hardest to obtain. For example in SetR, the combinations $n = 5$ & $T = 7$ and $n = 3$ & $T = 14$ each has a single unsolved instance which is for $P = P_{min}$.

Recognizing the difficulty of attaining optimal solutions for $P = P_{min}$ instances we proposed a heuristic algorithm. We now report on the performance of the heuristic algorithm.

To evaluate the performance, we assess the CPU times, the number of panels, total inventory holding cost values returned by the Improved Model II and by the heuristic procedure. We let $P(h)$ and $TIC(h)$ be the number of panels and total inventory holding cost (TIC) returned by the heuristic, respectively. $TIC^*(P_{min})$ and $TIC^*(P(h))$ are the optimal/best known TIC value with P_{min} and $P(h)$ panels, respectively. We perform our evaluations for T=14 instances as T=7 instances are easy-to-solve ones. Table 5.10 reports $P(h)$, P_{min}, $TIC(h)$ and $TIC^*(P(h))$ values for SetR. The CPU times of the model and the heuristic are also included in the table.

Table 5.10 Heuristic Performance, $T = 14$, SetR

Instance	$\boldsymbol{n}$	$\boldsymbol{P_{min}}$	$\boldsymbol{P(h)}$	**Improved Model II CPU Time (sec)**	**Heuristic CPU Time (sec)**	$\boldsymbol{TIC(h)}$	$\boldsymbol{TIC^*(P(h))}$
1	2	1*	2	0	0.02	168	168
2	3	6	7	0.08	0.06	180	180
3	3	1*	2	0	0.03	0	0
4	3	3	3	0.09	0	0	0
5	3	2	3	0.92	0.05	0	0
6	3	1*	2	0	0.06	72	72
7	4	1*	2	0	0.05	45	45
8	4	10	10	0.78	0.22	16	16
9	4	1*	2	0	0	63	63
10	4	22	22	14.92	5.2	72	32
11	4	1*	2	0	0	30	30
12	6	1*	2	0	0.05	264	243
13	7	2	3	0.08	0.05	45	45
14	9	12	12	24.66	5.11	390	330
15	9	1*	2	0.02	0	372	372
16	10	28	29	7200	68.95	922	563

* $P_{min} = 1$ instance

As can be observed from Table 5.10, the decomposition-based heuristic performs excellent producing either P_{min} or $P_{min} + 1$ panels. The resulting TIC values are also very close to the TIC values having the same number of panels.

We observe that the heuristic CPU times of the instances are much smaller for the hard-to-solve instances. The most notable instance is the last one where the optimal solution could not be solved in two hours and the heuristic results in a solution in about 1 minute. For that instance, the heuristic returns 29 panels which is only one unit more than the number of panels returned by the model at the termination time.

Another notable example is Instance 14 where the optimal solution is found in 24.66 seconds and the heuristic finds the optimal solution in 5.11 seconds.

Note that when $P_{min} = 1$, the heuristic algorithm returns exactly 2 panels and their TIC values are almost equal to the optimal TIC values of the solutions using 2 panels.

Tables 5.11 and 5.12 report the $P(h)$, $TIC(h)$, $TIC^*(P(h))$ values of SetL for $n = 3$ and $n = 5$, respectively. The CPU times of the model and the heuristic are also included in the tables.

Table 5.11 Heuristic Performance, $T = 14$, $n = 3$, SetL

Instance	$\boldsymbol{P_{min}}$	$\boldsymbol{P(h)}$	**Improved Model II CPU Time (sec)**	**Heuristic CPU Time (sec)**	$\boldsymbol{TIC(h)}$	$\boldsymbol{TIC^*(P(h))}$
A1	45	47	7200	8.89	46	36
A2	25	25	776.73	30.95	132	108
A3	19	20	81.66	1.38	75	75
A4	12	13	48.08	1.03	159	102
A5	21	22	611.86	3.34	40	40
CHL2	27	27	656.50	5.92	52	52
CHL5	6	7	12.09	0.98	157	140
Hchl3s	18	19	135.19	2.97	144	144
Hchl4s	8	8	68.09	1.05	480	378
Hchl6s	9	9	21.66	1.66	398	363
Hchl8s	11	12	8.28	0.75	35	30
HH	17	17	13.42	0.41	94	76
Of1	58	60	8.38	0.97	24	24
Of2	8	8	14.22	0.91	177	165
Sts4	17	17	34.31	1.77	81	66
W	40	40	80.28	8.72	100	86
2s	14	15	87.45	2.63	80	60
3s	28	28	21.52	3.06	166	114

Table 5.12 Heuristic Performance, $T = 14$, $n = 5$, SetL

Instance	P_{min}	$P(h)$	Improved Model II CPU Time (sec)	Heuristic CPU Time (sec)	$TIC(h)$	$TIC^{*}(P(h))$
A1	64	66	7200	127.25	50	49
A2	39	40	7200	904.53	78	82
A3	38	38	1040.80	54.28	39	39
A4	29	30	-	431.63	183	101
A5	21	22	5320.84	14.91	58	58
CHL2	43	43	1121.44	77.20	62	50
CHL5	12	13	5781.69	1198.61	95	94
Hchl3s	23	25	6195.52	68.97	213	213
Hchl4s	23	24	-	1024.92	572	488
Hchl6s	14	15	-	79.16	217	217
Hchl8s	16	17	142.89	5.02	53	25
HH	26	26	7200	447.64	442	388
Of1	74	75	7200	748.53	74	70
Of2	13	14	7200	14.94	110	110
Sts4	22	23	-	50.59	195	129
W	59	59	-	726.59	110	-
2s	22	22	7200	105.05	136	134
3s	32	33	7200	152.66	175	163

Table 5.11 and Table 5.12 show that the heuristic performs very satisfactory in producing near optimal number of panels. They produce at most one panel more than P_{min} in all instances, with a single exception. The exception is instance A1 where the heuristic produces only two more panels for both n values. The resulting TIC values are also very close to the TIC values of the solutions having the same number of panels.

We observe that the CPU times of the heuristics are several folds better than those of the model solutions. The CPU times of the heuristic for all instances –with one exception- are smaller than 10 seconds when n =3. The exception is instance A2 for which the heuristic runs in 30.95 seconds. The CPU time of the model for that instance is 776.73 seconds. Another notable difference is due to instance A1 where the model cannot find the optimal solution in 2 hours and the heuristic finds a solution (which is the best-known solution) in 8.89 seconds.

The difference in the CPU times between the model and heuristic becomes even more pronounced when *n* increases to 5. The model could find optimal solutions in only 6 out of 18 instances in 2 hours whereas the heuristic finds feasible solutions to all instances in less than 20 minutes.

CHAPTER 6

CONCLUSIONS

In this thesis, we consider an integrated cutting stock and lot sizing problem. Two dimensional items are to be cut from two dimensional blocks of identical size to satisfy all periodic demands over a specified planning horizon.

We propose a multi criteria approach with two objectives: minimizing the number of panels cut (cutting-stock related) and total inventory carrying cost of the items (lot sizing related) and aim to generate all nondominated objective vectors with respective to these two objectives.

We show that the selected objectives are conflicting and there is a nondominated objective vector corresponding to each number of panels in a specified range. To generate each objective vector, we propose a mixed integer linear programming model that minimizes the total inventory cost subject to a specified number of panels. We enhance the efficiency of the model by incorporating optimality properties and bounding mechanisms.

Our interest in the problem is from an aircraft manufacturing plant in Turkey to improve their steel cutting operations. The production planners of the company want to reduce the total purchasing and keeping costs of the big steel panels while carrying small amounts of fragile steel items. We illustrate our results on a trade-off curve over which the production planners of the company can see all reasonable solutions and make meaningful trade-offs between the number of big steel panels and the inventory holding cost over the small steel items. Hence, the decisions given using our multi criteria approach will give more insights compared to the ones that would be given using any single criterion approach. Our approaches might also be used to help the managers of other industries like furniture manufacturing having high demand for small wooden items that form final furniture products over long planning horizons.

We conduct experiments to assess the performance of our nondominated objective vectors generation method and reduction mechanisms using real data gathered from the industrial application and data taken from the cutting stock literature. The results have revealed that the instances with few items can be solved for up to 14 periods and the instances with more items can be solved for up to 7 periods, within our termination limit of two hours. We observe that when the number of items is high, an extreme efficient solution with the smallest number of panels could not be found within two hours. To handle this extreme case, we propose a decomposition-based heuristic approach and obtain favorable results.

To the best of our knowledge, we propose the first multi criteria approach for the two dimensional cutting stock and lot sizing problem. Future research may consider some extensions like allowing backorders (lot sizing problem) and non-guillotine cuts (cutting stock problem). A tri-objective problem could be defined once the backorders are allowed and penalized. Another promising extension may be to analyze the preferences of the company managers and find a representative set of nondominated objective vectors that favor those preferences.

REFERENCES

Ayasandır, U., & Azizoğlu, M. (2021). Two Dimensional Cutting Stock Problem with Multiple Stock Sizes. To appear in *International Journal of Planning and Scheduling*.

Campello, B. S. C., Ghidini, C. T. L. S., Ayres, A. O. C., & Oliveira, W. A. (2020). A multiobjective integrated model for lot sizing and cutting stock problems. *Journal of the Operational Research Society*, *71*(9), 1466–1478.

Cintra, G. F., Miyazawa, F. K., Wakabayashi, Y., & Xavier, E. C. (2008). Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. *European Journal of Operational Research*, *191*(1), 61–85.

Cui, Y., & Yang, Y. (2010). A heuristic for the one-dimensional cutting stock problem with usable leftover. *European Journal of Operational Research*, *204*(2), 245–250.

De Armas, J., Miranda, G., & León, C. (2011). A multi-objective approach for the 2D guillotine cutting stock problem. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6692 LNCS*(PART 2), 292–299.

Delorme, M., Iori, M., & Martello, S. (2016). Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, *255*(1), 1–20.

Farley, A. A. (1988). Mathematical Programming Models for Cutting-Stock Problems in the Clothing Industry. *Journal of the Operational Research Society*, *39*(1), 41–53.

Filho, A. A., Moretti, A. C., & Pato, M. V. (2018). A comparative study of exact methods for the bi-objective integer one-dimensional cutting stock problem.

Journal of the Operational Research Society, *69*(1), 91–107.

Furini, F., & Malaguti, E. (2013). Models for the two-dimensional two-stage cutting stock problem with multiple stock size. *Computers and Operations Research*, *40*(8), 1953–1962.

Furini, F., Malaguti, E., Medina Durán, R., Persiani, A., & Toth, P. (2012). A column generation heuristic for the two-dimensional two-staged guillotine cutting stock problem with multiple stock size. *European Journal of Operational Research*, *218*(1), 251–260.

Gilmore, P. C., & Gomory, R. E. (1965). Multistage Cutting Stock Problems of Two and More Dimensions. *Operations Research*, *13*(1), 94–120.

Gonzalez, Y., Miranda, G., & Leon, C. (2016). Multi-objective Multi-level Filling Evolutionary Algorithm for the 3D Cutting Stock Problem. *Procedia Computer Science*, *96*, 355–364.

Gramani, M. C. N., & França, P. M. (2006). The combined cutting stock and lot-sizing problem in industrial processes. *European Journal of Operational Research*, *174*(1), 509–521.

Gramani, M. C. N., França, P. M., & Arenales, M. N. (2009). A Lagrangian relaxation approach to a coupled lot-sizing and cutting stock problem. *International Journal of Production Economics*, *119*(2), 219–227.

Haessler, R. W. (1971). A Heuristic Programming Solution to a Nonlinear Cutting Stock Problem. *Management Science*, *17*(12), B-793-B-802.

Hendry, L. C., Fok, K. K., & Shek, K. W. (1996). A Cutting Stock and Scheduling Problem in the Copper Industry. *Journal of the Operational Research Society*, *47*(1), 38–47.

Hifi, M., & Roucairol, C. (2001). Approximate and Exact Algorithms for Constrained (Un) Weighted Two-dimensional Two-staged Cutting Stock Problems. *Journal of Combinatorial Optimization*, *5*(4), 465–494.

Leao, A. A. S., Furlan, M. M., & Toledo, F. M. B. (2017). Decomposition methods for the lot-sizing and cutting-stock problems in paper industries. *Applied Mathematical Modelling*, *48*, 250–268.

Melega, G. M., de Araujo, S. A., & Jans, R. (2018). Classification and literature review of integrated lot-sizing and cutting stock problems. *European Journal of Operational Research*, *271*(1), 1–19.

Mellouli, A., Mellouli, R., & Masmoudi, F. (2019). An Innovative Genetic Algorithm for a Multi-Objective Optimization of Two-Dimensional Cutting-Stock Problem. *Applied Artificial Intelligence*, *33*(6), 531–547.

Nonås, S. L., & Thorstenson, A. (2000). Combined cutting-stock and lot-sizing problem. *European Journal of Operational Research*, *120*(2), 327–342.

Nonås, S. L., & Thorstenson, A. (2008). Solving a combined cutting-stock and lot-sizing problem with a column generating procedure. *Computers and Operations Research*, *35*(10), 3371–3392.

Poltroniere, S. C., Poldi, K. C., Toledo, F. M. B., & Arenales, M. N. (2008). A coupling cutting stock-lot sizing problem in the paper industry. *Annals of Operations Research*, *157*(1), 91–104.

Silva, E., Alvelos, F., & Valério de Carvalho, J. M. (2010). An integer programming model for two- and three-stage two-dimensional cutting stock problems. *European Journal of Operational Research*, *205*(3), 699–708.

Silva, E., Alvelos, F., & Valério De Carvalho, J. M. (2014). Integrating two-dimensional cutting stock and lot-sizing problems. *Journal of the Operational Research Society*, *65*(1), 108–123.

Vanzela, M., Melega, G. M., Rangel, S., & Araujo, S. A. de. (2017). The integrated lot sizing and cutting stock problem with saw cycle constraints applied to furniture production. *Computers & Operations Research*, *79*, 148–160.