DEMAND DRIVEN DISASSEMBLY PLANNING

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

BY

TULÛ ERTEM

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

JANUARY 2007

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

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

Prof. Dr. Çağlar Güven
Head of Department

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

Assist. Prof. Z. Pelin Bayındır
Supervisor

Examining Committee Members

Prof. Dr. Meral Azizoğlu	(METU, IE)	
Assist. Prof. Dr. Z. Pelin Bayındır	(METU, IE)	
Assist. Prof. Dr. Sedef Meral	(METU, IE)	
Assoc. Prof. Dr. Levent Kandiller	(Çankaya Unv, IE)	
Bora Kat (M.S.)	(TÜBİTAK)	

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 : Tulû Ertem

Signature :

ABSTRACT

DEMAND DRIVEN DISASSEMBLY PLANNING

Ertem, Tulû
M.S., Department of Industrial Engineering
Supervisor : Assist. Prof. Z. Pelin Bayındır

January 2007, 122 pages

In this thesis, we deal with the demand driven disassembly planning. The main aim of the study is to construct heuristic approaches according to the suggested improvements in the literature. These heuristic approaches are further improved by recognizing the key points of the disassembly planning problem. All of the solution approaches aim minimizing total cost related to relevant costs of disassembly operations. Another subject given attention in this thesis is the importance of the setup cost on the disassembly planning, which has not been studied yet in the literature to the best of our knowledge. Computational studies are carried out to assess the performance of the heuristic procedures proposed.

Keywords: Disassembly planning, demand-driven disassembly, heuristic approach.

ÖZ

TALEP BAZLI DEMONTAJ PLANLAMASI

Ertem, Tulû
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi : Yrd. Doç. Z. Pelin Bayındır

Ocak 2007, 122 sayfa

Bu tezde talep bazlı demontaj planlamasıyla ilgilenilmiştir. Çalışmanın ana amacı literatürde önerilen ilerlemelere göre sezgisel yaklaşımlar oluşturmaktır. Demontaj planlaması problemlerinin kilit noktalarını belirleyerek, bu sezgisel yaklaşımlar daha da geliştirilmiştir. Bütün çözüm yaklaşımları demontaj planlamasıyla ilgili masraflarla ilişkili olan toplam maliyeti en aza indirmeyi amaçlar. Bu tezde ilgilenilen bir diğer konu ise, bildiğimiz kadarıyla literatürde henüz çalışılmamış olan kurulum maliyetinin demontaj planlamasındaki önemidir. Önerilen sezgisel yaklaşımların performansını değerlendirmek için deneysel çalışmalar yürütülmüştür.

Anahtar Kelimeler: Demontaj planlaması, talep bazlı demontaj, sezgisel yaklaşım.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Z. Pelin Bayındır for her guidance, patience, and support at every stage of the study. Without her friendly and helpful approach, I could never complete this study.

I would like to express my sincere thanks to Committee Members of my Thesis Defense, Prof. Dr. Meral Azizoğlu, Assist. Prof. Dr. Sedef Meral, Assoc. Prof. Dr. Levent Kandiller and Bora Kat, for their suggestions and corrections on the thesis. I also thank Baykal Hafızoğlu for helping me during the computational study.

My special thanks go to my parents and my brother who are always with me whatever I do.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Due to the fundamental changes in economics, huge reduction of natural sources and environmental issues, products' and materials' recovery are currently growing trends in most of the industrialized countries. The recovery of the end-of-life products is not only ecologically necessary and driven by legislation but also economically challenging by decreasing the utilization of virgin raw material and disposal costs. Bras and McIntosh (1999) designate the benefits of product recovery to the manufacturers as;

- an expanded share of the market,
- the trade-in value encourages customers loyalty and repeat business,
- information for product failures that can be used to create product improvements.

and they designate the benefits to the society as;

- cheaper products
- new job areas as disassembly and remanufacturing are labor intensive.

Environmental implications of product design and manufacturing processes became important for the environmental issues. In some European countries (especially in German and the Netherlands), legal applications exist to provide safe disposal of a product at the end of its life. The main reason for the legal preventions is the increasing shortage of landfill space. The combination of decreasing space and regulations make the disposal costs increase significantly in the next few years. In addition to this, governments increase the taxes on virgin raw material usage to motivate companies to try to re-use.

1.1 Recovery in the World

Environmental problems of waste management and disposal of discarded products have led many countries to take legislative action to improve reuse, recycling and other forms of recovery. Today, the majority of the studies about the recovery involve electronic waste. Electronic waste includes computers, entertainment electronics, mobile phones and other items that have been discarded by their original users. Despite its common classification as a waste, disposed electronics are a considerable category of secondary resource due to their significant suitability for direct reuse (for example, many fully functional computers and components are discarded during upgrades), refurbishing, and material recycling of its constituent raw materials. Because of great reuse opportunities they have, electronic waste is paid more attention in the world.

In the website of Wikipedia Foundation, http://en.wikipedia.org/wiki/WEEE, it is mentioned that some European countries banned the disposal of electronic waste in the landfills in the 1990s. This creates an e-waste processing industry in Europe. In 1991, the first electronic waste recycling system was implemented in Switzerland beginning with the collection of refrigerators. Over the years, all other electrical and electronic devices were gradually added to the system. Then, early in 2003 the European Union (EU) presented the Waste Electronic and Electric Equipment (WEEE) Directives for implementation be effective in 2005. The directive imposes the responsibility for the disposal of WEEE on the manufacturers of such equipment. The objective of the WEEE Directives is to prevent the generation of electrical and electronic waste and to support reuse, recycling and other forms of recovery in order to reduce the quantity of waste to be eliminated. WEEE Directives have to be applied seriously in the member states and if they do not apply, great monetary penalties are imposed to them. For instance, WEEE Directives will be enforced by UK Law on the January 1^{st}, 2007 and if the firms do not prepare themselves for the directives, they are going to pay great massing recycling bills. These directives include scope, collection, treatment, recovery and financial directives.

The directive includes large and small household appliances, IT and telecommunications equipment, consumer equipment, lighting equipment, electrical and electronic tools, toys, leisure and sports equipment, medical devices, monitoring and control instruments, automatic dispensers.

Member States set up several collection systems for WEEE, and according to the WEEE-Forum website (http://www.weee-forum.org/legislation_eu.htm#top) they have to ensure that;

- final holders and distributors can return such waste free of charge,
- distributors of new products ensure that waste of the same type of equipment can be returned to them free of charge on a one-to-one basis,
- producers are allowed to set up and operate individual or collective take-back systems,
- the return of contaminated waste presenting a risk to the health and safety of personnel may be refused.

Producers of electrical and electronic equipment must apply the best available treatment, recovery and recycling techniques. They have to set up systems for the recovery of WEEE collected separately. Producers must state the weight of the electrical and electronic waste entering and leaving treatment and recovery or recycling facilities. By 31 December 2008, the European Parliament and the Council are to set new targets for recovery, recycling and reuse.

Producers must provide for the financing of the collection, treatment, recovery and environmentally reliable disposal of WEEE. When a producer places a product into market, he must provide a guarantee concerning the financing the management of the waste of his product with a recycling insurance or a blocked bank account.

The WEEE Directive obliged the twenty-five European Union member states to transpose its provisions into national law by August 2004 but only one member met this deadline. Next year, all members except Malta and UK had transposed the framework of the directives. As mentioned above, UK will apply these directives from January 2007 on. Apart from the member states, countries which will try to involve in the EU organization must meet this directives in the negotiations and this WEEE problem is going to be recognized by them (This information is taken from the website of Wikipedia Foundation, http://en.wikipedia.org/wiki/Waste_Electrical_and_Electronic_Equipment_Directive).

The initiative and the biggest of the United Nations (UN) organizations is StEP (Solving the E-waste Problem). StEP is an organization with the members from industry, governments, international organizations and universities. According to the StEP Organization website (http://www.step-initiative.org/index.php), the objectives of this organization are;

- optimizing the life cycle of electric and electronic equipment by improving supply chains, closing material loops and reducing contamination,
- increasing utilization of resources and reuse of equipment,
- exercising concern about disparities such as the digital divide between the industrializing and industrialized countries,
- increasing public, scientific and business knowledge.

To perform its objectives, the organization has five taskforces; policy, redesign, reuse, recycle and knowledge. These five taskforces have different objectives and they involve in different projects and activities. This organization has lots of participants all over the world. Delft University of Technology (the Netherlands), United Nations University (Germany), University of California (Austria), University of Melbourne (Australia), Vienna University of Technology (Austria) and MIT (USA) are the participant universities. AEA Technology, Apple Germany and Dell are some of the companies that involve in the projects. In addition to these companies some governmental and environmental research institutes support their projects and activities.

Generally electronic companies take recycling and remanufacturing problems seriously. As they involve in the StEP Organization, Dell and HP are interested in the e-waste problem. In Hewlett-Packard Development Company's official website (http://h41131.www4.hp.com/uk/en/global_citizenship/globalrecycling.html), it is mentioned that HP has a Global Recycling Program and it expands the program to reach more customers and to create new ways to people to discard used or unwanted products. Some activities that HP is performing under this program are

- In countries that have implemented WEEE Directives, HP addresses its recycling responsibilities to European Recycling Platform which runs take back and recycling operations for WEEE in Austria, Ireland, Portugal, Spain, Germany and Poland.

- In countries such as the UK and Italy where the WEEE Directive has not yet been implemented, HP is running pilot product collection events to assess and prepare for the quantity and mix of equipment that consumers return.

- In Europe, HP also offers a free recycling service for its commercial and enterprise customers to return their IT hardware to designated collection points.

- HP will host a serious of product collection events in the United States similar to the number of pilot studies in Europe. With the majority in Europe and America, HP's recycling programme operates globally in more than 40 countries.

Similar to HP, Dell tries to meet the requirements of the WEEE directive and is engaged in the development of country-specific implementation schemes to comply with the national WEEE laws. They arrange consumer based campaigns with the slogan of 'No computer should go to waste'. They arrange donation and recycling options for their customers and assure them to pick up their products at their door. (This information is taken from Dell Inc.'s website, http://www.dell.com/content/topics/global.aspx/corp/environment/en/recycling_main?c=us&l=en&s=corp)

Apart from the electronic sector disassembly operations have a growing number of applications as a result of disassembly regulations. Taleb et al. (1997) mentions that German car manufacturer, BMW, has already opened a dismantling plant in Orlando, Florida, a number of plants in Europe and is planning to open fifteen more by 1995. Volkswagen also opened a dismantling plant in Europe in 1990 and is planning to open more. Their studies still continue as they have to meet the quota of the End-of-life Vehicles Directive by the year 2015. In addition to car manufacturers, EU legislation forces tire manufacturers to arrange for the environmentally friendly disposition of one used tire for every new tire sold.

Another current application of recovery is in dismantling weapons. It is no longer environmentally acceptable to discard weapons and ammunitions by blowing them up, burning or dumping them in the ocean. Meier (1993) said that the US Defense Department is hiring military contractors to dismantle and recycle unwanted weapons.

The applications of recovery are numerous and extending to almost every industry that deals with discrete parts products. The aerospace, construction, industrial equipment and electronics are all good examples. The reason for the extension is not only the environmental factors and legislation but also the profit of recovery.

1.2 What is disassembly?

Disassembly is a major activity performed in treatment and recovery facilities and it is the most important precedence of product and part recovery. Disassembly is defined as a systematic method of separating a product into its constituent parts and subassemblies. It is applied to recover pure material fractions, isolate hazardous substances, separate reusable parts and subassemblies. According to Lambert and Gupta (2002), disassembly is encouraged as a means to decrease the amount and cost of disposal and incineration by reclaiming valuable parts and materials and separating hazardous materials for processing in a responsible manner. There are two types of disassembly named as selective and complete disassembly. In the former one, one or more components are removed from a product in order to recover valuable parts of the product and to remove hazardous parts or subassemblies; whereas in the latter one, all of the parts or subassemblies are separated from a product. Another classification is about the destruction level and has three types. Lee et al. (2001) classifies disassembly as;

- non-destructive, involving no part demolition,
- partially destructive, with demolition of cheap parts,
- completely destructive, with uncontrolled destruction of the product structure.

Due to its critical role in recovery of products and materials, disassembly has recently become an active research area. Even though it sounds reasonable to call disassembly as the reverse of the assembly process, it has several differences. Lambert (2003) mentions the below differences:

- Disassembly is usually not performed to its full extent: incomplete disassembly is often preferred, which adds the disassembly depth to the decision variables.
- The assembly process is often not completely reversible.
- The value added in disassembly processes is usually modest compared to that obtained in assembly.
- Uncertainty exists with regard to the quality of the components.
- Uncertainty exists in the supply of discarded products from both qualitative and quantitative points of view.
- In disassembly, a variety in supplied products might be present.

Due to these features, disassembly is mainly carried out by human labor instead of by automated assembly lines or robots.

1.3 Motivation and General Approach Followed

The variety of the products that could be disassembled and remanufactured increases and operations begin to involve more complex products. In the field of disassembly, complexity means commonality and multiplicity. Commonality refers to the use of the same component in more than one root, an example of which could be a standard disk drive used in several computer models. Multiplicity refers to using the same component in more than one place in a certain product. A certain board may be used in several different assemblies within the same computer. While commonality and multiplicity make very good sense from an economic and environmental standpoint, they certainly complicate the planning decisions. It is no longer so easy to calculate the optimal number of roots; i.e., end of life products, needed to fulfill demand for the leaves; i.e., parts and subassemblies, as these leaves can be sourced from different cores. A need for powerful solution methods arises as a consequence of this complexity.

As the volume of recovery operations increases due to environmental legislation proposed and the value in terms of cost reduction understood well by the manufacturers, operation planning gains importance. A serious research effort has been put on operations management and planning issues for product and part recovery, as well as recycling. A recent book edited by Dekker et al. (2002) provides an extensive review of these studies.

As it is mentioned in the review and by several other authors working in the area, there is still a need for new techniques and procedures for part/product recovery planning problems at various levels of decision making.

In this thesis, we consider a part recovery situation: There is a known demand for recovered parts that can be obtained from a number of different end-of-life (EOL) products by disassembly. The aim is to determine the number of EOL products to be disassembled to satisfy parts' demands which are estimated (and assumed to be perfectly known) over a finite planning horizon. We consider two different decision levels: The first one seeks the number of EOL products to be disassembled when there is an option of procuring brand new parts whereas we consider disassembly lot sizing problem in the second problem.

Although we did not consider a specific recovery environment in this work, the problems that we consider are generic to any firm performing part recovery. As the volume of EOL products collected and the parts of them are used in either remanufacturing or as replacement parts in after sales service, the firms should make such disassembly decisions widely.

In order these proposed methods to be applicable by these companies, it is important that they are compatible with existing decision support and information tools, like ERP systems with modifications. It is also important that the developed methods make sense for the practioners. Therefore, our main aim in this thesis is to develop ‘easy to apply’ and ‘easy to understand’ heuristics that perform well with respect to both solution quality and computational requirements.

For this purpose, heuristics already developed in the literature for demand driven disassembly planning over a finite planning horizon are further improved (for the first problem) whereas we investigate the performance of the traditional lot sizing heuristics (for the second problem).

The outline of the thesis is as follows: In Chapter 2, an overview of the related literature on disassembly planning is given. The main focus of Chapter 3 is the solution methods for the disassembly planning problems. The first method is integer programming (IP). Our IP gives the optimal solution, however at an expense of long solution times that grow rapidly with the increasing number of the products and their leaves. In Section 3.2, an IP model constructed for the problem which is introduced in Section 3.1 is explained. The second method is the heuristic approaches. Heuristic approaches are simple and intuitive, but have no guarantee to find the optimal result unlike mathematical models. They usually give near optimal results but they are simple and quick. In Section 3.3, some of the heuristic approaches studied in the literature are explained in detail. In section 3.4, a heuristic is constructed according to the suggestions in the literature and explained in detail. Additionally, another variant that differs in inventory holding cost estimation is provided. Finally, in Section 3.5 the results of the computational study carried out are discussed. In

Chapter 4, a lot sizing problem for disassembly is considered. A setup cost is incurred every time a batch of root item is disassembled. It consists of engineering cost of setting up the disassembly equipments and labors. To the best of our knowledge, there is no study concerning disassembly lot sizing in the literature. In this work, our main aim to investigate the performance of the traditional lot sizing heuristics in disassembly environment. In Section 4.2, a mathematical model is provided for the problem. In Section 4.3, after explaining the general approach followed in the application of traditional lot sizing heuristics, the algorithms of these heuristics are described. In Section 4.4, the results of the computational study are discussed.

Finally in Chapter 5, we discuss our conclusion and present further research direction.

CHAPTER 2

LITERATURE REVIEW

The majority of the studies about disassembly involve disassembly scheduling and sequencing problems. According to Xirouchakis et al. (2001), disassembly scheduling problem can be defined as the problem of determining the order quantity to fulfill the demand of the individual parts or subassemblies where disassembly sequencing is the problem of generation of all possible sequences and determination of an optimal or near-optimal sequence considering the end-of-life options of disassembled parts and subassemblies. In this thesis, we deal with the disassembly scheduling problem by constructing a heuristic approach implementing the suggestions of Langella (2007) and modifying it with different point of views to some decision criteria. In addition to this, we performed a study to investigate the efficiency of the traditional lot-sizing heuristics in the disassembly scheduling problems. Apart from the heuristic approaches, mathematical models are provided for the problems. Both of the solution approaches determine the root item quantities in order to satisfy the individual parts', i.e. leaf items' demands.

Studies about the disassembly scheduling offer heuristic approaches and mathematical models to determine the quantity of root items in order to satisfy the demands of the parts and subassemblies with the main objective of cost minimization. For the sequencing problems, some graphical methods such as AND/OR and Liaison graphs, which show the precedence relations of the disassembly, are introduced. But the applications of these graphs are limited to simple products. In addition to this, some solution methods are provided in order to find an optimum sequence for a profitable end-of-life strategy of a product.

In this chapter, the studies on disassembly scheduling and sequencing are reviewed in Section 2.1 and Section 2.2, respectively.

2.1 Literature Review on Disassembly Scheduling

To begin with the mathematical models proposed for the scheduling problem, the studies of Clegg et al. (1995), Kongar and Gupta (2002), Spengler et al. (2002) and Langella (2007) are described.

Clegg et al. (1995) point out the importance of the disposal cost by providing examples from electronics industry. Recycling of products is an important way of decreasing disposal costs where remanufacturing is a major source of cost saving by partially rebuilding a product to give it a functionality. They claim that disposal cost is expected to increase in the next decade due to increasing shortage of landfill space and expanding regulation of waste disposal and the efforts have been directed at ways of decreasing disposal costs by recycling, remanufacturing or reusing which would push companies to build remanufacturing capability. They develop a linear programming model for a production system with remanufacturing capability. The aim of the model is to examine the effects of different factors such as disposal, disassembly and virgin material costs, limitations on disposal, disassembly capacity and restrictions on the relative amounts of remanufactured and new products in the product mix on the optimal production plan. The model is constructed under the assumptions listed below.

- The company recovers products from the field and has the options of
    - disposing them,
    - reusing some components after disassembly,
    - remanufacturing the product,
    - assembling the product from scratch.
- Both assembly and remanufacturing are carried out in the same manual assembly facility.
- All processing is carried out within a single period and single facility.

- Product demands are deterministic.
- The decision variables are the inventory level and the amount of products or components disposed, reused, remanufactured and assembled from scratch.

The objective is to maximize profit that takes the following into the account.

- The revenues come from the sale of remanufactured and new products,
- Inventory holding costs are incurred for returned products, partially disassembled products, finished remanufactured and new products,
- Purchasing cost,
- Waste disposal costs,
- Partial and total disassembly cost,
- Assembly cost for remanufactured and new products.

The constraints are based on;

1. inventory balance equations of returned product inventory, disassembled product inventory, reused modules inventory, finished remanufactured products and finished new products,
2. capacity constraints for disassembly capacity, waste disposal capacity and assembly capacity.

The authors claim that the model provides insights into the situations favorable to the implementations of reuse and remanufacturing programs, and provide companies with guidance in determining the areas of the operation and the costs need to be addressed to make such programs economically viable. There is no computational study provided in the article.

Kongar and Gupta (2002) provide a multi-criteria optimization model of a disassembly to order (DTO) system. The model determines the optimum number of each product type to be taken back at the end-of-life and disassembled to meet the demand with respect to the preemptive goals. The model is constructed for a single period case. The authors argue that in an environmentally conscious manufacturing environment, it is no longer realistic to use single objective and this model is an attempt to achieve a desired level of profit while also satisfying additional goals simultaneously. The preemptive goals to achieve are maximum total profit, maximum sales from materials, minimum number of disposed items, minimum number of stored items, minimum cost of disposal and preparation including sorting, refurbishing and cleaning. The total profit value is the difference between all the revenues and all the costs. Revenue sources are sales of demanded materials and items where costs are take back cost, transportation cost from collectors to facility and from facility to outside recycling plant, disposal site and storage location, preparation cost of end-of-life products, destructive and non-destructive disassembly cost, recycling cost, storage cost and disposal cost. The constraints are:

- Recycling can be either performed in plant or outsourced to an outside contractor.
- Number of items recycled has to be equal to the corresponding demand of items for recycling.
- The total space occupied by the stored items have to be less than or equal to the available space.
- The number of items retrieved from end-of-life products ordered has to be greater than or equal to the number of demanded items.
- The number of recycled items in plant should be less than or equal to its recycling capacity.
- The total number of disassembled items should be equal to the items that are reused, recycled, stored or disposed.

Deviation variables for the decision variables are introduced. Deviation variables are either positive or negative. When maximizing the decision variables in the objective function, the negative deviation is forced to zero and no restriction is put on the positive deviation variable to reach to the aspiration level and exceed it as much as possible. The opposite is done when minimizing the variables. Next, goal programming is applied.

Similar to Clegg et al. (1995), Spengler et al. (2003) give attention to the problems of electronics manufacturing. They predict that the Waste on Electrical and

Electronic Equipment (WEEE) increases by 3-5 % per year. They consider a system limited to a recovery center. The scrap of several types of products (TV, computers, radio, etc.) is generated by purchasing used products. Procurement is done regularly and the products purchased are kept in a designated storage. The recovery center orders the scrap from this input storage and gets them immediately. First step is the disassembly operation which is composed of manual and partially automated processes. Second step is bulk recycling designed to recover precious fractions from mixed electronic scrap. The scrap disassembled is either used internally in bulk recycling or marketed externally. To determine the amount of scrap to be recovered, disassembled and used in internal or external operations, they present a mixed-integer linear programming model for integrated planning of these stages. The basic assumptions are below:

- Single period planning horizon is considered.
- There is no order lead time between input storage and recovery center.
- Holding inventory is permitted for every item.
- There are limits for disassembly labor time, maximum mass of scrap type to be obtained from storage, sale capacity of scrap types to external recycling, equipment capacity of separation unit and sale capacity for isolated material fraction.

The objective function maximizes achievable marginal income by deciding the variables of the mass of scrap type to be taken, the number of executions of disassembly activity and the mass of scrap type directed to internal recycling and consists of

- acceptance revenues/costs,
- disassembly output revenues/costs,
- bulk recycling output revenues/costs,
- variable disassembly costs and variable process costs.

This model has some specific characteristics, especially the description of material flows throughout the bulk recycling units, the consideration of the choice of taken products in combination with capacity constraints, and the explicit modeling of interactions between disassembly and bulk recycling. After some numerical results are obtained, scenario analysis is made to derive recommendations for future planning including design improvement hints and promising strategic positions for a recycling enterprise. For that purpose, the system behavior under changing conditions is analyzed by the assessment of different scenarios based on the sensitivity analysis. Since the acceptance fees and the bulk recycling output revenues represent a major share of the objective value, price variations are examined. Besides, changes in price coefficients, which have relatively less impact, are analyzed. Apart from costs, capacity and labor force effects are studied. The scenarios are constructed by doubling or halving the costs or capacity values and investigate their effects. The major finding is that disassembly is only advisable if hazardous or very precious parts are removed.

The most recent study on the disassembly scheduling is performed by Langella (2007). In addition to some heuristic approaches which will be explained both later in this section and in Chapter 3, he proposes an IP model for the disassembly problem, but analytical properties of the model are not investigated. The model includes the assumptions mentioned below;

- They allow multiplicity and commonality. Multiplicity is the existence of several leaf items in one root item and commonality is the existence of the same leaf item in different root items.
- Demands of the intermediate and leaf items are deterministic and disassembly lead time occurs.
- There is no supply limit on root items.
- Cost parameters are procurement cost, disassembly costs and disposal costs.
- They have access to new components and this option may be chosen to avoid disassembly.
- Holding inventory is permitted for every item.
- Intermediate and leaf items are allowed to be disposed of at given costs.
- There are capacities for max inventory and return availability.

He constitutes the model formulation as cost minimization and the objective function includes

- item procurement,
- disassembly,
- item holding,
- item disposal costs.

The model has four types of constraint groups:

1. starting inventory levels,
2. inventory balance equations for the cores, intermediate and leaf items,
3. supply restriction of the cores,
4. maximum inventory level for intermediate and leaf items constraints.

The model gives the optimal solution but it gets slower in complex cases where the heuristic offers a faster solution with the risk of infeasibilities. Gupta and Taleb (1994) and Taleb et al. (1997) offer a reverse MRP like approaches where Taleb and Gupta (1997) and Langella (2007) provides original heuristic approaches for determining the disassembly order in order to satisfy the disassembled parts' demands.

Gupta and Taleb (1994) present an algorithm for scheduling the disassembly of discrete part products characterized by a well defined product structure. The objective in the disassembly case is the reverse of that of MRP but the algorithm is not the reverse of the MRP algorithm. The algorithm determines quantity and schedule of disassembly of the root item to fulfill the demand for its various parts. The basic assumptions are:

- All the required information is known with certainty.
- The lead times, when going from one level to the next, are constant, irrespective of the lot size.
- The disassembly processes are assumed to be perfect. Hence no defective parts are generated.
- Capacity is not considered.

They find the gross requirements of the sub-roots according to the net requirements of the leaf items. They found the gross requirements step by step according to the bill-of-materials structure of the products. The main problem arises in the reverse MRP is having multiple sources of demand. These demands are not independent and are satisfied by one root item and this situation leads to excess inventories for the components with relatively low demand compared to their brother items.

Further research for the above problem is done by Taleb et al. (1997) to accommodate parts and material commonality to the algorithm. The objective is to minimize the total number of root items to disassemble in order to fulfill the demand for these parts. The assumptions made in the algorithm are similar to the assumptions of Gupta and Taleb (1994). The model above is modified assuming a material commonality. For example, parts made of same material are grouped and the net requirements revealed by their total weight. In the non-commonality case, the modules are independent but under commonality, the demands and inventory levels are kept in a certain proportion so they are not independent and have to be processed in parallel for every time period.

Taleb and Gupta (1997) address the problem of scheduling the disassembly of discrete part products. The problem involves periodic demands of different leaves that could be obtained by disassembling some root items. The basic properties of their method are as below:

- They allow multiplicity and commonality.
- Demand is deterministic and disassembly lead time occurs.
- There exists limited different root items but there is no supply limit on them.
- Procurement cost, disassembly costs and disposal costs are included.

They propose two algorithms called ‘core algorithm’ and ‘allocation algorithm’. When solving the problem they want to show the proven benefits of commonality that are lower inventory costs and lower unit costs due to quantity discounts and the alternative use of parts/materials across several end products. Core algorithm aims to minimize the disassembly cost where allocation algorithm aims to minimize holding cost by delaying the disassembly as much as possible.

After Taleb and Gupta (1997), Langella (2007) studies the same problem with the same assumptions of Taleb and Gupta (1997) and tries to find solutions to some possible infeasibilities in the heuristics. Langella (2007) proposes an integral algorithm of core and allocation algorithms. The procedure is not different from the core algorithm but it solves lead time infeasibility in some cases that core and allocation algorithm cannot cope with. It again disassembles the most attractive root but this is done myopic, not over the whole planning horizon. Then, he improved the algorithm by considering the disposal cost, holding cost and external leaf procurement options in the heuristic but does not show the way he follows in these algorithms. These costs are added to item procurement and disassembly cost and become the part of the parameters for selecting the most attractive root to be disassembled. Under some assumptions on lead times and costs, heuristic is found to have an error of 4.55% compared to the optimal solution within the data set considered. The algorithms of Taleb and Gupta (1997) and Langella (2007) is explained widely in Chapter 3 with illustrative examples as well.

2.2 Literature Review on Disassembly Sequencing

Disassembly sequencing is an evolving field of interest. An assembly consists of many components which can be decomposed by a multitude of sequences and disassembly sequencing involves the search for all possible sequences and the selection of the optimum one. According to Lambert (2003), studies for disassembly sequencing are performed for

- optimal repair and maintenance,
- searching the optimal assembly sequence,
- design and optimize disassembly lines,
- optimum product design for disassembly, which is called Design for Disassembly (DFD).

De Mello and Sanderson (1990) introduce AND/OR graphs to represent all possible assembly and disassembly sequences of a product. It provides a useful tool for the selection of the best assembly, disassembly or repair plan. In an AND/OR graph there are nodes representing parts or other possible subassemblies of the product and there are edges representing the possible assembly/disassembly operations. In addition, three applications are discussed: a top-down search of the AND/OR graph which aims selection of the best assembly plan, a bottom-up search for recovery from execution errors and opportunistic scheduling. At last, they mention some related issues which are under investigation. Automatic construction of the AND/OR graph from design descriptions, development of evaluation function for selection among alternative plans and computational complexity are the important issues they give as further research.

Figure 2.1 is a simple AND/OR graph where Figure 2.1a shows an AND operation and Figure 2.1b shows an OR operation.

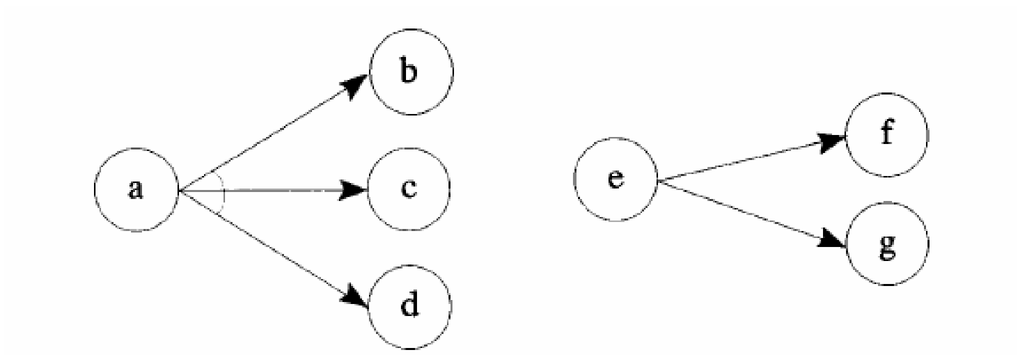

Figure 2.1 Branches of an AND/OR graph, Penev and De Ron (1996)

As De Mello and Sanderson (1990), Penev and De Ron (1996) deal with the graphical solution procedures in the disassembly sequence problems. According to Penev and De Ron (1996), the main task of the disassembly strategy is to determine the disassembly level and disassembly sequences which provide conditions of the generation of profit while the environmental conditions are maintained. They claim that the theory of graphs and dynamic programming method will be used for the evaluation of a good disassembly plan.

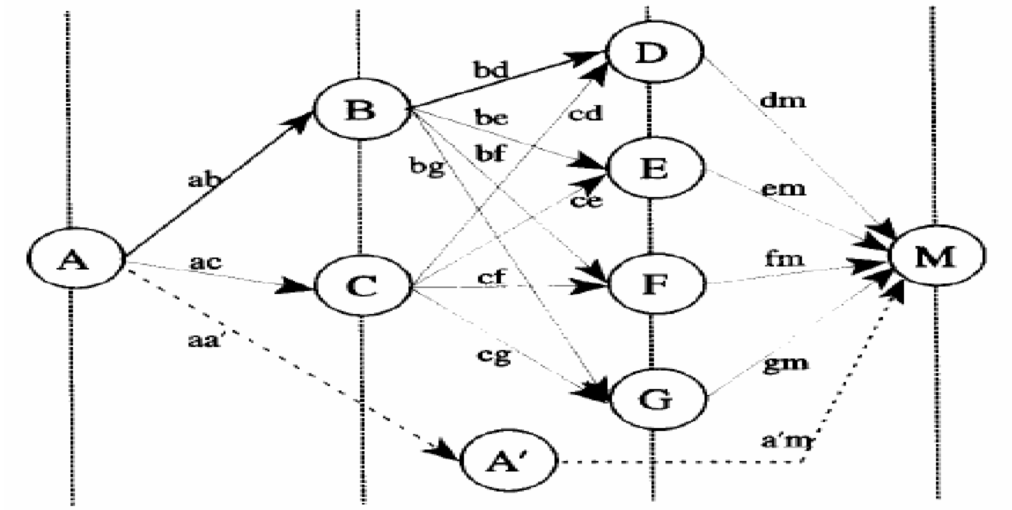

Figure 2.2 A disassembly process represented by a graph, Penev and De Ron (1996)

Figure 2.2 shows the possible disassembly operations between initial and final state, and shortest path is found by using dynamic programming. Between the stages, disassembly level (service, disassembly, dismantling, recycling, disposal) is chosen according to

- the value added to goods and materials,
- the disassembly costs per operation,
- the revenues per operation,
- the penalty if the poisonous materials are not completely removed.

After comparing the disassembly revenues, the maximum one is chosen and that operation is applied to the stage. To illustrate the applicability of their approach, they explain it by disassembling a bearing unit.

Xirouchakis et al. (2001) investigate the planning and scheduling problems in disassembly systems. At first, they give brief descriptions about disassembly, its classifications and factors limiting effectiveness. Secondly, they give some literature survey results about these two problems and give a high attention to some graphical methods named as AND/OR graph and Liaison graph.

As well, they compare the reverse MRP with MRP. In reverse MRP, the assumptions are:

- Any defective parts disassembled are not considered in the problem.
- A discrete timescale is used for the disassembly lead time and ordering lead time.
- Common parts occupy the same level in the disassembly product structure for a certain part.
- Capacity constraint is not considered as a part of the disassembly process.

At last, several research directions focusing on the methodology for coping with uncertainty, disassembly sequence generation and capacity for disassembly scheduling are suggested. One of the studies they give as a reference for the solution approaches of uncertainty in disassembly operations is Güngör and Gupta (1998).

Güngör and Gupta (1998) deals with disassembly sequence planning for products with defective parts. They discuss the sources of uncertainty in disassembly sequence planning and then identified an efficient approach of dealing with the uncertainty and developed a methodology to resolve uncertainty interactively during disassembly.

Lambert (1997) develops and describes a method for solving general optimal sequence generation problems by linear programming. The optimal disassembly sequence is the sequence of actions that generates maximum net revenue, subject to definite constraints. Based on LP techniques, a new approach to obtain optimal disassembly sequences is presented. This approach is applicable broader than the other approaches in the literature because the other approaches are restricted to determination of the optimal task sequence from a given initial state to a final state. But the model Lambert (1997) presents allows incomplete disassembly and final state is generated automatically. Clustering problem can be included in the model by an extension. Moreover, the model is adaptable to additional constraints like environmental or quality constraints. His approach is also useful in design, where preliminary insight in specific disassembly processes is desired, prior to subsequent phases that involve a more detailed approach.

Lambert and Gupta (2002) consider the problem of determining the optimal lot-sizes of end-of-life products to disassemble so as to fulfill the demand of various components from a mix of different product types that have a number of components in common and to disassemble these products optimally. They compare two methods named as disassembly graph approach and component-disassembly optimization model. In their study, they assume limited supply and all the demand are fulfilled. Their model considers disassembly cost, disposal cost and supply cost and it is appropriate for selective disassembly. In addition to these two methods, they suggest the third one which avoids the disadvantages of redundancy and nonlinearity and is applied to a multi-period, demand driven case that includes multiplicity and commonality.

Gonzalez and Adenso Diaz (2005) present an approach that determines the best end-of-life (EOL) strategy among disposal, recycling, reuse and disassembly for a product with the goal of enhancing the application of environmental criteria from the earliest stages of product design. This approach uses scatter search (SS) metaheuristic to determine the disassembly cost at each level of the BOM. It gains information from products' 3D CAD representations, BOM structure, technical and economical data. To determine the best EOL strategy, it calculates the profits for each EOL strategy and total disassembly time as the sum of the times spent breaking each joint and then chooses the most profitable one. The next step is the selecting the optimum disassembly sequence. And at last, solution approach modifies the solution in order to fulfill environmental criteria that are available resources and amount of waste generated. They describe the approach with a real example and mention some further research directions. As this approach is easy to use and has the possibility of modifying the encountered strategy according to the environmental problems, it is appropriate for industry applications.

After proposing a representation scheme that embraces the precedence relation representation, such as AND, OR predecessors and successors, for disassembly line balancing problem (DLBP), Altekin (2005) defines two problems, PC and PH. PC is defined as the assignment of disassembly tasks to an ordered sequence of stations such that the precedence relations are satisfied and the profit per disassembly cycle is maximized. PH defines a problem for the entire planning horizon by dividing into time zones and finds a different line balance for each zone. For each problem, she provides a mathematical programming model and a heuristic solution procedure. For the PC, the performance of the heuristic approach is investigated against the optimal solutions. For the PH, due to nonlinearities, restricted version of the problem is solved, and the quality of the heuristic approach is evaluated according to these problems. At last, she gives some further research directions related with the proposed solution procedures.

CHAPTER 3

DISASSEMBLY PLANNING WITH REGULAR PROCUREMENT OPTION WHEN THE SET-UP COSTS ARE NEGLIGIBLE

In this chapter, we present our work to further improve the disassembly planning over a finite planning horizon heuristics originally proposed by Taleb and Gupta (1997) and improved by Langella (2007).

In Section 3.1 the problem environment is discussed. In Section 3.2, a mathematical model for the problem in concern is provided. In Section 3.3 the heuristic approaches proposed in the literature are described. In Section 3.4 our improvements are presented. In Section 3.5, the computational study together with the results is discussed.

3.1 Environment

We consider a disassembly system that faces demand for the recovered parts. The source to satisfy the demand is either end-of-life (EOL) products or an outside supply, i.e., regular procurement of the parts is also an option. EOL products are referred to as root items, and the parts of EOL products that have demands are referred to as leaf items.

There are in total L leaf items demanded. The demand for those may have various sources. They can be demanded by independent remanufactures, or can be used as replacement parts at after sales service agents, or can be used in internal remanufacturing activities. We assume that the total demand for each leaf item is forecasted and available over a finite planning horizon, T, and hence they are treated as deterministic demand. The demand for leaf item $l \in \{1,...,L\}$ in period $t \in \{1,...,T\}$ is $\lambda_{l,t}$.

There are R root items that include these leaf items in their product structure. Root items must be collected, transported to the disassembly plant and then must be sorted and prepared for the disassembly operation by the disassembler. We assume that there is no supply constraint on root items, hence the disassembler can purchase as many root items as (s)he wants incurring a unit purchasing cost, PC_r. The purchasing lead time for root items is assumed to be negligible, without loss of generality.

After the preparation, disassembly, which is the process of systematic removal of the desired leaf items, starts. Root item $r \in \{1,...,R\}$ contains $\alpha_{r,l} \in \{0,1,...,\}$ of leaf item $l \in \{1,...,L\}$ on its product structure. A leaf item may be present in a number of different root items and this situation is referred to as *commonality*. Whereas the situation where a leaf item can only be obtained by disassembling only a particular root item is referred to as *non-commonality*. There may be more than one leaf item on the product structure of a particular root item, i.e., $\alpha_{r,l} > 1$, and this situation is referred to as *multiplicity*. We allow all commonality, non-commonality and multiplicity situations in the problem environment.

A simple bill-of-materials (*BOM*) structure to illustrate these definitions is depicted in Figure 3.1. In the figure, *A* and *B* are root items and *C, D* and *E* are leaf items. *A* extracts 2 *Cs* and 1 *D* whereas *B* extracts 1 *C*, 3 *Ds* and 2 *Es* as leaf items. *D* is a common item of *A* and *B*, that is its demand can be satisfied by disassembling either *A* or *B*. It is seen in Figure 3.1 that each root item satisfies multiplicity as *A* includes 2 *Cs* and *B* includes 3 *Ds* and 2 *Es*.

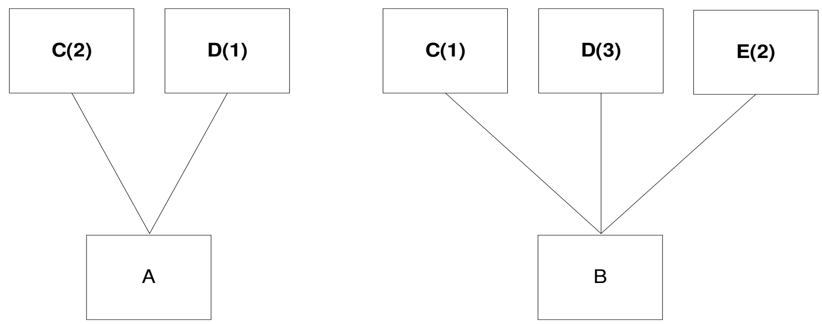

Figure 3.1 A sample product structure

We restrict our attention to the complete disassembly case. That is when a root item goes through the disassembly process; all of the leaf items that exist on its product structure are obtained. For instance, when 1 unit of root *A* is disassembled, exactly 2 units of leaf *C* and 1 unit of *D* are obtained (See Figure 3.1). The disassembly process is non-destructive, i.e., all leaf items are obtained in good state of quality. It is assumed that the yields of the root items are deterministic and as they are given in *BOM* structure.

Disassembly is a labor intensive process so its cost is mainly labor cost. For each unit of root *r* disassembled a unit disassembly cost, DC_r is incurred. We consider the case where disassembly cost is time stationary; i.e., the cost does not change with time. Unit disassembly cost includes pre-disassembly options' costs like sorting and cleaning.

Lead time for disassembly operation is assumed to be deterministic, and allowed to be non-identical for different root items. Disassembly lead time of root *r* is LT_r.

After disassembly, leaf items should be recovered before reuse. Unit recovery cost, RC_l, is incurred for leaf item *l* after the disassembly. It is assumed that every leaf item needs recovery except the items procured externally to meet the demand. Recovery cost is time stationary as well. We assume that lead time for recovery is less than a period. But this assumption can be relaxed easily. External procurement of leaf items is assumed to be available and its lead time is taken as zero. Externally procured items are brand new, hence they do not need a recovery operation. Unit cost of procuring leaf item l is EPC_l.

If the leaf items are not sold immediately, they are stored in the inventory for future use. As it is assumed that there is no damage and hazard on the leaf items, there is no item disposal. The leaf items leftover at the end of the planning horizon are also held in the inventory. We assume that inventory holding cost is time stationary and it is IC_l for leaf item l.

The disassembly planning problem that we focus is defined as to determine the type and number of root items to be disassembled and the number of leaf items procured from the outside supplier to satisfy the demand for the leaf items over a finite planning horizon.

3.2 Mathematical Model

In this section a mathematical model is provided to obtain the optimal solution to the disassembly planning problem.

<u>Notation</u>

<u>Parameters</u>

T : Length of the planning horizon.

t : Index for periods $t = 1,...,T$.

R : Number of root items.

r : Index for root items $r = 1,...,R$.

L : Number of leaf items.

l : Index for leaf items $l = 1,...,L$.

$\alpha_{r,l}$: Number of leaf item l that is obtained from root item r.

$\lambda_{l,t}$: Demand of leaf item l at period t.

IC_l : Unit inventory holding cost per period of leaf item l.

EPC_l : Unit external procurement cost of leaf item *l*.

RC_l : Unit recovery cost for leaf item *l*.

LT_r : Disassembly lead time for root item *r*.

DC_r : Unit disassembly cost of root item *r*.

PC_r : Unit procurement cost of root item *r*.

<u>*Decision variables*</u>

$x_{r,t}$: Number of root item *r* disassembled in period t, defined only if $t - LT_r \geq 1$.

$z_{l,t}$: Number of leaf item *l* held in inventory at the end of period t.

$p_{l,t}$: Number of leaf item *l* procured externally in period t.

The disassembly planning problem that we consider can be stated as follows.

Min

$$\sum_{r=1}^{R}\sum_{t=1}^{T}((DC_r + PC_r) * x_{r,t-LT_r}) + \sum_{l=1}^{L}\sum_{t=1}^{T}(IC_l * z_{l,t}) + \sum_{l=1}^{L}\sum_{t=1}^{T}(EPC_l * p_{l,t}) + \sum_{t=1}^{T}\sum_{l=1}^{L}\sum_{r=1}^{R} RC_l * \alpha_{r,l} * x_{r,t-LT_r} \tag{3.1}$$

s.t.

$$\sum_{k=1}^{t}\sum_{r=1}^{R}\alpha_{r,j} * x_{r,k-LT_r} + \sum_{k=1}^{t} p_{l,k} - \sum_{k=1}^{t}\lambda_{l,k} = z_{l,t} \quad for\ t = 1,...,T \quad l = 1,...,L \tag{3.2}$$

$$x_{r,t},\ p_{l,t},\ z_{l,t} \geq 0 \text{ and integer } for \quad \mathrm{r} = 1,...,R,\ l = 1,...,L,\ t = 1,...,T \tag{3.3}$$

Objective function of the model (3.1) aims to minimize the total costs incurred over the planning horizon. The relevant costs are total disassembly costs, inventory holding costs and external procurement costs of leaf items. Note that for every root item disassembled, unit disassembly and procurement costs are incurred. For all leaf items obtained by disassembly, unit recovery cost is incurred.

The set of equations given in (3.2) provides inventory balance for all leaf items at every period of the planning horizon. Inventory level of a leaf item *l* at the end of the period t is the difference between all inflows and outflows up to that period. Inflows are generated by disassembly and procurement and outflows are demand. $\sum_{k=1}^{t}\sum_{r=1}^{R}\alpha_{r,j} * x_{r,k-LT_r}$ is the number of leaf items *l* whose disassembly has been already finished, i.e., cumulative number of leaf items disassembled, in period *t.* $\sum_{k=1}^{t} p_{l,k}$ is the number of leaf items *l* that has already been procured, i.e., cumulative procurement quantity, in period *t.* $\sum_{k=1}^{t}\lambda_{l,k}$ is the cumulative demand in period *t* which constitutes an outflow from the inventory. Equation (3.3) represents non-negativity and integer constraints. For our problem environment, taking the number of root items as the only integer variables and relaxing the integer constraint for $p_{l,t}$ and $z_{l,t}$ does not result in different solutions.

In the environment that we consider all root items procured are disassembled, and all obtained leaf items are recovered. In addition unit recovery cost for a disassembled leaf is independent from the root from which the leaf is obtained. We do not allow any disposal for any items. This simplifies our costing scheme; instead of considering root procurement, root disassembly and recovery costs separately, one can consider a total disassembly cost, TDC_r for $r \in \{1,...,R\}$, to include all : $TDC_r = PC_r + DC_r + \sum_{l}\alpha_{rl} * RC_l$. According to this we can rewrite the model as below.

Min

$$\sum_{r=1}^{R}\sum_{t=1}^{T}(TDC_r * x_{r,t-LT_r}) + \sum_{l=1}^{L}\sum_{t=1}^{T}(IC_l * z_{l,t}) + \sum_{l=1}^{L}\sum_{t=1}^{T}(EPC_l * p_{l,t}) \tag{3.4}$$

s.t.

Equation (3.2)

Equation (3.3)

The model has $R*T+L*T+L*T$ decision variables. As it is discussed by Taleb and Gupta (1997) and Langella (2007), the complexity of the IP formulation increases exponentially with the amount of leaf and core items, and the length of the planning horizon. This encourages the development of heuristic approaches.

3.3 Heuristic Approaches

In real life, the model described in Section 3.2 should be solved at every period whenever a forecast update is made on a rolling horizon basis. As re-use practices get more popular, we expect a typical disassembly firm to deal with a large number of root and leaf items. In order to provide practical tools for the disassembly plan of these firms, several heuristic approaches are proposed in the literature. These heuristics should provide solutions that are close to the optimal solutions. Besides, they should be easy to be implemented on information systems. It is also important that the practioners can understand the idea and mechanics of the heuristics so that they can modify when needed.

In this thesis, our main objective is to implement the suggestions of Langella (2007) for the combination of heuristics proposed by Taleb and Gupta (1997) according to our approaches for including the cost parameters into the integral algorithm which is going to be described in Section 3.3.2 in detail.

The first heuristic is by Taleb and Gupta (1997). They suggest a two-step heuristic approach for the solution of the problem introduced in Section 3.2. The steps are called ‘core’ and ‘allocation’ algorithms. Allocation algorithm is executed after the core algorithm. Then, Langella (2007) suggests a single step integral algorithm to overcome some shortcomings of Taleb and Gupta (1997)’s approach. In addition to this heuristic, he suggested to further improve the heuristic taking inventory holding and disposal costs and the external leaf procurement option into account. These improvements are mentioned verbally and their application may differ with interpretation. One of our aims is to make a performance assessment for different interpretations. They are explained while providing the algorithms in the forthcoming sections.

3.3.1 Core and Allocation Algorithm

In two-step approach of Taleb and Gupta (1997), first, the number of cores to be disassembled is determined so as to satisfy the total demand over the planning horizon. Then, these aggregate numbers of cores are allocated to the periods. The algorithm determining the aggregate disassembly quantity for roots is called 'core algorithm' and the algorithm allocating these aggregate quantities to the periods of the planning horizon is called 'allocation algorithm'. In Section 3.3.1.1 and Section 3.3.1.2, the details of core and allocation algorithms are discussed, respectively.

3.3.1.1 Core Algorithm

The basic idea of the core algorithm depends on the selection of the source for common items based on the 'attractiveness' measure proposed. The attractiveness of a root is based on its benefit-to-cost ratio, called Contribution Factor. Taleb and Gupta (1997) defines the "benefit" as the percentage decrease in unfulfilled requirements, that is $\frac{\alpha_{r,l}}{\lambda_l}$ where λ_l is the unsatisfied portion of the aggregate demand for leaf item *l* over the entire planning horizon. The "cost" (in the benefit-to-cost ratio) is defined as the percentage increase in the *Total Disassembly Cost* of the root item. Taleb and Gupta (1997) determine the total number of root items to be disassembled over the planning horizon solely based on the priorities coming from the contribution factors.

Our implementation of the contribution factors has some differences compared to Taleb and Gupta (1997). The basic difference occurs in the definition of the benefit term. We define the benefit as the number of leaf item *l* that root item *r* includes, $\alpha_{r,l}$, the decrease in the number of unfulfilled leaf requirement if one unit of root item *r* is disassembled. We do this for the reason that Taleb and Gupta (1997)'s suggestion adds unnecessary operations that is the update of the parameter λ_l at each step of the algorithm. Besides, it has no guarantee to give better solutions with respect to our method. The other difference is the usage of total disassembly cost. We divide the benefit ratio to the total disassembly cost of that root item, not to the percentage increase in that cost. Note that, we re-define the contribution factor as the reciprocal of unit total disassembly cost per leaf item *l* if it is obtained from root item *r.* As a result our contribution factor is $\beta_{r,l} = \dfrac{\alpha_{r,l}}{TDC_r}$.

While determining the aggregate number of each root item to be disassembled, they suggest to start with non-common leaf items first. Notice that for a certain leaf item *l,* if $\sum_{k=1}^{R} \beta_{k,l} = \beta_{r,l}$ for a certain *r,* it means that leaf item *l* is a non-common item which exists only on root *r's* *BOM.* With the assumption of total disassembly costs of *10* and *11* for root items *A* and *B* respectively, contribution factors, $\beta_{r,l} = \dfrac{\alpha_{r,l}}{TDC_r}$, for the example provided in Figure 3.1 are given in Table 3.1.

Table 3.1 Contribution factors of root-leaf pairs and summations of them for each leaf items

Root Items	**Leaf Items**		
	C	**D**	**E**
A	2/10	1/10	0
B	1/11	3/11	2/11
$\sum_{r=1}^{R} \beta_{r,l} = (\beta_l)$	0.29	0.37	0.18

To satisfy the demand of leaf item *E*, only option is to disassemble root item *B*. The number of root item *B* required to be disassembled in order to satisfy the entire demand for *E* is $\left\lceil \dfrac{\lambda_E}{\alpha_{B,E}} \right\rceil$, where $\lceil x \rceil$ represents the smallest integer greater than or equal to *x*.

Notice that roots containing non-common leaves may also contain other leaves as well. Therefore, every time that a disassembly quantity is determined, requirements for these leaves should be updated. For instance, when disassembly quantity for *B* is set to $\left\lceil \frac{\lambda_E}{\alpha_{B,E}} \right\rceil$, the demand for *C* and *D,* which are included in root *B*'s *BOM,* should be updated accordingly. $\lambda_C \leftarrow \max\left\{\lambda_C - \alpha_{B,C} * \left\lceil \frac{\lambda_E}{\alpha_{B,E}} \right\rceil, 0\right\}$ and $\lambda_D \leftarrow \max\left\{\lambda_D - \alpha_{B,D} * \left\lceil \frac{\lambda_E}{\alpha_{B,E}} \right\rceil, 0\right\}$.

After determining the disassembly plan to get non-common leaf items, Taleb and Gupta (1997) suggests to consider the others. To do that, Taleb and Gupta (1997) select the most attractive root-leaf pair, i.e, the highest $\beta_{r,l}$. The disassembly quantity of the root having the highest $\beta_{r,l}$ is determined such that all requirements of the associated leaf item *l* are fully satisfied. Necessary remaining requirement updates are made for the leaves contained in the *BOM* of the root in concern. In addition to this, contribution factors for the leaves that have no demand after the requirement updates are set to zero in order not to consider for the next root selection.

The algorithm continues until all requirements are satisfied for all leaf items. Figure 3.2 gives a flowchart for the main steps of the core algorithm.

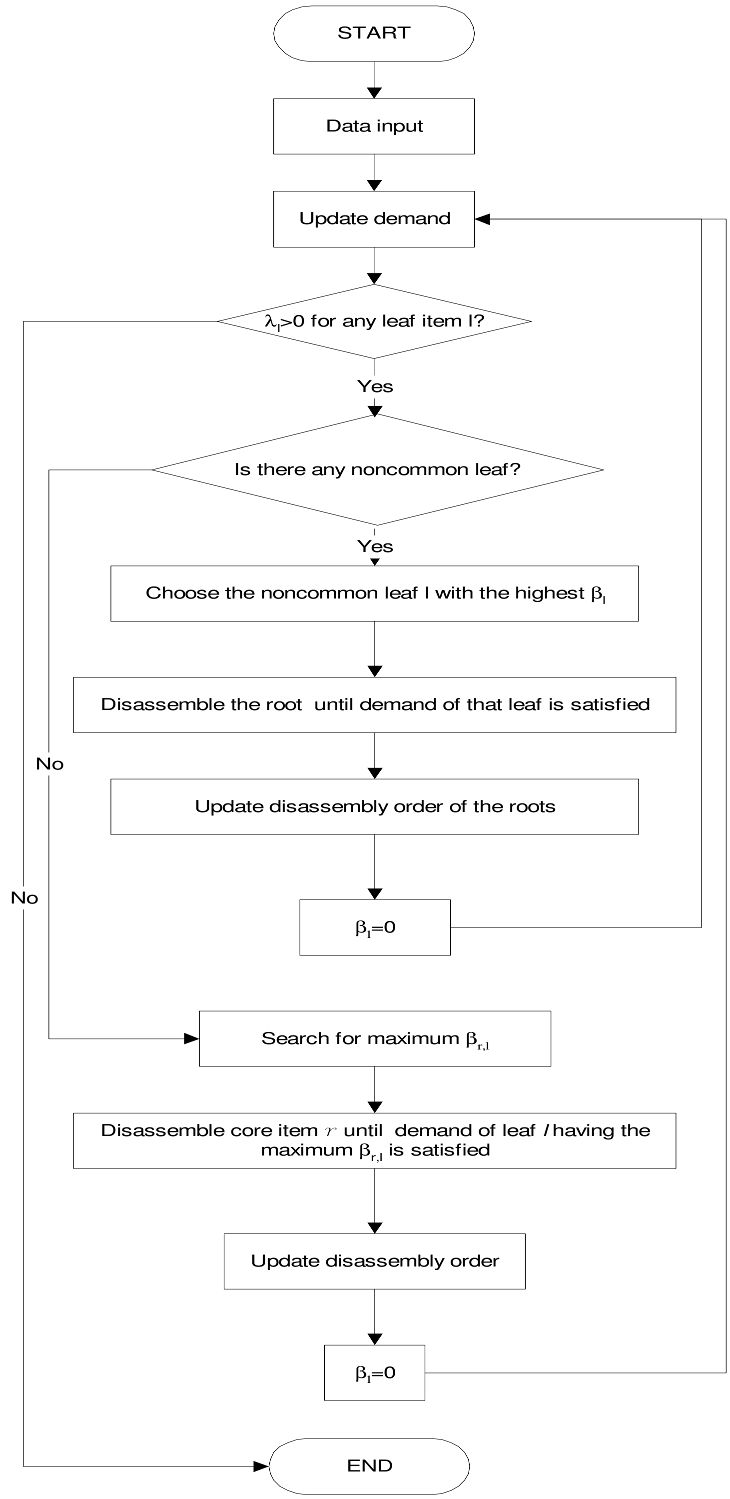

Figure 3.2 Flowchart of the core algorithm

In order to illustrate the core algorithm, suppose total demand over the entire planning horizon is available and given in Table 3.2 for leaf items in the example *BOM* in Figure 3.1. To represent the solution, let $d(A,B)$ be the solution vector representing the number of leaf A and leaf B disassembly quantities and initially $d = (0,0)$.

Table 3.2 Total demands for the leaf items over the planning horizon

	C	**D**	**E**
Updated Demand	15	20	10

First the algorithm satisfies the demand of the non-common leaf item E disassembling $\left\lceil \frac{\lambda_{\mathrm{E}}}{\alpha_{\mathrm{B,E}}} \right\rceil = \left\lceil \frac{10}{2} \right\rceil = 5$ units of root item *B*. $\beta_{B,E}$ is set to zero, since the demand for E is fully satisfied, i.e., there is no need to consider E further. Then algorithm updates the demands as in Table 3.3 and d is updated as (0,5).

Table 3.3 Updated demands of the leaf items after satisfying the demand of the non-common leaf *E*

	C	**D**	**E**
Updated Demand	10	5	0

As the demand of all non-common leaves is satisfied, algorithm examines the common leaf items. It finds the maximum contribution factor as 3/11, the alternative of getting *D* from *B*. 2 more item *B* need to be disassembled since $\left\lceil \frac{\lambda_D}{\alpha_{B,D}} \right\rceil = \left\lceil \frac{5}{3} \right\rceil = 2$.

As there is no demand for leaf *D* from now on, its contribution factors are set to zero and d is updated as (0,7). Again the demands and contribution factors are updated and the respective figures are given in Table 3.4 and Table 3.5. The algorithm updates the demand according to the formula $\lambda_l \leftarrow \lambda_l - \alpha_{r,l} * x_r$, where x_r is the number of root item disassembled to satisfy the demand of its leaf item which has the highest contribution factor.

Table 3.4 Updated demands of the leaf items

	C	D	E
Updated Demand	8	0	0

Table 3.5 Updated contribution factors of leaf items

Root Items	Leaf Items		
	C	D	E
A	2/10	0	0
B	1/11	0	0

To satisfy the remaining requirement of leaf *C*, the highest contribution factor is given by root *A*. $\left\lceil \frac{\lambda_C}{\alpha_{\mathrm{AC}}} \right\rceil = \left\lceil \frac{8}{2} \right\rceil = 4$ is the number of root *A* required to satisfy *C*'s demand and *d* is updated as (4,7). As a result core algorithm finds total disassembly quantities as in Table 3.6.

Table 3.6 Total disassembly quantities of root items

Root Item	Disassembly Quantity
A	4
B	7

3.3.1.2 Allocation Algorithm

Allocation algorithm is executed after the core algorithm in order to spread the disassembly quantities to periods to satisfy the periodic demands. Hence, it uses the output of the core algorithm as an input.

The procedure is similar to the core algorithm. As in the core algorithm, non-common leaf items are determined first. Then, disassembly quantities for the associated root items are determined. After updating the requirements for the current period, the allocation algorithm selects and releases a unit of the most "attractive" root. Again the attractiveness is the highest contribution factor of the root-leaf pairs. After that, the algorithm determines the receipts from disassembly and updates the requirements for the leaves. The previous step is repeated for the same time period until all the requirements for that time period are fulfilled. At that point, the algorithm moves to the next time period after arranging the demands for the next periods according to the inventories held previous period and repeats the same procedure. Once this procedure is completed over the entire planning horizon, the algorithm provides a disassembly schedule for the root items.

To continue to the example above, suppose that the demands over a planning horizon of 3 for all leaf items are given in Table 3.7. Core algorithm finds the result as 4 root *As* and 7 root *Bs* to satisfy the requirements of entire planning horizon. In other words, allocation algorithm has the limit of 4 *As* and 7 *Bs* to satisfy the periodic demands.

Table 3.7 Periodic demands of leaf items

Leaf	**Period**			
	1	**2**	**3**	**Total**
C	3	10	2	**15**
D	9	6	5	**20**
E	6	2	2	**10**

According to the demands, allocation algorithm finds the result in Table 3.8. For the first period, according to the contribution factors, root *B* is chosen and 3 of 7 root *B* is disassembled and the demand of the first period is totally satisfied without any inventory left. Now for the second period, allocation algorithm has 4 root *As* and 4 root *Bs* that it can possibly disassemble. As having a contribution factor of 3/11*,* root *B* is chosen again and 2 of 4 are disassembled. After disassembling 2 root *Bs,* there are only 8 leaf *Cs* remaining to be satisfied. To satisfy them, root *A* is chosen as it has a greater contribution with respect to root *B* for the leaf item *C*, i.e, $\beta_{A,C} > \beta_{B,C}$. 4 root *As* are disassembled which means that allocation algorithm uses all of them and has no root *A* for the last period. For the last period, there are 4 leaf *Ds* and 2 leaf *Es* carried in the inventory. After updating the demand data, the requirement for the last period becomes as 2 leaf *Cs* and 1 leaf *D.* For this period, we have only 2 *Bs* to disassemble. Again root *B* is chosen and one of them is enough to satisfy the demand of leaf *B.* Then, the only remaining demand is 1 leaf *C.* If we have enough quantity of each root, root *A* would be chosen to satisfy it. But root *A's* capacity is fully utilized by the algorithm. Hence, algorithm chooses the other alternative (if there were more than 2 roots, then the algorithm would choose the root item having the second greatest contribution factor) to satisfy leaf *C's* demand. This is the way how the capacity constraint affects the solution and it is one of the deficiencies of core and allocation algorithms.

Table 3.8 Schedule found by the Allocation algorithm

Root	**Period**			
	1	**2**	**3**	**Total**
A	0	4	0	**4**
B	3	2	2	**7**

The complexity of running the Core and the Allocation algorithms sequentially is reported to be $O(L)$ by Taleb and Gupta (1997).

3.3.2 Integral Algorithm

Integral algorithm is suggested by Langella (2007) to overcome some deficits of core and allocation algorithms. It is the combination of the core and allocation algorithm. The logic of the algorithm is same as the core algorithm and it can be thought as the core algorithm applied for each period successively.

Core algorithm does not consider the time aspect and determines the total number of root items according to the total disassembly costs. Allocation algorithm adds the time aspect to the solution of the core algorithm. On the other hand, integral algorithm considers time and cost aspects at the same time.

Allocation algorithm disperses the core items that are found by core algorithm so it has a supply constraint. Integral algorithm is not limited in this manner. It starts from the first period and finds the disassembly quantities of the root items, period by period.

When root items have non-identical lead times, allocation algorithm may result in an infeasible solution. Sometimes total number of one of the roots may not be enough for satisfying total demand of a leaf without disassembling another root. When different lead times exist between these roots involving a common leaf item, it sometimes becomes impossible to satisfy the demand of that leaf item. This may occur for the periods before the maximum lead time. The reason of this case is that the allocation algorithm only uses the roots found by core algorithm which considers total demands. But integral algorithm finds the number of root item for the demand of that period which prevents this kind of infeasibility. Tables 3.9 and 3.10 illustrate this infeasibility.

Table 3.9 Periodic demands of leaf items

Leaf Item	**Periods**				
	1	**2**	**3**	**4**	**Total**
3	0	6	2	1	**9**
4	0	6	3	2	**11**
5	0	0	3	2	**5**

Table 3.10 Yield, $\alpha_{r,l}$, for root-leaf pairs

Root Items	**Leaf Items**		
	3	**4**	**5**
1	1	2	0
2	1	1	2

For this problem instance, core algorithm finds disassembly quantities 4 and 5 for root items 1 and 2, respectively with the assumption of total disassembly costs *11* and *10* for root items 1 and 2, respectively. But if disassembly lead time is 1 and 2 periods for root items 1 and 2, respectively, disassembling 4 units of root item 1, is not sufficient to satisfy the demand of leaf item 3 in period 2. This infeasibility can be solved only by approaching periodically.

Integral algorithm solves this infeasibility in most of the time, but if we assume that there exists any demand in the first $L_{\min} = \min_r \{LT_r\}$ periods with all other data the same, then there is no root item to be disassembled for the first $L_{\min}$ periods' demand since they have lead times. Even the integral algorithm cannot solve this problem because Langella (2007) does not include external procurement option which is the only way to solve this kind of infeasibility. For instance, there is no option to satisfy the demand of the first period in the previous example if exists. Therefore, integral algorithm should be used whenever there is no demand for any root in the first $L_{\min}$ periods of the planning horizon.

The logic of the Integral algorithm is same as the Core algorithm but it finds schedules period by period so time-complexity of the Integral algorithm is $O(L*T)$. Steps of the Integral algorithm are provided in Appendix A.

3.3.3 Deficiencies of the Algorithms

These algorithms are very simple to execute even for large number of root and leaf items and their logic is very simple as it has few criteria. But they have some deficiencies that cause them to deviate from the optimal solution.

- They consider only total disassembly costs of the root items and the total demands of the leaf items. As they do not consider the inventory holding costs as one of the decision criteria, solutions may lead inventory accumulation.
- They are not appropriate for long horizon problems because when the planning horizon gets larger, the possibility of deviation from the optimal solution increases. The reason of this deviation is the accumulated inventories by periods.
- None of the algorithms includes the external leaf procurement option. As the core algorithm do not consider the periods of the planning horizon separately, allocation of the root items to early periods sometimes gives infeasible results when lead time exists. Similarly, in some cases it is impossible to satisfy a leaf item's demand because of lead times and as mentioned before, even integral algorithm cannot solve this problem because the only way to solve this infeasibility is to consider the external procurement option in the heuristic.
- In addition to fixing infeasibility, external procurement option may decrease the inventory costs. One of the reasons for high inventory levels yielded by these heuristics is the lack of external procurement option.

3.4 Improvements on Integral Algorithm

As mentioned above, the algorithms above only consider the total disassembly costs in finding a disassembly schedule. But as the inventory holding cost is not taken into consideration and the demand is satisfied only by root disassembly, solution may include high inventory levels that lead to high costs. To avoid high inventory carrying costs, as Langella (2007) suggested, some additional considerations, such as inventory holding, disposal and external procurement, must be included in the disassembly planning.

Most important of them is the external procurement option which may decrease the total cost over the planning horizon by decreasing both disassembly quantities and total inventory holding cost depending on the specific cost parameters. Langella (2007) compares unit disassembly and external procurement costs at the beginning and if external procurement option is cheaper with respect to his criteria, then the demand of leaf item(s) may be fully satisfied with only this option. In our suggestion, basic aim is to disassemble as much root items as possible, which is also the aim of a disassembly plant. And we consider the external procurement option according to the excess number of leaves after the disassembly schedule is determined. This approach not only disassembles more root items, but also expected to decrease the inventory levels.

Langella (2007) suggest to include all relevant costs of inventory holding and recovery cost of leaf items into TDC_r, so these costs directly affect the contribution factors. By this way, he makes them included in the selection criteria for root items. He puts inventory costs for each leaf item into TDC_r before checking whether the disassembly of a root causes inventories for leaves or not. In other words, he adds an inventory cost at the beginning when it is not certain for this cost to occur or not. But we did not include this cost into TDC_r and calculate the inventory costs if there are excess leaf items after the disassembly which is decided by the total disassembly costs only. Then after disassembly quantities are determined, total inventory holding cost of the excess items are added to the total disassembly cost. The comparison of external procurement and disassembly is made in this level, not in the beginning of the determination of root disassembly as Langella (2007) suggests.

We construct our heuristic in two ways for inventory holding cost. Both of them calculate the inventory holding cost after the disassembly schedule is determined. The first one is myopic, in which the inventory holding cost is calculated for only the first period after disassembly:

$$IHC_{l,t} = EX_{l,t} * IC_l \text{.} \tag{3.5}$$

In the second variant, the possibility of carrying some of this excess quantity more than one period is considered. That is;

$$IHC_{l,t} = IC_l * EX_{l,t} + \sum_{k=t+1}^{T} IC_l * \left(\max \left\{ 0, EX_{l,t} - \sum_{p=t+1}^{k} \lambda_{l,p} \right\} \right), \tag{3.6}$$

where $IHC_{l,t}$ is the inventory holding cost of leaf item *l* incurred in period *t.*

Considering the deficiencies of the previous algorithms mentioned in the previous sections and Langella (2007)'s suggestions, we construct a new heuristic approach called *Myopic NC-first* heuristic and its variant in terms of inventory holding cost calculation which is called *Non-Myopic NC-first* heuristic. Equation (3.5) is utilized in the inventory holding cost calculation of the *Myopic NC-first* algorithm where Equation (3.6) is used in the *Non-myopic NC-first* algorithm. These two heuristics give priority to the non-common leaf items in the order that the requirements are satisfied as integral algorithm does. We include this property in labeling the heuristics, because we relax this prioritization based on the observations on the performance. These are discussed in Section 3.5.

3.4.1 Overall Logic of the *Myopic NC-first* and *Non-myopic NC-first* Algorithms

Algorithms select the minimum-cost periodic schedule by considering the external procurement option and comparing the corresponding cost with respect to disassembly and inventory holding cost. As the Integral algorithm, it finds the schedule period by period. Algorithms use the logic of integral algorithm before deciding if external procurement should be chosen instead of disassembly. Hence, they either use the schedule found by integral algorithm or their own schedule if it provides a cost decrease. Steps of the *Myopic NC-first* and *Non-myopic NC-first* algorithms are provided in Appendix B

Algorithm starts with determining the leaf items whose demands cannot be satisfied with the root disassembly because of the lead times. After it determines the quantity of the external procurement of leaf items, it turns to the first period. Then it starts to search the appropriate roots to be disassembled to satisfy the demands of the current period. Afterwards, disassembly quantities of root items are tried to be decreased.

When a root item is withdrawn, i.e., its disassembly quantity is decreased, lacks are occurred in the leaf items, that is quantity of some leaf items becomes insufficient to satisfy its demand. The missing leaf item demands are fulfilled by external procurement option if this option results in a cost decrease.

Withdrawal of part starts with the calculation of the excess quantity of leaf items for the current period *t*, $EX_{l,t}$ for $l = 1,...,L$. This part resembles the integral algorithm in that non-common leaf items are considered first and the one with the highest selection ratio is chosen. Then, the disassembly quantity of the root item which involves that leaf item is decreased and the missing leaf items are filled by procurement. If the cost of this schedule is cheaper than the integral algorithm's schedule, the withdrawal continues until a larger cost is encountered, i.e., Step 5 is repeated until $\Delta > 0$, or disassembly quantity of that root item is set to zero by withdrawal. This method is applied to each non-common leaf item. If a decrease in the total cost is obtained by external procurement instead of root disassembly, then it is applied to the current period and excess leaf item quantities are updated according to the new disassembly and item procurement schedule.

Second step of this part is to check the common leaf items. Most of the problems involve more than one common leaf item. Likewise, real life problems involve more than two root items involving these leave items. Consequently, a criterion must be determined to choose a root-leaf pair. This criterion is called the selection ratio, $\beta_{r,l}$. Selection ratios are completely the same as the contribution factors used in integral algorithm but this time a root-leaf pair having the minimum ratio is selected due to the fact that root item having the minimum ratio for a leaf item has the minimum effect for satisfying that leaf items demand. The decision procedure is the same and if the external procurement is more profitable with respect to disassembly, it is applied instead of the disassembly of that root item. All common leaves are checked one by one starting from the lowest selection ratio. The flowchart of the withdrawal algorithm of the heuristics is given in Figure 3.3. This algorithm is applied just after the integral algorithm finds the disassembly quantities for a period.

Time complexity of *Myopic NC-first* and *Non-myopic NC-first* algorithms is heavily depends on problem data. It is equal to $O\left(T * L * \max\left\{R, LT, \max_{t,l}\left\{\lambda_{l,t}\right\}\right\}\right)$. The complexity is the same for two variants that are introduced in Section 3.5.

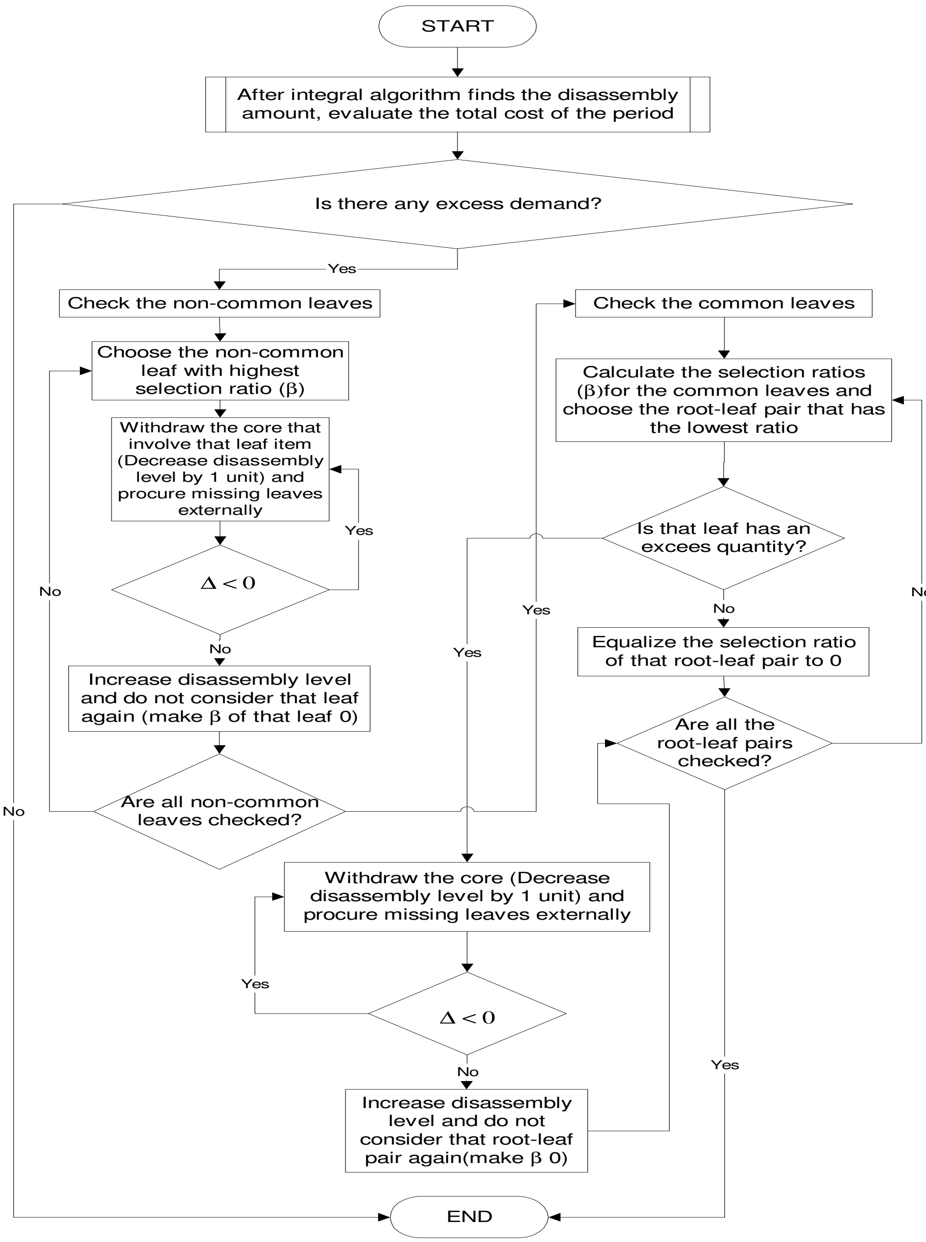

Figure 3.3 Flowchart of the *Myopic NC-first's* and *Non-myopic NC-first's* withdrawal algorithm

3.4.2 An Example

To illustrate the *Myopic NC-first* algorithm on an example, consider a problem instance for which the relevant costs are in Table 3.11. The *BOM* structure is the same as Figure 3.1 and the demand data is taken as in Table 3.7.

Table 3.11 Relevant cost used in *Myopic NC-first* algorithm

Items	Costs		
	Total disassembly	**Inventory holding**	**External Procurement**
A	10	-	-
B	11	-	-
C	-	2	8
D	-	2	8
E	-	2	8

There is no excess demand in the first period so the schedule is not changed. But for the second period integral algorithm finds the disassembly quantities as (4,2) resulting in excess quantities 0, 4, 2 for leaf items *C, D* and *E* respectively. The cost of this schedule is found by the formula $\sum_r S_{r,t} * TDC_r + \sum_l EX_l * IC_l$ and that is $4*10+2*11+4*2+2*2=74$. Because the non-common leaf item *E* has an excess of 2 units, the quantity of root item *B* is decreased 1 unit. This causes leaf *C* miss 1 unit. If that leaf is externally procured, according to the formula $\sum_r S_{r,t} * TDC_r + \sum_l EX_l * IC_l + \sum_l EP_l * EPC_l$ for $\forall\ (r,l)$, total cost becomes $4*10+1*11+1*2+1*8=61$. As a result external procurement to leaf *C* is applied with a cost decrease of 13. After this, there still exists an excess leaf of *D*. Root *A*, having the minimum ratio (1/11 < 3/10) is withdrawn and 2 units of leaf *C* is procured. But this time it does not decrease the total cost and it is not applied. Now, there is no any unchecked leaf item and *D* has an excess quantity of 1 unit in the second period. This makes the requirements of the third period equal to 2, 4 and 2 for leaf items *C, D* and *E,* respectively. Then after the next period's demands are updated, integral algorithm is applied and 2 *Bs* are disassembled which causes excess of 2 *Ds* and 2 *Es.* As non-common leaf item *E* has an excess, root item *B* is withdrawn 1 unit resulting in missing leaves of *C* and *D* by 1 unit each. Procurement option decreases the total cost of this period by 3 (calculations are totally the same as the second period's calculations), so it is applied. At the moment, there are no excess leaf items and the algorithm does not try to withdraw the other root *B*, so it finishes with the results shown in Tables 3.12, Table 3.13 and Table 3.14. In addition to the algorithms results, optimal planning schedule is provided in these tables.

Table 3.12 Disassembly schedules found by *Myopic NC-first* algorithm and the IP model.

Root Items	***Myopic NC-first***			**Total**	**Optimal**			**Total**
	Period 1	Period 2	Period 3		Period 1	Period 2	Period 3	
A	0	4	0	**4**	0	5	0	**5**
B	3	1	1	**5**	3	1	1	**5**

Table 3.13 External procurement schedules found by *Myopic NC-first* algorithm and the IP model.

Leaf Items	***Myopic NC-first***			**Total**	**Optimal**			**Total**
	Period 1	Period 2	Period 3		Period 1	Period 2	Period 3	
C	0	1	1	**2**	0	0	0	**0**
D	0	0	1	**1**	0	0	0	**0**
E	0	0	0	**0**	0	0	0	**0**

Table 3.14 Leaf item inventory schedules found by *Myopic NC-first* algorithm and the IP model.

Leaf Items	***Myopic NC-first***			**Total**	**Optimal**			**Total**
	Period 1	Period 2	Period 3		Period 1	Period 2	Period 3	
C	0	0	0	**0**	0	1	0	**1**
D	0	1	0	**1**	0	2	0	**2**
E	0	0	0	**0**	0	0	0	**0**

According to the values in Table 3.12, Table 3.13 and Table 3.14, total cost of the planning horizon is calculated as 121 and 111 by the *Myopic NC-first* algorithm and the IP model. For the same instance integral algorithm finds the disassembly and inventory holding schedule as in Table 3.15 and Table 3.16 with a total cost of 140. Note that *Myopic NC-first* algorithm improves the cost of the planning horizon found by the integral algorithm nearly 13.6%. And its solution has an error deviation of $\left(\frac{heuristic-optimal}{optimal}\right)=\left(\frac{121-111}{111}\right)*100=9.01\%$ with respect to the optimal cost of the planning horizon.

Table 3.15 Disassembly schedule found by Integral algorithm

Root Items	**Periods**			
	1	**2**	**3**	**Total**
A	0	4	1	**5**
B	3	2	1	**6**

Table 3.16 Leaf item inventory schedule found by Integral algorithm

Leaf Items	**Periods**			
	1	**2**	**3**	**Total**
C	0	0	1	**1**
D	0	4	3	**7**
E	0	2	2	**4**

3.5 Computational Study

3.5.1 Experimental Setting

In order to assess the performance of the proposed heuristic, we carried out a computational study. We consider a similar experimental setting to the one considered by Langella (2007) for our base case. 100 problem instances are generated in the same manner as he does. The parameter values for the problem instances are set as follows:

1. Number of root and leaf items, R=2, L=3
2. Planning horizon, T=12
3. Leaf procurement costs, EPC_l for $l = 1,...,L$, are drawn from a discrete uniform distribution with minimum and maximum values of 1 and 10, respectively.
4. Leaf holding cost for leaf l, IC_l, is 10% of the corresponding leaf procurement costs, EPC_l
5. Yield of root r for leaf l $\alpha_{r,l}$ are drawn from discrete uniform distribution between 0 and 3
6. Per period demand for leaves is generated using the normal distribution. The mean demand, $\mu_{\lambda,l}$, is set as: $\mu_{\lambda,l} = 100 * \sum_{r=1}^{R} \alpha_{r,l}$.

   The variance of per period demand, $\sigma^2_{\lambda,l}$, is set to $\mu_{\lambda,l} / 3$ for leaf $l = 1,...,L$.

Realizations which are less than zero, is set to zero.

7. Total disassembly cost, TDC_r, for root r is obtained from normal distribution. The mean disassembly cost, $\mu_{TDC,r}$ for root item $r = 1,...,R$ is set as:

$$\mu_{TDC,r} = \frac{\sum_{l=1}^{L} \alpha_{r,l} EPC_l}{2} \tag{3.7}$$

The variance of total disassembly cost, $\sigma^2_{TDC,r}$, is set to $\mu_{TDC,r}/3$ for root $r=1,...,R$. Realizations which are less than 1 is set to 1.

8. Disassembly lead time is set to zero for all roots.

After generating the instances, the instances where there is a leaf which cannot be obtained disassembling any of the roots, and for which there are root items containing no leaf items are deleted.

This experimental setting is the same as Langella (2007) except the followings:

- Yields are set from discrete uniform between 0 and 2 for all leaf root pairs by Langella (2007).
- Mean demand for leaves is set in the same manner, however the variance of the demand is equal to the mean by Langella (2007).
- Mean of total disassembly cost is set in the same manner, however the variance of total disassembly cost is equal to the mean by Langella (2007).

Due to the changes that we made on the experimental setting by Langella (2007), our instances include more commonality (since the probability that the yield is zero is less in our case) compared to his settings. In addition, demand shows less variability, and the probability that a negative total disassembly cost occurs is less compared to his study.

In order to assess the sensitivity of the proposed method to the problem parameters, 13 more data sets are constructed. While generating these problem sets, the way that only one or two parameters generated are changed at a time. The base case explained above is labeled as S1, and the other problem sets are labeled from S2 to S14. The way that these problem sets are generated is as follows:

- The problem sets S2-S4 are generated to assess the effect of relative value of total disassembly cost to the procurement costs.

S2: The problem instances have the same parameter values as S1, except, while generating the total cost of disassembly parameter, the mean value is set to ¼ of total procurement cost of the root, i.e., in Equation (3.7) 2 is replaced by 4.

S3: The problem instances have the same parameter values as S1, except, while generating the total cost of disassembly parameters, the variance is set to 1/5 of the respective mean value.

S4: The problem instances have the same parameter values as S3, except, while generating the total cost of disassembly parameter, the mean value is set to ¼ of total procurement cost of the root, i.e., in equation (3.7) 2 is replaced by 4.

- The problem sets S5-S6 are generated to assess the effect of problem size in terms of number of root and leaf items.
  S5: Same as S1, except R=4, L=6
  S6: Same as S4, except R=4, L=6

- The problem sets S7-S14 are generated to assess the trend in demand data. While generating these problem sets, on the generated demand data for S1, we include both positive and negative trend which is equal to either 10%, or 20% of the respective mean demand keeping the total demand over 12 periods as in S1 (or in S5). These data sets are:

  S7: Same as S1, except there is a 10% increasing trend in demand data.
  S8: Same as S1, except there is a 10% decreasing trend in demand data.
  S9: Same as S4, except there is a 10% decreasing trend in demand data.
  S10: Same as S4, except there is a 10% increasing trend in demand data.
  S11: Same as S1, except there is a 20% increasing trend in demand data.
  S12: Same as S1, except there is a 20% decreasing trend in demand data.

S13: Same as S4, except there is a 20% decreasing trend in demand data.

S14: Same as S4, except there is a 20% decreasing trend in demand data.

Although we generated the problem instances for 12 periods, considering the demand for the first 4 and 6 periods demands, we solve the instances for T=4 and T=6, in addition to T=12.

3.5.2 Discussion of the Results

All problem instances are solved by the heuristics explained in Section 3.4 (by both myopic and non-myopic approaches) and optimally as well. CPLEX is used for getting the optimal solutions.

When we examined percent deviation from optimal objective function value of heuristic solutions, we observed surprisingly big errors in some of the results and the related data for the most conspicuous one are given in Table 3.17 and Table 3.18. Table 3.19 summarizes the performance of the two algorithms and their two variants for this problem instance.

Table 3.17 Disassembly costs and number of leaves disassembled from roots

Roots	**Disassembly cost**	$\alpha_{r,l}$		
		Leaf 1	Leaf 2	Leaf 3
1	4.87	2	2	0
2	11.64	2	1	1

Table 3.18 Inventory holding and external procurement costs of leaf items

Leaf Items	**Inventory Holding Cost**	**External Procurement Cost**
1	0.302	3.02
2	0.918	9.18
3	0.107	1.07

Table 3.19 Optimal and heuristic approaches' results for the problem

	Total Cost	**% Deviation From the Optimal Cost**
Optimal	11574.72	
Myopic NC-first	30997.41	167.80
Non-myopic NC-first	30997.41	167.80
Myopic	11576.50	0.02
Non-myopic	11574.72	0.00

We suspect that the reason behind the high errors of *Myopic NC-first* and *Non-myopic NC-first* is due to giving priority to non-common items in the sequence that the solution is generated. Therefore, two versions of heuristics are constructed without giving the priority to non-common items. In other words, in the Integral algorithm provided in Section 3.3, Step 3 is omitted, and Step 4 is done for considering both common and non-common items instead of only common items. The versions based on the calculation of inventory holding costs, heuristics are named as *Myopic* and *Non-Myopic*.

All the problem instances are solved by this new idea with and without myopic inventory holding cost considerations. We report the results with the following convention of referring to these heuristics.

Myopic NC-first: The algorithm provided in Section 3.4 with myopic inventory holding cost consideration and priority of non-common leaf items.
Myopic: The same with *Myopic NC-first* in inventory holding cost consideration but do not give priority to non-common leaves.
Non-myopic NC-first: The algorithm provided in Section 3.4 with non-myopic inventory holding cost consideration and priority of non-common leaf items.
Non-myopic: The same with *Non-myopic NC-first* in inventory holding cost consideration but do not give priority to non-common leaves.

Myopic heuristic gives the worst average solutions except for some of the 4 and 6 period problems. *Non-myopic* gives the second worst solutions with respect to average behavior. This is expected because these two heuristics improve only some types of problems whose solutions have higher errors by beginning with non-common leaf items first.

For each set of problem instances, we report minimum, maximum and average percent deviation from optimal costs for all 4 heuristics in Table 3.21, Table 3.22 and Table 3.23 for T=4, T=6 and T=12, respectively, together with the best of 4 heuristic solutions. In addition, distribution of errors for 4 heuristics and for T=4, T=6 and T=12 are provided in Appendix C.

Since in almost all cases *Myopic-NC First* and *Non-myopic NC First* provide better results, for our observations we restrict ourselves to these two heuristics.

Table 3.20. Results of the algorithms for 4 period samples

% Errors	*Myopic NC-first*			*Myopic*			*Non-myopic NC-first*			*Non-myopic*			**Best of 4**		
	Max	**Min**	**Avg.**	**Max**	**Min**	**Avg.**	**Max**	**Min**	**Avg.**	**Max**	**Min**	**Avg.**	**Max**	**Min**	**Avg.**
S1	29.99	0	4.51	54.15	0	6.46	29.99	0	4.57	54.15	0	7.16	19.74	0	2.65
S2	41.09	0	4.93	77.87	0	7.52	19.81	0	3.41	92.39	0	7.46	19.74	0	2.68
S3	38.19	0	4.07	55.96	0	6.30	38.19	0	4.08	55.96	0	6.46	38.19	0	2.72
S4	26.18	0	4.93	71.67	0	6.44	26.17	0	3.21	71.67	0	6.05	17.11	0	2.51
S5	62.93	0	9.78	62.93	0	10.13	62.93	0	9.69	62.93	0	10.03	62.93	0	9.28
S6	84.91	0	15.52	84.91	0	17.10	85.49	0.01	16.03	85.49	0.01	17.73	84.91	0.01	14.76
S7	32.19	0	5.15	52.63	0	6.85	32.19	0	5.23	52.63	0	7.48	32.19	0	3.45
S8	35.37	0	4.34	56.48	0	6.17	35.37	0	4.43	58.62	0	7.07	16.05	0	2.36
S9	23.89	0	4.14	73.30	0	6.59	23.85	0	4.04	73.30	0	6.28	16.71	0	2.46
S10	22.38	0	4.05	68.49	0	6.29	17.37	0	2.93	68.49	0	5.70	17.37	0	2.49
S11	40.44	0	6.50	59.99	0	8.06	40.44	0	6.64	59.99	0	8.59	40.44	0	5.12
S12	35.50	0	4.21	57.84	0	6.13	35.50	0	4.31	60..58	0	7.07	18.93	0	2.21
S13	24.97	0	4.18	74.29	0	6.71	24.97	0	3.30	74.29	0	6.56	16.41	0	2.48
S14	40.18	0	5.09	59.53	0	7.17	40.18	0	4.18	59.53	0	6.51	40.18	0	3.85

Table 3.21. Results of the algorithms for 6 period samples

% Errors	***Myopic NC-first***			***Myopic***			***Non-myopic NC-first***			***Non-myopic***			**Best of 4**		
	Max	**Min**	**Avg.**	**Max**	**Min**	**Avg.**	**Max**	**Min**	**Avg.**	**Max**	**Min**	**Avg.**	**Max**	**Min**	**Avg.**
S1	30.06	0	6.41	55.65	0	6.99	30.06	0	4.98	55.65	0	7.67	20.35	0	3.02
S2	68.45	0	8.16	97.26	0	10.87	27.14	0	4.97	112.0	0	9.47	17.77	0	3.86
S3	37.45	0	4.44	58.69	0	6.64	37.45	0	4.27	58.69	0	6.78	37.45	0	2.95
S4	38.68	0	7.51	65.22	0	8.82	31.12	0	4.61	65.22	0	7.59	17.19	0	3.57
S5	91.49	0.10	10.79	65.03	0.10	11.02	65.27	0.10	10.59	65.27	0.10	10.83	65.03	0.10	9.98
S6	65.03	0.66	17.44	91.49	0.11	18.76	90.77	0.66	17.75	90.77	0.11	19.27	90.77	0.11	16.11
S7	30.69	0	5.34	53.26	0	7.41	30.69	0	5.23	53.26	0	7.82	21.29	0	3.53
S8	35.82	0	4.90	57.09	0	6.74	35.82	0	4.86	61.12	0	7.56	21.44	0	2.72
S9	40.84	0	7.02	69.08	0	9.01	34.55	0	4.73	69.08	0	7.73	20.14	0	3.50
S10	34.25	0	6.24	62.74	0	8.47	20.14	0	4.12	62.74	0	7.13	14.45	0	3.47
S11	29.71	0	5.61	50.89	0	7.91	29.71	0	5.64	50.89	0	8.26	24.49	0	4.10
S12	35.85	0	4.76	58.07	0	6.61	35.85	0	4.74	64.41	0	7.51	22.81	0	2.54
S13	41.75	0	7.23	71.69	0	9.17	37.98	0	4.84	71.69	0	7.95	22.12	0	3.52
S14	26.44	0	5.66	66.89	0	8.19	16.43	0	4.45	66.89	0	7.56	16.43	0	3.94

Table 3.22. Results of the algorithms for 12 period samples

% Errors	***Myopic NC-first***			***Myopic***			***Non-myopic NC-first***			***Non-myopic***			**Best of 4**		
	Max	**Min**	**Avg.**	**Max**	**Min**	**Avg.**	**Max**	**Min**	**Avg.**	**Max**	**Min**	**Avg.**	**Max**	**Min**	**Avg.**
S1	46.50	0	6.65	57.38	0	8.11	30.00	0	5.61	57.38	0	7.86	26.49	0	3.56
S2	163.97	0	17.62	167.36	0	20.32	52.48	0	8.46	143.2	0	12.28	42.67	0	6.32
S3	49.42	0	6.15	56.35	0	7.70	46.61	0	5.91	56.35	0	7.43	46.61	0	3.77
S4	62.48	0	14.72	64.56	0	15.37	60.16	0	7.98	64.56	0	9.87	25.93	0	5.90
S5	68.15	0.49	13.60	68.15	0.49	13.93	69.50	0.49	12.17	69.50	0.49	12.54	68.15	0.49	11.62
S6	124.9	1.67	24.51	124.9	1.31	25.53	109.7	2.50	21.68	109.7	1.31	23.09	107.0	1.31	19.75
S7	46.37	0	6.10	57.47	0	7.87	31.82	0	5.42	57.47	0	7.57	28.40	0	3.42
S8	49.22	0	7.26	57.10	0	8.49	34.07	0	5.92	57.10	0	8.36	24.50	0	3.78
S9	79.46	0	17.79	79.46	0	17.89	70.43	0	9.70	68.62	0	10.37	29.20	0	6.33
S10	47.65	0	11.69	66.63	0	13.02	46.51	0	6.94	66.63	0	9.19	19.66	0	5.31
S11	35.98	0	5.69	57.87	0	7.63	33.10	0.01	5.32	57.87	0	7.45	30.78	0	3.34
S12	56.88	0	7.57	56.40	0	8.71	36.31	0	5.92	64.74	0	8.64	23.88	0	3.73
S13	96.68	0	20.73	96.68	0	20.26	76.61	0	9.33	72.88	0	11.02	31.81	0	6.76
S14	36.83	0	9.12	69.79	0	11.00	36.82	0	6.22	69.79	0	8.86	21.67	0	4.88

Langella (2007) mentions that his heuristic approach gives the average error of 4.55% with respect to the optimal solution. He run the heuristic for only the case involving 2 root items, 3 leaf items and 4 periods. Note that for 4 period planning horizons, either *Myopic NC-first* or *Non-myopic NC-first* algorithm provides better results compared to Langella (2007)'s errors. In addition to this when the solutions of the *Myopic* and *Non-myopic* solution are taken into account, the best solution of these four heuristics decreases the percentage error to 2.21 % in one of the sample sets. Not only for the 4-period problems, but also for the 12-period problems, the best solution of the heuristics results in lower errors. That is the maximum value is 6.76% in S13 where the minimum average error is 3.34% in S11.

3.5.2.1 Effect of the Planning Horizon

As *Myopic NC-first* and *Non-myopic NC-first* algorithms have different approaches for inventory holding cost, they result in different error ranges according to the number of periods in the planning horizon. Generally for the long horizon problems (12 for our problem samples) lower results are expected with *Non-myopic NC-first* heuristic with respect to the *Myopic NC-first* heuristic. The reason of this is the effect of the inventory holding cost is larger for long horizons when leaf item inventory is accumulated and increases period by period. In another aspect, the effect of the cost may not be detected in short periods, so *Non-myopic NC-first* may not improve the solutions for short periods. This fact, i.e., larger inventory holding cost in the period, leads the *Non-myopic NC-first* algorithm externally procure leaf items and this decreases the inventory levels which is the basic result of the high costs. *Myopic NC-first* algorithm cannot detect inventory accumulation, so especially for long horizons it gives larger errors than *Non-myopic NC-first* algorithm gives.

In our computational study, we make observations to support these initial expectations. For the problems with 4-period planning horizon, we do not have the dominance of non-myopic approach. Whereas for longer horizon problems, non-myopic approach dominates the myopic one, unless the demand structure and cost parameters provide an opposite trade-off. These are discussed in related subsections.

3.5.2.2 Observations for the Samples with No Trends on Their Demands

- For S2 and S4, which have smaller disassembly costs, *Non-myopic NC-first* algorithm always gives better average results. When calculating inventory holding costs myopically, disassembly is seen more profitable compared to procurement option because of low disassembly costs and this leads to high inventory quantities and costs. However, when the effect of the inventory holding cost extended into periods (non-myopic approach), external procurement option becomes a profitable option (because it is compared with a higher cost term in *Non-myopic NC-first* algorithm). So, *Non-myopic NC-first* decreases the total planning horizon cost by decreasing the inventory levels and increasing the external procurement levels of leaf items.

- For S1 and S3, the reverse logic is more effective for short planning horizons. That is when disassembly costs are higher with respect to the external procurement costs, inventory carrying becomes more profitable option for short horizons. As the planning horizon becomes shorter, *Myopic NC-first* generates better results than *Non-myopic NC-first*. But the difference between their average results is very small S1 and S3 for 4 periods planning horizon instances.

- For the long horizon problems, we generally expect *Non-myopic NC-first* algorithm to give better results compared to *Myopic NC-first algorithm*. The reason is *Non-myopic NC-first* algorithm prevents high amount of inventories. But looking myopically, *Myopic NC-first* algorithm cannot prevent this inventory accumulation. But for the short period problems, carrying inventory may be the cheaper choice because inventories are carried for shorter times. External procurement option is still seems more profitable to *Non-myopic NC-first* algorithm for short horizons. But especially for S1 and S3, which have higher disassembly costs, *Myopic NC-first* algorithm gives better solutions by carrying more inventory. Our expectation comes true for the first 4 problem samples which have no increasing or decreasing demand trend.

3.5.2.3 Observations for the Samples Having Decreasing Trend Demand Data

- In S8 and S12 where we have negative trend in demand data and disassembly costs are relatively higher, *Myopic NC-first* algorithm gives more erroneous results as planning horizon gets larger. We also faced with this situation in S1. For the decreasing trend, more inventory is carried at first periods but the percentage error with respect to the demands is not so high because the disassembly amount is large to satisfy the demand. However as the planning horizon increases and the demand decreases, percentage of the inventories with respect to the disassembly and external procurement quantities increase. Hence, for larger periods *Myopic NC-first* algorithm gives bigger errors as it gives for the examples without trends in demands.

- For S9 and S13 where we have negative trend in demand data and disassembly costs are relatively smaller. In this case, increase in the error between 6 and 12 period problems is more than in S8 and S12. Low disassembly cost is the reason of this bigger difference because these costs orients *Myopic NC-first* algorithm to disassembly, which lead great inventory holding costs for the later periods. Small errors of *Non-myopic NC-first* algorithm compared to *Myopic NC-first* algorithm for 12 period problems of sample sets S9 and S13 proves our idea.

- For S8 and S12, errors of the *Non-myopic NC-first* algorithm increases as the planning horizon gets longer, too. In these sample sets, *Myopic NC-first* gives better solutions for the 4 periods planning horizon. The amount of increase in the solutions of *Non-myopic NC-first* is smaller than the *Myopic NC-first* algorithm's solutions. Hence we can conclude that, for longer horizon problems, the *Non-myopic NC-first* algorithm gives better solutions than *Myopic NC-first* algorithm, whereas for the short horizons *Myopic NC-first* gives just a little better average percent error.

3.5.2.4 Observations for the Samples Having Increasing Trend Demand Data

- For problem sets with increasing trends, *Non-myopic NC-first* algorithm provides better solutions for the log horizon problems. But for the short planning horizons, the difference of the solution of the *Myopic NC-first* and *Non-myopic NC-first* is negligible. As the demands are very low in first periods, inventory accumulation is not so high when compared to the decreasing trend demand data sample sets. Hence, the inventory accumulation does not affect the solutions seriously in short periods. For the longer horizons, inventory accumulation begins to increase after the mid-periods, so *Non-myopic NC-first* becomes the better alternative for these problems. As the effect of the inventory accumulation is limited by the planning horizon, the results of the two algorithms do not differs so much in average error values.

- For sample sets S10 and S14, which have lower disassembly costs, errors of the *Myopic NC-first* is high with respect to the sample sets S7 and S11. As the disassembly cost decreases, root disassembly seems more profitable option for *Myopic NC-first* but *Non-myopic NC-first* detects the advantage of external procurement option for these sample sets. And as *Myopic NC-first* prefers more disassembly in these samples, its average error values increase compared to *Non-myopic NC-first.* The resemblance of the error deviations between S8-S9 and S12-S13 proves our observation.

3.5.2.5 Observations for the Larger Problem Sets in terms of Number of Root-Leaf Items

- For these samples the average error values of *Myopic NC-first* and *Myopic* algorithms are very close to each other. Having four roots in the sample, the possibility of including a non-common leaf for a leaf item is very small. This causes two algorithms resemble to each other and find the same result in most of the problems.

- As the number of root and leaf items is more than the other samples, there are more feasible solutions. As a result, algorithm may choose different root items than the optimal solution chooses. So for samples S5 and S6, the algorithms' mistake possibility larger than the other samples. Because of this reason, it can be easily said that more root and leaf items causes bigger errors.

CHAPTER 4

DISASSEMBLY LOT SIZING

In Chapter 3, we consider the problem of determining the number of roots to be disassembled and brand new leaves to be externally procured in order to satisfy the demand for the leaf items. This problem corresponds to buy or make decisions that can be considered within the context of master production schedule in the traditional production planning hierarchy.

In this chapter, we consider lot sizing decisions for a disassembly firm. Lot sizing need arises from the fact that disassembly operations require a set up of the disassembly resources, like machinery, or work center that carries out the disassembly operations. Therefore, the model that we investigate in this chapter is more operational compared to the one in Chapter 3.

In this chapter, we restrict our attention to the environment where there is only disassembly option available for satisfying the demand for the leaf items. We include the set-up costs of disassembly of different root items, but ignore the set-up time. Our main objective is to investigate the performance of traditional lot sizing heuristics, namely Wagner-Whitin (WW) algorithm (which provides the optimal for the traditional problem), Silver-Meal (SM) and Least Unit Cost (LUC) when they run incorporation with the heuristics developed in Chapter 3. Notice that since we exclude the regular procurement option of leaf items, the heuristic approaches developed in Chapter 3 resemble the heuristic proposed by Langella (2007). Because, our improvements are based on the idea of comparing the benefits of external procurement option with what integral algorithm by Langella (2007) suggests.

We restrict our attention to only WW, SM and LUC heuristics, because from traditional lot sizing studies we know that they provide better results compared to others, like part period balancing (Teunter et al. (2007)). Similarly, we do not attempt to develop new heuristics for disassembly lot sizing, since our aim is to show the performance of these well known and widely used methods. If the performance of the heuristics can be proven to be at a certain acceptable level, with minor modifications on the existing Decision Support systems, these tools can be used in disassembly planning as well.

The outline of this chapter is as follows: In Section 4.1, the problem environment is described. In Section 4.2, a mathematical model for the problem under consideration is provided. Next, in Section 4.3, general approach followed in the application of the heuristics is explained in detail and modification of some lot sizing heuristic approaches that are used in our examples are described. In the last section, the computational study and its results are discussed.

4.1 Problem Environment

The main environmental assumptions employed on the following are exactly the same as the ones that are employed in the model constructed in Chapter 3. These are:

- *Demand structure:* The demand for recovered leaf items over a finite planning horizon are known in advance.
- *The product structure:* The number of leaf items that can be obtained from each leaf is known. There is no yield loss in disassembly process.

The main environmental differences that we consider in this chapter are as follows:

- *No external procurement option:* We exclude the regular procurement option, since our main aim is to investigate the effect of disassembly setup cost. Note that when there is regular procurement option of leaf items and a setup cost for disassembly operations, the comparisons of two options as we introduced in Chapter 3 becomes unfair. Under setup cost values for which lot sizing is required, i.e., when lot for lot is not reasonable, the disassembly option becomes inferior to regular procurement option. On the other hand, when there is also a setup cost for regular procurement option in addition to the setup cost of regular procurement, the problem gets more complicated; the attractiveness of disassembly option depends on many factors. Our main objective here is to investigate the performance of traditional well known lot sizing heuristics in disassembly environment. Therefore we restrict our attention to the case where the disassembly is the only source to satisfy the demand, and there is a setup cost associated with that.

- *Setup cost of disassembly:* We consider the situation that a setup is made for disassembling each type of root item. Time stationary cost data are assumed. A setup cost of SC_r, is incurred if root item r is disassembled in a period. The setup times are ignored. All other cost items are exactly the same as in Chapter 3.

4.2 Mathematical Model

In this section, we present the mathematical model that aims to find disassembly plan minimizing the sum of variable and fixed cost disassembly and inventory holding costs.

Notation

Parameters

T : Length of the planning horizon.

t : Index for periods $t = 1, ..., T$.

R : Number of root items.

r : Index for root items $r = 1, ..., R$.

L : Number of leaf items.

l : Index for leaf items $l = 1,...,L$.

$\alpha_{r,l}$: Number of leaf item *l* that is obtained from root item *r*

$\lambda_{l,t}$: Demand of leaf item *l* at period *t.*

IC_l : Unit inventory holding cost per period of leaf item *l.*

LT_r : Lead time for root item *r.*

SC_r : Setup cost for root item *r.*

<u>*Decision variables*</u>

$x_{r,t}$:Number of root item *r* disassembled at period *t.*

$z_{l,t}$: Number of leaf item *l* held in inventory at period *t.*

$y_{r,t}$: decision variable for set-up

$$y_{r,t} = \begin{cases} 1 \text{ if root } r \text{ is going to be disassembled at period } t \\ 0 \text{ if root } r \text{ is not going be disassembled at period } t \end{cases}$$

Min

$$\sum_{r=1}^{R}\sum_{t=1}^{T}(TDC_r * x_{r,t-LT_r}) + \sum_{l=1}^{L}\sum_{t=1}^{T}(IC_l * z_{l,t}) + \sum_{r=1}^{R}\sum_{t=1}^{T}(SC_r * y_{r,t}) \tag{4.1}$$

s.t.

$$\sum_{k=1}^{t}\sum_{r=1}^{R}\alpha_{r,j} * x_{r,k-LT_r} - \sum_{k=1}^{t}\lambda_{l,k} = z_{l,t} \quad for\ t = 1,...,T \quad l = 1,...,L \tag{4.2}$$

$$x_{r,t} \le My_{r,t} \quad \text{for } r = 1,...,R\ ,\ t = 1,...,T \tag{4.3}$$

$$x_{r,t},\ z_{l,t} \ge 0 \text{ and integer for } \ \mathrm{r} = 1,...,R,\ l = 1,...,L,\ t = 1,...,T \tag{4.4}$$

$$y_{r,t} \in \{0,1\}\ r = 1,...,R,\ t = 1,...,T \tag{4.5}$$

Objective function of the models aims to minimize the total costs incurred over the planning horizon. The relevant costs are setup costs, total disassembly costs and inventory holding costs.

Equation (4.2) is exactly the same as the Equation (3.2). Adding Equation (4.3) to IP ensures that if $x_{r,t} > 0$, i.e., if a root is disassembled in period *t*, than $y_{r,t} = 1$, i.e., setup cost for root *r* is incurred in period *t*. Equation (4.4) is non-negativity and integer constraints and Equation (4.5) restricts the value of $y_{r,t}$ to 0 or 1 for $r \in \{1,...,R\}$, $t \in \{1,...,T\}$ and $l \in \{1,...,L\}$. *M* in Equation (4.3) is a big positive number and it can be set to the maximum total demand of the leaf items, i.e., $\max\left\{\sum_{t=1}^{T} \lambda_{l,t}\right\}$ for $l \in \{1,...L\}$.

This mathematical model has $R*L+L*T$ integer variables and $R*T$ binary variables. Involving binary variables, this model is harder to solve compared to the model in Chapter 3. Therefore heuristic approaches are needed for the solution.

4.3 General Approach Followed in the Application of Traditional Lot Sizing Heuristics to Our Problem

The traditional lot sizing heuristics are constructed to determine the lot sizes that balance the tradeoff between fixed cost of production and inventory holding cost. As indicated by Winston (1993), the basic assumptions are:

- There is a single item in concern.
- Demand d_t during period *t=1,…,*T is known at the beginning of the planning horizon.
- Demand should be met on time.
- The cost of producing *x* units is *c(x)* is expressed as

  $$c(x) = \begin{cases} 0 \text{ if } x = 0 \\ K + c * x \text{ if } x > 0 \end{cases}$$

  where *K* is the setup cost and *c* is the unit cost of production.
- At the end of period *t,* the inventory level is observed, and a holding cost *h* is incurred.
- The objective is to determine a production level for each period *t* that minimizes the total cost of meeting the demands for periods *t*=1,…,*T.*

The traditional lot sizing problem can be expressed as follows:

Min

$$\sum_{t=1}^{T} K * y_t + c * x_t + h * I_t \qquad (4.6)$$

s.t.

$$I_{t-1} + x_t - d_t = I_t \quad \forall t \qquad (4.7)$$

$$M * y_t \geq x_t \ \forall t \qquad (4.8)$$

$$x_t \geq 0,\ y_t \in \{0,1\} \qquad (4.9)$$

Where the input parameters are;

T : Number of production periods,

K : Setup cost of production,

c : Cost of producing 1 unit item,

h : Cost of holding 1 unit item,

d_t : Demand in period *t,*

M : A big positive number

And the decision variables are;

x_t : Production quantity in period *t,*

I_t : Inventory level at the end of period *t.*

In the model, objective function minimizes the total cost of the production schedule. Equation (4.7) is the inventory balance constraint and equation (4.8) ensures that if a production occurs in period *t,* then a setup cost is incurred in that period by assigning y_t to 1.

According to the assumptions mentioned above, some heuristic approaches are constructed for the lot sizing problems. Wagner Whitin, Silver Meal and Least Unit Cost algorithms are described below.

Wagner-Whitin (WW) Algorithm :

It is well known that in the optimal solution, zero inventory property is satisfied, i.e., $I_{t-1} * d_t = 0$ for all $t = 1,...,T$. The implication of this property is that if a setup cost is made in period *t*, the production quantity is exactly equal to meet the demand for the periods $t, t+1,..., t+j < T$.

Wagner and Whitin (1958) suggest a dynamic programming recursion for the problem based on this optimality condition. Define f_t as the minimum cost incurred during period $t, t+1,...,T$, given that at the beginning of period *t,* the inventory level is zero. Then f_t must satisfy;

$$f_t = \min_{j=0,1,2,...,T-t} (c_{t,j} + f_{t+j+1}) \tag{4.10}$$

where $f_{T+1} = 0$ and $c_{t,j}$ is the total cost incurred during periods $t, t+1,..., t+j$. Thus $c_{t,j} = K + \sum_{p=k}^{k+t} (p-k) * h * d_p$, which is the sum of the setup cost and inventory holding cost incurred during periods $t, t+1,..., t+j$.

The original WW algorithm runs in $O(T^2)$. Wagelmans et al. (1992) propose an alternative algorithm that runs in $O(T \log T)$.

The Silver-Meal (SM) Heuristic :

Silver et al. (1998) mentions that the Silver-Meal heuristic selects the replenishment quantity in order to replicate a property that the basic economic order quantity (EOQ) possesses when the demand rate is constant with time, namely, the total relevant costs per unit time for the duration of the replenishment quantity are minimized. Suppose we are at the beginning of period *k* and are trying to determine how many periods of demand should be satisfied by k^{th} period's production quantity. If the production quantity at period *k* is going to satisfy the demand for the next *t* periods, then a cost of $C_{k,k+t} = K + \sum_{p=k}^{k+t} (p-k)*h*d_p$ will be incurred at period *k.*

Let $AC_{k,k+t} = \dfrac{K + \sum_{p=k}^{k+t} (p-k)*h*d_p}{t}$ be the average per-period cost incurred during the next *t* periods. This criterion has the desirable feature of not including, in the present replenishments, a large requirement well in the future, which makes the 'cost per unit time' measure too high. In most of the situations, an integer *t** can be found such that $AC_{k,k+t^*+1} \geq AC_{k,k+t^*}$. The S-M heuristic recommends that k^{th} period's production amounts be sufficient to meet the demands for periods *k,…,t**. If no *t** exists period *k* should satisfy the demand for periods *k,…,T.* If a *t** is found for period *k,* then algorithm goes for the determination of the *t*+1*st periods production amounts.

The Least Unit Cost (LUC) :

The Least Unit Cost (LUC) is identical to the Silver-Meal heuristic except that it accumulates requirements until the cost per unit criterion increases instead of cost per period. So the criterion for LUC is $AC_{k,k+t} = \dfrac{K + \sum_{p=k}^{k+t} (p-k)*h*d_p}{\sum_{p=k}^{k+t} d_p}$.

Both SM and LUC run in $O(T)$.

4.3.1 Application of Lot Sizing Heuristics to Disassembly Lot Sizing Problem

As these methods are for the traditional lot sizing problems, we have to make some modifications according to our problem environment. The main differences of our case from the traditional problem environment are as follows:

- Setup and unit cost of disassembly are defined for root items while the demand is estimated for the leaf items.
- Inventory holding costs are incurred for the leaf items, no root item inventory is carried.

Therefore, in order to utilize the traditional lot sizing heuristics in our environment, we need to have disassembly requirements, $S_{r,t}$, for each root item over the planning horizon. Thus in this problem, the disassembly quantities of the integral algorithm are treated as the demands, i.e. production quantities, of the traditional lot sizing heuristics. To obtain the quantities, we utilize the heuristics provided in Chapter 3. Note that since we do not consider the external procurement option here, all variations on the heuristic that we propose yield the same result which is exactly the same as the result of the original heuristic developed by Langella (2007), the integral algorithm. The disassembly plan obtained by the heuristics is treated as the requirements for the lot-sizing problem for each root item. We suggest a two-step approximate method to find disassembly lot sizes.

- Run Integral algorithm to determine the requirements for each root item r over the planning horizon.
- Run a traditional lot sizing algorithm for each root item r to satisfy the requirements.

In order to apply the lot sizing heuristics in our environment, we transform the holding cost defined for the leaf items into holding costs for root items as follows:

$$H_r = \sum_l \alpha_{r,l} * IC_l \quad \text{for } r = 1,\ldots,R \tag{4.11}$$

Note that H_r, as given in Equation (4.11) assumes that whenever a unit of root item r is disassembled, all leaves in the BOM of root r is held in the inventory individually.

In the lot sizing heuristics for the disassembly problem, the inputs are the disassembly quantities, $S_{r,t}$, of root items and the inventory holding costs, H_r, of the root items in addition to the setup costs of root items. Note that, treating the disassembly plan obtained by Integral algorithm as the demand for individual root items and setting the holding cost rates as given in Equation (4.11) is just an approximate approach. It assumes that whenever a root item *r* is "required", all the leaf items in the BOM of that root are demanded. In other words, it ignores the possibility that only some leaves have demand in a certain period of the planning horizon. In order to have an exact inventory holding cost calculation one should consider $\sum_{p=1}^{t}\sum_{r=1}^{R}(S_{r,t} * \alpha_{r,l} - \lambda_{l,p})$ in period *t,* while we assume that it is 0. However, in our two-step approach, we include holding costs into account in the second phase, where we do not know which roots are used to satisfy the demand for which leaves, and we consider each root's lot sizing decisions independent of others.

Therefore, the total cost obtained in the lot-sizing phase of our approach is not exact, it should be corrected to include additional leaf item inventory costs.

4.3.2 Solution Procedure

The first step of our approach is running the integral algorithm with the same input parameters as in Chapter 3. Integral algorithm finds the root items' quantities to satisfy the demands of the leaves and these quantities are the main inputs of the lot-sizing heuristics.

The next step is to run the lot-sizing heuristics for each root item separately to find their lot sizing quantities, $R_{r,t}$. These heuristics find the total lot sizes according to the cost parameters of setup cost and root inventory holding costs as given in Equation (4.11). Then, they calculate the cost of the schedule as the summation of setup costs incurred and inventory holding costs of leaves which occurred by the root items only. At last it updates the total cost by adding the disassembly costs and the inventory holding costs of individual excess leaves left as a result by the integral algorithm.

In Figure 4.1, our general approach in the application of traditional lot sizing heuristics to our problem is depicted.

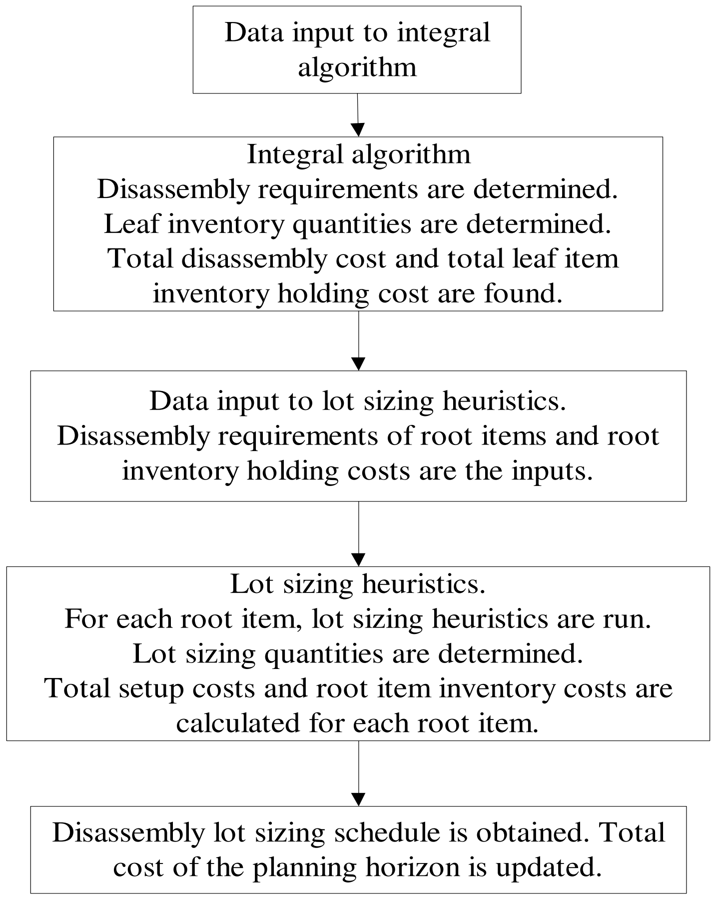

Figure 4.1 General approach followed

In order to adjust the traditional lot sizing heuristics to our problem, their cost calculations are modified as follows;

For the WW algorithm, the recursion is modified. Define f_t as the minimum cost incurred during periods *t, t+1,…, T,* given that the beginning of the period *t,* the inventory level for the root item in concern is zero. Then, f_t must satisfy

$$f_t = \min_{j=0,1,2,\ldots,T-t} (c_{t\,\mathrm{j}} + f_{t+j+1}),$$

where $f_{T+1} = 0$ and $c_{t\,j}$ is the total cost incurred during periods *t, t*+1, …, *t*+*j* if disassembly during period *t* is exactly sufficient to meet requirements for periods $t, t+1, ..., t+j$. Thus, $c_{t\,j} = SC_r + \sum_{p=t}^{j} (p-t) * H_r * S_{r,p}$ which is the sum of the setup cost and inventory holding cost incurred during periods $t, t+1, ..., t+j$.

The cost incurred for a period's disassembly quantity and the average cost per period equations are modified for the SM and LUC heuristic as follows. If the requirements for the root item in concern during the following *t* periods are disassembled, then a cost of $SC_r + \sum_{p=k}^{t} (p-k) * H_r * S_{r,p}$ will be incurred for root item *r* at period *k.* And the average cost will be equal to $AC_{k,k+t} = \dfrac{SC_r + \sum_{p=k}^{t} (p-k) * H_r * S_{r,p}}{t}$

And for the LUC, the average cost is equal to $AC_{k,k+t} = \dfrac{SC_r + \sum_{p=k}^{t} (p-k) * H_r * S_{r,p}}{\sum_{p=k}^{k+t} S_{r,p}}$.

Time complexities of our solution approaches are polynomial. If SM or LUC is used after the Integral algorithm, the time complexity of the overall algorithm is $O(L * T)$. If WW is used after the Integral algorithm, time complexity is equal to $O\left(\max\left\{L * T,\ T \log T\right\}\right)$ or $O\left(T * \max(T, L)\right)$ depending on the implementation of WW algorithm.

To illustrate the algorithm mathematically, consider the example given in Section 3.4.3 whose contribution factors and leaf item demands are given in Table 3.1 and Table 3.7, respectively. Disassembly costs and inventory holding cost of leaf items are shown in Table 3.11. For this problem, integral algorithm finds the disassembly schedule and the resulting leaf item inventories as in Table 4.1 and Table 4.2 below.

Table 4.1 Disassembly schedule found by the integral algorithm

Periods	1	2	3
A	0	4	1
B	3	2	1

Table 4.2 Inventory quantities of leaf items

Periods	1	2	3
C	0	0	1
D	0	4	3
E	0	2	2

According to the quantities in Table 4.1 total disassembly cost is found as $\sum_{r=1}^{R}(TDC_r * \sum_{t=1}^{T} S_{r,t}) = 10*5+11*6 = 116$. According to the inventory carrying schedule of the leaf items in Table 4.2, total inventory holding cost is $\sum_{l=1}^{L}(IC_l * \sum_{t=1}^{T} EX_{l,t}) = 2*1+2*7+2*4=24$. Hence our input for lot-sizing heuristics is the disassembly quantities in Table 4.1, inventory holding costs of root items and setup costs which are found and assumed as below.

$$H_A = \sum_{l=1}^{L} a_{r,l} * IC_l = 2*2+2*1 = 6$$

$$H_B = \sum_{l=1}^{L} a_{r,l} * IC_l = 2*2+2*3+2*1 = 12$$

Assume SC_A and SC_B be 500. According to these cost parameters heuristic finds (each of the lot-sizing heuristics find the same result with these cost parameters) the lot sizing quantities, $R_{r,t}$, as follows;

Table 4.3 Disassembly schedule found with lot-sizing heuristics

Periods	**1**	**2**	**3**
A	0	5	0
B	6	0	0

As seen in Table 4.3, for each root item, one disassembly operation occurs; hence, setup cost is incurred once for each. The total setup cost is 1000. For root *A* in 2nd period, 1 root is disassembled for the 3rd period, so inventory cost of 6 is incurred for root *A*. All the requirements of root *B* is disassembled in period 1. 2 of them are carried for 1 period, where 1 of them is carried for 2 periods. So, the inventory holding cost incurred for this root item is $\sum_{p=k}^{t}(p-k)*H_r*S_{r,p}=1*12*2+2*12*1=48$. Note that, inventory quantities in Table 4.2 are not taken into consideration in the inventory holding cost calculations of the lot-sizing heuristics. At last it updates the total cost as the summation of the costs it found and the costs that integral algorithm found. Total cost of the plan is $48+6+2*500+116+24=1194$.

4.4 Computational Study

A computational study is carried out to investigate the performance of the heuristics against the optimal values.

We consider a restricted class of problem sets considered in Chapter 3: S1, S4, S5 and S6. Notice that in S1 and S5 the disassembly costs are higher compared to S4 and S6. In the problem instances in sets S1 and S4, there are 3 leaf and 2 root items, whereas there are 6 leaf and 4 root items in S5 and S6.

In this study, we only consider T=12 as the planning horizon, since lot sizing decisions' effects are more substantial for long horizon problems.

Setup costs of the root items are set in 3 different levels as follows.

$$SC_r = H_r * 200 \text{ for } r = 1,...,R \tag{4.12}$$

$$SC_r = H_r * 800 \text{ for } r = 1,...,R \tag{4.13}$$

$$SC_r = H_r * 3200 \text{ for } r = 1,...,R \tag{4.14}$$

For the sample sets S1 and S4, we consider all 3 levels whereas, we consider the sample sets S5 and S6 with only the first 2 levels (the ones given in Equation 4.12 and Equation 4.13).

For each problem instance, the objective function values obtained under three lot sizing heuristics are compared with the optimal objective function values obtained by CPLEX. The minimum, maximum and average error values for each of the lot-sizing heuristics are reported in Table 4.4. In addition, distributions of errors for these heuristics are provided in Appendix D.

Table 4.4 Summary of results

% Errors	Wagner-Whitin			Silver Meal			Least Unit Cost			Best of 3		
	Min.	**Max.**	**Avg.**	**Min.**	**Max.**	**Avg.**	**Min.**	**Max.**	**Avg.**	**Min.**	**Max.**	**Avg.**
S1(K_r=IC_r*200)	0	11.71	1.68	0	11.71	1.70	0	14.23	2.25	0	11.71	1.68
S1(K_r=IC_r*800)	0	15.29	2.48	0	16.49	2.83	0	20.03	3.97	0	15.29	2.48
S1(K_r=IC_r*3200)	0	17.30	3.49	0	21.71	4.71	0	21.39	5.75	0	17.30	3.49
S4(K_r=IC_r*200)	0	42.25	2.82	0	42.25	2.84	0	42.25	3.77	0	42.25	2.82
S4(K_r=IC_r*800)	0	34.64	3.89	0	35.20	4.41	0	34.65	6.17	0	34.64	3.89
S4(K_r=IC_r*3200)	0	23.63	5.18	0	28.24	6.74	0	29.47	8.24	0	23.63	5.18
S5(K_r=IC_r*200)	0	94.83	14.81	0	94.83	14.81	0	94.83	15.05	0	94.83	14.81
S5(K_r=IC_r*800)	0	110.26	15.08	0	110.26	15.12	0	111.20	16.19	0	110.26	15.08
S6(K_r=IC_r*200)	0	108.95	21.33	0	108.95	21.34	0	108.95	21.69	0	108.95	21.33
S6(K_r=IC_r*800)	0	104.13	19.94	0	104.26	20.01	0	104.13	21.55	0	104.13	19.94

Some of the problem instances could not be solved due to the long running times on CPLEX. There are 10 problem instances for S5 under $K_r = IC_r * 200$, 25 for S5 under $K_r = IC_r * 800$, 15 for S6 under $K_r = IC_r * 200$ and 17 for S6 under $K_r = IC_r * 800$ could not be solved in 30 minutes. In Figure 4.1, maximum, minimum and the average error values of the remaining problem instances are shown.

For the problem sets with two root items and three leaf items, S1 and S4, the integral algorithm provides closer results compared to optimal values. The type and the amount of root items disassembled is generally the same, since there is no so much option. In these sample sets, WW algorithm finds the optimal solution in 39.33% of the problem instances in S1 and S4. The percentages for SM and LUC for these sample sets are 22% and 9.84% respectively. For the sample sets S5 and S6, there is a sharp decrease to 3% for both WW and SM, and to 1.82% for LUC.

For the same sample set S1 and S4, when the setup cost increases, amount of error increases as expected. Note that, the integral algorithm provides the same requirements under different setup cost values for a specific problem instance since it does not take the setup cost into account. Therefore, the main difference here is the handling of setup cost; that is, the error increases as the setup cost increases since the weight of the setup cost increases with respect to the total costs of disassembly.

When the setup costs are the same but the disassembly cost of the root items differ, the situation is more complicated. With the same setup costs, sample sets having lower disassembly costs have larger errors. There are two reasons for these errors. First reason is the same: setup cost increases relatively with respect to the disassembly cost, so the percent error increases. The other reason is the inventory holding cost: integral algorithm provides very close (even exactly the same disassembly amounts in most of the problems) results, i.e., requirements for root items, with changing disassembly costs. Hence, since the setup costs are equal, lot sizing heuristics do not find much different solutions, either. But the importance of inventory holding, i.e., the effect of the inventory holding cost compared to disassembly cost, increases when the disassembly costs decrease. As the sensitivity of the model to the inventory holding costs increases, it may find different disassembly amounts and schedules with respect to the solutions of S1. With this consideration, model finds lower total costs for S4 than expected from compared to S1. An example is shown in Table 4.5 below;

Table 4.5 Disassembly costs and percent errors of problem instances with respect to optimal solution.

	Disassembly Costs		**Percent Error**
	Root A	Root B	WW
Problem instance 1	21.71	5.47	10.68
Problem instance 2	12.64	7.07	42.25

Problem instances are taken from the S1 under $K_r = IC_r*200$ and S4 under $K_r = IC_r*200$ respectively. For both problem instances, all the parameters except disassembly costs are same. Integral algorithm chooses root *B* in each problem instance and satisfies all the demand by disassembling only root *B.* For the second problem, with the decrease in the disassembly cost of the root *A*, model chooses more root item *A* to prevent the inventory accumulation and this approach provides a crucial decrease in the total inventory holding cost at the same time disassembling cheaper root item *A,* which provides a decrease in the total disassembly cost compared to instance *1*. As a result, when lot sizing heuristics find higher cost for instance *2* according to the increasing disassembly cost of root *B*, model finds a more profitable way to satisfy the demand. Hence, the percent error increases in the second problem.

The other sample sets have higher errors. They have four root items and six leaf items. These sample sets (S5 and S6) have higher errors in the previous chapter, too. As they have four root items, the possibility of selecting the wrong root item increases the error made by the integral algorithm. As expected, the reason of high errors is not the effect of the setup cost. Hence, the main difference begins before running the lot-sizing heuristics.

To conclude, WW algorithm always results in smaller or equal error values with respect to SM and LUC as it finds the optimal solution for the traditional lot sizing problems. But in terms of percent errors, there is little difference (between 0.02% and 0.07%) between WW and SM although SM does not find the optimal solution for as many problem instances as WW.

CHAPTER 5

CONCLUSIONS

Nowadays, no product or distribution strategy is designed without considering the future of that product. A plan must be made about what to do with the resulting waste at the products' end of life. As environmental regulations come into effect and procedures are obliged to collect their end-of-life products and recover the parts and materials, disassembly strategy becomes a growing trend. Hence, as the variety of the products that can be disassembled and remanufactured increases, a need for powerful solution procedures arises.

In this thesis, our main aim is to improve some of the heuristic approaches that were studied in the literature and find some new approaches for demand driven disassembly problems.

In Chapter 3, the disassembly planning problem is defined as to determine the number of root items to be disassembled and number of leaf items to be procured externally from an outside source in order to satisfy the total demand of the leaf items. The number of root items and the number of leaf items they include are deterministic. In the BOM structure of the root items, subassemblies are not taken into account and it is assumed that only leaf items are extracted with a complete disassembly operation. In addition, we allow multiplicity and commonality.

We improve Integral algorithm proposed by Langella (2007) according to his suggestions about adding the inventory holding cost and external procurement option to the heuristic. We suggest to first determine the type of root items to be disassembled to satisfy leaf demand period by period, then investigate the possible benefits of external procurement. While doing that as it is originally proposed by Taleb and Gupta (1997) and Langella (2007), we start with non-common leaf items. Two variants based on the calculation of inventory holding costs are considered: *Myopic NC-first* and *Non-myopic NC-first.* After observing some deficiencies of prioritizing non-common leaves, we constructed two new algorithms which treat all leaf items as common leaves and called them *Myopic* and *Non-myopic*.

We apply these heuristic approaches to different problem environments. The main parameters used in generating the problem instances are the number of root items, number of leaf items, the length of the planning horizon and the relevant cost parameters. Generally we deal with the instances having two root items and three leaf items. We try to investigate the effects of the relevant costs in the solutions. Then, other sets of problem instances having more root and leaf items are solved. In addition to this, decreasing and increasing trends in demand are considered and their results are compared with the problems having no trend in their demand data. Apart from the heuristic solutions, we solve the same problems optimally according to an IP model and compare all of the results. The main findings of this study can be summarized as follows:

- With the new ways applied to the heuristic approaches, the algorithms find better solutions compared to the solutions found by the heuristic provided by Langella (2007).
- Algorithms aim to find if external procurement is profitable with respect to disassembly. If algorithms find external procurement more profitable, then disassembly and inventory quantities decrease. Hence, inventory accumulation is prevented if it is profitable. As there are myopic and non-myopic approaches for the inventory holding cost, algorithms find different solutions. When these solutions are compared, one can determine the better solution whether the inventory carrying is advantageous or not.

- The heuristics which do not give priority to non-common leaf items are very important for the lower number root item problems. In some of these problems, it provides enormous decrease in the percent error with respect to the heuristics giving priority to the non-common leaves. As the number of root item increase, the possibility of having a non-common leaf item decreases. When all the leaf items are common leaves, *Myopic* and *Non-myopic* heuristics give exactly the same disassembly quantities compared to the other two heuristic approaches, which gives priority to the non-common leaf items. As the algorithms of the heuristics do not differ for the determination of the root items for satisfying the demands of the common leaf items, results are exactly the same. Hence the possibility of improving the solution decreases for *Myopic* and *Non-myopic* when the number of root items increase

- Generally demands of goods fluctuate in a planning horizon. There will be both descents and ascents in different seasons, months, etc. With this aspect, the effect of the increasing and decreasing trends in the demands on the heuristics' behavior is investigated. The different behaviors of the algorithms are observed for the different length planning horizons.

- Mostly, heuristics are solved for the problems with two root items having three leaf items in their product structure. In addition to this, problems with more root and leaf items are solved to see the applicability of the heuristic to these problems. Due to the increase in the number of root items, the possibility of choosing wrong root items for the algorithms increases. Although, the errors of these sample sets are higher from the sample sets having two root items as expected, they are in an acceptable range.

The objective of the Chapter 4 is to investigate the effect of the setup cost on the lot sizing decisions of disassembly planning. The demands and the product structures are exactly the same as the problems of Chapter 3. But there are two differences that there is no external leaf procurement option and setup cost and root inventory costs are included. Without the external procurement option, we restrict the solution to the root disassembly which becomes the only source to satisfy the demand. Setup cost is incurred when a batch of root item is disassembled in any period.

To investigate the effect of the setup cost on the disassembly plan, we use the traditional lot sizing heuristics, namely Wagner-Whitin, Silver Meal and Least Unit Cost. But before running these heuristics, the disassembly plan is found by the integral algorithm of Langella (2007), which gives the same result with the *Myopic NC-first* algorithm when external procurement option is omitted. The disassembly plan of the integral algorithm becomes the requirement of the lot sizing heuristics. After the requirements are put into lot sizing heuristics as input, these heuristics find a disassembly schedule for the planning horizon. At last we compare these results with the optimal solutions found by MIP model using CPLEX. The main outcomes of this computational study can be summarized as follows.

- The main reason of the errors is the different solutions of the integral algorithm from the optimal solutions. For the small number of root items in the problem (in our samples there are two root items), the root items chosen to be disassembled is generally the same with the optimal. When this situation occurs, lot sizing heuristics may find the optimal disassembly schedule.

- As the optimal solution considers the inventory holding cost, it sometimes finds different root items to be disassembled. When this happens, the percent error is not high for these problem samples and increases if the setup cost increases as the ratio of setup cost over disassembly cost increases.

- For more complex problems which involve more root and leaf items, the amount of errors gets larger. As the number of root items increase, the possibility of choosing wrong root item increases for the integral algorithm. Hence, similarly the reason of the errors is the different solution of the integral algorithm. But the range of these errors is in an acceptable range apart from some exceptional problems.

The main contributions of this thesis to the literature can be stated as follows;

- Improves the existing heuristic approaches and adds new heuristics considering different insights of inventory holding cost and priority to non-common leaf items with different approaches.
- The effect of disassembly setup cost is first studied.
- We modify the traditional lot-sizing heuristic approaches for the disassembly environment with changing the inventory holding cost consideration.

It is possible to extend this work in various ways. The model and the heuristic approaches can be improved for the problem considered in the thesis and some new heuristics may be constructed for the more realistic problems.

In our problem complete disassembly is assumed. Both for some cases, subassemblies are needed and incomplete disassembly may be more profitable option in most of the times. Especially when there are options of material recycling or reuse apart from remanufacturing for the disassembled parts, subassemblies become more important. A new model and a heuristic may be constructed by taking the selective disassembly option into account if the profits of the each option are known for all components and subassemblies. In this approach, disassembly sequences are found according to the precedence relations, which must be known exactly for disassembly operations, in addition to the schedules when satisfying the demands of the leaf items.

In our study, we assume no restriction on supply and inventory values. But for the recovery environment, collecting the utilized goods is very hard and this may lead to serious supply shortages. Likewise, inventory capacity may be a problem for the disassemblers because in some cases an inventory accumulation is observed in our problem instances.

We assume at most 12 period planning horizon for the problems. But for real life problems, we have to consider infinite planning horizons as the production never stops in a factory. Similarly, in our case demands always exist and disassembly operations never stop. Hence, it is not realistic to assume a finite planning horizon. Considering this, with the updated demand forecasts, our heuristic approaches can be adapted to a rolling horizon basis.

Another outstanding issue in disassembly is the high amount of uncertainty. Uncertainty is always crucial in all traditional production environments for both demands and manufactured product. But it is higher in recovery applications not only in terms of demand but also in terms of supply as the supply of this work is the second hand goods. The quality and the quantity of the returned product affect yield amounts from root items, making the yields stochastic. As a result, disassembly environment is stochastic in terms of demand, supply and yield and this case needs more powerful heuristic approaches and mathematical models.

Materials and product recovery evolve basically for the environmental factors. Most of the manufacturers involve in this job because of the legal preventions and environmental regulations. According to these regulations, they have to recover at least a certain amount of their goods. Hence, when maximizing profit or minimizing cost, environmental factors can be added to the solution approaches as constraints or multi-objective models can be written.

REFERENCES

A. J. D. Lambert, 1997, "Linear Programming in Disassembly/Clustering Sequence Generation", Computers and Industrial Engineering, Vol .36, 723-738.

Allen J. Clegg and Davis J. Williams, 1995, "Production Planning for Companies with Remanufacturing Capability", IEEE International Sysmposium, 186-191.

Altekin F. T., 2005, Profit Oriented Disassembly Line Balancing, Ph.D. Dissertation, Middle East Technical University, Ankara.

Aşkıner Güngör and Surendra M. Gupta, 1998, "Disassembly Sequence Planning for Products with Defective Parts in Products Recovery", Computers and Industrial Engineering, Vol. 35, No. 1-2, 161-164.

B. Gonzalez and B. Adenso-Diaz, 2005, "A Bill of Materials-based Approach for End-of-life Decision Making in Design for the Environment", International Journal of Production Research, Vol. 43, No. 10, 2071-2099.

Bras B. and McIntosh M. W., 1999, "Product, Process and Organitional Design for Remanufacture-an Overview of Research", Robotics and Computer Integrated Manufacturing, Vol. 15, 165-178.

D-H Lee, J-G Kang and P. Xirouchakis, 2001, "Disassembly Planning and Scheduling: Review and Further Research", Proc. Instn. Mech. Engnr, Vol. 215 Part B, 695-708.

Edward A. Silver, David F. Pyke, Rein Peterson, “Inventory Management and Production Planning and Scheduling”,John Wiley and Sons. Inc., 1998, 210-215.

Elif Kongar and Surendra M. Gupta, 2002, "A Multi Criteria Decision Making Approach for Disassembly-to-Order Systems", Journal of Electronics Manufacturing, Vol. 11,No. 2, 171-183.

Ian M. Langella, 2007, "Heuristics for Demand-Driven Disassembly Planning", Computers and Operations Research, Vol. 34, No. 2, 552-577.

K. D. Penev and A. J. De Ron, 1996, "Determination of Disassembly Strategy", International Journal of Production Research, Vol. 34, No. 2, 495-506.

Karim N. Taleb and Surendra M. Gupta, 1997, "Disassembly of Multiple Product Structures", Computer and Industrial Engineering, Vol. 32, No. 4, 949-961.

Karim N. Taleb, Surendra M. Gupta and Louis Brennan, 1997, "Disassembly of Complex Product Structures with Parts and Materials Commonality", Production Planning and Control, Vol. 8, No. 3, 255-269.

Luiz S. Homem De Mello and Arthur C. Sanderson, 1990, "AND/OR Graph Representation of Assembly Plans", IEEE Transactions on Robotics and Automation, Vol. 6, No. 2, 188-199.

Surendra M. Gupta and A. J. D. Lambert, 2002, "Demand-Driven Disassembly Optimization for Electronic Products", Journal of Electronics Manufacturing, Vol. 11, No. 2, 121-135.

Ruud H. Teunter, Z. Pelin Bayındır, Wilco van den Heuvel (2006), "Dynamic lot sizing with product returns and remanufacturing", International Journal of Production Research 44 (20): 4377-4400

Thomas Spengler, Martin Ploog and Markus Shröter, 2003, "Integrated Planning of Acquisition, Disassembly and Bulk Recycling: A Case Study on Electronic Scrap Recovery", OR Spectrum Vol. 25, 413-442.

V. Daniel R. Guide, Jr., Ruud H. Teunter and Luk N. Van Wassenhove, 2003, "Matching Demand and Supply to Maximize Profits from Remanufacturing", Manufacturing and Service Operations Management, Vol. 5, No. 4, 303-316.

Wagelmans, A., Van Hoesel, CPM, and Kolen A., 1992, Economic Lot-Sizing: an $O(n \log n)$ Algorithm That Runs in Linear in the Wagner-Whitin Case, Operations Research, 40 (Supp-1), 145-156.

Wayne L. Winston, “Operations Research : Applications and Algorithms”, International Thomson Publishing, 1993, 1046-1048.

Website of Dell Inc., http://www.dell.com/content/topics/global.aspx/corp/environment/en/recycling_main?c=us&l=en&s=corp, (last accessed in February 2006).

Website of Hewlett-Packard Development Company, http://h41131.www4.hp.com/uk/en/global_citizenship/globalrecycling.html, (last accessed in February 2006).

Website of StEP Organisation, http://www.step-initiative.org/index.php, (last accessed in February 2006).

Website of WEEE-Forum, http://www.weee-forum.org/legislation_eu.htm#top, (last accessed in February 2006).

Website of Wikipedia Foundation, Inc., http://en.wikipedia.org/wiki/WEEE, (last accessed in February 2006), http://en.wikipedia.org/wiki/Waste_Electrical_and_Electronic _Equipment_Directive (last accessed in February 2006).

APPENDIX A

STEPS OF THE INTEGRAL ALGORITHM

Step 0. Data input.

Input the parameters $R,\ L,\ T,\ TDC_r,\ LT_r,\ \alpha_{r,l},\ \lambda_{l,t}$ for $r = 1,...,R,\ l = 1,...,L,$ $t = 1,...,T$,

Step 1. Initialization

Step 1a. Set t=0 and $S_{r,t} = 0$ for $\forall\ (r,t)$

Step 1b. Calculate the contribution factors for each root-leaf pair.

$$\beta_{r,l} = \frac{\alpha_{r,l}}{TDC_r} \text{ for } \forall\ (r,l)$$

Step 1c. Determine the non-common leaves.

Calculate the total contribution factor for each leaf

$$\beta_l = \sum_r \beta_{r,l} \text{ for } \forall\ l$$

If for any l, $\beta_l = \beta_{r,l}$ for some r, then l is a non-common leaf that is only obtained from root r. Therefore, the set of all non-common leaves (and corresponding roots), *NC,* can be formed by

$$NC = \left\{(l,r) \middle| \beta_l = \beta_{r,l} \quad l = 1,...L,\ r = 1,...R\right\}$$

Step 2. *t=t+1*

Step 3. Determine and update the disassembly plan to satisfy the demand for non-common leaf items for all $(l,r) \in NC$ with $LT_r \leq t$

Step 3a. Determine the disassembly quantities for root items required to satisfy the demand for corresponding non-common leaves.

The number of r to disassemble in order to satisfy the current period requirements of l is $\left\lceil \dfrac{\lambda_{l,t}}{\alpha_{r,l}} \right\rceil = x_{r,t}$.

Update disassembly quantity accordingly, $S_{r,t} \leftarrow S_{r,t} + x_{r,t}$.

Step 3b. Update demand for all leaves according to the current disassembly plan's updated quantity, i.e., the current values of $x_{r,t}$'s.

$$\lambda_{l,t} \leftarrow \lambda_{l,t} - \sum_{r} x_{r,t} * \alpha_{r,l} \quad \text{for } l = 1, \ldots, L$$

Step 3c. Update total contribution factors.

$$\beta_l = \begin{cases} 0 & if \quad \lambda_{l,t} \leq 0 \\ \beta_l & if \quad \lambda_{l,t} > 0 \end{cases}$$

If there exists any $\beta_l > 0$ for some l which is included in *NC* then go to Step 3a.

Else go to Step 4.

Step 4. Determine the disassembly plan to satisfy demand for common leaves.

Step 4a. Update selection ratios under current partial plan

$$\beta_{r,l} \leftarrow \begin{cases} 0 & \text{if} \quad LT_r > t \\ 0 & \text{if} \quad \lambda_{l,t} \leq 0 \\ \beta_{r,l} & \text{if} \quad \lambda_{l,t} > 0 \text{ and } LT_r \leq t \end{cases} \qquad \text{for } \forall\, l$$

Determine the leaf and corresponding root giving the maximum of contribution factors.

$$l' = \text{argmax}\{\beta_{r,l}\}$$
$$r' = \text{argmax}\{\beta_{r,l}\}$$

Step 4b. Determine the disassembly quantity of root r' such that the demand for l' is completely satisfied for period t.

The amount of r' to disassemble in order to satisfy the current period requirements of l' is $\left\lceil \dfrac{\lambda_{l',t}}{\alpha_{r',l'}} \right\rceil = x_{r',t}$.

Update disassembly quantity accordingly, $S_{r',t} \leftarrow S_{r',t} + x_{r',t}$.

Step 4c. Update demand of leaves for period t.

$$\lambda_{l,t} \leftarrow \lambda_{l,t} - \sum_{r} x_{r',t} * \alpha_{r,l} \quad for\ \forall\ l$$

Step 4d. If $\exists$ any $\lambda_{l,t} > 0$ then go to Step 4. Else go to Step 5.

Step 5. Determine the excess demand for the current period and update the future periods' demands.

$$EX_{l,k} = \max\left\{0, \sum_{r=1}^{R}\sum_{p=1}^{t} S_{r,p} * \alpha_{r,l} - \sum_{p=1}^{k} \lambda_{l,p}\right\} \quad for\ \forall\ l \text{ and } k = t,...,T$$

$$\lambda_{l,k+1} \leftarrow \max\left\{0, \lambda_{l,k+1} \text{ - } EX_{l,k}\right\} \quad for\ \forall\ l \text{ and } k = t,...,T-1$$

Step 6. If $t<T$ then go to Step 2. Else update disassembly plan according to lead times and stop.

$$S_{r,t} \leftarrow S_{r,t+LT_r} \text{ for } r = 1,..., R \text{ and } t = 1,...T$$

APPENDIX B

STEPS OF THE *MYOPIC NC-FIRST* AND *NON-MYOPIC NC-FIRST* ALGORITHMS

Step 0 determines the leaf items which are going to be procured externally as their demands cannot be satisfied by disassembly because of the lead times of root items. Step 3 is the application of Integral Algorithm. Between Steps 4-6, withdrawal of root items involving non-common leaves are considered and between Steps 7-9, withdrawal of all root items are considered according to the excess demands of common leaves. Step 10 updates the demands and excess leaf items and at the end of planning horizon Step 11 shifts the schedule according to the lead times of root items and ends the algorithm.

Step 0. Set $t=0$ and determine the leaves whose demands can be satisfied only by external procurement.

Step 0a. Set $t=t+1$.

For any $l \in \{1,...,L\}$, if $LT_r > t$ for all *(r,l)* pair with $\beta_{r,l} > 0$, then leaf item l cannot be obtained by disassembly. So it is externally procured.

$EP_{l,t} \leftarrow \lambda_{l,t}$ for all l satisfying above.

Repeat this step for the whole planning horizon and then go to Step 1.

Step 1. $t=0$

Step 2. Set $t=t+1$.

Step 3. Evaluate total cost of the current period t by considering only disassembly option.

Step 3a. Apply the integral algorithm's first 4 steps for period t until all the requirements are satisfied and calculate $EX_{l,t}$ for $\forall\ l$.

$$EX_{l,t} = \sum_{r=1}^{R} S_{r,t} * \alpha_{r,l} - \lambda_{l,t} \quad \text{for } l = 1,...,L$$

Step 3b.Calculate the total relevant costs (disassembly + inventory holding) for the current period *t*.

If myopic approach is applied, then total cost is,

$$TC = \sum_{r} TDC_r * S_{r,t} + \sum_{l} IC_l * EX_{l,t}$$

If non-myopic approach is applied, then total cost is,

$$TC = \sum_{r} TDC_r * S_{r,t} + \sum_{l} \left[IC_l * EX_{l,t} + \sum_{k=t+1}^{T} IC_l * \left(\max\left\{ 0, EX_{l,t} - \sum_{p=t+1}^{k} \lambda_{l,p} \right\} \right) \right]$$

Step 4. Selection ratio calculation.

Step 4a. Initialize selection ratios as

$$\beta_{r,l} = \frac{\alpha_{r,l}}{TDC_r} \; for \; \forall \; (r,l)$$

Step 4b. Determine the non-common leaf and corresponding root pair giving the maximum selection ratio after recalculating them as below.

$$\beta_{r,l} = \begin{cases} \beta_{r,l} & if \;\; l \in NC \\ 0 & if \;\; l \notin NC \\ 0 & if \;\; EX_{l,t} = 0 \end{cases}$$

If $\nexists$ any $\beta_{r,l} > 0$ then go to step 7.

Step 5.

Step 5a. Decrease the disassembly quantity of root *r* which has the maximum $\beta_{r,l}$ by 1.

$$S_{r,t} \leftarrow S_{r,t} - 1$$

Step 5b. Recalculate excess demands.

$$EX_{l,t} = \sum_{r} S_{r,t} * \alpha_{r,l} \text{-} \lambda_{l,t} \quad \text{for } \forall \; l$$

Step 5c. Calculate the cost difference by considering external procurement of leaf items having $EX_{l,k} < 0$

If myopic approach is used,

$$\Delta = \sum_{l} \left\{ (EX_{l,t})^{+} * IC_{l} + (EX_{l,t})^{-} * EPC_{l} - (EX_{l,t} + \alpha_{r,l}) * IC_{l} \right\} - TDC_{r}$$

If non-myopic approach is used,

$$\Delta = \sum_{l} \left\{ \begin{array}{l} \left[IC_{l} * EX_{l,t}^{+} + \sum_{k=t+1}^{T} IC_{l} * \left(\max\left\{ 0, EX_{l,t} - \sum_{p=t+1}^{k} \lambda_{l,p} \right\} \right) \right] + (EX_{l,t})^{-} * EPC_{l} \\ - \left[IC_{l} * (EX_{l,t} + \alpha_{r,l}) + \sum_{k=t+1}^{T} IC_{l} * \left(\max\left\{ 0, (EX_{l,t} + \alpha_{r,l}) - \sum_{p=t+1}^{k} \lambda_{l,p} \right\} \right) \right] \end{array} \right\} - TDC_{r}$$

where $x^{+} = \max\{0, x\}$ and $x^{-} = \max\{0, -x\}$

Step 6. If $\Delta < 0$ then update external procurement quantity of leaves and excess quantity of leaves and go to step 5a.

$$EP_{l,t} \leftarrow EP_{l,t} + EX_{l,t}^{-} \quad \text{for } \forall\ l$$

$$EX_{l,t} = \begin{cases} 0 & if\ EX_{l,t} \le 0 \\ EX_{l,t} & if\ EX_{l,t} \ge 0 \end{cases}$$

Else update disassembly quantity of root *r* and excess demands. Also set $\beta_{r,l} = 0$ and go to step 4b.

$$S_{r,t} \leftarrow S_{r,t} + 1$$

$$EX_{l,t} \leftarrow EX_{l,t} + \alpha_{r,l} \quad for\ \forall\ l$$

Step 7. Selection ratio calculation.

Step 7a. Initialize selection ratios as

$$\beta_{r,l} = \frac{\alpha_{r,l}}{TDC_{r}}\ for\ \forall\ (r,l)$$

Step 7b. Recalculate selection ratios.

$$\beta_{r,l} = \begin{cases} \beta_{r,l} & if\ l \notin NC \\ 0 & if\ l \in NC \\ 0 & if\ EX_{l,t} = 0 \end{cases}$$

If $\nexists$ any $\beta_{r,l} > 0$ then go to step 10.

Else determine the leaf *l* and corresponding root pair *r* having the minimum selection ratio from the selection ratios greater than 0.

Step 8.

Step 8a. Decrease the disassembly quantity of root *r* by 1.

$S_{r,t} \leftarrow S_{r,t} - 1$

Step 8b. Recalculate excess demands.

$$EX_{l,t} = \sum_r S_{r,t} * \alpha_{r,l} \text{-} \lambda_{l,t} \quad \text{for } \forall\, l$$

Step 8c. Calculate the cost difference by considering external procurement of leaf items having $EX_{l,t} < 0$

If myopic approach is used,

$$\Delta = \sum_l \left\{ (EX_{l,t})^+ * IC_l + (EX_{l,t})^- * EPC_l \text{ - } (EX_{l,t} + \alpha_{r,l}) * IC_l \right\} \text{-} TDC_r$$

If non-myopic approach is used,

$$\Delta = \sum_l \left\{ \begin{array}{l} \left[IC_l * EX_{l,t}^+ + \sum_{k=t+1}^{T} IC_l * \left(\max\left\{ 0, EX_{l,t} - \sum_{p=t+1}^{k} \lambda_{l,p} \right\} \right) \right] + (EX_{l,t})^- * EPC_l \\ \text{-} \left[IC_l * (EX_{l,t} + \alpha_{r,l}) + \sum_{k=t+1}^{T} IC_l * \left(\max\left\{ 0, (EX_{l,t} + \alpha_{r,l}) - \sum_{p=t+1}^{k} \lambda_{l,p} \right\} \right) \right] \end{array} \right\} \text{-} TDC_r$$

Step 9. If $\Delta < 0$ then update external procurement quantity of leaves and excess quantity of leaves and go to step 8a.

$EP_{l,t} \leftarrow EP_{l,t} + EX_{l,t}^- \quad \text{for } \forall\, l$

$$EX_{l,t} = \begin{cases} 0 & if\ EX_{l,t} \le 0 \\ EX_{l,t} & if\ EX_{l,t} \ge 0 \end{cases}$$

Else update disassembly quantity of root *r*, excess demands and set $\beta_{r,l}$=0 and go to step 7b.

$S_{r,t} \leftarrow S_{r,t} + 1$

$EX_{l,t} \leftarrow EX_{l,t} + \alpha_{r,l} \quad for\ \forall\, l$

$\beta_{r,l} = 0$

Step 10. Determine the excess demand for the current period and update the future periods' demands.

$$EX_{l,k} = \max\left\{ 0, \sum_{r=1}^{R} \sum_{p=1}^{t} S_{r,p} * \alpha_{r,l} - \sum_{p=1}^{k} \lambda_{l,p} \right\} \quad for\ \forall\, l \text{ and } k = t, \ldots, T$$

$$\lambda_{l,k+1} \leftarrow \max\left\{ 0, \lambda_{l,k+1} \text{ - } EX_{l,k} \right\} \quad for\ \forall\, l \text{ and } k = t, \ldots, T-1$$

Step 11. If *t*<*T* then go to step 2. Else update the disassembly schedule according to lead times and stop.

$S_{r,t} \leftarrow S_{r,t+LT_r}$ for $r = 1,...,R$ and $t = 1,...T$

APPENDIX C

ERROR DISTRIBUTIONS OF HEURISTIC APPROACHES COSTRUCTED IN CHAPTER 3

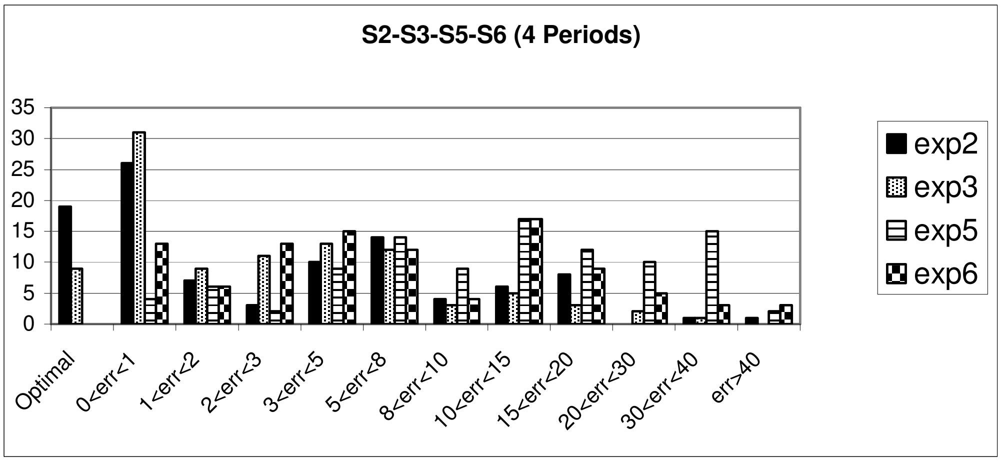

Figure A.1 Error distributions of Myopic NC-first algorithm for sample sets S2, S3, S5 and S6 for 4 periods planning horizon.

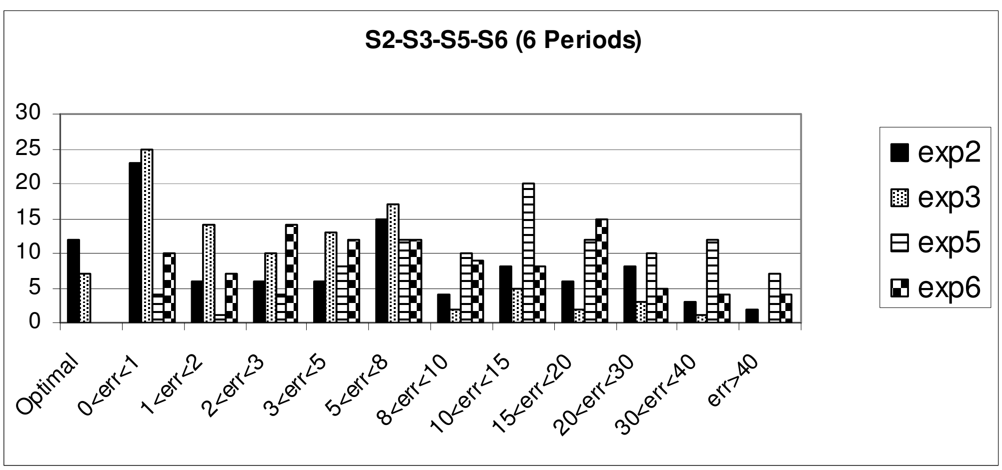

Figure A.2 Error distributions of Myopic NC-first algorithm for sample sets S2, S3, S5 and S6 for 6 periods planning horizon.

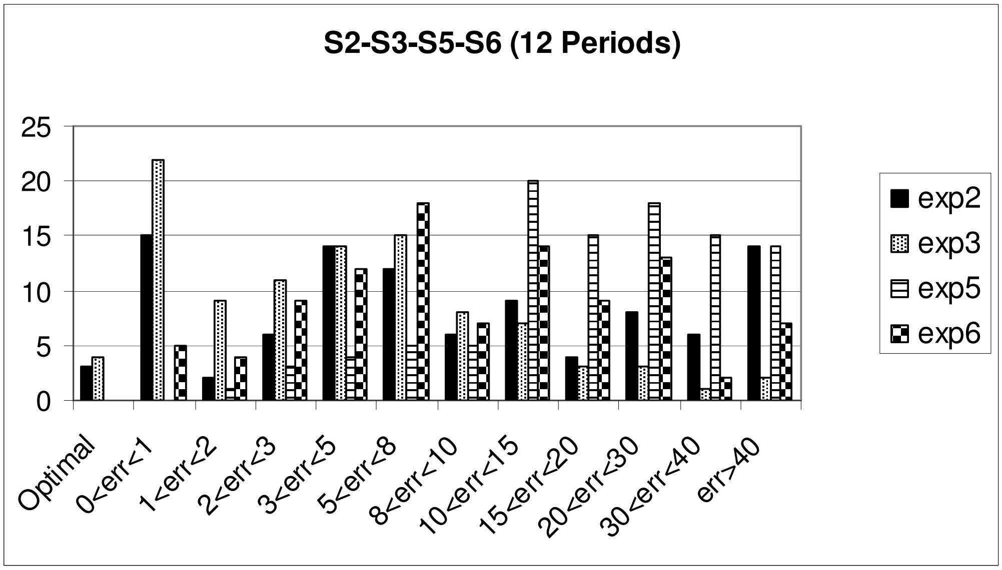

Figure A.3 Error distributions of Myopic NC-first algorithm for sample sets S2, S3, S5 and S6 for 12 periods planning horizon.

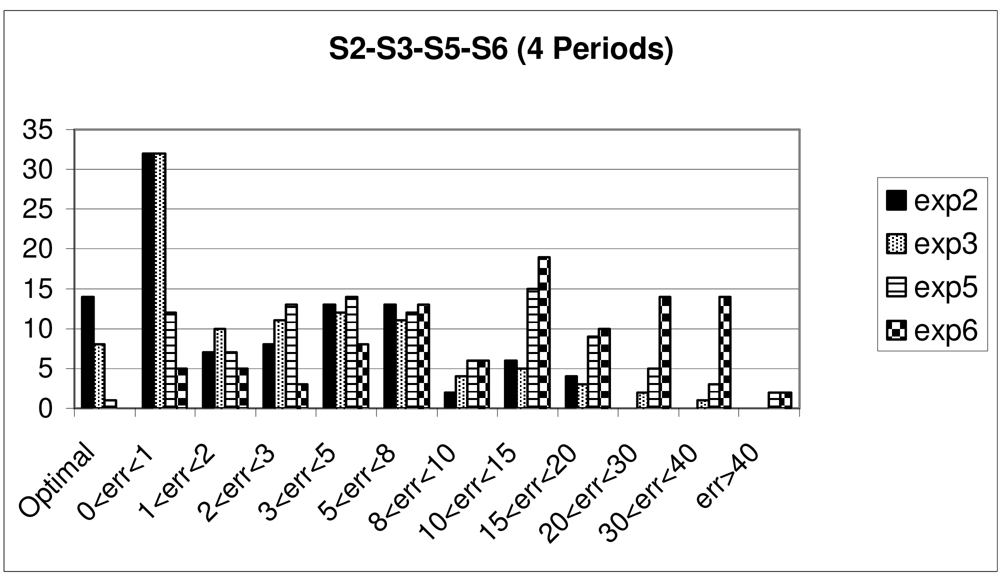

Figure A.4 Error distributions of Non-myopic NC-first algorithm for sample sets S2, S3, S5 and S6 for 4 periods planning horizon.

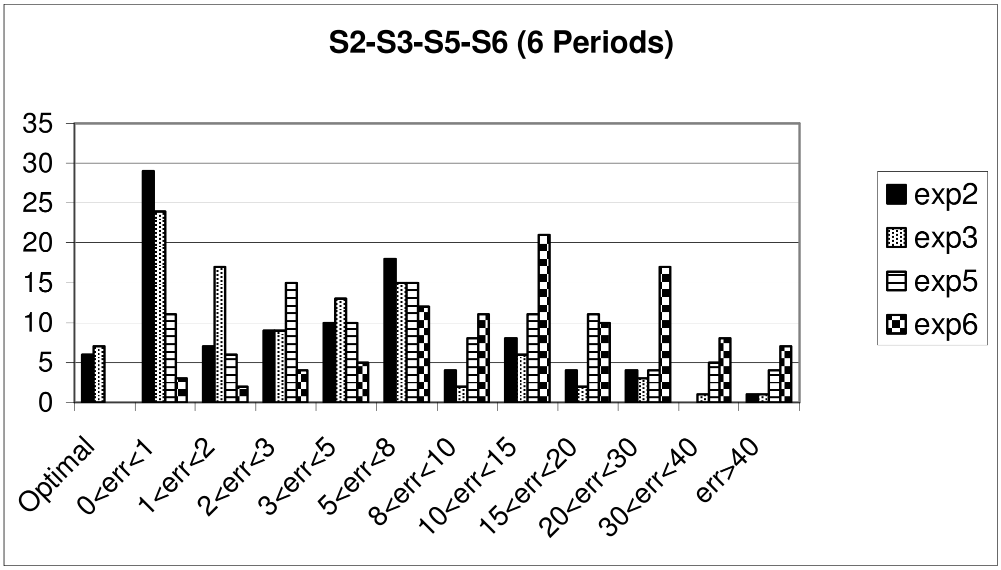

Figure A.5 Error distributions of Non-myopic NC-first algorithm for sample sets S2, S3, S5 and S6 for 6 periods planning horizon.

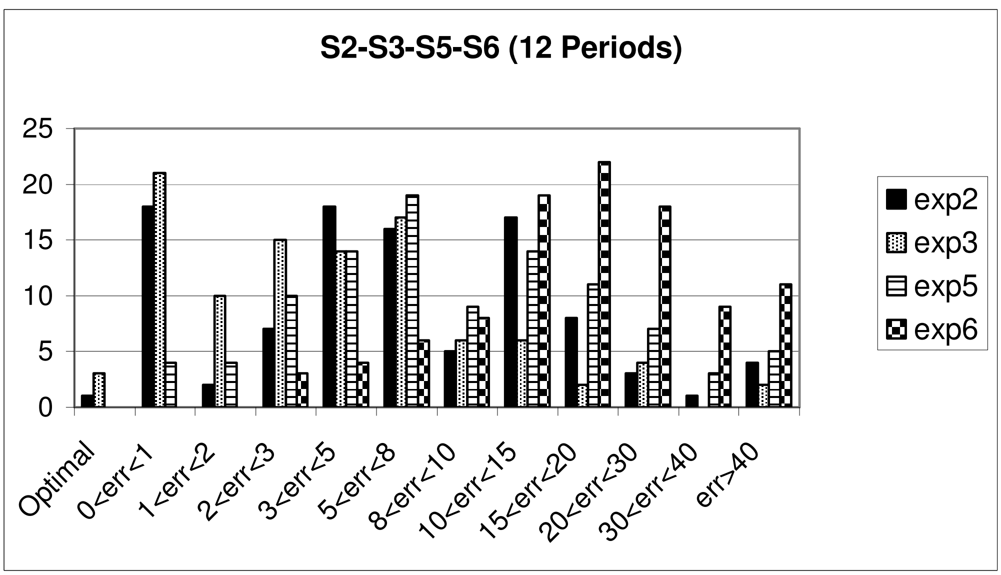

Figure A.6 Error distributions of Non-myopic NC-first algorithm for sample sets S2, S3, S5 and S6 for 12 periods planning horizon.

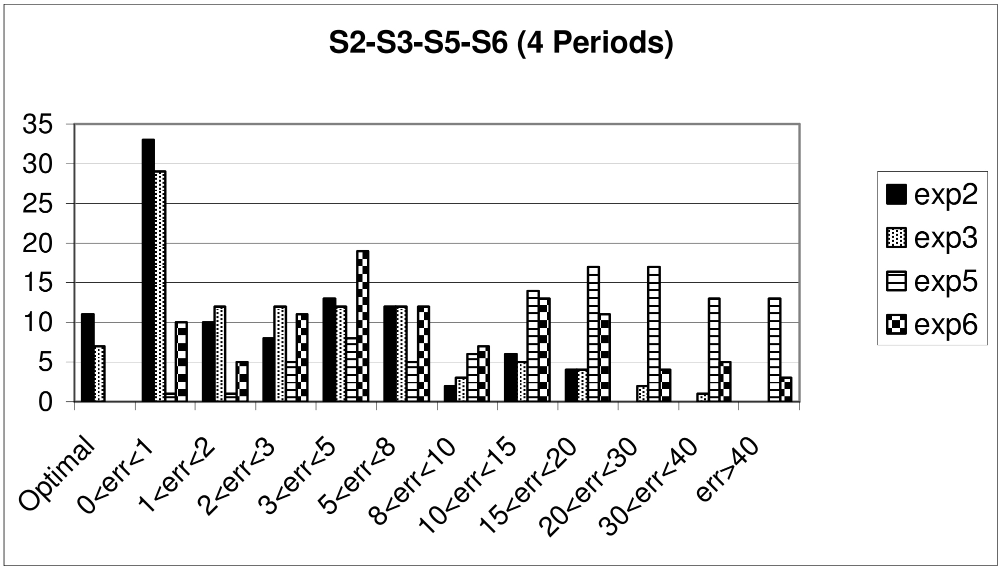

Figure A.7 Error distributions of Myopic algorithm for sample sets S2, S3, S5 and S6 for 4 periods planning horizon.

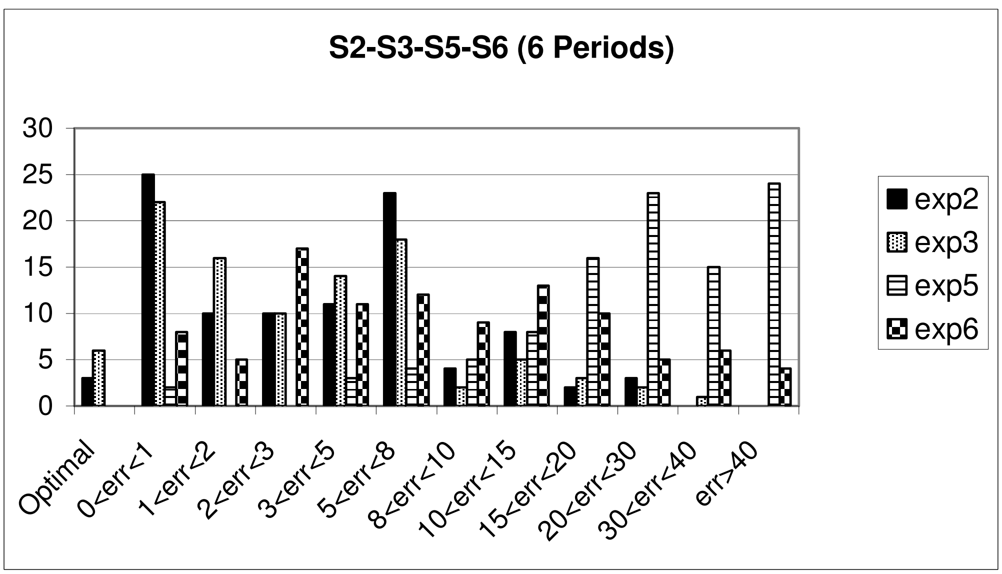

Figure A.8 Error distributions of Myopic algorithm for sample sets S2, S3, S5 and S6 for 6 periods planning horizon.

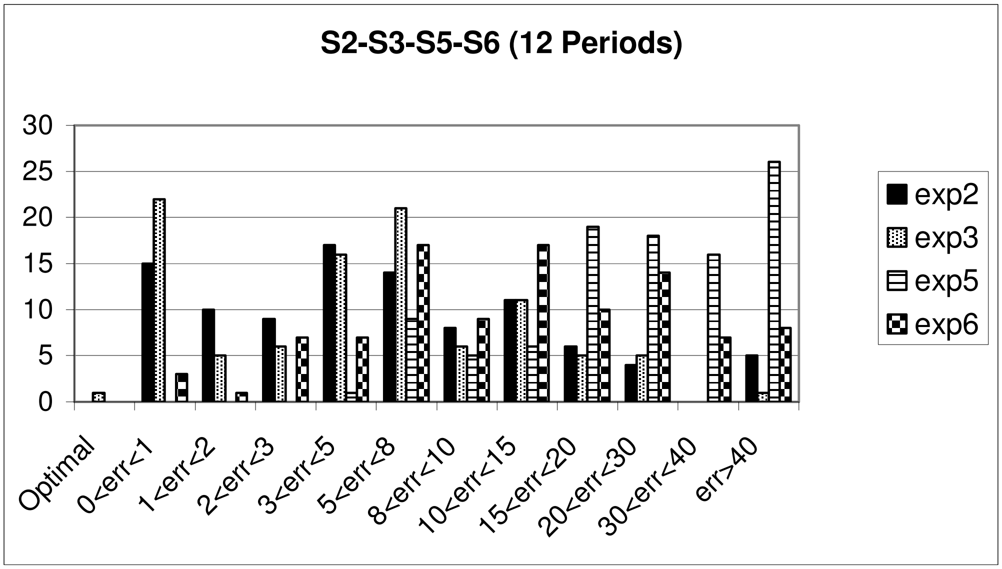

Figure A.9 Error distributions of Myopic algorithm for sample sets S2, S3, S5 and S6 for 12 periods planning horizon.

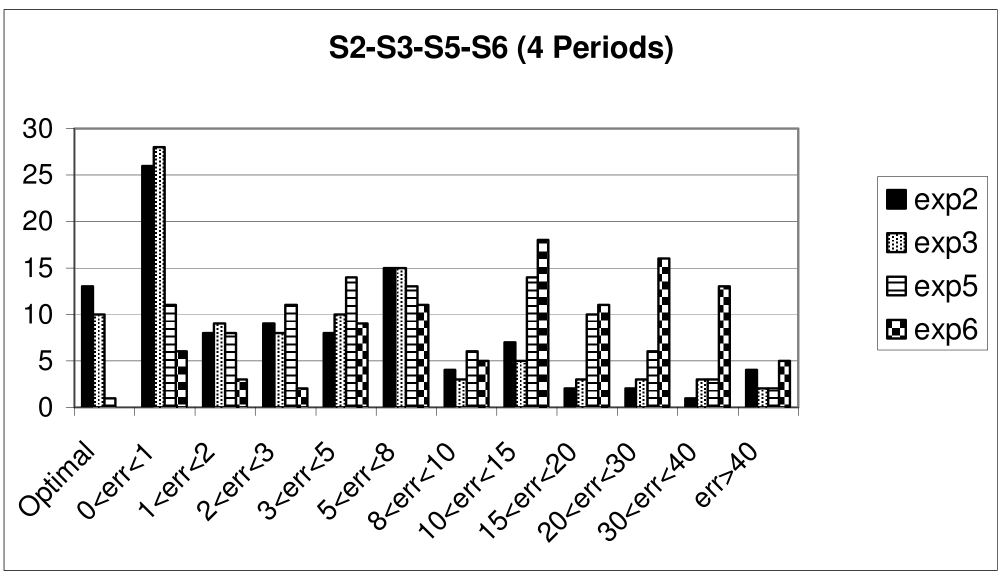

Figure A.10 Error distributions of Non-myopic algorithm for sample sets S2, S3, S5 and S6 for 4 periods planning horizon.

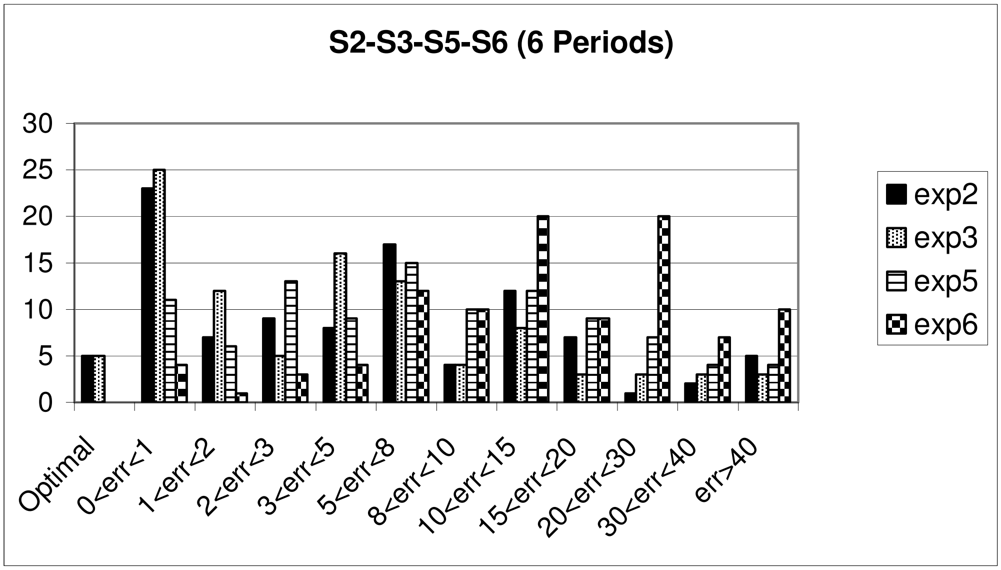

Figure A.11 Error distributions of Non-myopic algorithm for sample sets S2, S3, S5 and S6 for 6periods planning horizon.

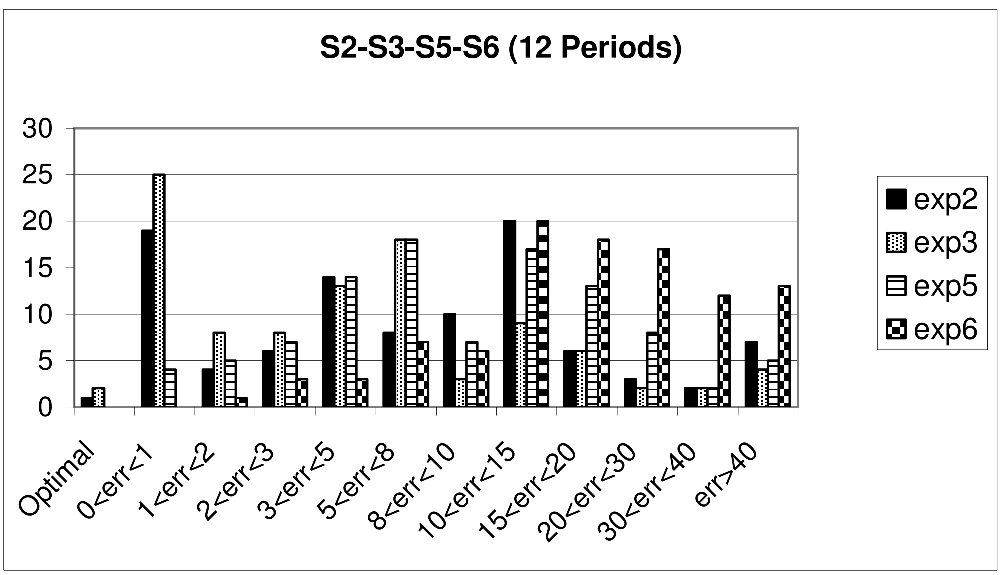

Figure A.12 Error distributions of Non-myopic algorithm for sample sets S2, S3, S5 and S6 for 12 periods planning horizon.

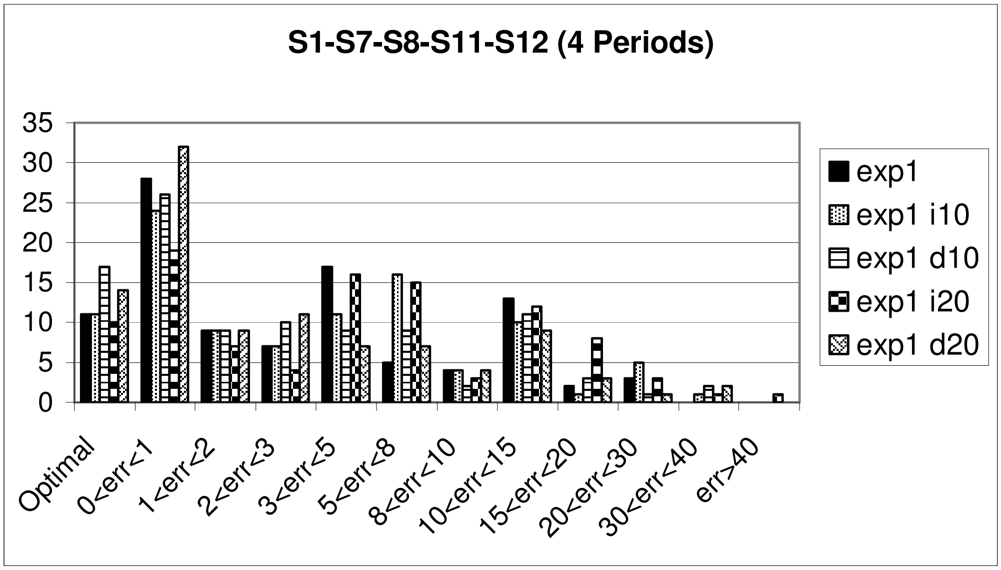

Figure A.13 Error distributions of Myopic NC-first algorithm for sample sets S1, S7, S8, S11 and S12 for 4 periods planning horizon.

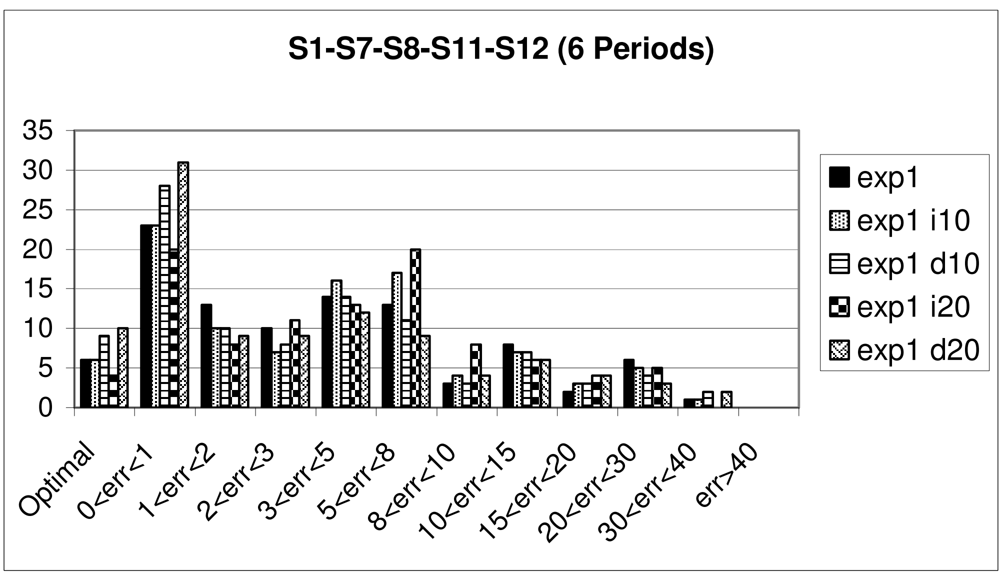

Figure A.14 Error distributions of Myopic NC-first algorithm for sample sets S1, S7, S8, S11 and S12 for 6 periods planning horizon.

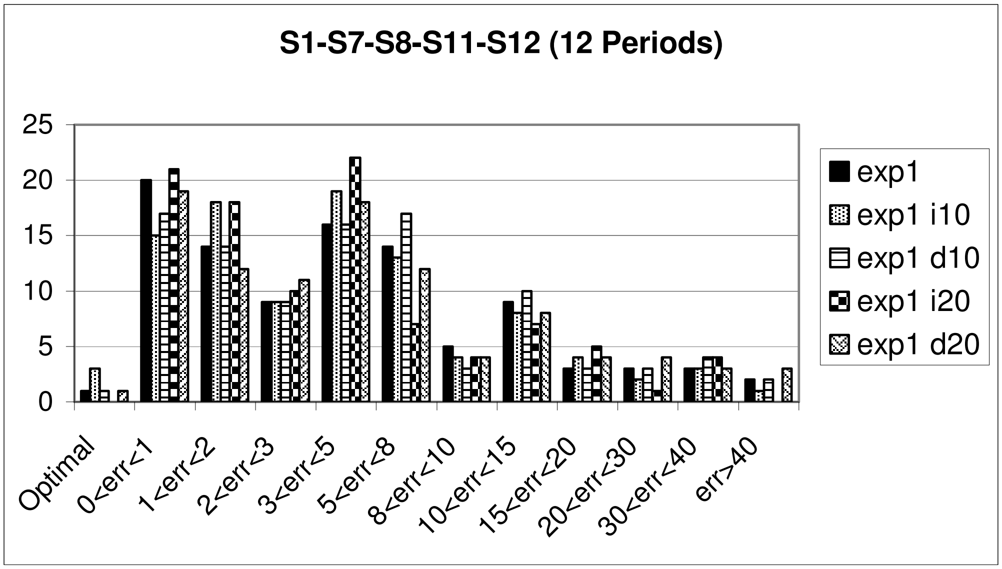

Figure A.15 Error distributions of Myopic NC-first algorithm for sample sets S1, S7, S8, S11 and S12 for 12 periods planning horizon.

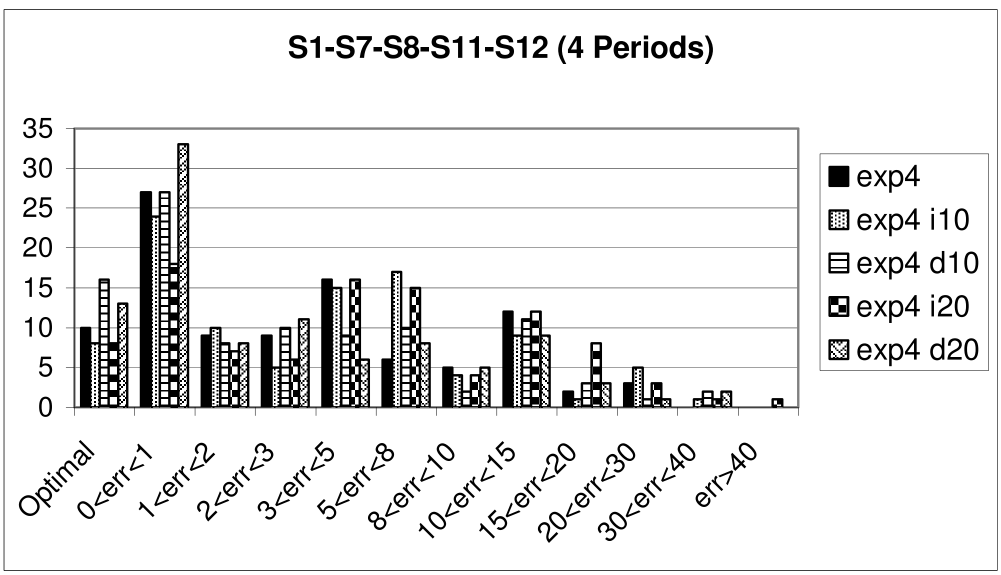

Figure A.16 Error distributions of Non-myopic NC-first algorithm for sample sets S1, S7, S8, S11 and S12 for 4 periods planning horizon.

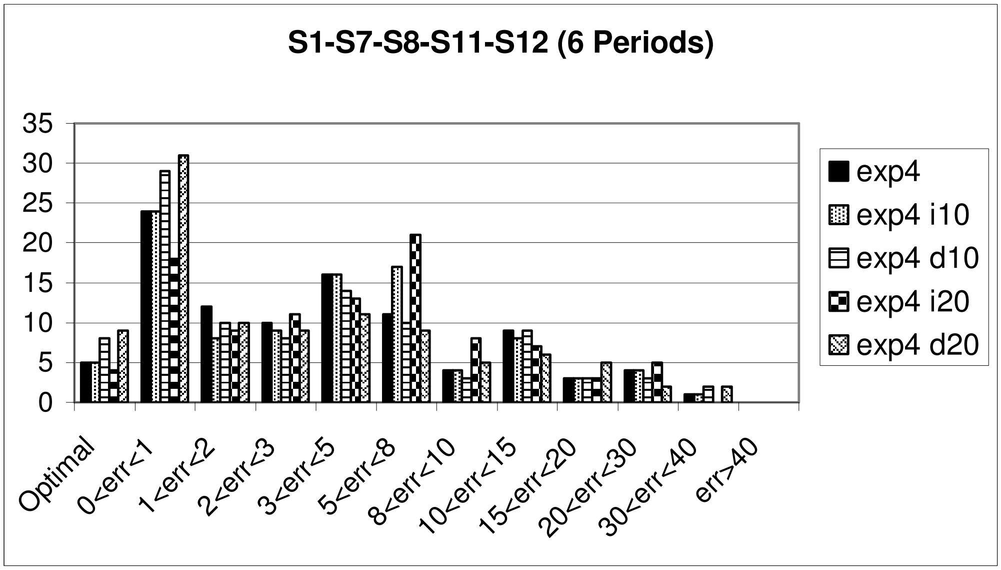

Figure A.17 Error distributions of Non-myopic NC-first algorithm for sample sets S1, S7, S8, S11 and S12 for 6 periods planning horizon.

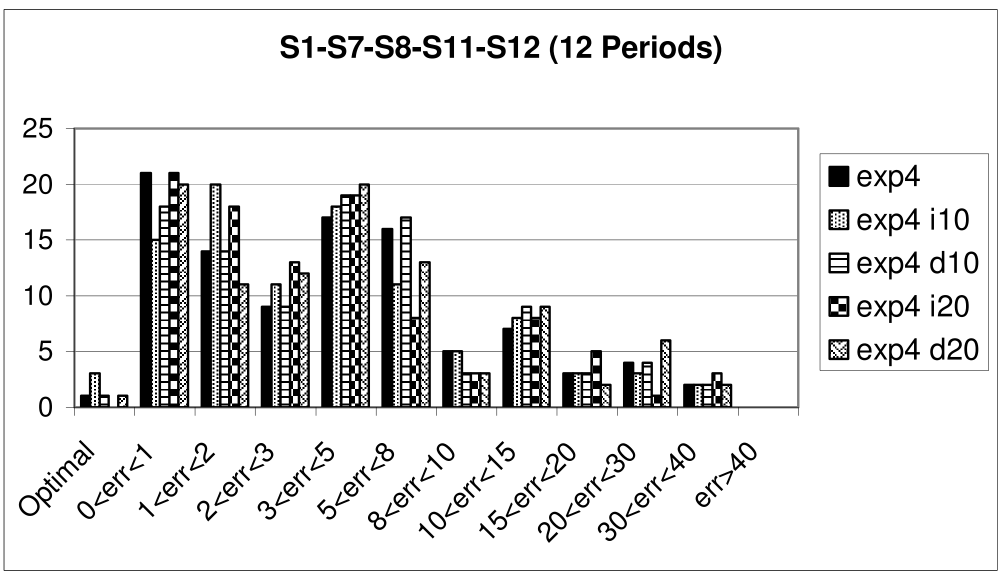

Figure A.18 Error distributions of Non-myopic NC-first algorithm for sample sets S1, S7, S8, S11 and S12 for 12 periods planning horizon.

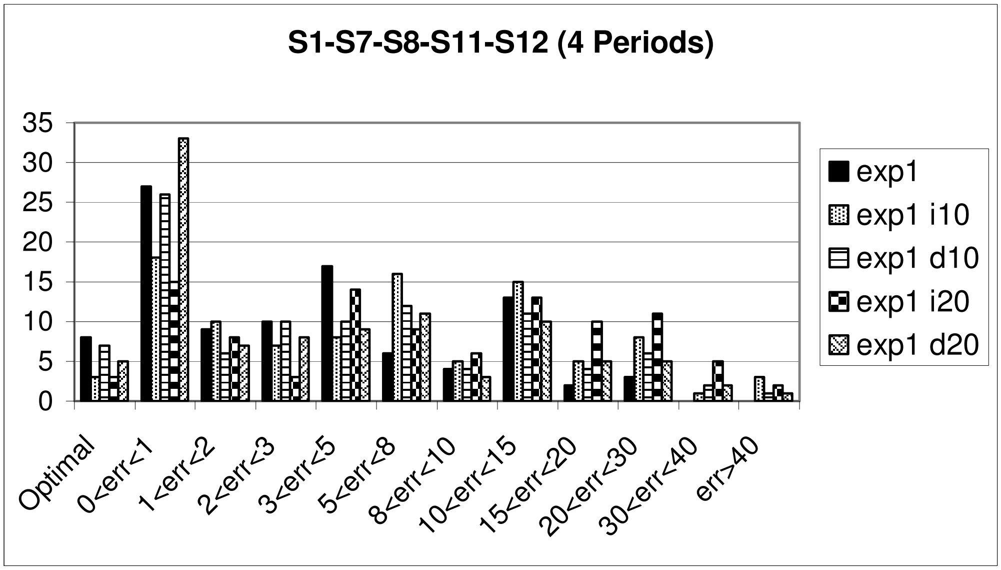

Figure A.19 Error distributions of Myopic algorithm for sample sets S1, S7, S8, S11 and S12 for 4 periods planning horizon.

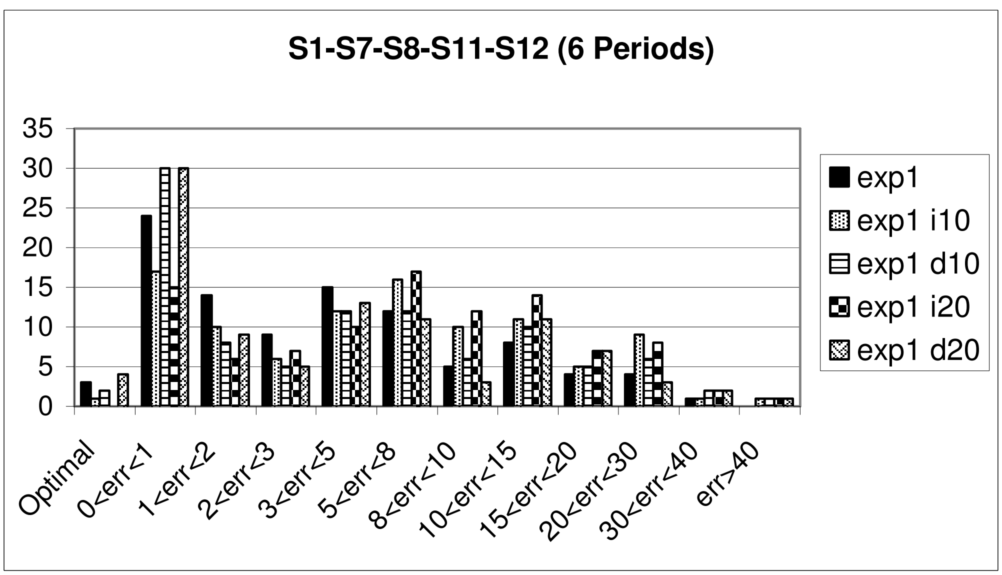

Figure A.20 Error distributions of Myopic algorithm for sample sets S1, S7, S8, S11 and S12 for 6 periods planning horizon.

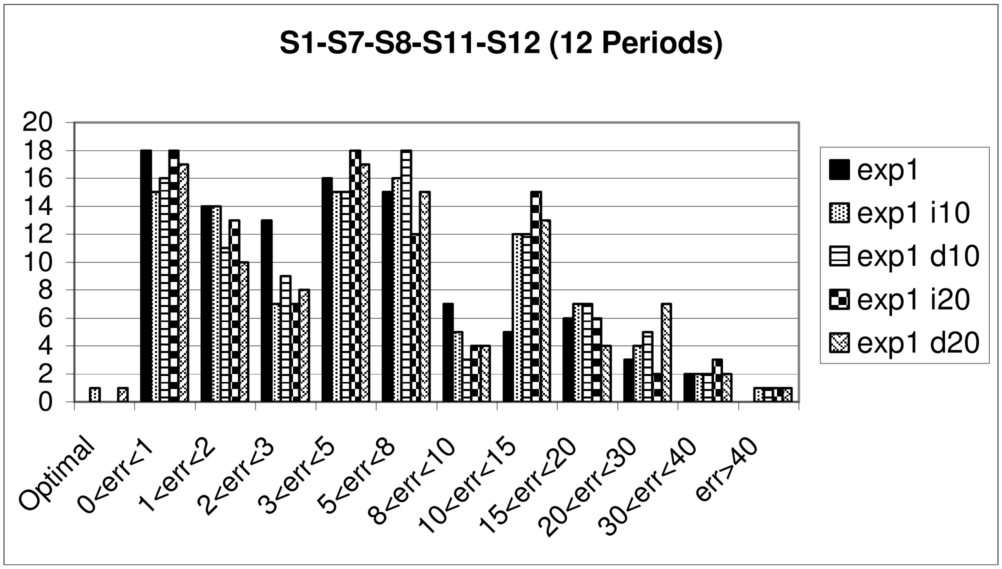

Figure A.21 Error distributions of Myopic algorithm for sample sets S1, S7, S8, S11 and S12 for 12 periods planning horizon.

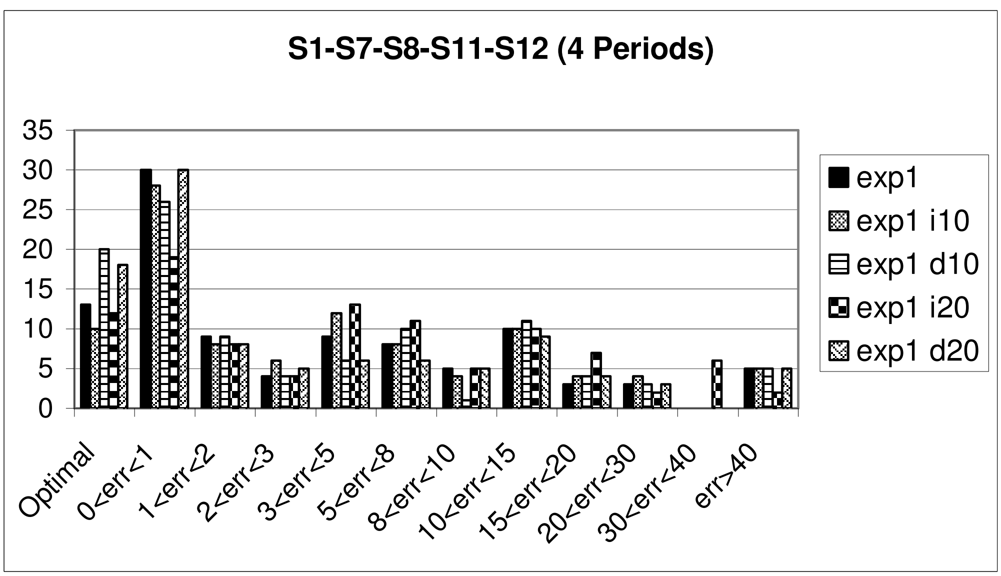

Figure A.22 Error distributions of Non-myopic algorithm for sample sets S1, S7, S8, S11 and S12 for 4 periods planning horizon.

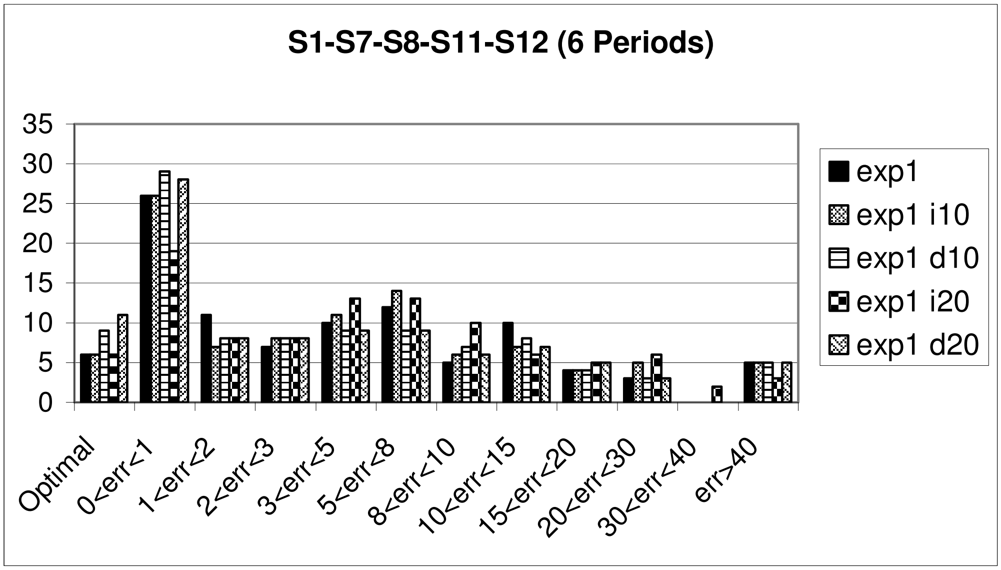

Figure A.23 Error distributions of Non-myopic algorithm for sample sets S1, S7, S8, S11 and S12 for 6 periods planning horizon.

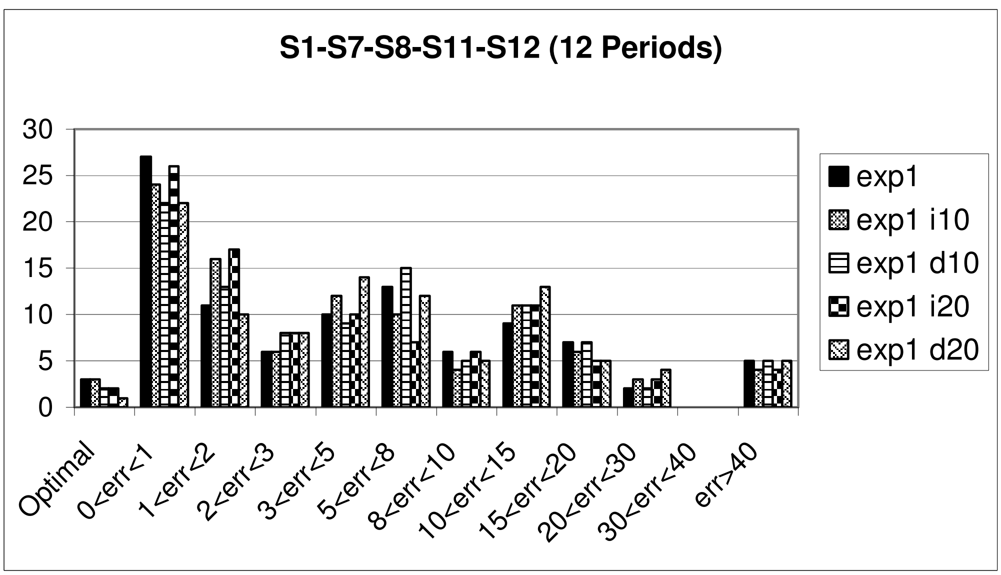

Figure A.24 Error distributions of Non-myopic algorithm for sample sets S1, S7, S8, S11 and S12 for 12 periods planning horizon.

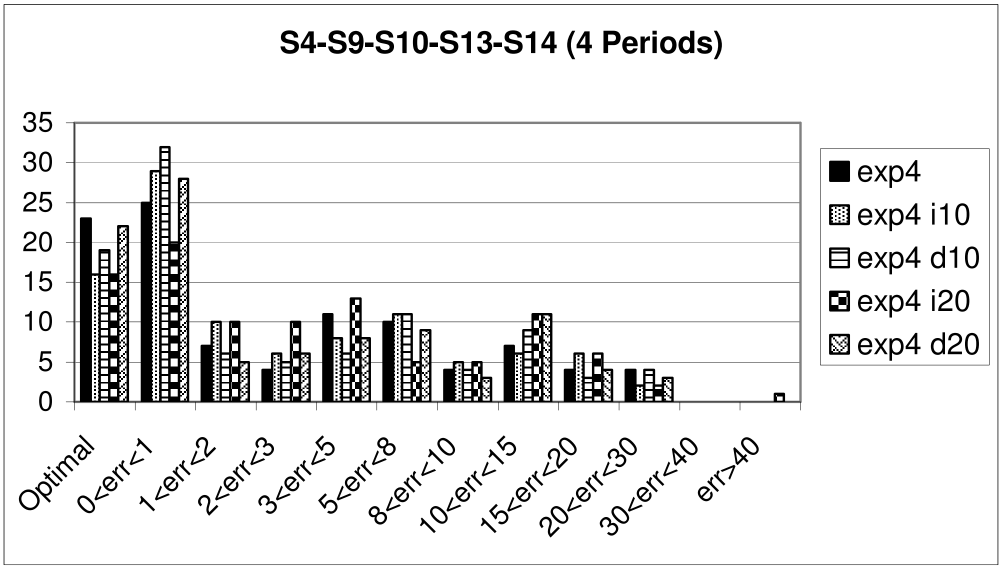

Figure A.25 Error distributions of Myopic NC-first algorithm for sample sets S4, S9, S10, S13 and S14 for 4 periods planning horizon.

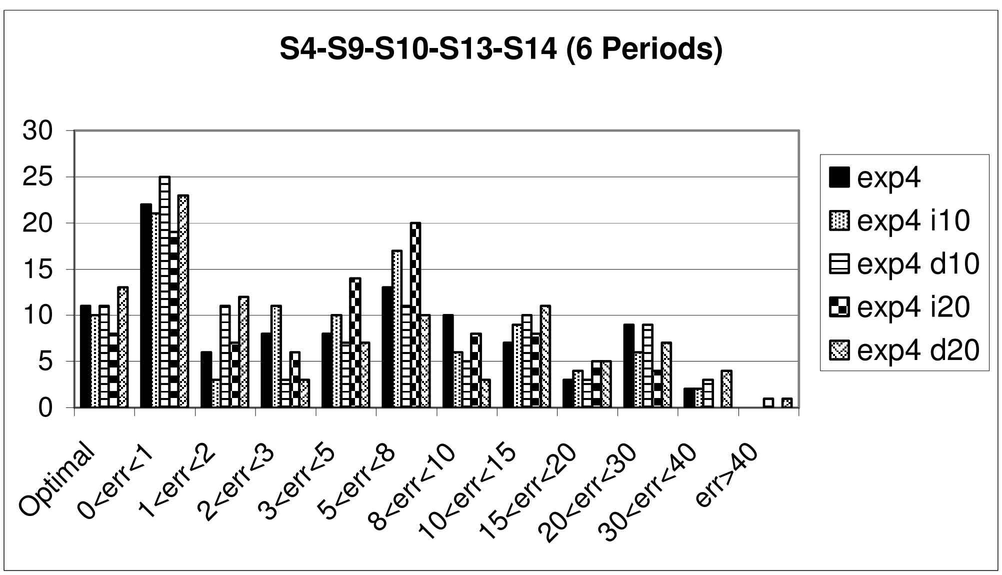

Figure A.26 Error distributions of Myopic NC-first algorithm for sample sets S4, S9, S10, S13 and S14 for 6 periods planning horizon.

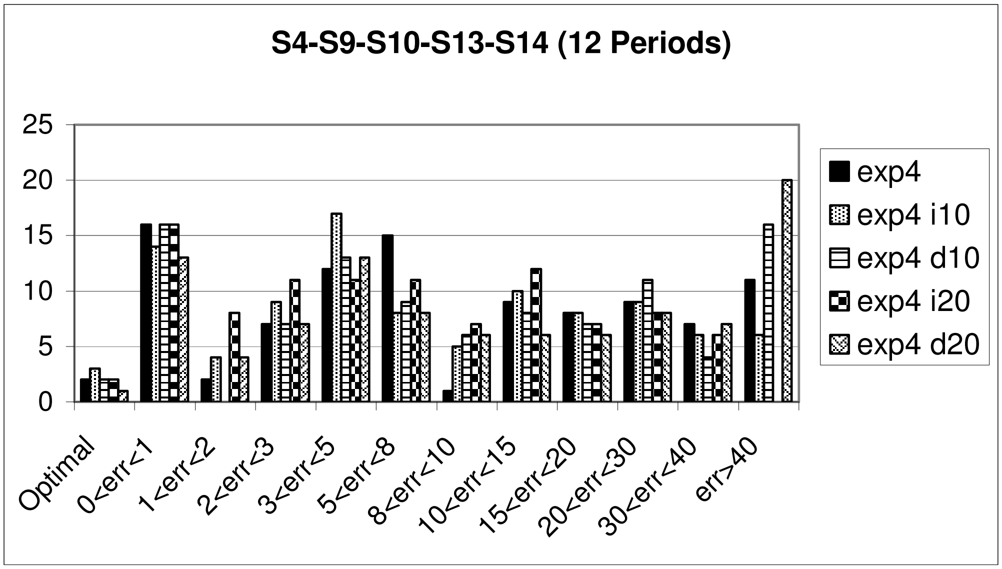

Figure A.27 Error distributions of Myopic NC-first algorithm for sample sets S4, S9, S10, S13 and S14 for 12 periods planning horizon.

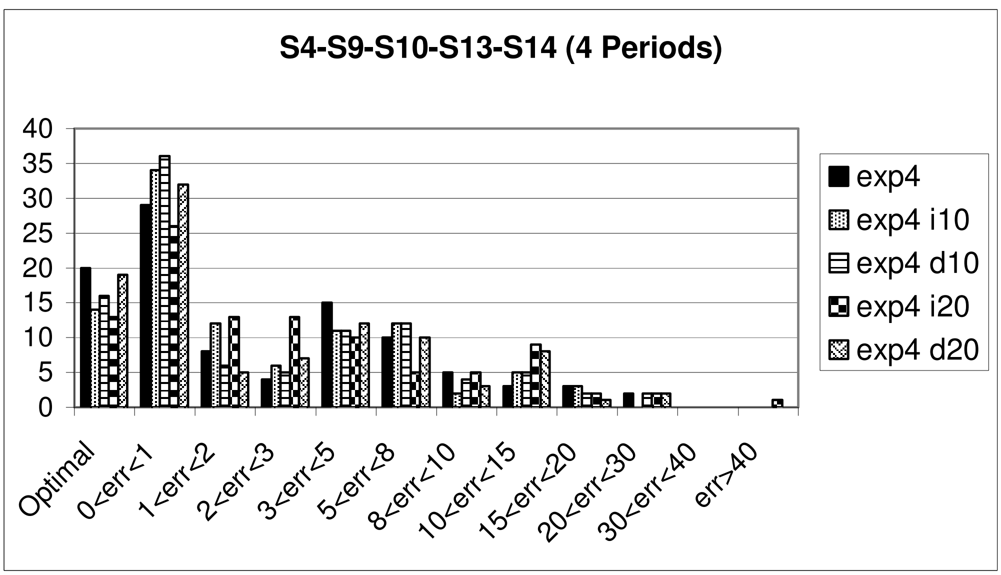

Figure A.28 Error distributions of Non-myopic NC-first algorithm for sample sets S4, S9, S10, S13 and S14 for 4 periods planning horizon.

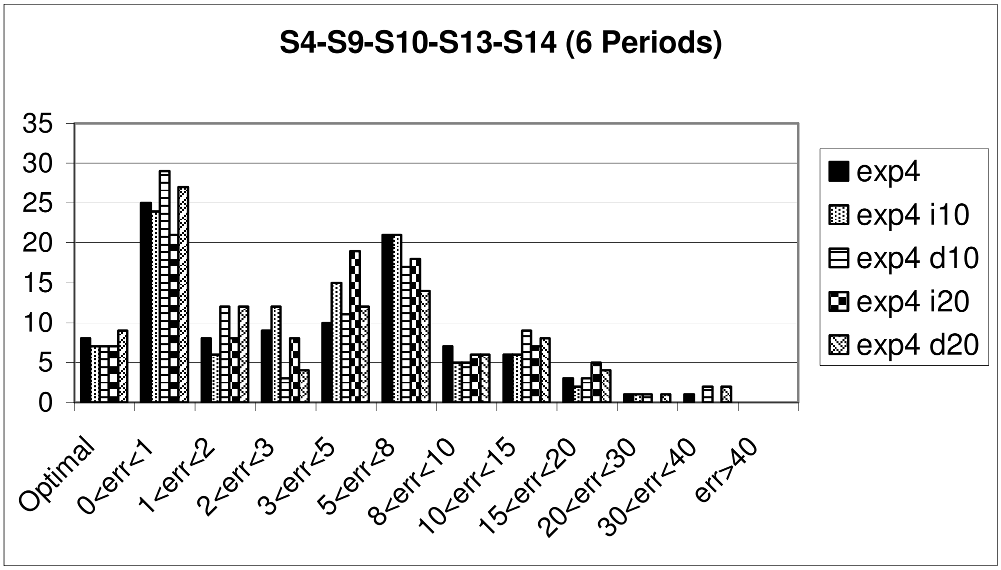

Figure A.29 Error distributions of Non-myopic NC-first algorithm for sample sets S4, S9, S10, S13 and S14 for 6 periods planning horizon.

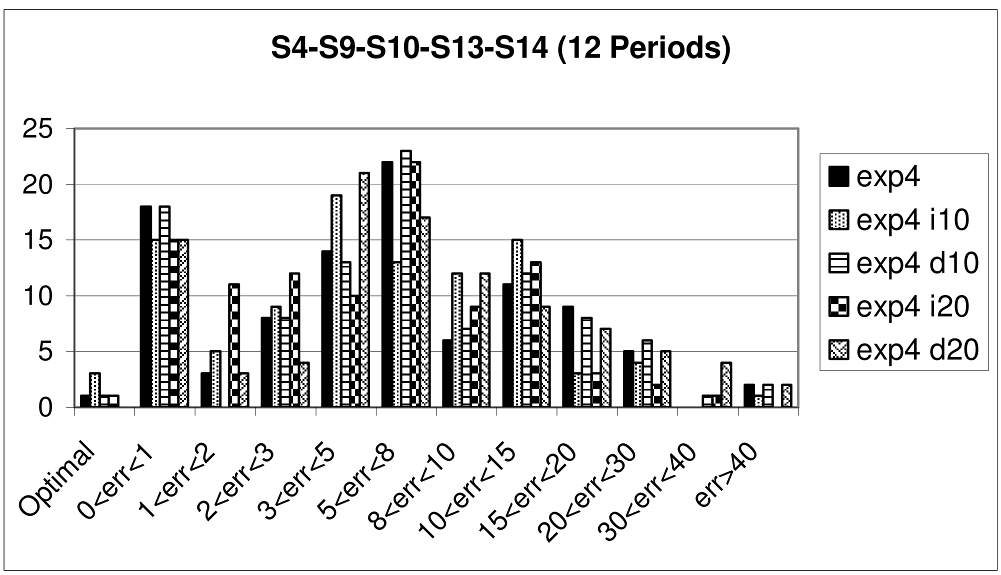

Figure A.30 Error distributions of Non-myopic NC-first algorithm for sample sets S4, S9, S10, S13 and S14 for 12 periods planning horizon.

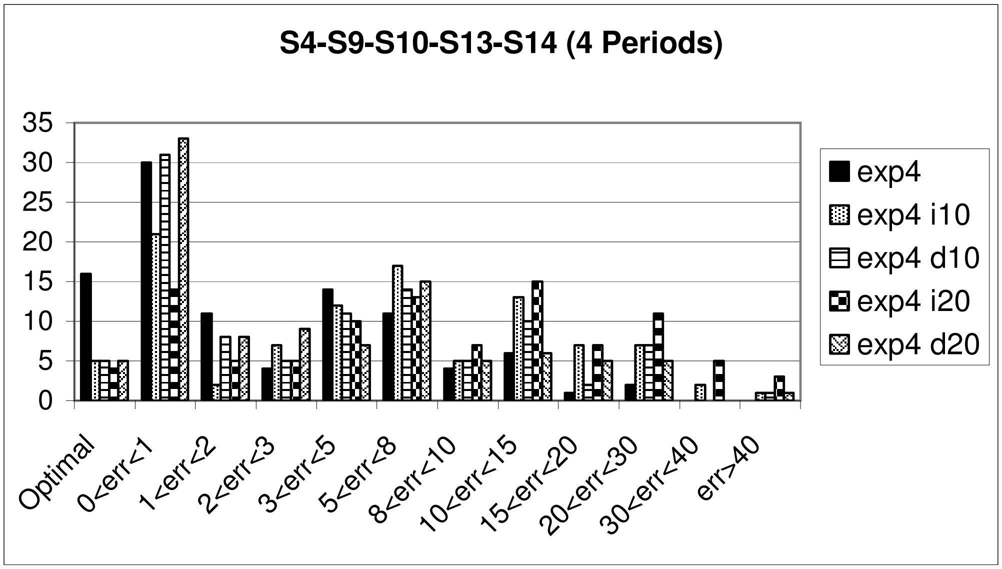

Figure A.31 Error distributions of Myopic algorithm for sample sets S4, S9, S10, S13 and S14 for 4 periods planning horizon.

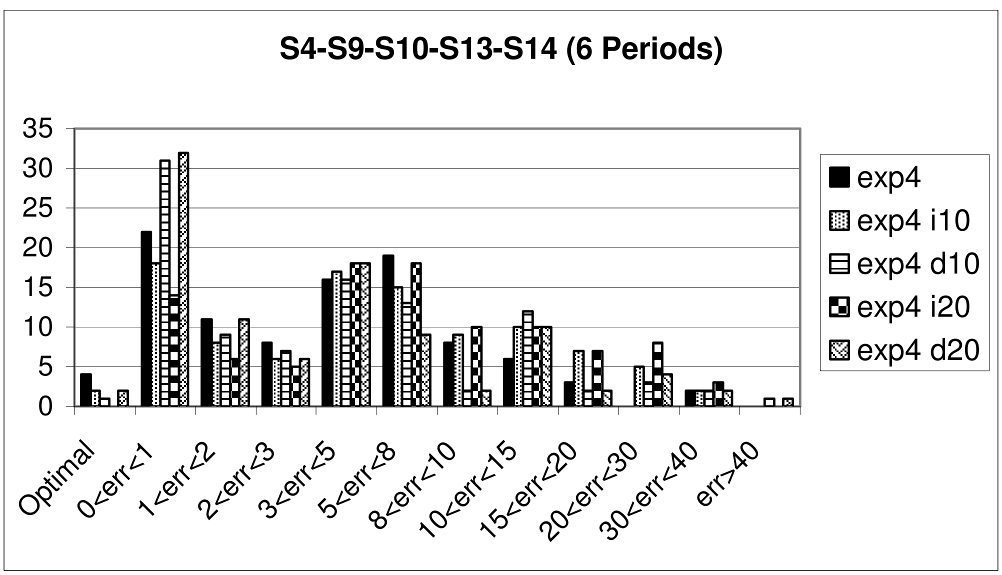

Figure A.32 Error distributions of Myopic algorithm for sample sets S4, S9, S10, S13 and S14 for 6 periods planning horizon.

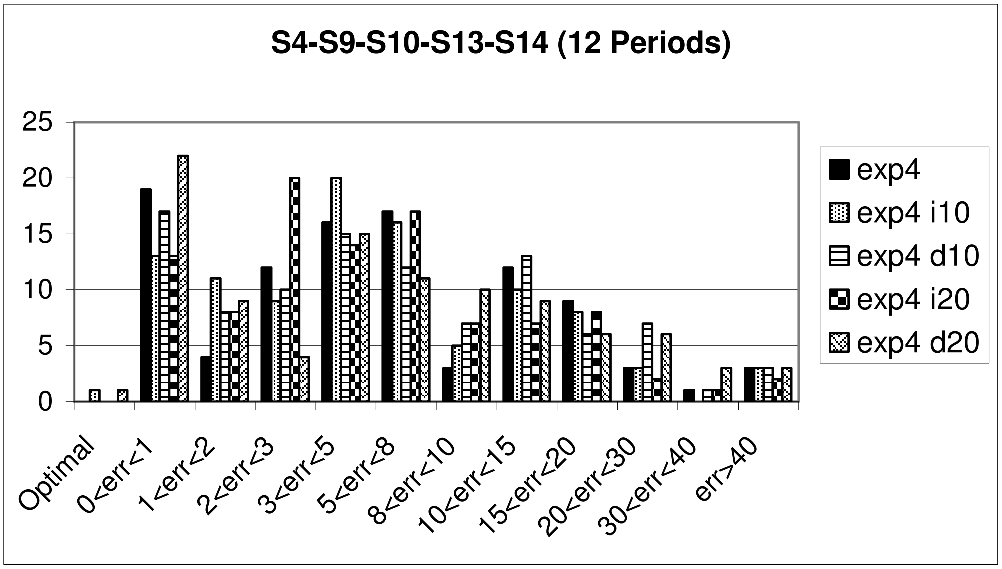

Figure A.33 Error distributions of Myopic algorithm for sample sets S4, S9, S10, S13 and S14 for 12 periods planning horizon.

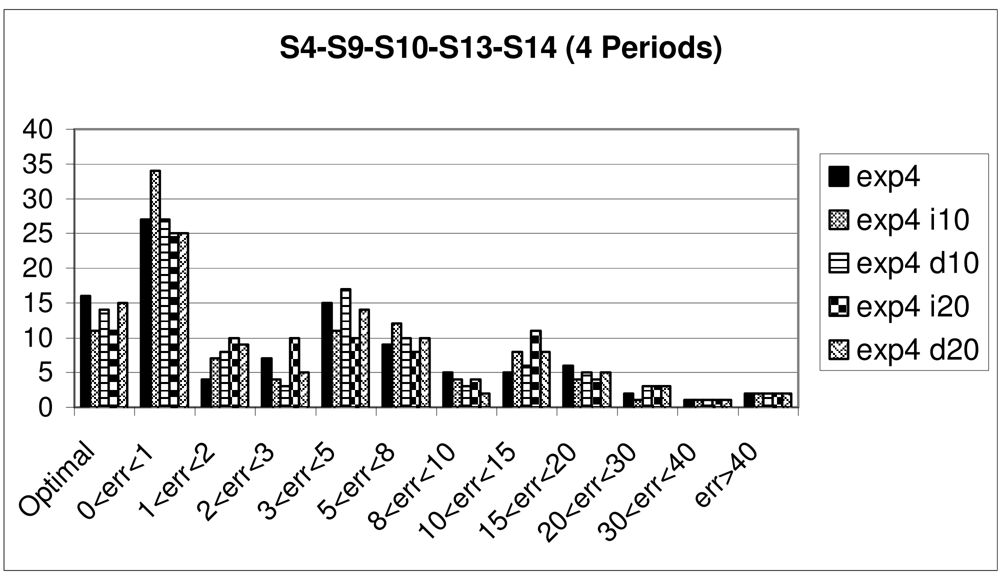

Figure A.34 Error distributions of Non-myopic algorithm for sample sets S4, S9, S10, S13 and S14 for 4 periods planning horizon.

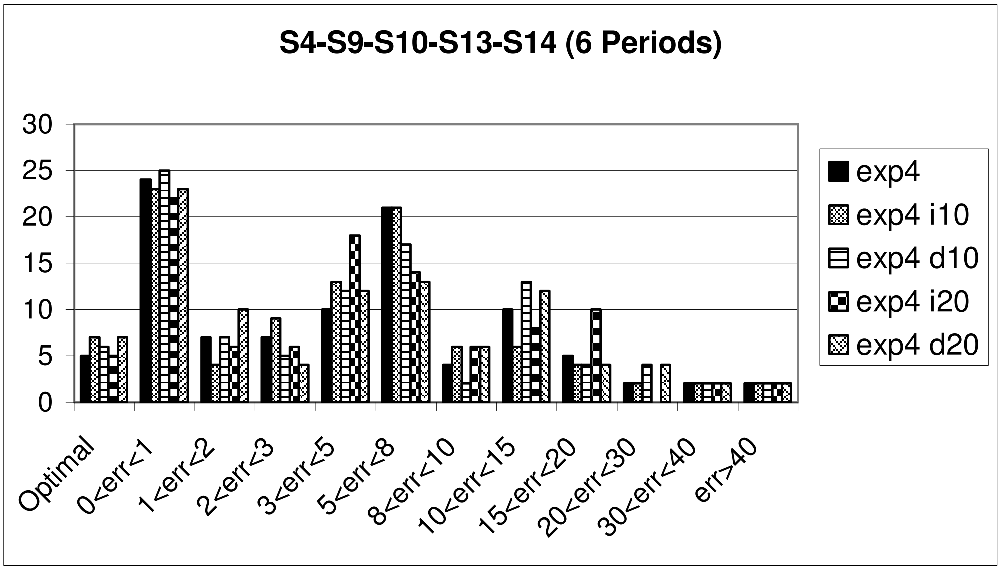

Figure A.35 Error distributions of Non-myopic algorithm for sample sets S4, S9, S10, S13 and S14 for 6 periods planning horizon.

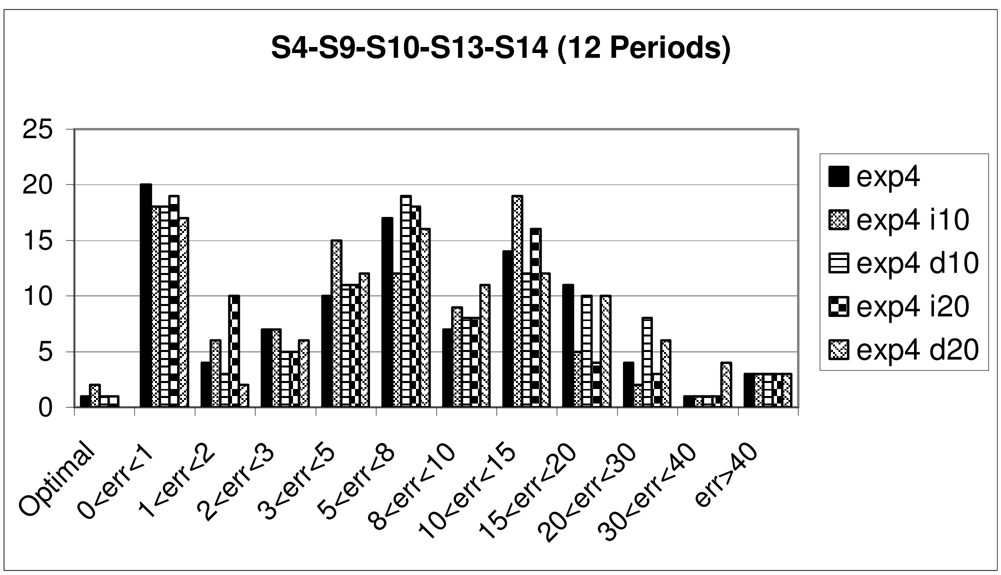

Figure A.36 Error distributions of Non-myopic algorithm for sample sets S4, S9, S10, S13 and S14 for 12 periods planning horizon.

APPENDIX D

ERROR DISTRIBUTIONS OF LOT SIZING HEURISTICS

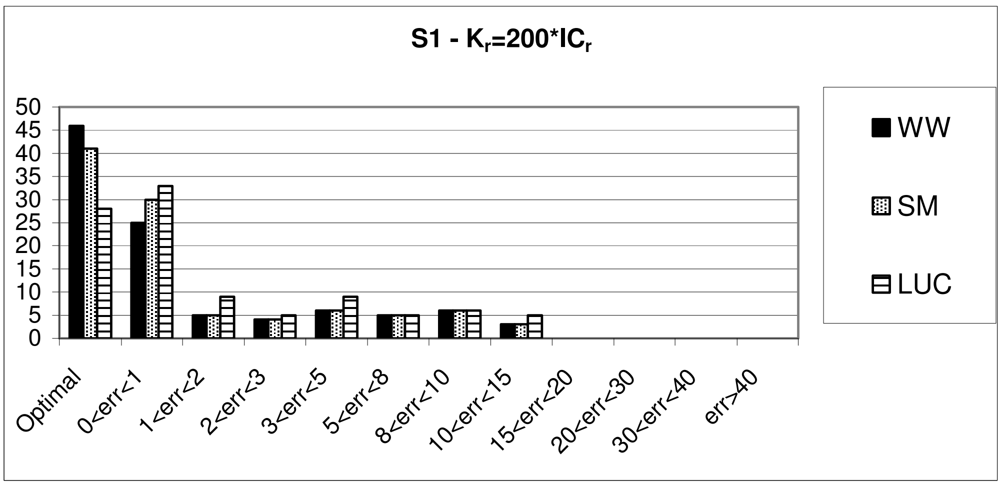

Figure B.1 Error distribution of lot sizing heuristics for S1 with $K_r = 200 * IC_r$

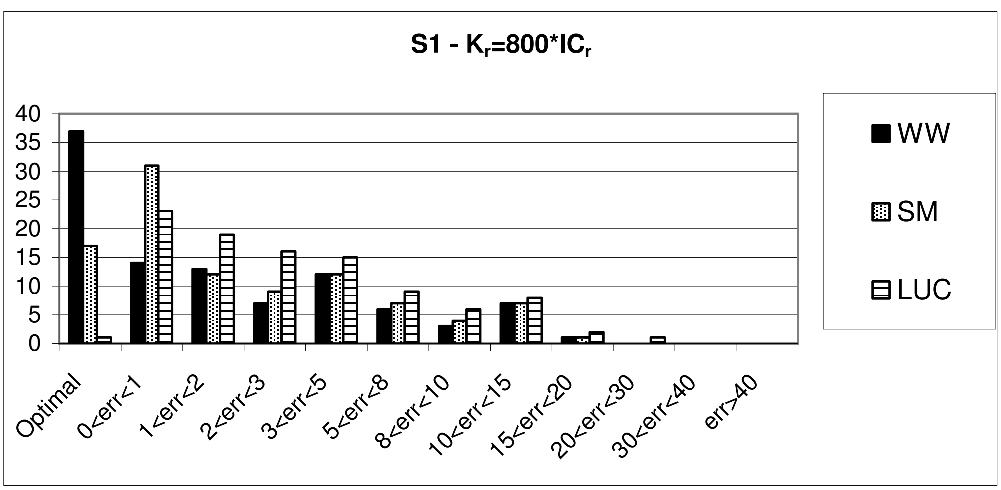

Figure B.2 Error distribution of lot sizing heuristics for S1 with $K_r = 800 * IC_r$

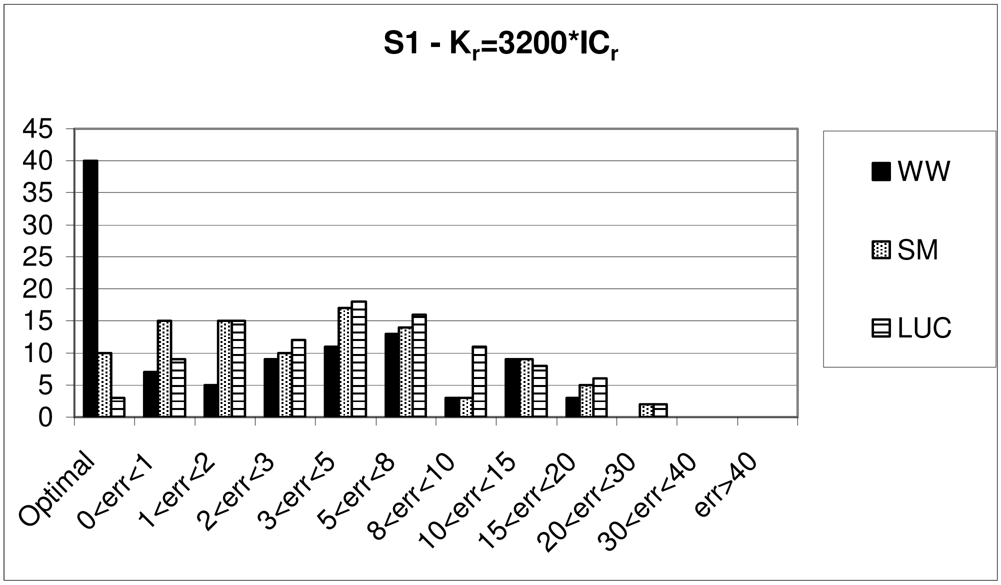

Figure B.3 Error distribution of lot sizing heuristics for S1 with $K_r = 3200 * IC_r$

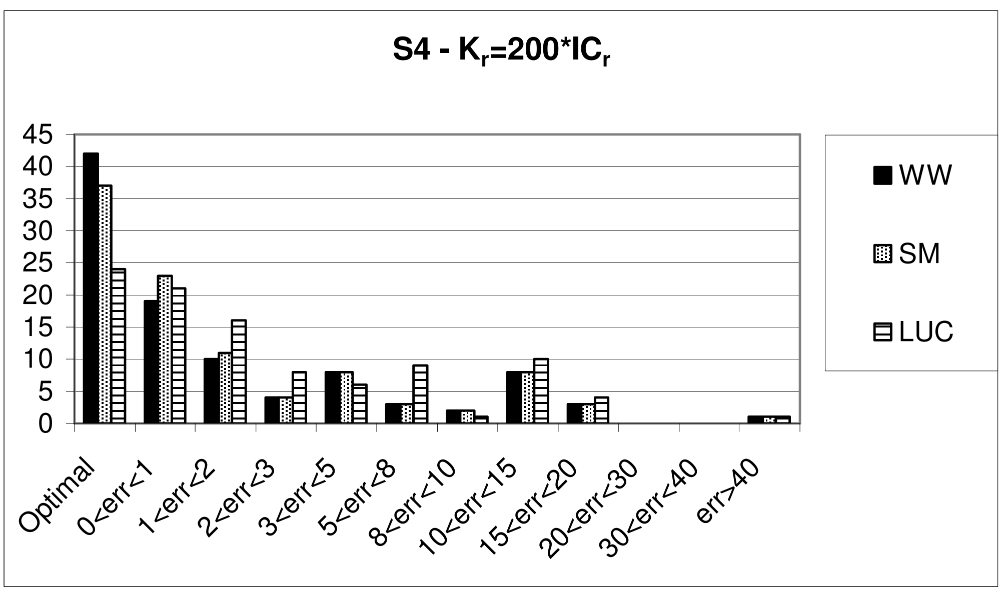

Figure B.4 Error distribution of lot sizing heuristics for S4 with $K_r = 200 * IC_r$

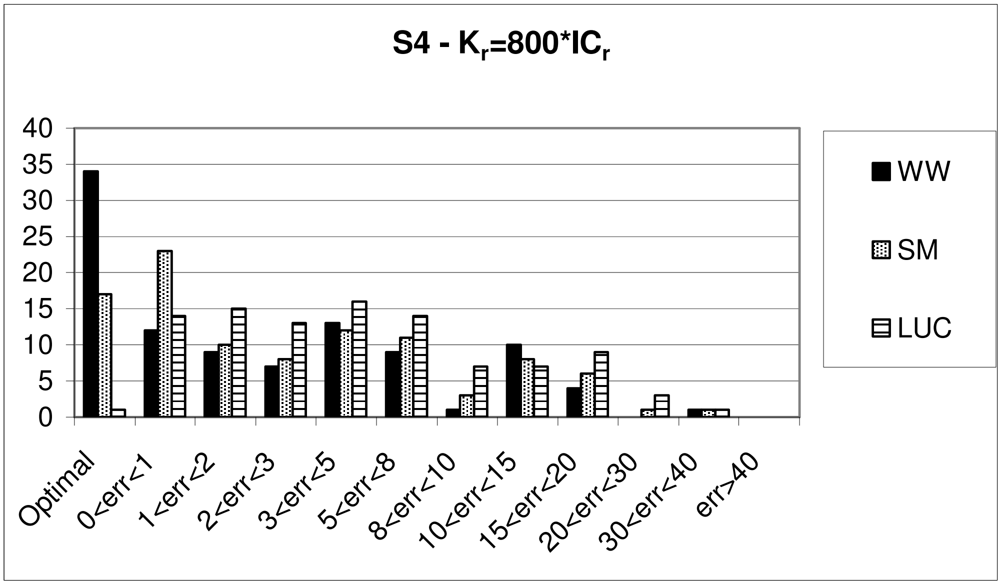

Figure B.5 Error distribution of lot sizing heuristics for S4 with $K_r = 800 * IC_r$

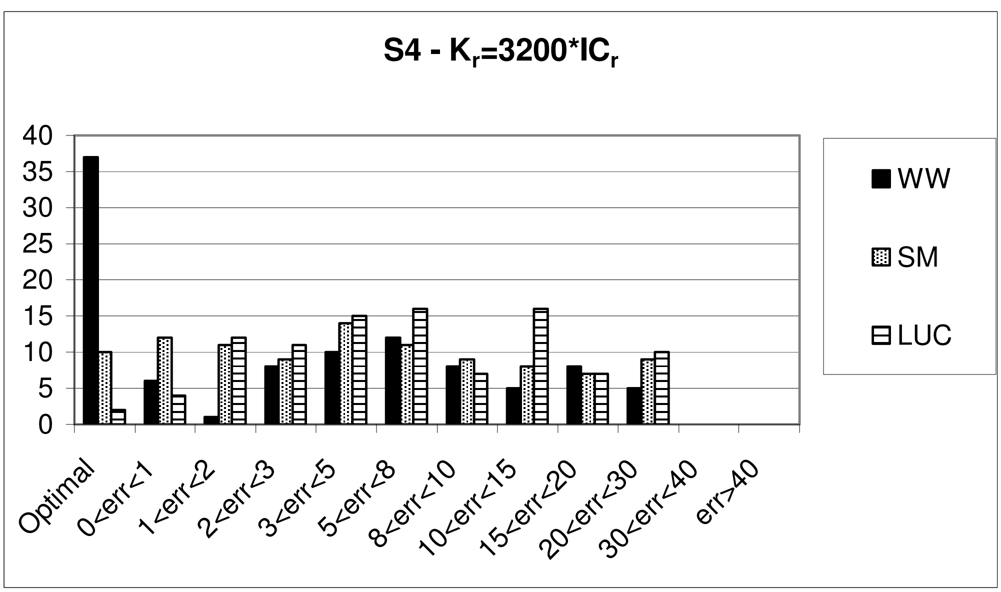

Figure B.6 Error distribution of lot sizing heuristics for S4 with $K_r = 3200 * IC_r$

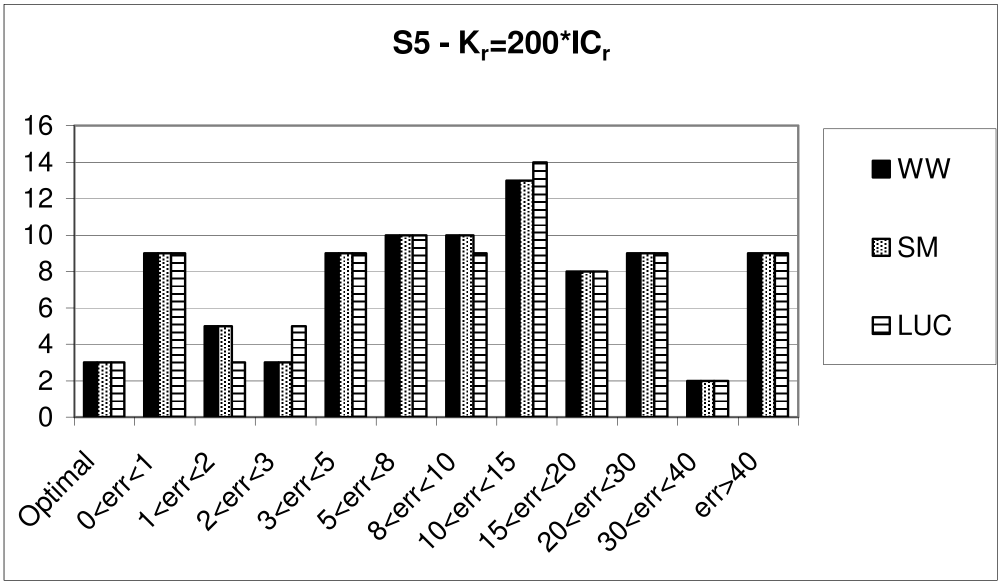

Figure B.7 Error distribution of lot sizing heuristics for S5 with $K_r = 200 * IC_r$

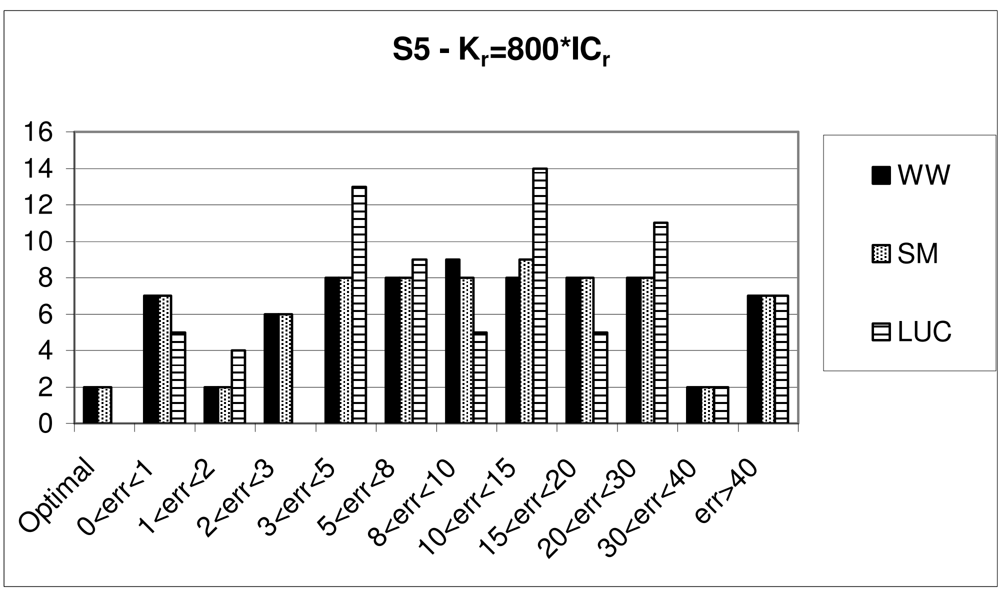

Figure B.8 Error distribution of lot sizing heuristics for S5 with $K_r = 800 * IC_r$

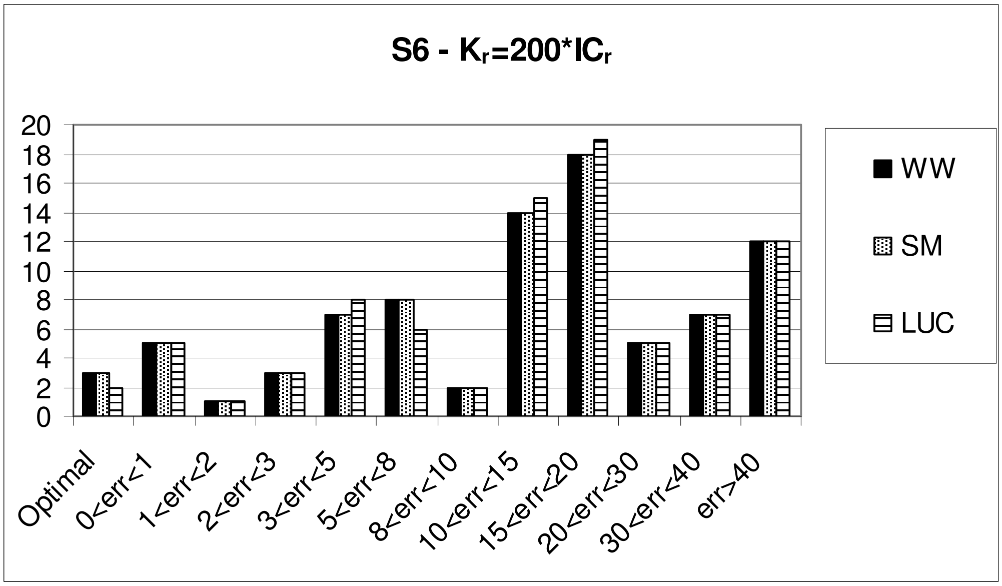

Figure B.9 Error distribution of lot sizing heuristics for S6 with $K_r = 200 * IC_r$

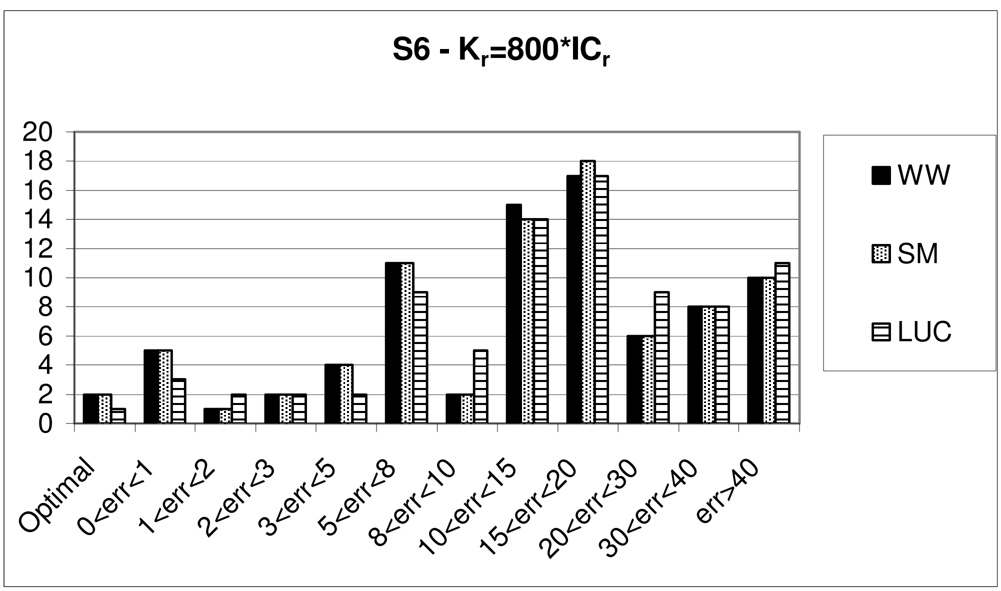

Figure B.10 Error distribution of lot sizing heuristics for S6 with $K_r = 800 * IC_r$