FORMULATIONS AND VALID INEQUALITIES FOR ECONOMIC LOT SIZING PROBLEMS WITH REMANUFACTURING (ELSR)

by

Sharifah Aishah Binti Syed Ali
Department of Management Science
University of Strathclyde

A thesis presented in fulfilment of the requirement for the degree of
Doctor of Philosophy
(Management Science)
2016

Declarations of Authenticity and Author Rights

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Some parts of this thesis have been presented in academic event and conferences:

(i) Syed Ali, S. A., Akartunalı, K. and van der Meer, R. [2014]. Computational analysis of lower bounds for economic lot-sizing problem with remanufacturing (ELSR), *Presented at 20th Conference of the International Federation of Operations Research Societies (IFORS), Barcelona, Spain.*

(ii) Syed Ali, S. A., Doostmohammadi, M., Akartunalı, K. and van der Meer, R. [2015]. Valid inequalities for economic lot-sizing problem with remanufacturing (ELSR): Separate setup case, *Presented at 27th European Conference on Operational Research (EURO), Glasgow, United Kingdom.*

(iii) Syed Ali, S. A., Akartunalı, K., Doostmohammadi, M. and van der Meer, R. [2015]. Valid inequalities for economic lot-sizing problem with remanufacturing (ELSR): Joint setup case, *Presented at 2015 INFORMS Annual Meeting Conference, Philadelphia, Pennsylvania, USA.*

Signed:

Date:

Abstract

Nowadays, many manufacturers are beginning to establish remanufacturing facilities due to the stricter government regulations on end-of-life product treatment, and the increasing public awareness towards environmental issues. Remanufacturing offers a huge potential for employment, and provides profitable business opportunities. However, production planning activities are more complex for remanufacturing, as they incorporate greater uncertainties and greater risk associated with product returns and demands. These activities become even more intricate in hybrid remanufacturing and manufacturing systems.

For this reason, we have investigated two variants of production planning of the hybrid remanufacturing and manufacturing systems, they are: i) Economic Lot Sizing Problem with Remanufacturing and Separate Setups (ELSRs) and ii) Economic Lot Sizing Problem with Remanufacturing and Joint Setups (ELSRj). In each period, the demands can be fulfilled by either remanufactured, or new products, or both. These problems have been proven to be NP-hard in general. Therefore, we study different approaches to tackle these problems.

First, we propose several traditional methodologies to obtain better lower bounds for both problems, namely $(\ell, S) - like$ inequalities and reformulation techniques, such as facility location (FL) reformulation, multi-commodity (MC) reformulation, and shortest path (SP) reformulation. Both theoretical and computational comparisons of different lower bounding techniques are discussed. The results show that the reformulation techniques demonstrate better performance than other formulations for the separate setups case when the setup cost for remanufacturing is equivalent to the setup cost for manufacturing. For the joint setups case, our $(\ell, S) - like$ inequalities, which have the same lower bounds as the reformulation techniques, are the most efficient methods to quickly solve the problem.

Motivated by the previous chapter, we further investigate the polyhedral structure of a simpler mixed integer set, arising from the feasible set of ELSRs and ELSRj problems, in order to derive several existing and new valid inequalities. These mixed integer sets are variants of the well-known single node fixed-charge network set, where two knapsack sets are considered simultaneously. Our main contributions for these problems rely upon on identifying the facet-defining conditions of the proposed inequalities, and discussing their separation problems. For each problem, comparisons of computational experiments between different traditional methodologies introduced earlier, and the proposed inequalities, are presented to test their effectiveness. The results indicate that the valid inequalities, with embedded $(\ell, S) - like$ inequalities for the separate setups case, have significantly improved the lower bounds in almost (all) instances tested, compared with other formulations when the setup cost for remanufacturing is, at most, the setup cost for manufacturing. As regards to the joint setups case, the results show that $(\ell, S) - like$ inequalities remain provide stronger lower bounds than the proposed inequalities for those randomly generated instances.

(434 words)

Keywords: Remanufacturing, Lot Sizing, Mixed Integer Programming, Polyhedral Study, Valid Inequalities

Acknowledgements

In the name of Allah, The Most Gracious and The Most Merciful

Alhamdulliah, all praise be to Almighty God, Allah for the blessing He bestowed on me to enable me to successfully finish this thesis. There is nothing easy except what He makes easy and He makes the difficult easy if it be His Will. I believe Him more than myself, and without Him, I am nothing.

My deepest gratitude and sincere appreciation go to the many individuals who have supported me throughout the completion of this thesis, especially to my mother, Rohani Hassan, and my family for their \textit{dua',} encouragement and inspiration. My family has always been my support system and taught me to never give up pursuing my passions and dreams.

I am grateful to my superb supervisors, Dr. Kerem Akartunali and Dr Robert van der Meer, for their invaluable guidance and continuous support throughout my research. They have seen my struggles during my studies and have always be understanding and patient with me. Thank you very much! Further, special thanks to Dr Mahdi Doostmohammadi for helping me develop an essential understanding of mathematical proofs and motivating me to work harder.

I also would like to express my thanks to my wonderful colleagues and friends, Noorseha Ayob, Norasmiha Mohd Nor, Ruzanna Mat Jusoh, Nita Ali, Noor Wini Mazlan, Hilya Mudrika Arini, Aby Subin, Seda Sucu, Erfan Rahimian, Junchi Tan, Christoph Werner, and many others I have not mentioned here for their friendship and their direct and indirect support. Thank you also to the staff in the department for being so kind and helpful. I should also not forget my lovely flatmate, Arie Restu Wardhani, for always encouraging me to do the best whenever I feel down. I am so delighted for her words of understanding and encouragement.

Last but not least, I am indebted to the Ministry of Education in Malaysia for their financial assistance throughout my four years of study at the University of Strathclyde and to my employer, National Defence University of Malaysia (NDUM), for providing me the opportunity to study abroad and allowing me to experience other countries' cultures, to meet great people, and to challenge my fear of learning new things. Thank you very much to all of you from the bottom of my heart!

SASA, Strathclyde, Glasgow'16

"No two things have been combined together better than Knowledge and Patience"

Prophet Muhammad P.B.U.H

"Two roads diverged in a wood, and I took the one less traveled by, And that has made all the difference"

Robert Frost

This thesis is affectionately dedicated to the memory of my late father, Syed Ali Syed Mohamed and to my beloved mother, Rohani Hassan, families and friends for their love, support and pray of day and night.

List of Algorithms

List of Figures

List of Tables

Abbreviations

AI	Average Improvement
B&B	Branch-and-Bound
B&C	Branch-and-Cut
FL	Facility Location reformulation
LB	Lower Bound
LP	Linear Programming
MC	Multi-Commodity reformulation
MIP	Mixed Integer Programming
SP	Shortest Path reformulation
UB	Upper Bound

Nomenclature

N	Number of periods in the production planning problem
$\mathbb{R}^n$	The n-dimensional space of real values
$\mathbb{R}^n_+$	The n-dimensional space of nonnegative real values
$\mathbb{Z}^n$	The n-dimensional space of integer values
$\{0,1\}^n$	The n-dimensional space of binary values
$conv(X)$	Convex hull of the feasible set of points X
$dim(X)$	Dimension of a polyhedron X
$O\left(\right)$	Big-O notation for problem complexity
NP	The complexity class NP

Contents

Chapter 1

Introduction

1.1 Motivation

The increasing scarcity of the earth's natural resources and disposal capacity are global environmental problems. This is driven by technological development of new products, which has led to the excessive consumption of raw materials and energy in many industry sectors. Due to this, original equipment manufacturers (OEMs) in many industries are beginning to remanufacture used products, namely single-use cameras, machine tools, copiers, ink cartridges, computers, automotive parts, tires, aviation equipment and medical instruments (Ferrer, 1997; Guide Jr et al., 1997; Lebreton and Tuma, 2006; Matsumoto and Umeda, 2011; Cao et al., 2012; Ahmadi et al., 2013; Xia et al., 2015).

Remanufacturing is an industrial process that brings used products to at the least OEM functioning order with a warranty to match (Ijomah, 2009). It is the most advanced product recovery option and offers value-added recovery, extends product's life cycles, reduces landfill waste, raw materials, and energy consumption, and involves specialized labour (Shi et al., 2011). As reported by the Centre for Remanufacturing and Reuse, the UK remanufacturing industry contributes around £5 billion per annum to the economy, creates jobs for more than 500,000 people and saves 270,000 tonnes of materials (mostly metals) from recycling or scrapping (Chapman et al., 2010). Furthermore, in March 2014, the former Environment Secretary Caroline Spelman, on behalf of the All-Party Parliamentary Sustainable Resource Group, said that the UK remanufacturing industry has huge financial potential to increase from the current value of £2.4 billion to £5.6 billion , creating of thousands of skilled jobs (The All-Party Parliamentary Sustainable Resource Group, 2014).

In addition, in October 2015, the European Remanufacturing Network's researchers working on the EU Horizon2020 project carried out a survey of the current level of remanufacturing in the EU in nine main sectors, including the aerospace, medical equipment, electronics, furniture and rail sectors. The findings show that remanufacturing benefits from greater profit margins, generates an estimated €30

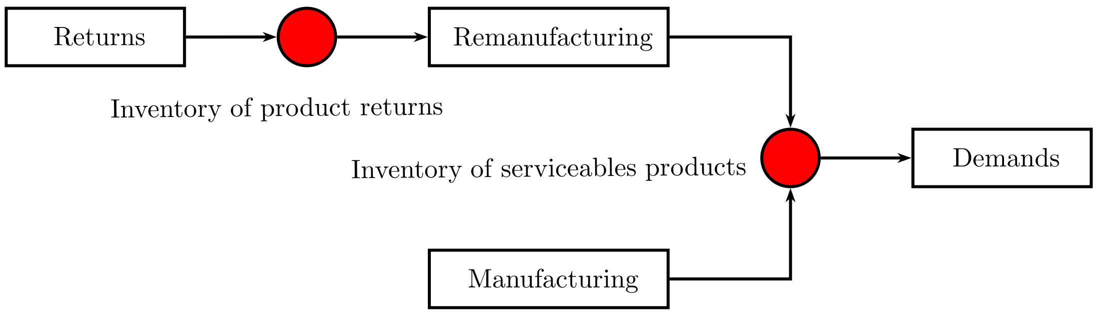

Figure 1.1.1: Material flows in a hybrid model

billion in annual since over two-thirds of remanufactured products sell for between 41% and 80% of the cost of a new product, and employs around 190,000 people. Apart from this, remanufacturing offers new alternative business models (rental and service-based) that create better relationships with customers and a flexible workforce (Parker et al., 2015).

In remanufacturing systems, there are two types of business strategies: a dedicated model (remanufacturing) and a hybrid model (manufacturing-remanufacturing). OEMs that employ a dedicated model normally outsource their operations to third-party remanufacturers. This is because remanufacturing is much more reactive and less visible compared to manufacturing. It involves an inherently complex kind of a manufacturing process that requires specific tools, high-technology machinery and multi-skilled labour. Moreover, the three main sub-processes of remanufacturing—disassembly, reprocessing and reassembly–incorporate a higher degree of uncertainty and risk associated with end-of-life products; this complicates production planning and control activities (Guide Jr, 2000). These planning activities become even more complex in a hybrid model when remanufacturing is carried out in combination with original manufacturing. According to Patel (2006), remanufacturing in North America generally follows a dedicated model; in contrast, most remanufacturing operations in European countries employ a hybrid model (Li et al., 2009).

In this thesis, our main interest is investigating the economic lot-sizing problem of hybrid remanufacturing-manufacturing systems that arise in production planning. The problem is to find an effective production plan that meets demand for remanufactured and new products on time as minimises total setup, production and inventory holding costs. The material flows of a hybrid model are illustrated in Figure 1.1.1.

Three major assumptions are present in our model, namely The single-level, single-item uncapacitated lot-sizing problem; deterministic returns and demand over a finite planning horizon; and ensuring that the quality of remanufactured products is as good as that of new products. The first assumption is the consideration of the single-stage, single-item uncapacitated lot-sizing problem. Even single-stage systems do not describe most real-life production systems; however, they provide good insights and ideas about coping with more complex problems. In single-stage lot-sizing problems, the remanufacturing or manufacturing process is characterized by a single-level product structure in which products are directly produced from used products (remanufacturing) or raw materials (manufacturing) without intermediate stocking points or subassembly.

The second major assumption is that both returns and demand are deterministic. According to Souza (2012), the assumption of deterministic returns is possible in a situation when returns are retrieved from leasing operations. Moreover, returns can be also forecasted for the entire planning horizon within an appropriate approximation. An example of a realistic case of deterministic returns is found in Golany et al. (2001), where the demand for or returns of the packaging and shipping materials (such as pallets or containers) used in shipments are known as the shipments are planned in advance. Obviously, the assumption of known demand is not reasonable; however, it can be determined for a rolling horizon. In other words, the quantities of current inventory and future returns and demands are approximated and updated by period (Ferguson, 2010).

The last major assumption is that demand can be satisfied by either remanufactured or new products. This is referred to as serviceable products, where unfulfilled demand for remanufactured products can also be satisfied by new products. The new products and remanufactured products cannot be distinguished since all products may consist of reused parts. For example, Fuji Xerox's remanufacturing operations in Japan integrates reused parts in new products (Matsumoto and Umeda, 2011). Another interesting example is the Kodak line of single-use cameras, where the parts can be reused multiple times, including the polymer, which is used to cast new parts, and film is the only new material required. Since consumers are mainly concerned about the quality of the film, they are not aware of the parts used in the camera even though they are obviously labelled as remanufactured parts on the packaging (Atasu et al., 2010). Lastly, as stated by Thierry et al. (1995), if products have a service contract, demand can be satisfied from both sources as the remanufactured products are treated in the same way as new products, with similar warranties and service contracts and identical lease prices (Retel Helmrich et al., 2013).

In this study, we consider two different setup cost schemes: separate setups costs for remanufacturing and manufacturing (dedicated production line) and a joint setup cost (single production line). There is a pressing need to study both types of production lines to support decision making in closed-loop supply chains. For instance, in the case of joint setups, Tang and Teunter (2006) studied an actual case company, Autopart, which manufactures and remanufacturers car parts, with both remanufacturing and manufacturing operations performed on a single production line. Teunter et al. (2008) further investigated the same case company and the use of separate production lines. Using the same set of cases considered by Tang and Teunter (2006), they analysed the cost benefits of switching from a single line to separate lines.

The next section gives a brief overview of the simple lot-sizing problem and provides general insights that are useful for our models.

1.2 Simple Lot Sizing Problem

The dynamic economic lot size model was firstly introduced in the seminal paper of Wagner and Whitin in 1958. The model is also known as an uncapacitated single-item lot sizing problem, which aims to determine when and how much of a product to produce, such that total sum of the costs, i.e. a fixed setup cost, a nonnegative inventory holding cost and a constant production cost, are minimized while assuming that deterministic demands are satisfied. This problem is modeled as a mixed integer programming formulation; and basic decision variables and parameters used in this formulation are given as follows.

Decision variables

x_t is the amount of products produced in period t,

y_t is 1 if the production takes place in period t, 0 otherwise,

I_t is the inventory of products at the end of period t.

Parameters

h_t is unit holding cost for inventory in period t,

K_t is unit setup cost in period t,

M is the big-M value, an upper bound on x_t,

d_t is the amount of products demanded in period t,

n is the number of periods in the planning horizon such that $N = 1, ..., n$.

$$Z^c = \min \sum_{t=1}^{n} (K_t y_t + h_t I_t) \tag{1.1}$$

$$\text{s.t.} \quad I_t = I_{t-1} + x_t - d_t \qquad \forall t \in N \tag{1.2}$$

$$x_t \leq M y_t \qquad \forall t \in N \tag{1.3}$$

$$x_t, I_t \geq 0 \qquad \forall t \in N \tag{1.4}$$

$$y_t \in \{0, 1\}^n \qquad \forall t \in N \tag{1.5}$$

$$I_0 = I_n = 0 \tag{1.6}$$

The objective function defined by (1.1) is the minimization of the total costs, i.e. the setup costs and the holding costs. Constraint (1.2) represents the inventory flow balance.Constraint (1.3) forces variable y_t to be 1 if production occurs in a given period t. The big-M value here refers to a large positive number, which is known as an upper bound on the maximum lot size in period t. Constraint (1.4) ensure nonnegativity of production and inventory. Constraint (1.5) ensure the binary nature of the setup variable. Finally, without loss of generality, we assume there is no inventory on hand initially and at the end of time period n. This simple lot sizing problem can be illustrated in Figure 1.2.1.

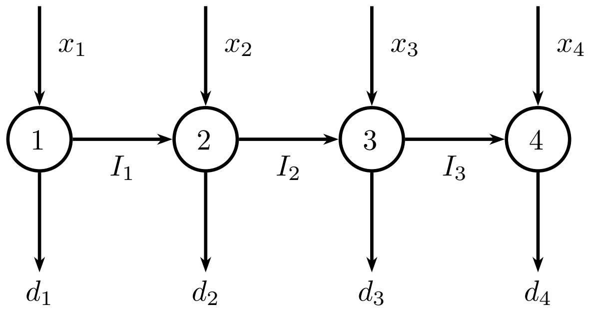

Figure 1.2.1: Network representation of the classical economic lot sizing problem with period, $n = 4$

Then, in Section 1.3, we will discuss some basic concepts of our solution approach i.e. mixed integer programming techniques used in this thesis.

1.3 Mixed Integer Programming (MIP)

Lot sizing problems are often formulated as Mixed Integer Programming (MIP) models. In this section, some important definitions and theorems of MIP used throughout the remainder of the thesis are discussed.

A MIP problem can be defined as an optimization problem with linear constraints and a linear objective function, which consists of continuous and integer variables as follows.

$$Z = \min_{(x,y)} \{cx + ky : (x, y) \in X\} \quad \text{where} \quad X = \{(x, y) \in \mathbb{R}^n \times \mathbb{Z}^p : Ax + By \leq d\}$$

where, Z represents the objective value, X corresponds to the set of the feasible solutions (feasible region) such that x is the $n - dimensional$ (column) vector of real variables and y is the $p - dimensional$ (column) vector of integer variables including 0 and 1. The parameters, $c \in \mathbb{R}^n$ and $k \in \mathbb{Z}^p$ are the (row) vectors of the objective function coefficients. $d \in \mathbb{R}^m$ is the (column) vector of the right hand side coefficients of the m linear constraints. A and B are the matrices of constraints of size $(m \times n)$ and $(m \times p)$, respectively. In a matrix form, this can be illustrated in the following example.

Example 1. Let

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix}_{3\times 2} \quad \text{and} \quad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}_{3\times 3} \quad \text{where } d = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix}_{3\times 1}$$

The matrix form can be presented as linear constraints, $m = 3$, continuous variables with $n = 2$ and integer variables, $p = 3$.

$$\begin{aligned} \overbrace{a_{11}x_{11} + a_{12}x_{12}} + \overbrace{b_{11}y_{11} + b_{12}y_{12} + b_{13}y_{13}} &\leq d_1 \\ a_{21}x_{21} + a_{22}x_{22} + b_{21}y_{21} + b_{22}y_{22} + b_{23}y_{23} &\leq d_2 \\ a_{31}x_{31} + a_{32}x_{32} + b_{31}y_{31} + b_{32}y_{32} + b_{33}y_{33} &\leq d_3 \end{aligned}$$

where the objective function is:

$$\begin{aligned} Z = \min\{c_1(x_{11} + x_{12}) + c_2(x_{21} + x_{22}) + c_3(x_{31} + x_{32}) + k_1(y_{11} + y_{12} + y_{13}) \\ + k_2(y_{21} + y_{22} + y_{23}) + k_3(y_{31} + y_{32} + y_{33})\} \end{aligned}$$

Observe that, if all integer variables are restricted to take binary values, the problem is called as a mixed binary linear program or a mixed 0-1 program, where the feasible set $X = \{(x, y) \in \mathbb{R}^n \times \{0, 1\}^p : Ax + By \leq d\}$. Meanwhile, if $n = 0$, the problem is then called as a pure integer program (IP) or a 0-1 program.

Next, the linear programming (LP) relaxation of the MIP is obtained by removing the integrality restrictions on the y variables. By solving the relaxation of a problem, a bound on the optimal value of the original problem is obtained. It plays a very important role in the optimization algorithm. The LP relaxation of the MIP problem can be defined as follows.

Definition 1. If a program $z^{LP} = \min_{x,y} \left\{c^T x + k^T y | (x, y) \in X^{LP}\right\}$ is a relaxation of program $z = \min_{x,y} \left\{c^T x + k^T y | (x, y) \in X\right\}$, where $X = \{Ax + By \leq d, x \in \mathbb{R}^n, y \in \mathbb{Z}^p\}$ and $X^{LP} = \{Ax+By \leq d, x \in \mathbb{R}^n, y \in \mathbb{R}^p\}$. Then, $z^{LP} \leq z$ as $conv(X) \subseteq X^{LP}$.

Now, we introduce some basics on polyhedral theory, which provide better insights into the problem structures addressed in this thesis.

Definition 2. Let $x_1, ..., x_k \in \mathbb{R}^n$ be any point. Then, x is a convex combination if:

$$x = \sum_{t=1}^{k} \lambda_t x_t = \lambda_1 x_1 + \lambda_2 x_2 + ... + \lambda_k x_k$$

with $\lambda_1 + ... + \lambda_k = 1$ and nonnegativity, $\lambda_t \geq 0$, $t = 1, ..., k$.

Definition 3. Let a set $X \in \mathbb{R}^n$, the convex hull of X, denoted as $conv(X)$ is the set of all convex combinations of points in X or

$$conv(X) = \left\{ x \in \mathbb{R}^n : x_t \in X \text{ and } \lambda_t \geq 0, t = 1, .., n \text{ such that } x = \sum_{t=1}^{n} \lambda_t x_t \right.$$
$$\left. \text{and } \sum_{t=1}^{n} \lambda_t = 1 \right\}$$

Definition 4. Let $X \subseteq \mathbb{R}^n$ be a set. X is a convex set if it contains a line segment (or any convex combination) between any two points $x_1, x_2 \in X$ in the set X, such that $0 \leq \lambda \leq 1$ and $\lambda x_1 + (1 - \lambda)x_2 \in X$.

Proposition 1. *The convex hull of two points is a line segment.*

Then, we provide the concept of a polyhedron, P.

Definition 5. Given $P \in \mathbb{R}^n$ is a set of points that satisfies a finite set of linear inequalities, $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ is a polyhedron.

Definition 6. A polyhedron $P \subseteq \mathbb{R}^n$ is bounded if there exists a constant $r \in \mathbb{R}_+$ such that

$$P \subseteq \{x \in \mathbb{R}^n : |x_t| < r, \forall t \in 1, ..., n\}$$

A bounded polyhedron is called a polytope.

Definition 7. A polyhedron, P is called a formulation for X if $X = P \cap \mathbb{Z}^n$; that is X is precisely the set of integer points in P.

Observe that, we can have an infinity of formulations in a set X. Now, in Figure 1.3.1, we present two formulations, P_1 and P_2 for X.

Definition 8. Suppose that P_1 is better than P_2 if $P_1 \subset P_2$. Then, for any objective function $c^T \in \mathbb{R}^n$ and $k^T \in \mathbb{R}^n$, we obtain

$$z \geq \min \left\{ c^T x + k^T y | (x, y) \in P_1 \right\} \geq \min \left\{ c^T x + k^T y : (x, y) \in P_2 \right\}$$

Example 2 (continued)**.** In Figure 1.3.1, it can be clearly seen that formulation P_1 is better than formulation P_2.

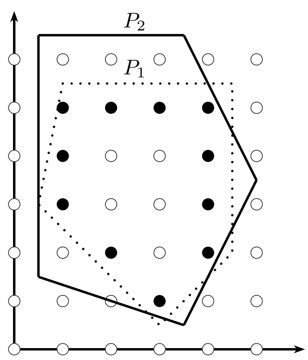

Figure 1.3.1: Two formulations for X

Definition 9. Let a set $X \in \mathbb{R}^n$. Then, the convex hull of X, denoted as $conv(X)$ is the set of all convex combinations of points in X or

$$conv(X) = \left\{ x \in \mathbb{R}^n : x_t \in X \text{ and } \lambda_t \geq 0, t = 1, .., n \text{ such that } x = \sum_{t=1}^{n} \lambda_t x_t \text{ and } \sum_{t=1}^{n} \lambda_t = 1 \right\}$$

For IP and MIP problems, $conv(X)$ is the smallest polyhedron containing X. It follows that $conv(X)$ is the best of all formulations for X, and

$$z = \min \left\{ c^T x + k^T y | (x, y) \in conv(X) \right\} \geq \min \left\{ c^T x + k^T y : (x, y) \in P \right\}$$

for all formulations P of X.

Example 3. Refer to the Figure 1.3.2, the convex of hull of set X is represented by the shaded area, where

$$X = \{(1,3), (1,4), (1,5), (2,2), (2,3), (2,4), (2,5), (3,1), (3,2), (3,3), (3,4), (3,5), (4,2), (4,3), (4,4), (4,5)\}$$

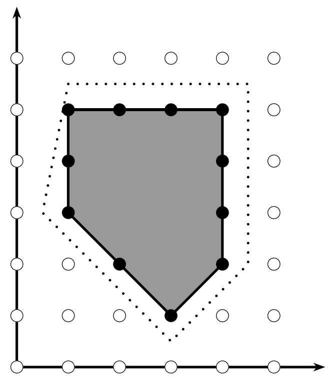

Figure 1.3.2: The convex hull of X

The following propositions state the importance of convex hull X. X is the feasible region of the general MIP that will help us to find an optimal solution.

Proposition 2 (**Pochet and Wolsey (2006)**). *Let* $X \subseteq \mathbb{R}^n$ *and suppose the MIP problem* $\min \left\{ c^T x + k^T y : (x, y) \in X \right\}$ *has an optimal solution then*

$$\min_{x,y} \left\{ c^T x + k^T y | (x, y) \in X \right\} = \min_{x,y} \left\{ c^T x + k^T y | (x, y) \in conv(X) \right\}$$

This proposition states that in order to solve this MIP problem, it suffices to solve it over the convex hull, X of its feasible region. Therefore, our main interest is to identify several families of valid linear inequalities that can describe partially or completely the convex hull of mixed integer sets of our problems. This convex hull can be best described in terms of valid inequalities and extreme points and rays.

1.3.1 Defining Polyhedra by Valid Inequality

In this section, we describe the concept of valid inequality.

Definition 10. A linear inequality $\pi x + \mu y \leq \pi_0$ is a valid inequality if and only if it is satisfied by all points in X , where $(\pi, \mu) \in \mathbb{R}^n \times \mathbb{R}^p$ and π_0 is a scalar.

Definition 11. The inequality $\pi x + \mu y \leq \pi_0$ is valid for a feasible set X if and only if it is valid for $conv(X)$.

Definition 12. The inequality $\pi x + \mu y \leq \pi_0$ is violated by the points (x^*, y^*) if $\pi x^* + \mu y^* > \pi_0$.

The concept of linear independence and affinely independence are defined as follows.

Definition 13. A finite collection of points $x^1, ..., x^k \in \mathbb{R}^n$ is linearly independent if the unique solution to $\sum_{i=1}^{k} \lambda_i x^i = 0$ is $\lambda_i = 0$, $\forall i = 1, 2, ..., k$. Otherwise, the points are linearly dependent.

Definition 14. A set $x^0, x^1, ..., x^k$ of $k+1$ points in $\mathbb{R}^n$ is affinely independent if the unique solution to $\sum_{i=1}^{k} \lambda_i x^i = 0$, $\sum_{i=1}^{k} \lambda_i = 0$ is $\lambda_i = 0$, $\forall i = 1, 2, ..., k$ or equivalently $x^1 - x^0, ..., x^k - x^0$ in $\mathbb{R}^n$ is linearly independent.

Dimension of a polyhedron P, $dim(X)$ can be expressed in the following way.

Definition 15. A polyhedron P is of the dimension k, denoted as $dim(P) = k$ if the maximum number of affinely independent points in P is $k+1$.

Definition 16. A polyhedron $P \subseteq \mathbb{R}^n$ is full-dimensional if $dim(P) = n$.

Example 4 (continued)**.** $dim(conv(X)) = 2$ because (1,3), (2,3) and (2,4) are affinely independent. Therefore, the polyhedron $conv(X)$ is full-dimensional.

Definition 17. Let $conv(X) \subseteq \mathbb{R}^n$ and $\pi x + \mu y \leq \pi_0$ be a valid inequality for X. Then, a face of $conv(X)$ is non-empty set of points $F = \{x, y \in conv(X) : \pi x + \mu y = \pi_0\} \neq \emptyset$. A face of F is said to be a proper face if $F \neq \emptyset$ and $F \neq conv(X)$. A face F is called a facet of $conv(X)$ if $dim(F) = dim(conv(X)) - 1$. Then, the valid inequality $\pi x + \mu y \leq \pi_0$ is said to describe the face.

For an inequality to be strong, the face should have as high dimension as possible. The facet-defining inequalities which dominate all other inequalities are those of maximal dimension. i.e. dimension one less than the dimension of the polyhedron. It is sufficient to exhibit $dim(P)$ affinely independent points belonging to the set

$\{x, y \in conv(X) : \pi x + \mu y = \pi_0\}$ in order to show that a valid inequality $\pi x+\mu y \leq \pi_0$ defines a facet for P. This idea is used in our study especially in Chapter 4 and 5 to establish that certain valid inequalities define facets. By showing all valid inequality is facet-defining, it is adequate to describe the convex hull of the problem. However, it is a challenge task to find all facet-defining inequalities if the problem is NP-hard. In some cases, the valid inequalities can be proven theoretically as facet-defining but they are computationally hard to find as the number of cuts generated by these inequalities grows exponentially. However, adding some good valid inequalities to a formulation necessarily increases its strength.

1.3.2 Defining Polyhedra by Extreme Points and Extreme Rays

Alternatively, we can describe a polyhedra by its extreme points and extreme rays. The polyhedron P has a finite number of extreme points and extreme rays.

Definition 18. $x \in P$ is an extreme point of polyhedron P if it cannot be written as a convex combination of two points in P or in other words there do not exist two points $x^1, x^2 \in P, x^1 \neq x^2$ with $x = \frac{1}{2}x^1 + \frac{1}{2}x^2$.

An extreme point can also be called as a vertex of a polyhedron and has 0-dimensional face of a polyhedron that represents a point.

Definition 19. Let $r \in \mathbb{R}^n$. Then, $r \neq 0$ is a ray of polyhedron, $P \neq \emptyset$ if for each $x \in P$, the set $\{x + \lambda r | \lambda \geq 0\}$ is contained in P. In other words, a ray r of P is an extreme ray if there do not exist two linearly independent rays, r^1, r^2 of P , $r^1 \neq \lambda r^2$ for some $\lambda > 0$ with $r = \frac{1}{2}r^1 + \frac{1}{2}r^2$.

Theorem 1 (Minkowski's Theorem)**.** *Every polyhedron $P \neq \emptyset$ can be represented as a convex combination of extreme points $\{x^t\}_{t=1}^T$ and a non-negative combination of extreme rays $\{r^s\}_{s=1}^S$:*

$$P = \left\{ x : x = \sum_{t=1}^{T} \lambda_t x^t + \sum_{s=1}^{S} \mu_s r^s, \sum_{t=1}^{T} \lambda_t = 1, \lambda \in \mathbb{R}_+^T, \mu \in \mathbb{R}_+^S \right\}$$

A characteristic cone of a polyhedron is also called as an extreme ray of P, defined as follows.

Definition 20. Let $P = \{x \in \mathbb{R}^n : Ax \leq b\}$. Then

$$char.cone(P) = \{r \in \mathbb{R}^n : Ar \leq 0\}$$

In the next section, we will discuss several optimization approaches used to solve MIP problems.

1.3.3 Optimization Algorithms

There are three most commonly used optimization algorithms for solving MIP problems: (i) Branch-and-Bound (B&B) algorithm; ii) Branch-and-Cut (B&C) algorithm i.e. cutting plane and separation algorithm and iii) Extended Formulations.

Branch-and-Bound (B&B) algorithm

Branch-and-Bound (B&B) algorithm is the traditional solution approach used in mixed integer programming problems. This algorithm is basically a tree, where each node of the tree is an LP problem. The algorithm procedure, i.e. minimization problem, can be described as follows.

The value of the feasible solution found so far is called the incumbent, which represents the upper bound of the value of the optimal solution. We set the incumbent to ∞ if there is no feasible solution found. B&B solves LP relaxation at the root node and in case a fractional solution k for an integer variable y is obtained, a constraint $y \leq \lfloor k \rfloor$ or $y \leq \lceil k \rceil$ is added to the LP relaxation to obtain two child nodes (two subproblems).

At each tree node, the LP relaxation is solved. First, if the solution found is integral, the incumbent is updated and tree node is pruned. Second, in the case of solution is infeasible, the tree node will be also pruned as the following subproblems are infeasible. When the value of the incumbent is less than the value of the LP solution, the tree node can be pruned if the optimal solution of the subproblem is worse than a known feasible solution. Otherwise, we choose a variable with the fractional value in the LP solution to be branched into two subproblems. Lastly, B&B algorithm will stop if the set of subproblems is empty in which the optimal solution of the problem is found. If not, this algorithm will continue search tree node recursively. The B&B scheme in Figure 1.3.3 is summarized next.

Branch-and-Cut (B&C) algorithm

As regards Branch-and-Cut (B&C) algorithm, the use of cutting planes is implemented within the Branch-and-Bound (B&B) algorithm so as to strengthen the bounds of the LP solution to the actual feasible integer solutions.

The fundamental idea behind the cutting plane method and separation algorithm is to generate valid inequalities and added them to the original formulation when they are needed (in most cases, there are exponentially many of these inequalities which are inactive and useless) and only when they are not satisfied at the optimal solution of LP relaxation. In other words, the constraints (valid inequalities) are added to a linear program until the optimal basic feasible solution takes on integer values.

The separation algorithm for a valid inequality is given by the following definition.

Definition 21. Given a point $(x^*, y^*) \in \mathbb{R}^p \times \mathbb{R}^q$ with $(x^*, y^*) \notin conv(X)$ of a mixed integer set, then the separation problem, denoted by $SEP(X, x^*, y^*)$ is the problem of finding a valid inequality $\pi x + \mu y \leq \pi_0$ cutting off points (x^*, y^*) such that $\pi x^* + \mu y^* > \pi_0$ or deciding that there is no such inequality.

Next, we remark some important results that relate to the optimization and separation problems.

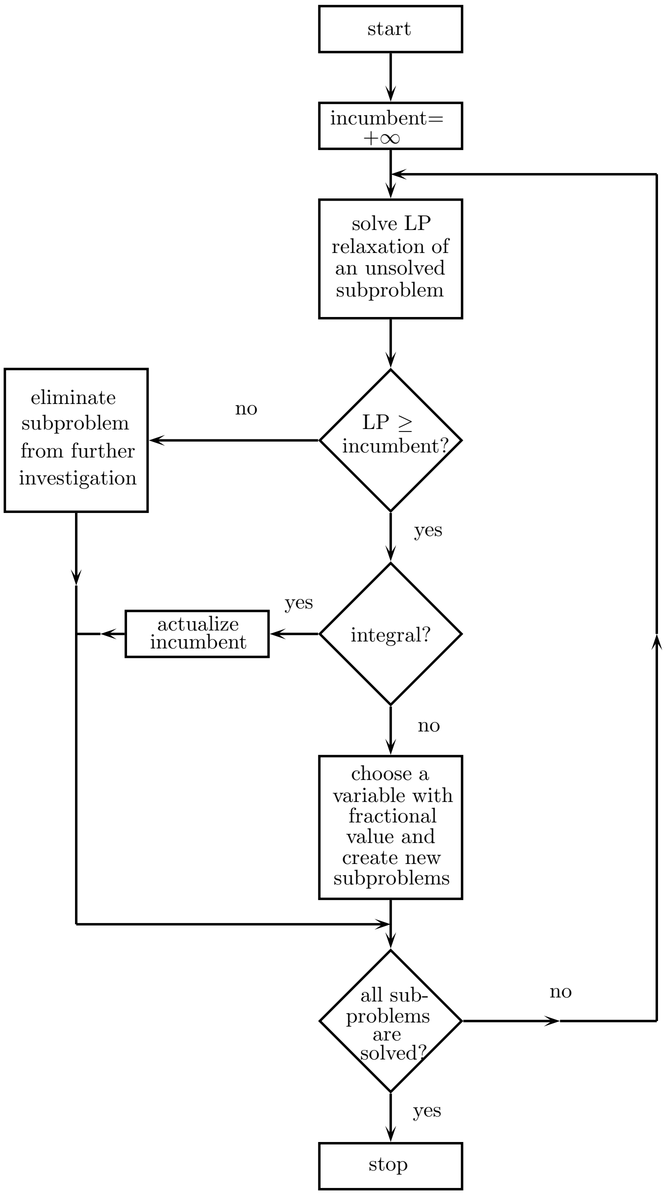

Figure 1.3.3: Branch-and-bound algorithm (Doostmohammadi, 2014)

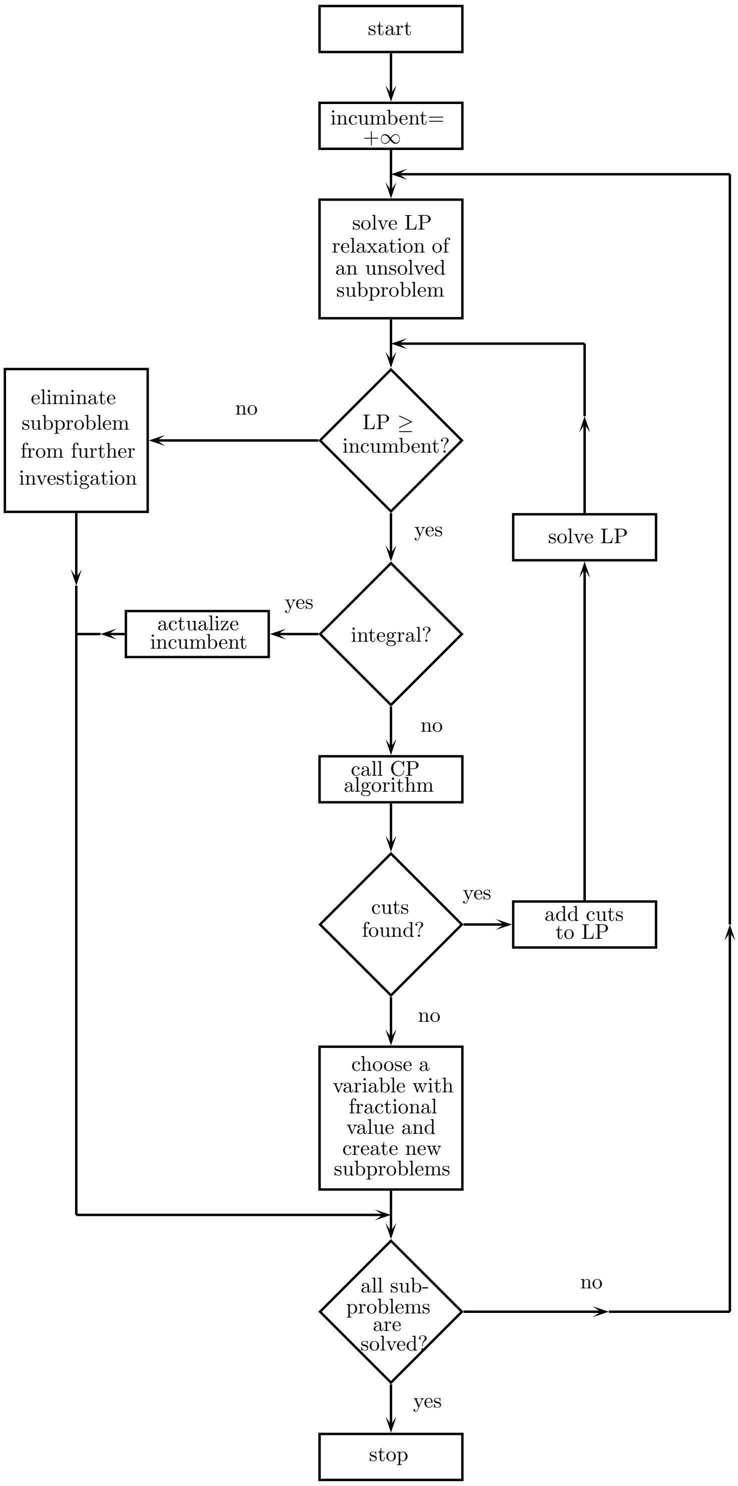

Figure 1.3.4: Branch-and-cut algorithm (Doostmohammadi, 2014)

Proposition 3 (Pochet and Wolsey (2006)). *Optimization problem and separation problem are polynomially equivalent, where*

- Solving the optimization problem, $\min_{x,y} \left\{c^T x + k^T y | (x, y) \in conv(X)\right\}$ is solvable in polynomial time,
- Separating $(x^*, y^*) \in \mathbb{R}^p \times \mathbb{R}^q$ over $conv(X)$ is also solvable in polynomial time.

From this, if the optimization and separation problem are polynomially solvable, we can possibly find the complete description of $conv(X)$. However, for the complex case i.e. NP hard problem, we can hope that at least the partial description of $conv(X)$ can be obtained.

Next, we discuss the steps of implementing the cutting plane method in the following ways.

(i) Find the LP relaxation of the MIP problem is solved. Then, if LP relaxation solution obtained gives the convex hull of feasible region, then STOP, otherwise go to Step (ii),

(ii) Solve the separation problem by finding the violated valid inequalities (or a family of valid inequalities) that cut off a fractional point of the LP relaxation solution. Then, add them directly to the original formulation. If no violated inequality is found STOP, otherwise go back to Step (i).

The flowchart of B&C algorithm is illustrated in Figure 1.3.4.

Extended Formulations

Alternatively, extended reformulation can also be used to strengthen a formulation by introducing new variables. For $X = \left\{x \in \mathbb{Z}^n_+ : Ax \leq b\right\}$, suppose that

$$X = \left\{x \in \mathbb{Z}^n_+ : Bx + Gz \leq b \text{ for some } z \in \mathbb{R}^q\right\}$$

Definition 22. Let $Q = \{(x, z) \in \mathbb{R}^n_+ \times \mathbb{R}^q : Bx + Gz \leq b\}$. Then, the projection of Q into the $x-$variable space denoted, by $proj_x(Q)$ is the polyhedron given by

$$\tilde{P} = proj_x(Q) = \{x \in \mathbb{R}^n : \exists z \in \mathbb{R}^q \text{ with } (x, z) \in Q\}$$

which is a formulation for X as $X = \tilde{P} \cap \mathbb{Z}^n$. Figure 1.3.5 illustrates such a projection.

The extended formulation can be defined as follows.

Definition 23. The polyhedron $Q = \{(x, z) \in \mathbb{R}^n_+ \times \mathbb{R}^q : Bx + Gz \leq b\}$ is an extended formulation for $X = \left\{x \in \mathbb{Z}^n_+ : Ax \leq b\right\}$ if $proj_x(Q)$ is a formulation for X.

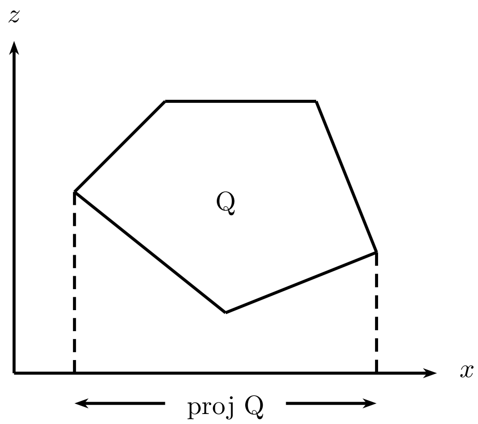

Figure 1.3.5: Extended formulation and projection

Definition 24. The extended reformulation, $Q \subset \mathbb{R}^{n+q}$ is a tight formulation for X if

$$proj_x(Q) = conv(X)$$

and compact if its size is polynomial in the size of X.

Interestingly, the number of inequalities needed to describe $conv(X)$ with an extended formulation may be small (perhaps polynomial) compared to the number of facet-defining inequalities (possibly exponential) generated to describe $conv(X)$ in the original space (Pochet and Wolsey, 2006).

1.4 Problem Formulations for ELSR

As in Teunter et al. (2006) and Retel Helmrich et al. (2013), we consider two variants of models of economic lot sizing problem with remanufacturing which are NP-hard in general (see literature review chapter for the details). In the first model, remanufacturing and manufacturing processes each operate on dedicated production lines, each with its own setup cost. This problem is called ELSRs (Economic Lot Sizing Problem with Remanufacturing and Separate Setups). In the second model, remanufacturing and manufacturing processes perform on the same production line with a single setup cost, known as ELSRj (Economic Lot Sizing Problem with Remanufacturing and Joint Setups).

These models seek to find an optimal production plan that satisfies customer demands such that the total costs (production, inventory and setup costs) are minimized. Now, we shall refer to the original formulations of ELSRs and ELSRj problems as stated in Teunter et al. (2006) and Retel Helmrich et al. (2013). The problems are formulated as mixed integer programs. First, we define the decision variables and parameters used in the model formulations.

Decision variables

x_t^r is the amount of remanufactured products produced in period t,

x_t^m is the amount of new products produced in period t,

y_t^r is 1 if remanufacturing process takes place in period t, 0 otherwise,

y_t^m is 1 if manufacturing process takes place in period t, 0 otherwise,

y_t is 1 if remanufacturing and manufacturing process both take place in period t, 0 otherwise,

I_t^r is the inventory of product returns at the end of period t,

I_t^s is the inventory of serviceable products at the end of period t.

Parameters

p_t^r is unit production cost of remanufacturing in period t,

p_t^m is unit production cost of manufacturing in period t,

h_t^r is unit holding cost for inventory of product returns in period t,

h_t^s is unit holding cost for inventory of serviceable products in period t,

K_t^r is unit separate setup cost for remanufacturing in period t,

K_t^m is unit separate setup cost for manufacturing in period t,

K_t is unit joint setup cost for remanufacturing and manufacturing in period t,

d_t is the amount of demands in period t, where $d_{t,t'} = \sum_{i=t}^{t'} d_i$,

r_t is the incoming amount of returns to be remanufactured in period t, where $r_{t,t'} = \sum_{i=t}^{t'} r_i$.

1.4.1 Separate Setups

We present the original formulation of ELSRs:

$$Z^{ss} = \min \sum_{t=1}^{n} (K_t^r y_t^r + K_t^m y_t^m + p_t^r x_t^r + p_t^m x_t^m + h_t^r I_t^r + h_t^s I_t^s) \tag{1.7}$$

$$\text{s.t.} \quad I_t^r = I_{t-1}^r + r_t - x_t^r \qquad \forall t \in N \tag{1.8}$$

$$I_t^s = I_{t-1}^s + x_t^r + x_t^m - d_t \qquad \forall t \in N \tag{1.9}$$

$$x_t^r \leq d_{t,n} y_t^r \qquad \forall t \in N \tag{1.10}$$

$$x_t^m \leq d_{t,n} y_t^m \qquad \forall t \in N \tag{1.11}$$

$$y_t^r, y_t^m \in \{0,1\}^n \qquad \forall t \in N \tag{1.12}$$

$$x_t^r, x_t^m, I_t^r, I_t^s \geq 0 \qquad \forall t \in N \tag{1.13}$$

$$I_0^r = I_0^s = 0 \tag{1.14}$$

The objective (1.7) is to minimize the total of setup costs, production costs for remanufacturing and manufacturing processes; and holding costs for product returns and serviceable products. Constraint (1.8) represents flow conversation (inventory balance) for product returns. Constraint (1.9) indicates flow conversation (inventory balance) for serviceable products. Constraint (1.10) is setup forcing constraint for remanufacturing. Constraint (1.11) is setup forcing constraint for manufacturing. Next, (1.12) provide the integrality of remanufacturing and manufacturing. Then, (1.13) denotes nonnegativity requirements of production of remanufactured and new products and inventory variables of product returns and serviceable products. Lastly, without loss of generality, we assume no initial inventory for product returns and inventory of serviceable products on hand as stated in constraint (1.14). The illustration of a network representation for ELSRs is given in Figure 1.4.1.

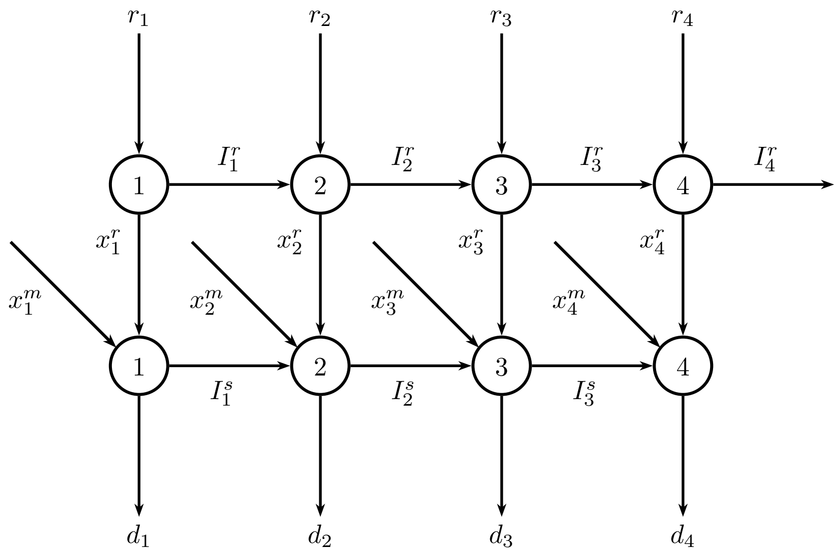

Figure 1.4.1: Network representation of ELSRs problem with period, $n = 4$ (Retel Helmrich et al. (2013))

As the remanufacturing operation depends on the amount of returns at the beginning of production period t, we obtain the following variable upper bound on x^r.

$$x_t^r \leq \min\left(r_{1,t}, d_{t,n}\right) y_t^r \qquad \forall t \in N \tag{1.15}$$

This new valid upper bounds (1.15) on x^r indicates that remanufactured products can be produced up to the total amount of returns from period 1 to t but it is restricted to the total amount of demands from period t to n.

As for now, we obtain the feasible region of the basic formulation for ELSRs:

$$X^{ss} = \{(x^r, x^m, y^r, y^m, I^r, I^s)|(1.8), (1.9), (1.11) - (1.15)\}$$

with the objective function $Z^{ss} = \min\{(1.7)|(x^r, x^m, y^r, y^m, I^r, I^s) \in X^{ss}\}$.

1.4.2 Joint Setups

As for joint setups case, we use the same formulation as in separate setups case except that a single setup variable, y_t and a single setup cost parameter, K_t are considered.

$$Z^{js} = \min \sum_{t=1}^{n} (K_t y_t + p_t^r x_t^r + p_t^m x_t^m + h_t^r I_t^r + h_t^s I_t^s) \tag{1.16}$$

$$\text{s.t. (1.8),(1.9), (1.13), (1.14)} \qquad x_t^r + x_t^m \leq d_{t,n} y_t, \qquad \forall t \in N \tag{1.17}$$

$$y_t \in \{0,1\}^n, \qquad \forall t \in N \tag{1.18}$$

Then, we have the following feasible region of the basic formulation for ELSRj:

$$X^{js} = \{(x^r, x^m, y, I^r, I^s)|(1.8), (1.9), (1.13), (1.14), (1.17), (1.18)\}$$

and the objective function of $Z^{js} = \min\{(1.16)|(x^r, x^m, y, I^r, I^s) \in X^{js}\}$. The following Figure 1.4.2 represents a network representation for ELSRs as a special case of ELSRj. The details explanation of this figure can be found in Retel Helmrich et al. (2013).

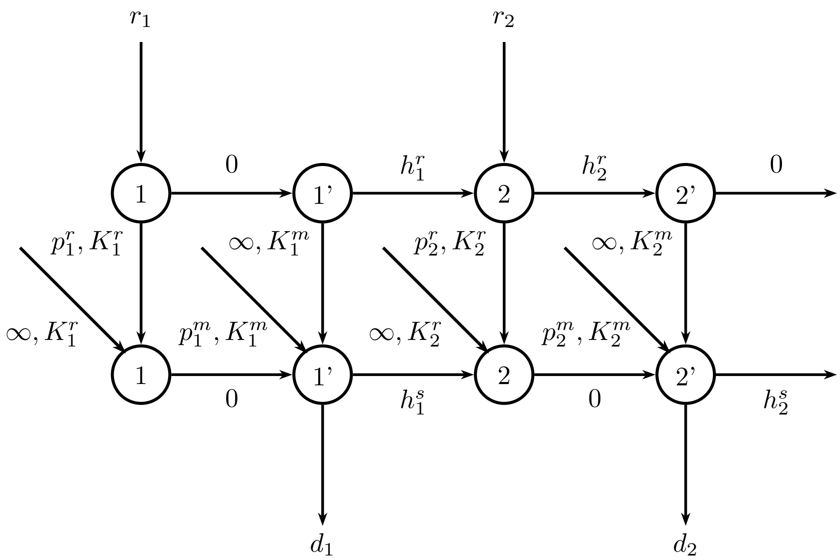

Figure 1.4.2: ELSRs as a special case of ELSRj with period, $n = 4$ (Retel Helmrich et al. (2013))

1.5 Outline of the Thesis

Chapter 2 presents the relevant literature on several mathematical programming techniques used to solve the classical single-item lot-sizing problem and its extension (i.e., the remanufacturing option). In Chapter 3, we propose several traditional solution techniques to obtain better lower bounds for ELSR problems. Theoretical and computational comparisons between these different lower bounding techniques are presented. Finally, we further investigate the polyhedral structure of both problems in Chapters 4 and 5, following the findings from the previous chapter. Several families of valid inequalities for the problems are derived and their facet-defining conditions are identified. Finally, these cuts are computationally tested in order to observe their effectiveness.

Chapter 2

Literature Review

This chapter discusses the literature survey on several solution techniques, specifically, mathematical programming approaches that are commonly used to solve a wide variety of lot-sizing problems. Since ELSR problems have been proven to be NP-hard, this generally causes them to be computationally inefficient so there is a need to develop and improve solution procedures. In this thesis, we review three main solution techniques, namely polynomial-time algorithms (e.g., Wagner-Whitin algorithm and its extensions) for the special cases in Section 2.1, mixed integer programming (MIP) methods in Section 2.2 and heuristic methods in Section 2.3. The literature review of this classical lot-sizing problem develops an essential understanding of ELSR problems' substructures.

2.1 Polynomial Algorithms for Special Cases

Wagner and Whitin (1958) were the first to present an $O(n^2)$ dynamic programming algorithm for the single-item uncapacitated lot sizing problem with constant production costs and nonnegative inventory holding costs. This dynamic lot-size model is a generalization of the economic order quantity (EOQ) model, which allows deterministic demand rate, costs and lot sizes for a single item to vary from period to period throughout the planning horizon. The key element of this model is stated in the following property.

Definition 25. The Wagner-Whitin property, also known as the zero-inventory property, states that $x_t I_{t-1} = 0$, which can be either $x_t = 0$ or $I_{t-1} = 0$ or both. (We produce only if the entering inventory is zero.)

Definition 26. The problem is Wagner-Whitin if the production costs, $p_t^{'}$ in period t and holding costs, $h_t^{'}$ at the end of period t satisfy $h_t^{'} + p_t^{'} - p_{t+1}^{'} \geq 0$ for all $t \in [0, n]$, where $p_0^{'} = p_{n+1}^{'} = 0$. Note that it is optimal to produce as late as possible because it is costlier to produce in period t and retain until period $t+1$ than to produce in period $t+1$.

Further, an earlier study on basic extensions of the uncapacitated single-item problem can be found in Zangwill (1969). The author generalized the Wagner-Whitin model, first with backlogging and second with a multi-level problem, both without capacities and with concave cost functions. Both models are represented by single-source networks to develop efficient dynamic programming algorithms. These single-item problems were then extended by Florian and Klein (1971) in the case of constant capacities with and without backlogging. The authors developed an $O(n^4)$ dynamic programming based on a shortest path algorithm. Later, Bitran and Yanasse (1982) studied the computational complexity of capacitated lot sizing under various assumptions of costs and capacity structures. The complexity of the problem is given by the notation, $\alpha/\beta/\gamma/\delta$, where α, β, γ and δ represent setup cost, holding cost, production cost and capacity, respectively. The values for each notation are G, C, ND, NI and Z, which indicate general structure, constant, non-decreasing, non-increasing and zero, respectively. For instance, the notation ND/C/NI/G is a family of problems, where the setup cost is non-decreasing, the holding cost is constant, the production cost is non-increasing and capacity is not restricted to a prespecified structure. Their findings are shown in Table 2.1.1.

Table 2.1.1: Results of problem complexity (Bitran and Yanasse (1982))

Problem	Complexity
NI/G/NI/ND	$O(n^4)$
NI/G/NI/C	$O(n^3)$
C/Z/C/G	$O(n \log n)$
ND/Z/ND/NI	$O(n)$

In the 1990s, three independent studies discovered new algorithms that reduce the computational complexity of the Wagner-Whitin algorithm. They improved $O(n^2)$ time to $O(n \log n)$ and $O(n^2)$ time to $O(n)$ for some special cases of Wagner-Whitin costs. Firstly, Federgruen and Tzur (1991) proposed a forward-recursion dynamic programming algorithm for the general single-item lot-sizing problem with fixed and linear costs. The authors also proposed an $O(n)$ simple algorithm for two important special cases of: (i) no speculative motives for carrying stock and (ii) non-decreasing setup costs. Further, Wagelmans et al. (1992) used a backward-recursion dynamic programming algorithm to solve a Wagner-Whitin case in linear time, $O(n)$ where the cost functions are linear and not restricted in sign (may have negative costs). Lastly, Aggarwal and Park (1993) investigated the case with several cost structures, with and without backlogging for the single-item uncapacitated lot-sizing problem. The authors provided efficient algorithms using dynamic programming and array searching that improve on those in some previous studies. In the case of backlogging with arbitrary concave-cost functions, this study reduced the computational complexity, $O(n^3)$ of Zangwill's algorithm of 1969 to $O(n^2)$. However, the authors were unable to improve their running time to the

$O(n^2)$ algorithm for the case with non-decreasing concave-cost functions of holding and backlogging costs, and constant production cost. Next, Van Hoesel and Wagelmans (1996), who studied the constant capacities economic lot-sizing problem with concave production costs and linear holding costs, improved Florian and Klein's algorithm that runs in $O(n^3)$ time. In the case of the general problem and both constant and arbitrary capacities, the dynamic programming algorithms proposed by Kirca (1990) can be performed at least three times faster than Florian and Klein's algorithm.

Several studies have improved the complexity of problems addressed by Bitran and Yanasse (1982). Referring to Table 2.1.1, for the case of both non-increasing setup and production costs (NI/G/NI/C), the problem is solved in $O(n^3)$ time. The same complexity was obtained byVan Hoesel and Wagelmans (1996), in the case where production costs are concave and holding costs are linear. When setup costs have an arbitrary pattern, holding costs are constant, the production costs are non-increasing, and the capacities are non-decreasing (NI/G/NI/ND), the problem complexity obtained by Chung and Lin (1988) outperforms Bitran and Yannasse's algorithm from $O(n^4)$ to $O(n^2)$. Van den Heuvel and Wagelmans (2006) also addressed the same problem as Chung and Lin (1988) to derive a new $O(n^2)$ algorithm. This new algorithm considered fewer candidate solutions in each iteration than Chung and Lin (1988) and becomes more effective when the capacities are relatively large. Numerical tests show the effectiveness of the proposed algorithm compared to Chung et al.'s algorithm.

Fleischmann (1990) proposed a dynamic programming algorithm for a special case in order to solve the relaxed problem of the discrete lot-sizing and scheduling problem. Van Hoesel et al. (1994b) also investigated the same problem and presented an efficient dynamic programming algorithm that uses properties of its optimal solutions. Vanderbeck (1998) presented an $O(n^6)$ dynamic programming algorithm for the single-item lot-sizing model with stationary capacities and start-up times for both discrete and continuous setup models. This algorithm reduces the problem complexity to $O(n^4)$ when the production and holding costs satisfy the Wagner-Whitin costs.

Some extensions of lot-sizing problems include start-up costs and time windows. Shaw and Wagelmans (1998) solved the capacitated economic lot-sizing problem with piecewise linear production costs, general holding costs, backlogging and start-up costs that run in pseudo-polynomial time. The authors proposed the $O(n^2\bar{p}\bar{d})$ algorithm, where n is the number of periods and $\bar{d}$ and $\bar{p}$ are the average demand and average number of pieces of the production cost functions, respectively. Lee et al. (2001) studied dynamic lot sizing with demand time windows, with and without backlogging. The complexity of the problem is $O(n^2)$ if the no-backlogging case is considered. Otherwise, $O(n^3)$ is obtained. Hwang (2007) also considered the same problem with backlogging. The same complexity $O(n^3)$ algorithm as Lee et al. is obtained in the case of non-speculative cost structure. For a somewhat general cost structure, the algorithm is improved to $O(\max n^2, dn)$ time, where d is the demand scheduled for n periods.

In the problem with bounded inventory, Gutiérrez et al. (2003) presented an algorithm that runs in $O(n)$ expected time when the demands vary between the interval of zero and the storage capacity. This algorithm runs almost 30 times faster than the algorithm proposed by Love (1973). Later, Gutiérrez et al. (2008) further addressed the problem with time-varying storage capacities and costs whose running time is $O(n \log n)$. The authors showed that thee an optimal plan exists that satisfies the zero-inventory ordering (ZIO) property in the case of constant production/ordering unit costs (i.e., the Wagner\textendash Whitin case). Liu (2008) who studied the economic lot-sizing problem with both upper and lower inventory bounds, proposed an $O(n^2)$ algorithm for the general problem and an $O(n)$ algorithm for a special case with non-speculative motives. Önal et al. (2012) argued that their algorithms do not provide an optimal solution in general. They proposed an improved algorithm that also runs in $O(n^2)$ time. For a more realistic model, Chu and Chu (2007) were the first to consider a single-item dynamic lot-sizing problem with the integration of outsourcing, backlogging decisions and inventory capacity in real-life crude-oil procurement problems that arise in refineries. When the inventory holding and backlogging cost functions are concave and the production cost functions are linear with fixed charges or concave piecewise linear, the problem is solved in$O(n^2)$ time whereas $O(n \log n)$ time is obtained if unbounded inventory is considered. Furthermore, the outsourcing model is solved in $O(n^2 \log n)$ time when the inventory holding and the outsourcing cost functions are linear. In addition, Chu et al. (2013) adapted their study to the case of real-life production planning problems of luxury goods. The problem is solved in $O(n^4 \log n)$ time using dynamic programming algorithm.

The problem of minimum order requirements was explored by the study of Okhrin and Richter (2009). The authors presented a dynamic programming algorithm for the single-item lot-sizing problem with both capacity constraints and minimum order quantity requirements. They showed that this general problem is NP-hard. The problems with constant capacity and minimum order quantities and with general minimum order quantities and infinite capacities are all polynomially solvable. They also developed a fully polynomial time-approximation scheme in the presence of linear cost functions and possible fixed procurement costs. Okhrin and Richter (2011) then proposed an $O(n^3)$ algorithm for the single-item capacitated lot-sizing problem with minimum order quantities by investigating the properties of the optimal solution structure to obtain sub-problems of an explicit solution.

Lastly, the study of a simple lot-sizing problem has also been extended to two-level lot sizing and carbon emission constraints. Melo and Wolsey (2010) developed a forward dynamic programming algorithm for the uncapacitated two-level lot-sizing problem with running time of $O(n^2 \log n)$. The authors also provided a compact and tight extended formulation for the problem. Next, Absi et al. (2013) proposed new lot-sizing problems with four different carbon-emissions constraints. They developed a dynamic programming algorithm to obtain an optimal solution for the multi-sourcing uncapacitated lot-sizing problem with a periodic carbon-emissions constraint. The remaining three constraints have been proven to be NP-hard.

We now review the existing literature related to lot-sizing problem with remanufacturing. There are two types of demand streams addressed in the literature: (i) demand that can be satisfied by either new or remanufactured products (ii) different demand streams for new and remanufactured products (i.e., demand for remanufactured products can be also satisfied by new products but not vice versa). In this thesis, the first type of demand streams is of interest.

Some special cases of ELSR problems, such as whether there are sufficient returns to satisfy demand, no production or no speculative motives on costs (also called Wagner-Whitin cost), can be solved in polynomial time. To the best of our knowledge, Richter and Sombrutzki (2000) was the first to study the reverse version of Wagner et al.'s classical algorithm in the case of a large quantity of low inventory cost of product returns. In this study, manufacturing and remanufacturing are performed in different production lines, which each has its own setup costs, and demand can be satisfied by either remanufactured or new products. The authors assumed that the amount of returns at the beginning of the production period is sufficient to meet total demand over the entire horizon; therefore, the manufacturing process is not necessary. Accordingly, some modifications of the classical zero Wagner-Whitin inventory property hold as follows:

Lemma 1. *Any optimal solution satisfies the following property:* $x_t^r x_t^m = 0$ *and* $I_{t-1}^s(x_t^r + x_t^m) = 0$ *for all* $t \in N$.

From this, it is clearly seen that the optimal solution can be obtained when either manufacturing or remanufacturing activities take place during a particular period. In other words, these activities can never occur during the same period. The selected activity can only be performed if the ending inventory of serviceable products in the previous period is empty. This study was later extended by Richter and Weber (2001) with additional variable manufacturing and remanufacturing costs. Using the same property as mentioned previously, the authors derived conditions that exclude one of these activities as they can never occur during the same period. For time-constant costs and demands, they proved that the optimal policy begins with remanufacturing before switching to manufacturing and found that there is only one switching point from remanufacturing to manufacturing.

Following this, Golany et al. (2001) investigated the production-planning problem with manufacturing, remanufacturing and disposal options without restrictive assumptions on the amount of returns. The problem has a network flow formulation and is solved using dynamic programming. This study proves that the problem is NP-hard for the general concave cost structure. For the case of linear costs and zero setup costs, an exact algorithm of $O(n^3)$ is obtained when transforming the problem into a transportation problem in a special way. Yang et al. (2005) used settings similar to those in Golany et al. (2001) to develop an effective polynomial-time heuristic algorithm using the extreme-point optimal solutions of the feasible region and showed that the concave-cost problem is also NP-hard, even for the case of stationary concave cost functions.

Heuvel (2004) investigated the complexity of the economic lot-sizing problem with a remanufacturing option and separate setup costs and proved that the problem is NP-hard in general, even under stationary cost parameters. Then, Teunter et al. (2006) studied the dynamic lot-sizing problem with remanufacturing with joint and separate setup cases. The first model of joint setup costs for manufacturing and remanufacturing was solved using an exact polynomial-time dynamic programming algorithm based on zero-inventory and remanufacture-first properties. With this model, the authors presented the same property addressed by Richter and Sombrutzki (2000). Further, they provide a second lemma that gives priority to a remanufacturing activity.

Lemma 2. *Any optimal solution satisfies the following property: in every period where products are manufactured, the stock of returns at th end of that period is zero, i.e.* $I_t^r x_t^m = 0$ *for all* $t \in N$.

This lemma tells us that the production of new products in a particular period can take place if and only if the inventory of returns at the end of that period is zero. As regards the second model, the heuristic approach is adopted, which is discussed in the next section.

Pan et al. (2009) extended the basic problem with the separate setups case addressed by Heuvel (2004) and Teunter et al. (2006) to a capacitated problem. The authors addressed the capacitated dynamic lot-sizing problem with remanufacturing and disposal options. Several useful properties of the problem are characterized when the cost functions are concave. The findings show that the dynamic lot-sizing problem with only disposal or remanufacturing can be converted into the traditional capacitated lot-sizing problem and solved using polynomial algorithms if constant capacities are considered. In addition, the author proposed a pseudo-polynomial algorithm for the problem with both capacitated disposal and remanufacturing.

Lastly, Wang et al. (2011) presented the single-item dynamic lot-sizing problem with remanufacturing and outsourcing, where demand and returns are deterministic over a finite planning horizon. Outsourcing is used to meet unfulfilled demand and thus no backlogging is allowed. An $O(n^2)$ dynamic programming is developed to solve this problem if a large amount of returns is considered. We look at alternative approaches that use mixed integer programming in Section 2.2.

2.2 Mixed Integer Programming

One way to obtain better lower bounds within the MIP approach is by finding good formulations that can give us a better approximation of the convex hull of the problem. There are two types of exact methods: (i) adding valid inequalities into an original space and (ii) extending a formulation into different variable spaces.

2.2.1 Valid Inequalities

Adding valid inequalities or constraints a priori to the original formulation provides a tightened formulation that improves the lower bounds provided by linear relaxations solved at a node root, reduces the branch-and-bound (B&B) nodes required to solve the MIP problem, and increases the efficiency of computation times.

There is a vast literature on the polyhedral properties of the uncapacitated lot-sizing problem and many of its variants. The first polyhedral study of the uncapacitated lot-sizing problem was introduced by Barany et al. (1984a). They proposed a family of valid inequalities, namely (ℓ, S) inequalities as follows:

Proposition 4 (Barany et al. (1984a))**.** *For any* $\ell = 1, ..., n$, $L = \{1, ..., \ell\}$, *and* $S \subseteq L$, *the family of valid inequalities*

$$\sum_{i \in S} x_i \leq \sum_{i \in S} d_{i,\ell} y_i + I_\ell \tag{2.1}$$

is called the (ℓ, S) *inequalities. The proof for this type of inequality can be referred to in the cited paper.*

Note that there exists an exponential number of (ℓ, S) inequalities added into the formulation and hence a cutting-plane approach should be used to avoid adding all these inequalities a priori to the formulation. The feasible region of the original MIP for this single-item uncapacitated lot-sizing problem is $X = \{(x, y, I|(1.2) - (1.6)\}$, and the LP relaxation with added (ℓ, S) inequalities is given by $X_{LS} = \{(x, y, I)|(1.2) - (1.6), (2.1) : 0 \leq y \leq 1\}$. Accordingly, the convex hull of this problem is denoted as $X_{LS} = conv(X)$. Solving LP relaxation with the violated (ℓ, S) inequalities added into the original formulation suffices to obtain the complete linear description of the convex hull of its feasible region (Barany et al., 1984b). A simple polynomial separation algorithm presented in Algorithm 2.1 is used to enumerate over all possible values of ℓ, whose running time is $O(n^2)$.

Algorithm 2.1 (ℓ, S) separation algorithm for simple lot sizing problem

Input: LP relaxation solution (x^*, y^*, I^*)
Output: Violated (ℓ, S) inequalities
for all $\ell = 1$ to n **do**
  Initialize S is an empty set
  for all $i = 1$ to ℓ **do**
    if $x_i^* > d_{i,\ell} y_i^*$ **then**
      $S \leftarrow S \cup \{i\}$
    end if
  end for
  if $\sum_{i \in S} x_i^* > \sum_{i \in S} d_{i,\ell} y_i^* + I_\ell$ **then**
    Add violated (ℓ, S) inequality
  end if
end for

Given a LP relaxation solution (x^*, y^*, I^*), the separation algorithm can be solved by either:

- Find an (ℓ, S) inequality violated by (x^*, y^*, I^*) or,
- Prove that all (ℓ, S) inequality are satisfied by (x^*, y^*, I^*).

We rewrite the (ℓ, S) inequality as $\sum_{i \in S}(x_i - d_{i,\ell} y_i) \leq I_\ell$. Then, we can find the most violated (ℓ, S) inequality for the fixed interval $\ell \in \{1, ..., n\}$. It suffices to set

$$S^* = \{i \in \{1, ..., \ell\} : x_i^* - d_{i,\ell} y_i^* > 0\}$$

and test whether $\sum_{i \in S^*}(x_i^* - d_{i,\ell} y_i^*) > I_\ell^*$. The (ℓ, S^*) inequality is the most violated inequality for the given value of ℓ if this test holds. Otherwise, there is no violated (ℓ, S) inequality for a given value of ℓ. Interested readers can refer to Pochet and Wolsey (2006).

Example 5. Given that the optimal solution of the linear relaxation of (1.1) - (1.6) in Figure 2.2.1. Note that the missing arcs correspond to arcs with zero flow. We will find an (ℓ, S) inequality cutting off the point.

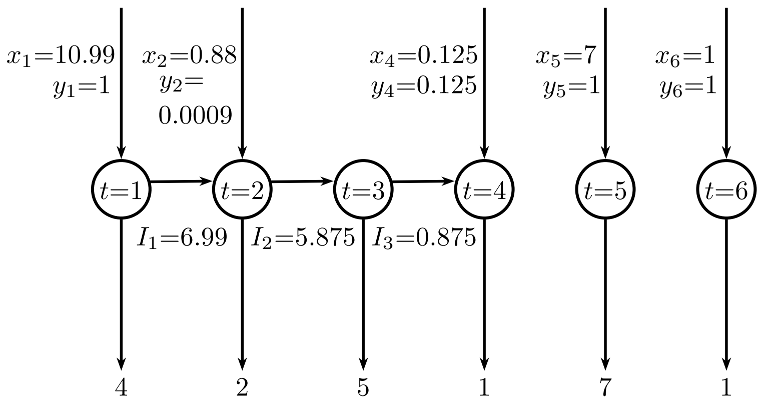

Figure 2.2.1: The solution of linear relaxation of (1.1) - (1.6)

Let $\ell = 3$ and $S \subseteq \{1, ..., 3\}$, therefore we have following valid inequalities.

$$\begin{aligned}
S &= \{1\}, & x_1 &\leq 11y_1 &+ I_3 \\
S &= \{2\}, & x_2 &\leq 7y_2 &+ I_3 \\
S &= \{3\}, & x_3 &\leq 5y_3 &+ I_3 \\
S &= \{1,2\}, & x_1 + x_2 &\leq 11y_1 + 7y_2 &+ I_3 \\
S &= \{1,3\}, & x_1 + x_3 &\leq 11y_1 + 5y_3 &+ I_3 \\
S &= \{2,3\}, & x_2 + x_3 &\leq 7y_2 + 5y_3 &+ I_3 \\
S &= \{1,2,3\}, & x_1 + x_2 + x_3 &\leq 11y_1 + 7y_2 + 5y_3 &+ I_3
\end{aligned}$$

By substituting the fractional solutions from Figure 2.2.1, we set

$$
\begin{array}{lll}
S^* = \{1\}, & x_1^* - 11y_1^* & > 0 \rightarrow \text{Not satisfied} \\
S^* = \{2\}, & x_2^* - 7y_2^* & > 0 \rightarrow 0.8737 > 0 \\
S^* = \{3\}, & x_3^* - 5y_3^* & > 0 \rightarrow \text{Not satisfied} \\
S^* = \{1,2\}, & x_1^* + x_2^* - 11y_1^* - 7y_2^* & > 0 \rightarrow 0.8637 > 0 \\
S^* = \{1,3\}, & x_1^* + x_3^* - 11y_1^* - 5y_3^* & > 0 \rightarrow \text{Not satisfied} \\
S^* = \{2,3\}, & x_2^* + x_3^* - 7y_2^* - 5y_3^* & > 0 \rightarrow \text{Not satisfied} \\
S^* = \{1,2,3\}, & x_1^* + x_2^* + x_3^* - 11y_1^* - 7y_2^* - 5y_3^* & > 0 \rightarrow 0.8637 > 0
\end{array}
$$

Then, we test whether $\sum_{i \in S^*}(x_i^* - d_{i,\ell}y_i^*) > I_\ell^*$.

$$
\begin{array}{lll}
S^* = \{2\}, & x_2^* - 7y_2^* & > I_3^* \rightarrow 0.8737 \not> 0.875 \\
S^* = \{1,2\}, & x_1^* + x_2^* - 11y_1^* - 7y_2^* & > I_3^* \rightarrow 0.8637 \not> 0.875 \\
S^* = \{1,2,3\}, & x_1^* + x_2^* + x_3^* - 11y_1^* - 7y_2^* - 5y_3^* & > I_3^* \rightarrow 0.8637 \not> 0.875
\end{array}
$$

There is no such violated inequalities exists for the given value of ℓ.

An alternative way to write the above inequality (2.1) given by:

Corollary 1 (Pochet and Wolsey (2006)). *The (ℓ, S) inequality (2.1) can be written as:*

$$\sum_{i \in L \setminus S} x_i + \sum_{i \in S} d_{i,\ell} y_i \geq d_{1,\ell} \qquad S \subseteq [1, \ell], \forall \ell \in N$$

Proof. Substitute $\sum_{i \in L} x_i = d_{1,\ell} + I_\ell$. □

Next, as discussed earlier, the problem is said to have Wagner-Whitin costs if $p_t^{'} + h_t^{'} \geq p_{t+1}^{'}$ for all t and $p_0^{'} = p_{n+1}^{'} = 0$. In this case, it is optimal to produce as late as possible, where the setup period will be either before or equal to t in order to satisfy the demand in period t. Alternatively, I_{k-1} contains demand d_i for period $i \geq k$ only if no setup occurs in the time interval $[k, ..., i]$.

Proposition 5 (Pochet and Wolsey (2006)). *In the case of Wagner-Whitin costs, we obtain (ℓ, S, WW) inequality as follows:*

$$I_{k-1} + \sum_{i=k}^{\ell} d_{i,\ell} y_i \geq d_{k,\ell} \qquad 1 \leq k \leq \ell \leq n \tag{2.2}$$

is valid. It can be rewritten as an (ℓ, S) inequality

$$\sum_{i=1}^{k-1} x_i + \sum_{i=k}^{\ell} d_{i,\ell} y_i \geq d_{1,\ell} \qquad 1 \leq k \leq \ell \leq n$$

In this study, we derive (ℓ, S, WW) inequality from (ℓ, S) inequality for both ELSR problems. We now review several polyhedral studies of different variants of the lot-sizing problem.

Firstly, the lot-sizing problem can be formulated as a fixed-charge network flow problem. A number of polyhedral studies investigating this problem are available in the literature. Van Roy and Wolsey (1985) proposed a family of valid inequalities for single-item uncapacitated fixed-charge networks. The findings show that these inequalities are sufficient to describe the convex hull of solutions. They present a heuristic separation algorithm for this class of inequalities. Padberg et al. (1985) studied three 0-1 mixed integer sets arising from capacitated fixed charge problems. They derived two classes of facet-defining inequalities of the convex hull of the problem, and the second of these classes provides a complete description of the convex hull when the capacity is $m_t = m$ for all $t \in N$. These facets, called 'flow cover' inequalities, are used as cutting planes to tighten the formulation of a certain mixed integer problem. We extend these findings into our ELSR problems.

Pochet (1988) combined Padberg et al.'s approaches by introducing a network structure in a capacitated fixed-charge problem. In the equal capacity case, he obtained a large number of facet-defining inequalities; however, he was unable to find an efficient algorithm (polynomial algorithm). Therefore, he applied a heuristic separation algorithm. Leung et al. (1989) further extended the problem with the multi-item capacitated lot-sizing problem. The authors firstly studied the polyhedral structure of the single-item capacitated lot-sizing problem and then used the results obtained to develop methods for the multi-item case. They proposed a set of valid inequalities that defines the facets of the problem. Next, Ortega and Wolsey (2003) described dicut inequalities and their variants as well as the complexity of the separation problem of the single-commodity uncapacitated fixed-charge network flow problem. The proposed branch-and-cut algorithm was then tested computationally.

Other extensions of the lot-sizing problems include start-up costs, inventory bounds, fixed charges on stocks, step-wise production costs, one-way substitution and supplier selections. Van Hoesel et al. (1994a) provided a complete linear description of the economic lot-sizing problem with start-up costs. The authors generalized the (ℓ, S) inequalities proposed by Barany et al. (1984a) to (ℓ, R, S)-inequalities, where R be a subset of S. The separation problem was solved by formulating the problem as a shortest path problem. Escalante et al. (2011) extended the problem to continuous start-up costs. They studied the polyhedral structure of the problem by providing some general properties and deriving facet-inducing inequalities.

The polyhedral structure of the lot-sizing problem with inventory bounds was first studied by Atamtürk and Küçükyavuz (2005). Two models are considered in their study: first with linear inventory costs and second with linear and fixed inventory costs. They defined facet-defining inequalities and presented exact separation algorithms for both problems. Van Vyve and Ortega (2004) provided a polyhedral analysis of the uncapacitated lot-sizing problem with fixed charges on stocks. The authors extended the (ℓ, S)-inequalities for a complete description of the convex hull of the problem. Akbalik and Pochet (2009) considered the single-item capacitated lot-sizing problem with step-wise production costs to develop a new class of valid inequalities, namely mixed flow cover. This mixed flow cover is derived from two well-known classes of valid inequalities, namely flow cover inequalities and integer cover inequalities. They proposed a cutting-plane algorithm within a branch-and-cut procedure, where an exact polynomial separation is implemented.

Yaman (2009) investigated a polyhedral analysis for the two-item uncapacitated lot-sizing problem with one-way substitution, where the demand of a low-quality item can be substituted by a high-quality item. A family of facet-defining inequalities is derived from the projection of the feasible set onto the space of production and setup variables. In the case of two periods, these inequalities, together with the trivial facet-defining inequalities, define the convex hull of the stated projection. Zhao and Klabjan (2012) studied the polyhedral structure of both uncapacitated and capacitated lot sizing with supplier selection problems. For the uncapacitated case, a full description of the convex hull of the problem is obtained. The author defines several families of valid inequalities for the general capacitated case. Lastly, Gicquel and Minoux (2014) proposed a family of valid inequalities for the multi-product discrete lot-sizing and scheduling problem with sequence-dependent changeover costs and provided both exact and heuristic separation algorithms. The efficiency of both algorithms at strengthening the linear relaxation was tested.

In this research, we basically extend the study of Barany et al. (1984b) to our ELSR problem with separate setups and joint setups cases in Chapter 3. Then, in Chapter 4 and Chapter 5, we further investigate the polyhedral structure of a mixed integer set arising from these problems, originally motivated by Padberg et al.'s study.

2.2.2 Extended Reformulations

Extended reformulation can also be used to tighten the original formulation in order to obtain better lower bounds for mixed integer problems. The new variables are defined and added to the problem in different variable spaces. As mentioned in Chapter 1, the number of inequalities required to describe $conv(X)$ with an extended reformulation may be relatively small compared to the number of facet-defining inequalities generated to define $conv(X)$ in the original space that grows exponentially. However, the problem size of an extended reformulation is greater than adding valid inequalities due to additional new variables.

There are some well-known extended reformulation techniques in the classical lot-sizing literature. The first extended reformulation was introduced by Krarup and Bilde (1977), which is facility location (FL) reformulation for the single-item uncapacitated lot-sizing problem. The authors decomposed the production variable x_t by defining a new decision variable, $w_{t,t'}$, which is the amount produced in period

t to satisfy demand in period t'. Then, the amount of production x_t in period t is $x_t = \sum_{t'=t}^{n} w_{t,t'}$. This reformulation has $O(n^2)$ variables and $O(n^2)$ constraints, which suffices to solve LP relaxation. Barany et al. (1984b) also examined this reformulation technique for the same problem to obtain the convex hull of solutions. The second reformulation technique is more compact as it has only $O(n)$ constraints regardless of its nonnegativity constraints. This reformulation, called shortest path (SP) reformulation, was proposed by Eppen and Martin (1987). The authors define, a new variable, $z_{t,t'}$, which is the fraction of demand in periods t until t' to be satisfied by the production in period t. This reformulation has been proven to be equivalent to FL reformulation as it provides integral solutions in an extended space. Another reformulation technique that has been found useful in solving lot-sizing problem is multi-commodity (MC) reformulation, which was suggested by Rardin and Wolsey (1993) for fixed-charge network flow problems. The reformulation basically has the same formulation as FL reformulation, but the inventory variables are included in the formulation. They decompose the production flow, x_t , as a function of its destination node (demand node) at time interval $[t, t+1, ..., n]$ and the inventory flow, I_t as a function of its destination node (demand node) at time interval $[t+1, t+2, ..., n]$.

Several polyhedral studies have addressed extended reformulations for special cases of the lot-sizing problem. Pochet and Wolsey (1988) investigated the uncapacitated lot-sizing problem with backlogging. The authors used a facility location reformulation technique to define a family of valid inequalities and then used a heuristic separation algorithm to find an optimal solution. Küçükyavuz and Pochet (2009) derived a relationship between Pochet and Wolsey's FL reformulation and their facets in its natural space of production, setup, inventory and backlogging variables. Next, Pochet and Wolsey (1994) examined four different cases of the single-item lot-sizing problem with Wagner-Whitin costs: the uncapacitated problem (ULS), the uncapacitated problem with backlogging (BLS), the uncapacitated problem with start-up costs (ULSS) and the constant capacity problem (CLS). For each model, they studied the structure of the stock-minimal solutions in order to derive the extended reformulations of the problem. The extended reformulations with Wagner-Whitin costs were projected onto original spaces, which were then used to define convex hull of the stock-minimal solutions and solve separation problems.

Later, Pochet and Wolsey (2010) constructed an extended formulation for the single-item lot-sizing problem with non-decreasing capacities using mixing sets and obtained the convex hull of solutions when capacities are constant over time. The authors tested this formulation with different instances, including with and without Wagner-Whitin costs and with both non-decreasing and arbitrary capacities over time so as to observe its effectiveness. Vyve et al. (2014) proposed exact and approximate extended formulations for several variants of two-level multi-item discrete lot-sizing problems. The performance of an extended formulation for the problem with uncapacitated at both levels and start-up costs was found to be better than an existing formulation. Due to a large-size formulation of the problem with uncapacitated at the upper level and constant capacity at the lower level, they projected the formulation onto the variable space. Additionally, they constructed an extended formulation for relaxation in the case of constant capacity at both levels.

Lastly, in regard to the case of ELSR problems, Retel Helmrich et al. (2013) was the first to present a good mixed integer programming formulation for both variants of ELSR problems. Firstly, they showed that both variants are NP-hard and then followed by proposing several alternative formulations for both variants such as shortest path formulation, a partial shortest path formulation and an adaptation of the (l, S, WW)-inequalities for the classic problem with Wagner-Whitin costs to tighten the original formulation. The authors tested the efficiency of all formulations was tested on a large number of data sets and found that a (partial) shortest path type formulation outperforms the original formulation for both variants in terms of LP gaps (%), MIP computation times and number of optimal solutions.

In our research, we also propose several extended formulations for both variants of ELSR problems, namely facility location (FL) reformulation, MC reformulation and shortest path (SP) reformulation. Note that our SP formulation is slightly different from the formulation proposed by Retel Helmrich et al. (2013). Additionally, we provide theoretical and computational comparisons between these different lower bounding techniques to prove the equivalence of the formulations and to test the effectiveness of the formulations. We discuss all of these formulations in Chapter 3.

2.3 Heuristics

With the hope of obtaining good solutions in the least amount of time, a heuristic approach is an alternative method to solve a wide variety of lot-sizing problems. In this section, we provide the reader with a brief overview of mixed integer programming heuristics and other types of heuristics. As for our research, we do not use this approach to tackle our problems.

2.3.1 Mixed Integer Programming (MIP) Heuristics

There are two commonly used MIP heuristics, namely construction heuristic and improvement heuristic as discussed in Pochet and Wolsey (2006). The interested reader can refer to their article for more in-depth explanations on this heuristic. This class of heuristic aims to obtain better quality solutions in a reasonable computation time.

- Construction heuristic: This heuristic starts with no solution and constructs it step-by-step from scratch.

  (i) LP-and-Fix: The integral values in the LP relaxation solution are fixed.

(ii) Relax-and-Fix: The integrality restriction of some variables is relaxed to continuous, then the integrality of other variables is fixed at each iteration.

- Improvement heuristic: This heuristic always begins with an initial solution and the aim is to improve it.

  (i) Relaxation Induced Neighborhood Search (RINS): This heuristic was discussed in Danna et al. (2005). The idea is to explore the neighbourhood between the LP relaxation solution and the MIP solution. If both solutions produce the same value of an integer variable, then that value is fixed. This heuristic is an improved version of the LP-and-Fix heuristic.

  (ii) Local Branching (LB): This heuristic was initially introduced by Fischetti and Lodi (2003) and constructs the branching of neighboorhood using MIP solution.

  (iii) Exchange (EXCH): This is an improvement version of the relax-and-fix heuristic. At each iteration, some integer variables are fixed at their values in the best current MIP solutions, except for the variables, which are restricted to take integer values.

Several studies have considered MIP-based heuristics with decomposition of time windows. Mercé and Fontan (2003) introduced two MIP-based heuristic algorithms within a rolling horizon framework for the multi-item capacitated lot-sizing problem. Absi and Kedad-Sidhoum (2007) proposed two MIP-based heuristics, namely fix-and-relax and double-fix-and-relax using horizon decomposition, for the same problem. Beraldi et al. (2008) introduced new rolling horizon and fix-and-relax heuristics for the identical parallel machine lot-sizing and scheduling problem with sequence-dependent setup costs. Akartunalı and Miller (2009) combined LP-and-fix and relax-and-fix heuristics to develop a heuristic framework using decomposition of time windows for the big-bucket multi-level production planning problem.

These MIP heuristic approaches are useful for solving our ELSR problems in terms of reducing computation times and providing stronger lower bounds as they offer a good trade-off between solution quality and solution time.

2.3.2 Other Types of Heuristics

This section summarizes other types of heuristics. One heuristic approach is the Lagrangian relaxation heuristic (Toledo and Armentano, 2006; Rizk et al., 2006; Haugen et al., 2007a,b; Absi and Kedad-Sidhoum, 2009). It considers a relaxation of the capacity constraints, where the sub-problems are generated and can be easily solved through the Wagner–Whitin algorithm (or any other uncapacitated single-item algorithms). Another type is the branch-and-bound heuristic method (Gelders et al., 1986; Chen and Thizy, 1990; Diaby et al., 1992; Chung et al., 1994; Lotfi and Yoon, 1994; Hindi, 1995; Armentano et al., 1999). In this heuristic, nodes that are close (within a percentage) to the best current upper bound are fathomed. The optimal value of the Lagrangian relaxation heuristic can be used as a lower bound in the branch-and- bound procedure. Furthermore, some studies have used metaheuristics (Özdamar and Bozyel, 2000; Taşgetiren and Liang, 2003; Chang et al., 2006; Gaafar, 2006; Süer et al., 2008; Gaafar et al., 2009; Chandrasekaran et al., 2009) and some use classical heuristics such as silver meal (SM) and Least Unit Cost (Dixon and Silver, 1981; Dogramaci et al., 1981; Senyiğit, 2009) to solve the problems.

In the context of ELSR problems, most authors have adapted several well-known classical heuristics. Teunter et al. (2006). conjectured that the ELSR problem with separate setups case is NP-hard. Therefore, the authors suggested some modifications and comparisons of the well-known SM, Least Unit Cost (LUC) and Part Period Balancing (PPB) heuristics. They also implemented these methods for the case of joint setups. Furthermore, Schulz (2011) developed a new SM-based heuristic based on the work of Teunter et al. (2006). The findings show that the average percentage gap to the optimal solution can be reduced to less than half of the original value obtained by Teunter et al. (2006).

Metaheuristic approaches have also been utilized to tackle ELSR problems. Li et al. (2013) studied the dynamic lot-sizing problem with product returns and remanufacturing and found that the problem with general setup cost functions is an NP-hard problem. A Tabu search was suggested to produce high-quality solutions that include some new features. Their findings show that the proposed approach is better than other algorithms. Next, Baki et al. (2014) proposed a new dynamic programming-based heuristic for the same problem by analysing the properties of the block structure of optimal solutions. Sifaleras et al. (2015) investigated the same problem studied by Teunter et al. (2006), and Schulz (2011) proposed two novel variable neighbourhoods search (VNS) metaheuristic approaches. The results demonstrate that their method outperforms existing heuristic methods in the literature. Finally, Parsopoulos et al. (2015) also addressed the same problem using a different type of metaheuristic approach, namely differential evolution (DE), a promising alternative for solving the lot-sizing problem.

For a more comprehensive literature survey of variants of lot-sizing models along with their methods and industrial applications, interested readers can refer to interesting articles such as Maes and Van Wassenhove (1988), Drexl and Kimms (1997), Staggemeier and Clark (2001), Karimi et al. (2003), Brahimi et al. (2006), Quadt and Kuhn (2008), Gicquel et al. (2008) Ullah and Parveen (2010), Buschkühl et al. (2010), Clark et al. (2011) and Almada-Lobo et al. (2015).

In the following three chapters, we discuss several of the possible solution techniques addressed in this chapter to tackle both ELSRs and ELSRj problems. In each chapter, theoretical and computational test results are presented to demonstrate the effectiveness of the proposed formulations.

Chapter 3

Computational Analysis of Lower Bounds for Economic Lot Sizing Problems with Remanufacturing (ELSR)

This chapter evaluates and discusses the strength of different MIP formulations. There are two MIP-exact techniques can be used to solve ELSRs and ELSRj problems: (1) add valid inequalities into an original formulation or (2) introduce new variables into the model. These techniques provide stronger lower bounds, i.e. linear programming (LP) relaxation and reduce computation times for both problems. We organize this chapter as follows. In Section 3.1, we firstly describe the families of $(\ell, S) - like$ inequalities along with the efficient separation algorithms for both problems. The relationship between $(\ell, S) - like$ inequalities and $(\ell, S, WW) - like$ inequalities is also presented. Then, in Section 3.2, the extended reformulation techniques which are Facility Location (FL) reformulation, Multi-Commodity (MC) reformulation and Shortest Path (SP) reformulation are discussed. In Section 3.3, we provide theoretical comparisons between all the formulations. Next, Section 3.4 presents computational results for each problem and lastly, we conclude in Section 3.5.

3.1 Valid Inequalities for ELSR

In this section, we identify several families of $(\ell, S) - like$ inequalities and $(\ell, S, WW) - like$ inequalities for ELSR with separate setups and joint setups cases. We compare (ℓ, S, WW) inequalities of Retel Helmrich et al. (2013) with our $(\ell, S, WW) - like$ inequalities in order to identify the differences.

3.1.1 $(\ell, S) - like$ Inequalities for ELSR

In this section, we aim to approximate convex hull of feasible solutions for ELSR problems. We propose several families of valid inequalities, initially introduced by Barany et al. (1984a) for single-item uncapacitated problem. Adding these valid inequalities into the original formulation have been found useful to improve lower bounds and computation times. Now, we introduce several families of $(\ell, S) - like$ inequalities for both ELSRs and ELSRj problems.

Separate Setups

There are four families of $(\ell, S) - like$ inequalities have been derived for ELSRs problem.

Proposition 6. *For any* $1 \leq k \leq \ell \leq n$, *suppose that* $L = \{k, .., \ell\}$ *and* $S \subseteq L$, *then the following inequalities are valid for* X^{ss} :

$$\sum_{i \in S} x_i^r \leq \sum_{i \in S} r_{k,i} y_i^r + I_{k-1}^r \tag{3.1}$$

$$\sum_{i \in S} (x_i^r + x_i^m) \leq \sum_{i \in S} d_{i,\ell} (y_i^r + y_i^m) + I_l^s \tag{3.2}$$

$$\sum_{i \in S} x_i^r \leq \sum_{i \in S} d_{i,\ell} y_i^r + I_\ell^s \tag{3.3}$$

$$\sum_{i \in S} x_i^m \leq \sum_{i \in S} d_{i,\ell} y_i^m + I_\ell^s \tag{3.4}$$

Proof. Consider a point $(x^r, x^m, y^r, y^m, I^r, I^s) \in X^{ss}$. If $\sum_{i \in S} y_i^r = 0$, then $x_i^r = 0$, $\forall i \in S$ and $I_{k-1}^r \geq 0$, hence the inequality is satisfied. Let $\sum_{i \in S} y_i^r \geq 1$ and $p = \max \{i \in S\}$. Then $\sum_{i \in S} x_i^r \leq \sum_{i=k}^{p} x_i^r \leq r_{k,p} + I_{k-1}^r \leq \sum_{i \in S} r_{k,i} y_i^r + I_{k-1}^r$ such that $p \leq \ell$. The first inequality follows the definition S and the nonnegativity of x_i^r, second inequality shows the constraint of flow conversation for product returns and lastly using $y_p^r = 1$ and the nonnegativity of y_i^r.

For the second inequality, we use a similar technique of proofing as discussed earlier. Given a point $(x^r, x^m, y^r, y^m, I^r, I^s) \in X^{ss}$, the inequality is satisfied if both $\sum_{i \in S} y_i^r = \sum_{i \in S} y_i^m = 0$, then $x_i^r = x_i^m = 0, \forall i \in S$ and $I_\ell^s \geq 0$. Let $q = \min \{i \in S\}$ and also $\sum_{i \in S} (y_i^r + y_i^m) \geq 1$. Then $\sum_{i \in S} (x_i^r + x_i^m) \leq \sum_{i=q}^{\ell} (x_i^r + x_i^m) \leq d_{q,\ell} + I_\ell^s \leq \sum_{i \in S} d_{i,\ell} (y_i^r + y_i^m) + I_\ell^s$, where the first inequality follows the definition S and the nonnegativity of both x_i^r and x_i^m, second inequality shows the constraint of flow conversation for serviceable products and lastly using $y_q^r = 1$ and $y_q^m = 1$; and the nonnegativity of both y_i^r and y_i^m.

As regards the third inequality, suppose that we have a point $(x^r, x^m, y^r, y^m, I^r, I^s) \in X^{ss}$. Then, if $\sum_{i \in S} y_i^r = 0$, then $x_i^r = 0$, $\forall i \in S$ and $I_\ell^s \geq 0$. Let $a = \min \{i \in S\}$ and $\sum_{i \in S} y_i^r \geq 1$. We obtain $\sum_{i \in S} x_i^r \leq \sum_{i=a}^{\ell} x_i^r \leq d_{a,\ell} + I_\ell^s \leq \sum_{i \in S} d_{i,\ell} y_i^r + I_\ell^s$. The interpretation is similar to the previous ones. The proof for the last inequality can be handled in a similar manner. □

Then, we define a new feasible region of ELSRs problem associated with these family inequalities as:

$$X_{LS}^{ss} = \{(x^r, x^m, y^r, y^m, I^r, I^s)|(1.8), (1.9), (1.11) - (1.15), (3.1) - (3.4)\}$$

and the objective function is $Z_{LS}^{ss} = \min\{(1.7)|(x^r, x^m, y^r, y^m, I^r, I^s) \in X_{LS}^{ss}\}$.

Joint Setups

Next, we describe two families of $(\ell, S) - like$ inequalities for ELSRj problem in the following proposition.

Proposition 7. *Suppose that* $1 \leq k \leq \ell \leq n$, $L = \{k, .., \ell\}$ *and* $S \subseteq L$, *then the inequalities are valid for* X^{js} :

$$\sum_{i \in S} x_i^r \leq \sum_{i \in S} r_{k,i} y_i + I_{k-1}^r \tag{3.5}$$

$$\sum_{i \in S} (x_i^r + x_i^m) \leq \sum_{i \in S} d_{i,\ell} y_i + I_\ell^s \tag{3.6}$$

Proof. The interpretation for these inequalities (3.5) and (3.6) is basically similar to those in separate setups case. □

Next, we define the feasible region of ELSRj problem associated with these families of valid inequalities as:

$$X_{LS}^{js} = \{(x^r, x^m, y, I^r, I^s)|(1.8), (1.9), (1.13), (1.14), (1.17), (1.18), (3.5), (3.6)\}$$

and the objective function is $Z_{LS}^{js} = \min\left\{(1.16)|(x^r, x^m, y, I^r, I^s) \in X_{LS}^{js}\right\}$.

Since these formulations contain an exponential number of $(\ell, S) - like$ inequalities and they are not possible to add all the $(\ell, S) - like$ inequalities a priori in the both original formulation, then we can use them in a cutting plane approach. Given the fractional solutions obtained from LP relaxation at a root node, we solve the separation problem associated to the $(\ell, S) - like$ inequalities to test whether any $(\ell, S) - like$ inequality is violated or not. Algorithms 3.1 and 3.2 depict a simple polynomial separation algorithm for ELSRs and ELSRj problems, respectively. The details on separation algorithms of a simple problem can be found in Barany et al. (1984a) and Pochet and Wolsey (2006) for a single-item lot sizing problem; and a complex problem can be referred to Akartunalı and Miller (2012) for a multi-level production planning problem.

3.1.2 $(\ell, S, WW) - like$ Inequalities for ELSR

This section discusses several families of $(\ell, S, WW) - like$ inequalities for both problems, derived from $(\ell, S) - like$ inequalities.

Algorithm 3.1 (ℓ, S) separation algorithm for ELSRs problem

Input: LP relaxation solution $(x^{r*}, x^{m*}, y^{r*}, y^{m*}, I^{r*}, I^{s*})$
Output: Violated $(\ell, S) - like$ inequalities for ELSRs
for all $\ell = 1$ to n **do**
  Initialize S to be an empty set
  for all $k = 1$ to ℓ **do**
    for all $i = k$ to ℓ **do**
      if $x_i^{r*} > r_{k,i} y_i^{r*}$ or $x_i^{r*} > d_{i,\ell} y_i^{r*}$ or $x_i^{m*} > d_{i,\ell} y_i^{m*}$ or $x_i^{r*} + x_i^{m*} > d_{i,\ell}(y^{r*} + y_i^{m*})$ **then**
        $S \leftarrow S \cup \{i\}$
      end if
    end for
  end for
  if $\sum_{i \in S} x_i^{r*} > \sum_{i \in S} r_{k,i} y_i^{r*} + I_{k-1}^{r*}$ **then**
    Add violated first $(\ell, S) - like$ inequality
  end if
  if $\sum_{i \in S} (x_i^{r*} + x_i^{m*}) > \sum_{i \in S} d_{i,\ell} (y_i^{r*} + y_i^{m*}) + I_\ell^{s*}$ **then**
    Add violated second $(\ell, S) - like$ inequality
  end if
  if $\sum_{i \in S} x_i^{r*} > \sum_{i \in S} d_{i,\ell} y_i^{r*} + I_\ell^{s*}$ **then**
    Add violated third $(\ell, S) - like$ inequality
  end if
  if $\sum_{i \in S} x_i^{m*} > \sum_{i \in S} d_{i,\ell} y_i^{m*} + I_\ell^{s*}$ **then**
    Add violated fourth $(\ell, S) - like$ inequality
  end if
end for

Algorithm 3.2 (ℓ, S) separation algorithm for ELSRj problem

Input: LP relaxation solution $(x^{r*}, x^{m*}, y^{*}, I^{r*}, I^{s*})$
Output: Violated $(\ell, S) - like$ inequalities for ELSRj
for all $\ell = 1$ to n **do**
  Initialize S to be an empty set
  for all $k = 1$ to ℓ **do**
    for all $i = k$ to ℓ **do**
      if $x_i^{r*} > r_{k,i} y_i^{*}$ or $x_i^{r*} > d_{i,\ell} y_i^{*}$ or $x_i^{m*} > d_{i,\ell} y_i^{*}$ or $x_i^{r*} + x_i^{m*} > d_{i,\ell} y^{*}$ **then**
        $S \leftarrow S \cup \{i\}$
      end if
    end for
  end for
  if $\sum_{i \in S} x_i^{r*} > \sum_{i \in S} r_{k,i} y_i^{*} + I_{k-1}^{r*}$ **then**
    Add violated first $(\ell, S) - like$ inequality
  end if
  if $\sum_{i \in S} (x_i^{r*} + x_i^{m*}) > \sum_{i \in S} d_{i,\ell} y_i^{*} + I_\ell^{s*}$ **then**
    Add violated second $(\ell, S) - like$ inequality
  end if
end for

Separate Setups

To begin with, we present $(\ell, S, WW) - like$ inequalities for ELSR problem with separate setups case.

Corollary 2. *Let* $1 \leq k \leq \ell \leq n$, *then* $(\ell, S) - like$ *inequalities,* (3.1) - (3.4) *for ELSRs problem can be rewritten as:*

$$\sum_{i=k}^{\ell} x_i^r \leq \sum_{i=k}^{\ell} r_{k,i} y_i^r + I_{k-1}^r \tag{3.7}$$

$$\sum_{i=k}^{\ell} (x_i^r + x_i^m) \leq \sum_{i=k}^{\ell} d_{i,\ell} (y_i^r + y_i^m) + I_l^s \tag{3.8}$$

$$\sum_{i=k}^{\ell} x_i^r \leq \sum_{i=k}^{\ell} d_{i,\ell} y_i^r + I_\ell^s \tag{3.9}$$

$$\sum_{i=k}^{\ell} x_i^m \leq \sum_{i=k}^{\ell} d_{i,\ell} y_i^m + I_\ell^s \tag{3.10}$$

or as $(\ell, S, WW) - like$ *inequalities*

$$I_\ell^r + \sum_{i=k}^{\ell} r_{k,i} y_i^r \geq r_{k,\ell} \tag{3.11}$$

$$I_{k-1}^s + \sum_{i=k}^{\ell} d_{i,\ell} (y_i^r + y_i^m) \geq d_{k,\ell} \tag{3.12}$$

$$I_{k-1}^s + \sum_{i=k}^{\ell} d_{i,\ell} y_i^r + \sum_{i=k}^{\ell} \left(I_i^s - I_{i-1}^s - I_{i-1}^r - r_i + I_i^r + d_i \right) \geq d_{k,\ell} \tag{3.13}$$

$$I_{k-1}^s + \sum_{i=k}^{\ell} d_{i,\ell} y_i^m + \sum_{i=k}^{\ell} \left(r_i + I_{i-1}^r - I_i^r \right) \geq d_{k,\ell} \tag{3.14}$$

Proof. We prove the first valid inequality by substituting $I_\ell^r = I_{k-1}^r + r_{k,\ell} - \sum_{i=k}^{\ell} x_i^r$ into (3.11) as follows:

$$I_\ell^r + \sum_{i=k}^{\ell} r_{k,i} y_i^r \geq r_{k,\ell}$$

$$I_{k-1}^r + r_{k,\ell} - \sum_{i=k}^{\ell} x_i^r + \sum_{i=k}^{\ell} r_{k,i} y_i^r \geq r_{k,\ell}$$

$$I^r_{k-1} + \sum_{i=k}^{\ell} r_{k,i} y^r_i \geq \sum_{i=k}^{\ell} x^r_i$$

$$\sum_{i=k}^{\ell} x^r_i \leq \sum_{i=k}^{\ell} r_{k,i} y^r_i + I^r_{k-1}$$

The remaining valid inequalities will use the similar techniques of proofing by substituting the constraints, (1.8), (1.9) and $I^s_{k-1} = I^s_\ell + d_{k,\ell} - \sum_{i=k}^{\ell} (x^r_i + x^m_i)$ into (3.12) - (3.14). □

Joint Setups

In the case of joint setups, we obtain the following inequalities.

Corollary 3. *Suppose that* $1 \leq k \leq \ell \leq n$, *then we can rewrite* $(\ell, S) - like$ *inequalities,* (3.5) - (3.6) *for ELSRj problem as:*

$$\sum_{i=k}^{\ell} x^r_i \leq \sum_{i=k}^{\ell} r_{k,i} y_i + I^r_{k-1} \tag{3.15}$$

$$\sum_{i=k}^{\ell} (x^r_i + x^m_i) \leq \sum_{i=k}^{\ell} d_{i,\ell} y_i + I^s_\ell \tag{3.16}$$

or as $(\ell, S, WW) - like$ *inequalities*

$$I^r_\ell + \sum_{i=k}^{\ell} r_{k,i} y_i \geq r_{k,\ell} \tag{3.17}$$

$$I^s_{k-1} + \sum_{i=k}^{\ell} d_{i,\ell} y_i \geq d_{k,\ell} \tag{3.18}$$

Proof. The same intepretations as previous. □

Next, we discuss the similarities and differences between our proposed $(\ell, S, WW) - like$ inequalities and (ℓ, S, WW) inequalities proposed by Retel Helmrich et al. (2013).

Proposition 8 (Retel Helmrich et al. (2013))**.** *The* (ℓ, S, WW) *inequalities for ELSRs problem below are valid:*

$$I^r_\ell + \sum_{i=k}^{\ell} r_{k,i} y^r_i \geq r_{k,\ell} \qquad 1 \leq k \leq \ell \leq n \tag{3.19}$$

$$I^s_{k-1} + \sum_{i=k}^{\ell} d_{i,\ell} (y^r_i + y^m_i) \geq d_{k,\ell} \qquad 2 \leq k \leq \ell \leq n \tag{3.20}$$

Proposition 9 (Retel Helmrich et al. (2013))**.** *The following* (ℓ, S, WW) *inequalities for ELSRj problem are valid:*

$$I^r_\ell + \sum_{i=k}^{\ell} r_{k,i} y_i \geq r_{k,\ell} \qquad 1 \leq k \leq \ell \leq n \qquad (3.21)$$

$$I^s_{k-1} + \sum_{i=k}^{\ell} d_{i,\ell} y_i \geq d_{k,\ell} \qquad 2 \leq k \leq \ell \leq n \qquad (3.22)$$

We observe that their valid inequalities, (3.19) and (3.21) are identical with our valid inequalities, (3.11) and (3.17). However, the planning horizon of our valid inequalities, (3.12) for separate setups and (3.18) for joint setups include period 1. There is a need to take into an account the first period because the inventory of product returns and serviceable products are assumed to be zero at the beginning of period 1 ($I^r_0 = I^s_0 = 0$) in the original formulations, thus the remanufactured or new products should be produced at period 1 in order to satisfy the demand at that particular period.

In the following section, we introduce several extended reformulation techniques for both ELSR problems such as facility location (FL) reformulation, multi-commodity (MC) reformulation and shortest path (SP) reformulation.

3.2 Extended Reformulations for ELSR

In this section, we propose and examine three reformulation techniques to solve ELSR problems. Firstly, we suggest facility location (FL) reformulation, which is originally developed by Krarup and Bilde (1977) for single-item uncapacitated problem. Next, the second formulation is based on multi-commodity (MC) reformulation, introduced by Rardin and Wolsey (1993) for fixed-charge network problems is discussed. Lastly, we propose shortest path (SP) reformulation as similar to Retel Helmrich et al. (2013), yet we present an alternative formulation. This formulation is originally introduced by Eppen and Martin (1987) for a classical capacitated lot sizing problem.

3.2.1 Facility Location Reformulation

The first formulation is based on the single-item facility location (FL) reformulation problem, proposed by Krarup and Bilde (1977). This reformulation disaggregates the production variables of remanufacturing, x^r_t and manufacturing, x^m_t by defining new decision variables as follows:

$w^{sr}_{t,t'}$ — is the amount of remanufactured products produced in period t to satisfy the demand in period t', where $t' \geq t$,

$w^{sm}_{t,t'}$ — is the amount of new products produced in period t to satisfy the demand in period t', where $t' \geq t$.

We also introduce a new decision variable involving returns that is used for linking the variable $w^{sr}_{t,t'}$:

$w^{r}_{t,t'}$ is the amount of remanufactured products produced in period t' based on used products were returned in period t, where $t' \geq t$.

Separate Setups

In an extended variable space of ELSRs, the following constraints are added into the original formulation.

$$x^r_t = \sum_{t'=1}^{t} w^r_{t',t} \qquad \forall t \in [1,n] \tag{3.23}$$

$$x^m_t = \sum_{t'=t}^{n} w^{sm}_{t,t'} \qquad \forall t \in [1,n] \tag{3.24}$$

$$w^{sr}_{t,t'} \leq d_{t'} y^r_t \qquad \forall t \in [1,n],\ \forall t' \in [t,n] \tag{3.25}$$

$$w^{sm}_{t,t'} \leq d_{t'} y^m_t \qquad \forall t \in [1,n],\ \forall t' \in [t,n] \tag{3.26}$$

$$w^{r}_{t',t} \leq r_{t'} y^r_t \qquad \forall t \in [1,n],\ \forall t' \in [1,t] \tag{3.27}$$

$$\sum_{t'=1}^{t} \left(w^{sr}_{t',t} + w^{sm}_{t',t} \right) = d_t \qquad \forall t \in [1,n] \tag{3.28}$$

$$\sum_{t'=t}^{n} w^r_{t,t'} \leq r_t \qquad \forall t \in [1,n] \tag{3.29}$$

$$\sum_{t'=1}^{t} w^r_{t',t} = \sum_{t'=t}^{n} w^{sr}_{t,t'} \qquad \forall t \in [1,n] \tag{3.30}$$

$$w^{sr}, w^{sm}, w^r \geq 0 \tag{3.31}$$

Constraints (3.23) and (3.24) indicate the relationship between old and new variables. Constraints (3.25) - (3.27) ensure positive production of remanufactured and new products, where $w^{sr}_{t,t'} \geq 0$, $w^{sm}_{t,t'} \geq 0$ and $w^{r}_{t',t} \geq 0$ respectively. Constraint (3.28) guarantees the demands of remanufactured and new products are satisfied. Constraint (3.29) limits the production of remanufactured products by the amount of product returns. Constraint (3.30) links $w^r_{t,t'}$ to the $w^{sr}_{t,t'}$ variables, which means that the total amount of returns retrieved from period 1 to t is remanufactured at period t to satisfy the total amout of demands from period t to n. Lastly, (3.31) denotes the nonnegativity constraints. We then define the feasible region and objective function associated with this formulation as:

$$X^{ss}_{FL} = \{x^r, x^m, y^r, y^m, I^r, I^s, w^r, w^{sr}, w^{sm})|(1.8),(1.9),(1.11)-(1.15), (3.23)-(3.31)\}$$

and $Z^{ss}_{FL} = \min\{(1.7)|(x^r, x^m, y^r, y^m, I^r, I^s, w^r, w^{sr}, w^{sm}) \in X^{ss}_{FL}\}$, respectively.

Joint Setups

As one single setup variable is considered in ELSRj problem, the constraints (3.25) - (3.27) are replaced with constraints (3.32) and (3.33).

$$w^{sr}_{t,t'} + w^{sm}_{t,t'} \leq d_{t'} y_t \qquad \forall t \in [1, n], \qquad \forall t' \in [t, n] \tag{3.32}$$

$$w^{r}_{t',t} \leq r_{t'} y_t \qquad \forall t \in [1, n], \qquad \forall t' \in [1, t] \tag{3.33}$$

The feasible region associated with this formulation can be defined as:

$$X^{js}_{FL} = \{(x^r, x^m, y, I^r, I^s, w^r, w^{sr}, w^{sm}) | (1.8), (1.9), (1.13), (1.14), (1.17), (1.18), (3.27), (3.29) - (3.33)\}$$

with the objective function is:

$$Z^{js}_{FL} = \min \left\{ (1.16) | (x^r, x^m, y, I^r, I^s, w^r, w^{sr}, w^{sm}) \in X^{js}_{FL} \right\}$$

This formulation adds $O(n^2)$ variables and $O(n^2)$ constraints to the problem.

3.2.2 Multi-commodity Reformulation

The original formulation of ELSRs problem can be reformulated as multi-commodity (MC) reformulation. This alternative approach, defined by Rardin and Wolsey (1993) has been found to be effective in tightening the formulation of fixed-charge network flow problems. We decompose the production flow for both remanufacturing, x^r_t and manufacturing, x^m_t as functions of their destination nodes (return and demand periods) at $t, t+1, .., n$. The inventory flow for both product returns, I^r_t and serviceable products, I^s_t are also decomposed at $t+1, t+2, .., n$. Unlike a classical lot sizing problem, we consider two types of commodities which are:

(i) Commodity, A_t represents the return delivered onto the system in period t, where $t \leq t$',

(ii) Commodity $B_{t'}$ corresponds the demand delivered onto the system in period t', where $t \leq t$'.

Now, we define new decision variables as follows:

$u^{sr}_{t,t'}$ is the amount of remanufactured products in period t of commodity $B_{t'}$,

$u^{sm}_{t,t'}$ is the amount of new products in period t of commodity $B_{t'}$,

$v^{rp}_{t,t'}$ is the inventory of product returns at the end of period t' of commodity A_t,

$v^{sp}_{t,t'}$ is the inventory of serviceable products at the end of period t of commodity $B_{t'}$.

As similar to facility location reformulation, we also include a new linking decision variable that is:

$u^{rr}_{t,t'}$ is the amount of remanufactured products in period t' produced from commodity A_t.

Note that both inventory variables, $v^{rp}_{t,t} = 0$ and $v^{sp}_{t,t} = 0$ for all $t = 1, .., n$ do not exist as the commodity cannot be both returned to the system or delivered and hold in stock in period t, respectively. Also, the inventory stock of both product returns and serviceable products at the end of the planning horizon are assumed to be zero.

Separate Setups

For the case of separate setups, we add the following constraints into the original formulation.

$$x^r_t = \sum_{t'=1}^{t} u^{rr}_{t',t} \qquad t \in [1, n] \tag{3.34}$$

$$x^m_t = \sum_{t'=t}^{n} u^{sm}_{t,t'} \qquad t \in [1, n] \tag{3.35}$$

$$u^{sr}_{t,t'} \leq d_{t'} y^r_t \qquad t \in [1, n] \qquad t' \in [t, n] \tag{3.36}$$

$$u^{sm}_{t,t'} \leq d_{t'} y^m_t \qquad t \in [1, n] \qquad t' \in [t, n] \tag{3.37}$$

$$u^{rr}_{t',t} \leq r_{t'} y^r_t \qquad t \in [1, n], \qquad t' \in [1, t] \tag{3.38}$$

$$v^{sp}_{t-1,t} + u^{sr}_{t,t} + u^{sm}_{t,t} = d_t \qquad t \in [1, n] \tag{3.39}$$

$$v^{sp}_{t-1,t'} + u^{sr}_{t,t'} + u^{sm}_{t,t'} = v^{sp}_{t,t'} \qquad t \in [1, n-1], \qquad t' \in [(t+1), n] \tag{3.40}$$

$$\sum_{t'=t}^{n} v^{rp}_{t,t'} + \sum_{t'=t}^{n} u^{rr}_{t,t'} = r_t \qquad t \in [1, n] \tag{3.41}$$

$$\sum_{t'=1}^{t} u^{rr}_{t',t} = \sum_{t'=t}^{n} u^{sr}_{t,t'} \qquad t \in [1, n] \tag{3.42}$$

$$u^{sr}, u^{sm}, u^{rr}, v^{rp}, v^{sp} \geq 0 \tag{3.43}$$

Constraints (3.34) and (3.35) represent the relationship between old and new variables. Constraints (3.36) - (3.38) are setup forcing constraints. Constraints (3.39) and (3.40) are inventory flow balance for serviceable products and constraint (3.41) is for inventory flow balance for product returns. Constraint (3.42) links the variables between $u^{rr}_{t,t'}$ and $u^{sr}_{t,t'}$. Lastly, (3.43) provides the nonnegativity constraints. Then, the feasible region and objective function of this new formulation are:

$$X^{ss}_{MC} = \{(x^r, x^m, y^r, y^m, u^{rr}, u^{sr}, u^{sm}, v^{rp}, v^{sp}) | (1.8), (1.9), (1.11) - (1.15), (3.34) - (3.43)\}$$

and $Z_{MC}^{ss} = \min\{(1.7)|(x^r, x^m, y^r, y^m, u^{rr}, u^{sr}, u^{sm}, v^{rp}, v^{sp}) \in X_{MC}^{ss}\}$, respectively. This formulation also adds $O(n^2)$ variables and $O(n^2)$ constraints to the problem.

Joint Setups

As regards joint setups case, we exclude the constraints (3.36) - (3.38) and replace with the following constraints.

$$u_{t,t'}^{sr} + u_{t,t'}^{sm} \le d_{t'} y_t \qquad t \in [1,n], \qquad t' \in [t,n] \tag{3.44}$$

$$u_{t',t}^{rr} \le r_{t'} y_t \qquad t \in [1,n], \qquad t' \in [1,t] \tag{3.45}$$

From this, we define the feasible region as:

$$X_{MC}^{js} = \{(x^r, x^m, y, u^{rr}, u^{sr}, u^{sm}, v^{rp}, v^{sp})|(1.8),(1.9),(1.13),(1.14),(1.17),(1.18), (3.34)-(3.36),(3.39)-(3.45)\}$$

with the objective function is:

$$Z_{MC}^{js} = \min\left\{(1.16)|(x^r, x^m, y, u^{rr}, u^{sr}, u^{sm}, v^{rp}, v^{sp}) \in X_{MC}^{js}\right\}$$

3.2.3 Shortest Path Reformulation

The last reformulation technique is shortest path (SP) reformulation defined by Eppen and Martin (1987) for classical capacitated lot-sizing problem. With respect to ELSR problem, Retel Helmrich et al. (2013) is the first ones to introduce shortest path reformulation techniques. We basically benefit from their ideas to find an alternative way of developing SP formulation. We define the decision variables as follows:

$z_{t,t'}^{sr}$ is the fraction of demand in each period t until t' that is satisfied by production of remanufactured products in period t,

$z_{t,t'}^{sm}$ is the fraction of demand in each period t until t' that is satisfied by production of new products in period t,

$z_{t,t'}^{r}$ is the fraction of return in each period t until t' that is remanufactured in period t'.

These new variables, $z_{t,t'}^{sr}$ and $z_{t,t'}^{sm}$ are 1 if production occurs in period t to satisfy all demands in periods $t, ..., t'$ and 0 otherwise. Also, the variable, $z_{t,t'}^{r}$ is 1 if used products returned in periods $t, .., t'$ to be remanufactured in period t' and 0 otherwise.

Then, as discussed by Retel Helmrich et al. (2013), it is possible to have the final inventory of product returns, i.e. not all returns need to be remanufactured within the planning horizon. They define:

f_t is the fraction of return in each of the periods t until n that is added to the final inventory of product returns at the end of period n.

where $I_t^r = \sum_{t=1}^{n} r_{t,n} f_t$.

Separate Setups

As regards separate setups case, the objective function (3.46) that consists of setup and production costs for both remanufacturing and manufacturing still remains the same as in the original formulation (1.7). However, the formulation for holding costs for product returns and serviceable products are redefined, presented in (3.47).

$$\min \sum_{t=1}^{n} \left(K_t^r y_t^r + K_t^m y_t^m + p_t^r x_t^r + p_t^m x_t^m\right) \tag{3.46}$$

$$\min \sum_{t=1}^{n} \sum_{t'=t}^{n} \left(c_{t,t'}^r z_{t,t'}^r + c_{t,t'}^s \left(z_{t,t'}^{sr} + z_{t,t'}^{sm}\right)\right) + \sum_{t=1}^{n} c_t^f f_t \tag{3.47}$$

where, $c_{t,t'}^r = \sum_{u=t}^{t'-1} h_t^r r_{t,u}$, $c_{t,t'}^s = \sum_{u=t}^{t'-1} h_t^s d_{u+1,t'}$ and $c_t^f = \sum_{u=t}^{n} h_u^r r_{t,u}$. Then, the constraints are:

$$\text{s.t. (1.12)} \quad x_t^r = \sum_{i=1}^{t} r_{i,t} z_{i,t}^{sr} \qquad t \in [1,n] \tag{3.48}$$

$$x_t^m = \sum_{i=t}^{n} d_{t,i} z_{t,i}^{sm} \qquad t \in [1,n] \tag{3.49}$$

$$\sum_{t'=t:d_{t,t'}>0}^{n} z_{t,t'}^{sr} \leq y_t^r \qquad t \in [1,n] \tag{3.50}$$

$$\sum_{t'=t:d_{t,t'}>0}^{n} z_{t,t'}^{sm} \leq y_t^m \qquad t \in [1,n] \tag{3.51}$$

$$\sum_{t'=1:r_{t',t}\geq 0}^{t} z_{t',t}^{r} \leq y_t^r \qquad t \in [1,n] \tag{3.52}$$

$$\sum_{t=1}^{n} \left(z_{t,n}^{sr} + z_{t,n}^{sm}\right) = 1 \tag{3.53}$$

$$-\sum_{t=1}^{n} \left(z_{1,t}^{sr} + z_{1,t}^{sm}\right) = -1 \tag{3.54}$$

$$\sum_{t=1}^{t'} \left(z_{t,t'}^{sr} + z_{t,t'}^{sm}\right) = \sum_{t=t'+1}^{n} \left(z_{t'+1,t}^{sr} + z_{t+1'.t}^{sm}\right) \qquad t' \in [1,n-1] \tag{3.55}$$

$$\sum_{t=1}^{n} z_{t,n}^{r} + f_t = 1 \qquad t \in [1,n] \tag{3.56}$$

$$-\sum_{t=1}^{n} z^r_{1,t} - f_1 = -1 \tag{3.57}$$

$$\sum_{t=1}^{t'} z^r_{t,t'} = \sum_{t=t'+1}^{n} z^r_{t'+1,t} + f_{t'+1} \qquad t' \in [1, n-1] \tag{3.58}$$

$$\sum_{t=1}^{t'} r_{t,t'} z^r_{t,t'} = \sum_{t=t'}^{n} d_{t',t} z^{sr}_{t',t} \qquad t' \in [1, n] \tag{3.59}$$

$$z^{sr}, z^{sm}, z^r \geq 0 \tag{3.60}$$

The constraints (3.48) and (3.49) are added into shortest path reformulation as a linking variable between old and new variables. Constraints (3.50) - (3.52) represent the setup forcing constraints between the linear and binary variables. Then, the constraints (3.53) - (3.57) are flow conservation constraints and (3.59) links between z^r and z^{sr} variables. Finally, (3.60) is nonnegativitivity constraints. The feasible region associated with this formulation can be defined as:

$$X^{ss}_{SP} = \{(x^r, x^m, y^r, y^m, z^r, z^{sr}, z^{sm}) | (1.12), (3.48) - (3.60)\}$$

and the problem is $Z^{ss}_{SP} = \min\{(3.46), (3.47) | \ (x^r, x^m, y^r, y^m, z^r, z^{sr}, z^{sm}) \in X^{ss}_{SP}\}$.

Joint Setups

In the case of joint setups, we consider one flow variables $z^{sp}_{t,t'}$:

$z^{sp}_{t,t'}$ is the fraction of the demand in each period t until t' that is satisfied by remanufactured products and new products in period t.

We also use the returns variable, $z^r_{t,t'}$ addressed earlier. Then, the objective function of ELSRj problem is given as:

$$\min \sum_{t=1}^{n} (K_t y_t + p^r_t x^r_t + p^m_t x^m_t) \tag{3.61}$$

$$\min \sum_{t=1}^{n} \sum_{t'=t}^{n} \left(c^r_{t,t'} z^r_{t,t'} + c^s_{t,t'} z^{sp}_{t,t'} \right) + \sum_{t=1}^{n} c^f_t f_t \tag{3.62}$$

This is followed by the constraints.

$$\text{s.t. } (1.18),(3.48),(3.49),(3.56) - (3.58), \qquad \sum_{t'=t:d_{t,t'}>0}^{n} z^{sp}_{t,t'} \leq y_t \quad t \in [1, n] \tag{3.63}$$

$$\sum_{t'=1:r_{t',t}\geq 0}^{t} z^r_{t',t} \leq y_t \quad t \in [1, n] \tag{3.64}$$

$$\sum_{t=1}^{n} z^{sp}_{t,n} = 1 \tag{3.65}$$

$$-\sum_{t=1}^{n} z^{sp}_{1,t} = -1 \tag{3.66}$$

$$\sum_{t=1}^{t'} z^{sp}_{t,t'} = \sum_{t=t'+1}^{n} z^{sp}_{t'+1,t} \qquad t' \in [1, n-1] \tag{3.67}$$

$$\sum_{t=1}^{t'} r_{t,t'} z^{r}_{t,t'} \leq \sum_{t=t'}^{n} d_{t',t} z^{sp}_{t',t} \qquad t' \in [1, n] \tag{3.68}$$

$$z^{sp}, z^{r} \geq 0 \tag{3.69}$$

Constraints (3.63) and (3.64) are setup forcing constraints for manufacturing and remanufacturing processes, respectively. Next, constraints (3.65) - (3.67) are flow conservation constraints. (3.68) represents the relationships between z^r and z^{sp} and (3.69) indicates nonnegativity constraints. The feasible region associated with this formulation can be defined as:

$$X^{js}_{SP} = \{(x^r, x^m, y, z^r, z^{sp}) | (1.18), (3.48), (3.49), (3.56) - (3.58), (3.63) - (3.69)\}$$

and the objective function is $Z^{js}_{SP} = \min \left\{ (3.61), (3.62) | (x^r, x^m, y, z^r, z^{sp}) \in X^{js}_{SP} \right\}$.

3.3 Theoretical Comparisons between Formulations

In this section, we establish the equivalence between the solution approaches. Note that the binary setup variable, y is restricted to take integer value of 0 or 1. If we relax the integrality constraint of binary variable y to be continuous, in which it can take any value between the interval $0 \leq y \leq 1$, we call this as LP relaxation. Let superscript LP denotes as the LP relaxation of a problem. For instance, Z^{LPj}_{FL} indicates the problem Z^{js}_{FL} with the relaxed binary variable, y.

Proposition 10. $Z^{LPs}_{FL} = Z^{LPs}_{MC} = Z^{LPs}_{SP}$.
These results show that identical lower bounds are obtained by three reformulation techniques for the original ELSRs problem.

Proof. We will prove that $Z^{LPs}_{FL} = Z^{LPs}_{MC}$. In order to prove these two formulations are identical, we can show that facility reformulation (3.23) - (3.31) is equivalent to the multi-commodity reformulation (3.34) - (3.43). Firstly, we eliminate inventory variables in the constraints of MC, (3.39) - (3.41). As a result, the addition of two flow conservation constraints (3.39) and (3.40) of MC is equivalent to the constraint (3.28) of FL. Also, the constraint (3.41) of MC has the same formulation as the constraint (3.29) of FL as a result of removing inventory variables.

Lastly, as $Z_{FL}^{LPs} = Z_{MC}^{LPs}$ then we prove that $Z_{FL}^{LPs} = Z_{SP}^{LPs}$. The shortest path reformulation is equivalent to the facility location reformulation such that $w_{t-1,t'}^{r} \geq w_{t,t'}^{r}$ for any $1 < t \leq t' \leq n$; and $w_{t,t'}^{sr} \geq w_{t,t'+1}^{sr}$ and $w_{t,t'}^{sm} \geq w_{t,t'+1}^{sm}$ for any $1 \leq t \leq t' < n$ using the substitution of variables changes $z_{t,t'}^{r} = w_{t-1,t'}^{r} - w_{t,t'}^{r}$ for any $1 < t \leq t' \leq n$; and $z_{t,t'}^{sr} = w_{t,t'}^{sr} - w_{t,t'+1}^{sr}$ and $z_{t,t'}^{sm} = w_{t,t'}^{sm} - w_{t,t'+1}^{sm}$ for any $1 \leq t \leq t' \leq n$. This demonstrates the equivalence of three reformulation techniques. Interested readers can be referred to Pochet and Wolsey (1988) on the proofing of a simple lot sizing problem. □

Proposition 11. $Z_{FL}^{LPj} = Z_{MC}^{LPj} = Z_{SP}^{LPj}$.
This shows that facility location reformulation, multi-commodity reformulation and shortest path reformulation provide the same lower bounds for the original ELSRj problem.

Proof. Using the same technique of proofing addressed in the previous proposition, we obtain $Z_{FL}^{LPj} = Z_{MC}^{LPj}$. Then, we will prove that $Z_{FL}^{LPj} = Z_{SP}^{LPj}$. Similarly, these two formulations are equivalent, augmented with $w_{t-1,t'}^{r} \geq w_{t,t'}^{r}$ for any $1 < t \leq t' \leq n$; and $w_{t,t'}^{sr} + w_{t,t'}^{sm} \geq w_{t,t'+1}^{sr} + w_{t,t'+1}^{sm}$ for any $1 \leq t \leq t' < n$ and using the substitution of variables changes $z_{t,t'}^{r} = w_{t-1,t'}^{r} - w_{t,t'}^{r}$ for any $1 < t \leq t' \leq n$ and $z_{t,t'}^{sp} = w_{t,t'}^{sr} + w_{t,t'}^{sm} - w_{t,t'+1}^{sr} - w_{t,t'+1}^{sm}$ for any $1 \leq t \leq t' \leq n$. This completes the proof. □

Proposition 12. $Z_{LS}^{LPj} = Z_{FL}^{LPj}$. *This indicates that the lower bounds provided by $(\ell, S) - like$ inequalities for joint setups case is identical with extended reformulation, namely facility location reformulation. Note that all reformulation techniques are equivalent; therefore, only facility location reformulation is considered in this proof.*

Proof. In order to prove the equivalence of $(\ell, S) - like$ inequalities and facility location reformulation, we will show that the separation algorithm of $(\ell, S) - like$ inequalities can be used to derive an extended formulation of uncapacitated lot sizing problem that is facility location reformulation as similar to Pochet et al. (1995) for uncapacitated lot sizing problem. Note that we have the same objective function for both $(\ell, S) - like$ inequalities and facility location reformulation.

We firstly eliminate stock variables from the constraints (1.8) and (1.9) of the original formulation and we obtain the equivalent formulation as follows.

$$r_{1,t-1} + \sum_{j=t}^{\ell} r_{t,j} y_j \geq \sum_{j=1}^{\ell} x_j^r \qquad \text{for } 1 \leq t \leq \ell \leq n \tag{3.70}$$

$$\sum_{j=1}^{t-1} \left(x_j^r + x_j^m\right) + \sum_{j=t}^{\ell} d_{j,\ell} y_j \geq d_{1,\ell} \qquad \text{for } 1 \leq t \leq \ell \leq n \tag{3.71}$$

$$y_1 = 1 \tag{3.72}$$

$$x_j^r, x_j^m \geq 0,\ 0 \leq y_j \leq 1 \qquad \text{for all } j \tag{3.73}$$

The inequalities $I^r_\ell = r_{1,\ell} - \sum_{j=1}^{\ell} x^r_j \geq 0$ and $I^s_\ell = \sum_{j=1}^{\ell} (x^r_j + x^m_j) - d_{1,\ell} \geq 0$ with $y_j = 0, \forall j \in \{t, ..., \ell\}$ correspond to (3.70) and (3.71), respectively. Then, $y_1 = 1$ comes from $x^r_1 + x^m_1 = d_1 + I^s_1 \geq d_1 > 0$. Lastly, (3.73) ensure nonnegativity and integrality, respectively.

Now, we establish the relationship between this equivalent formulation and our facility location reformulation. We introduce new variables, $\pi^{sr}_{j,\ell}$ and $\pi^{sm}_{j,\ell}$ to represent the production of remanufactured and new products in period j for periods j up to ℓ, respectively. We also consider variable, $\pi^r_{j,\ell}$ to represent the amount of returns in periods j up to ℓ, where at period, ℓ the production of remanufactured products will occur. This variable is used as linking variables to the variables, $\pi^{sr}_{j,\ell}$. Then, we have the following formulation, Q:

$$
\begin{aligned}
(Q) \qquad & \sum_{j=1}^{\ell} \pi^r_{j,\ell} \geq \sum_{j=1}^{\ell} x^r_j && \text{for } 1 \leq \ell \leq n && (3.74)\\
& \pi^r_{j,\ell} \leq r_j && \text{for } 1 \leq j \leq \ell \leq n && (3.75)\\
& \pi^r_{j,\ell} \leq r_j y_\ell && \text{for } 1 \leq j \leq \ell \leq n && (3.76)\\
& \sum_{j=1}^{\ell} \left(\pi^{sr}_j + \pi^{sm}_j\right) \geq d_{1,\ell} && \text{for } 1 \leq \ell \leq n && (3.77)\\
& \pi^{sr}_{j,\ell} + \pi^{sm}_{j,\ell} \leq x^r_j + x^m_j && \text{for } 1 \leq j \leq \ell \leq n && (3.78)\\
& \pi^{sr}_{j,\ell} + \pi^{sm}_{j,\ell} \leq d_{j,\ell} y_j && \text{for } 1 \leq j \leq \ell \leq n && (3.79)\\
& \pi^r_{1,j} = \pi^{sr}_{j,n} && \text{for } 1 \leq j \leq n && (3.80)\\
& \pi^r_{j,\ell}, \pi^{sr}_{j,\ell}, \pi^{sm}_{j,\ell} \geq 0, \; 0 \leq y_j \leq 1 && \text{for } 1 \leq j \leq \ell \leq n && (3.81)
\end{aligned}
$$

where $(x^r, x^m, y, \pi^r, \pi^{sr}, \pi^{sm}) \in Q$ and $\min\{(1.16)|(x^r, x^m, y, \pi^r, \pi^{sr}, \pi^{sm}) \in Q\}$ is an extended reformulation of ELSRj.

With regard to the relationship between Q and the facility location (FL) reformulation, we consider the definitions, $\pi^r_{j,\ell} = \sum_{t=j}^{\ell} w^r_{j,t}$, $\pi^{sr}_{j,\ell} = \sum_{t=j}^{\ell} w^{sr}_{j,t}$ and $\pi^{sm}_{j,\ell} = \sum_{t=j}^{\ell} w^{sm}_{j,t}$. By using these definitions of variable changes, it suffices to show that any solution $(x^r, x^m, y, w^r, w^{sr}, w^{sm}) \in X^{js}_{FL}$ of the linear programming relaxation of FL, (3.23), (3.24) and (3.28) - (3.33) correlate to a point $(x^r, x^m, y, \pi^r, \pi^{sr}, \pi^{sm}) \in Q$ with the same objective function value.

Suppose that any $(x^r, x^m, y, w^r, w^{sr}, w^{sm})$ satisfying (3.23), (3.24) and (3.28) - (3.33), then we check whether the point $(x^r, x^m, y, \pi^r, \pi^{sr}, \pi^{sm})$ belongs to Q. Firstly, constraints (3.23) and (3.24), $x^r_t = \sum_{t=j}^{n} w^{sr}_{j,t} \geq \sum_{t=j}^{\ell} w^{sr}_{j,t} = \pi^{sr}_{j,\ell}$ and $x^m_t = \sum_{t=j}^{n} w^{sm}_{j,t} \geq \sum_{t=j}^{\ell} w^{sm}_{j,t} = \pi^{sm}_{j,\ell}$ for all $1 \leq j \leq \ell \leq n$. Then, summing the constraint (3.28) over $t = 1, ..., \ell$ gives $\sum_{t=1}^{\ell}\sum_{j=1}^{t} \left(w^{sr}_{j,t} + w^{sm}_{j,t}\right) = \sum_{j=1}^{\ell}\sum_{t=j}^{\ell} \left(w^{sr}_{j,t} + w^{sm}_{j,t}\right) = \sum_{j=1}^{\ell} \left(\pi^{sr}_{j,\ell} + \pi^{sm}_{j,\ell}\right) =$

$d_{1,\ell}$ for all $\ell = 1, ..., n$. Next, for constraint (3.29), let $\ell = n$ then we have $\sum_{j=t}^{n} w_{t,j}^{r} = \pi_{t,n}^{r} \leq r_t$ for all $t = 1, ..., n$. As regards constraint (3.30), since $\pi_{1,j}^{r} = \pi_{j,n}^{sr}$ then $\sum_{j=1}^{t} w_{j,t}^{r} = \sum_{j=t}^{n} w_{t,j}^{sr}$ holds true. Also, summing the constraint (3.32) over $t = j, ..., \ell$, then $\sum_{t=j}^{\ell} \left(w_{j,t}^{sr} + w_{j,t}^{sm} \right) = \pi_{j,\ell}^{sr} + \pi_{j,\ell}^{sm} \leq d_{j,\ell} y_j = d_t y_j$ for all $t = j, ..., \ell$. Finally, the constraint (3.33), $w_{t,j}^{r} = \pi_{t,j}^{r} \leq r_t y_j$ for all $1 \leq t \leq j \leq n$. These complete the proof. □

In the next section, we present computational analysis of lower bounds for both ELSRs and ELSRj problems, where the strength of different lower bounding techniques, $(\ell, S) - like$ inequalities and extended reformulations are tested using a great extent of data sets available from the literature.

3.4 Computational Testing of Lower Bounds

The primary aim of this section is to computationally test the theoretical results discussed earlier and examine their effectiveness in improving lower bounds for ELSR problems. We run 360 test instances obtained from Retel Helmrich et al. (2013) on a PC with Intel (R) Core(TM) i7-4500U CPU 2.40 GHz processor and 8 GB RAM. All problems are solved by FICO (R) Xpress Optimization Suite in the Mosel modelling language version 7.7 without any solver cuts. The default time is set to 600 seconds for each test instance.

The planning horizons are 25, 50 and 75 periods. The demands are drawn randomly from a normal distribution with mean, $\mu = 100$, and standard deviation, $\sigma = 50$. We also assume the returns parameter is normally distributed with three different parameter settings: low return($\mu = 10, \sigma = 5$), medium return ($\mu = 50, \sigma = 25$), and high return ($\mu = 90, \sigma = 45$). This gives us nine possible parameter combination settings, where each is replicated 10 times, resulting in 90 different data sets. We assume that the demands and the returns values are nonnegative and the cost parameters are time-invariant. The setup costs for both remanufacturing and manufacturing take the values of 125, 250, 500 and 1000. Then, the holding costs for both product returns and serviceable products are equal to 1 for all test instances. Lastly, the production costs for both remanufacturing and manufacturing are assumed to be zero.

The detailed results of lower bounds for ELSRs and ELSRj are provided in Appendix C.1 - C.9 and Appendix D.1 - D.9, respectively. In Appendix C.1 - C.9:

- The first column indicates the variation of setup costs, SC, and is followed by the number of iterations.
- The next three main columns demonstrate the lower bounds, LB, at the root node solution of the branch-and-bound tree and the upper bounds, UB, obtained from the original formulation (Teunter et al., 2006; Retel Helmrich

et al., 2013), $(\ell, S)-like$ inequalities, (ℓ, S, WW) inequalities (Retel Helmrich et al., 2013), facility location (FL) reformulation, multi-commodity (MC) reformulation and SP reformulation, terminated within 600 seconds. Note that we also rerun computational experiments on the (ℓ, S, WW) inequalities proposed by Retel Helmrich et al. (2013) in order to avoid bias in a computational comparison.

In this chapter, the computational results for both ELSRs and ELSRj problems are divided into two main parts:

(i) The pairwise comparisons of lower bounds in terms of average improvement (in percentage) are summarized in Table 3.4.1 for separate setups and joint setups cases. The first column represents return variability, namely low, medium and high returns. This is followed by the number of periods, n, and the variation of setup costs. For simplicity, we use (ℓ, S) bound notation to represent $(\ell, S)-like$ bound. The "(ℓ, S) vs (ℓ, S, WW)" indicates that the (ℓ, S) bound improves the (ℓ, S, WW) bound to (ℓ, S) bound or that there is simply average improvement of lower bounds from (ℓ, S, WW) bound to (ℓ, S) bound. The average improvement (%) can be calculated as:

$$\text{AI (\%)}= \frac{(\ell, S) \text{ bound} - (\ell, S, WW) \text{ bound}}{(\ell, S) \text{ bound}} \times 100$$

for each test instance regardless of the solution optimality. Also note that the lower bounds provided by FL are identical to those of MC and SP for both ELSRs and ELSRj problems. These relationships were proven theoretically in the previous section. Further, the interpretation of "FL vs (ℓ, S)" is similar to "(ℓ, S) vs (ℓ, S, WW)".

(ii) The performance analysis of all formulations is presented in Table 3.4.2 for separate setups and in Tables 3.4.3 - 3.4.4 for joint setups. We examine the linear programming relaxation gap (%), also known as the duality gap, for each test instance:

$$\text{LP gap (\%)}= \frac{\text{Best UB} - \text{Best LB}}{\text{Best UB}} \times 100$$

where, Best LB and Best UB are the best values found for the lower bound (LP relaxation) and the upper bound, respectively, when the enumeration is terminated at a preassigned time. The number of times the LP relaxation found the integer solutions is also provided. Furthermore, we also include the number of optimal solutions found (out of ten replications) that could be solved to optimality within the preassigned time of 600 seconds. Next, the average solution times of the MIPs are presented. If a test instance could not be solved to optimality within the given time, the solution time is counted as 600 seconds. Lastly, the best performance among all formulations is highlighted in bold-face.

The results of the pairwise comparisons of lower bounds for both ELSR problems are presented in Table 3.4.1. We first discuss the average improvement (AI) from the (ℓ, S, WW) bound provided by Retel Helmrich et al. (2013) to the $(\ell, S) - like$ bound. Generally, the AI from the (ℓ, S, WW) bound to the $(\ell, S) - like$ bound deteriorates when the number of periods or the return variability is increased for both problems. Specifically, if the amount of returns is large, then the remanufacturing operation dominate production to satisfy the demand. Therefore, the family of valid inequalities involving returns introduced by Retel Helmrich et al. (2013) and also considered in our study becomes more effective in improving the lower bounds. As a result, (ℓ, S, WW) bound improves slightly by our $(\ell, S) - like$ bound, which is only 6% maximum on average. Compared to a low return scenario, the $(\ell, S) - like$ bound improves the (ℓ, S, WW) bound significantly, up to 22% on average. One of the reasons we obtain a large AI is because the second inequality (3.12) involving demands introduced by Retel Helmrich et al. (2013) does not include production during the first period, causing demand in this period to not be satisfied. Normally, with low returns, manufacturing will dominate production over remanufacturing. This causes a valid inequality involving returns to become less effective.

Next, we examine the average improvement from the $(\ell, S) - like$ bound to FL bound for ELSRs. The results show that the $(\ell, S) - like$ bound in general improves notably by the FL bound in the case of medium returns with large setup costs, up to approximately 2% for all periods. In the case of low and high returns, the FL bound shows less significant average improvement over the $(\ell, S) - like$ bound, which shows less than 1% improvement. This indicates that our proposed $(\ell, S) - like$ inequalities provide better lower bounds since we found some identical bounds with FL in some data instances.

In short, the lower bound provided by our $(\ell, S) - like$ bound is at least as strong as the FL bound. Furthermore, for each period and scenario, the AI from the $(\ell, S) - like$ bound to the FL bound in general increases gradually as we increase the amount of setup costs. In regard to the joint setups case, we review only the pairwise comparisons of lower bounds between (ℓ, S, WW) bound and the $(\ell, S) - like$ bound. We do not present the results of average improvement of the FL bound over the $(\ell, S) - like$ bound since the lower bounds of all the proposed formulations are identical.

Firstly, we review the performance analysis of all formulations for the ELSRs problem, which are presented in Table 3.4.2. We note that most test instances in the case of medium and high return scenarios with large periods 50 and 75 could not be solved to optimality within the default time of 600 seconds. Overall, three equivalent reformulation techniques have better LP relaxations in the sense that they have smaller LP gaps compared to $(\ell, S) - like$ inequalities, at most a 3% difference, followed by (ℓ, S, WW) inequalities and the original formulation for each ten replications. Although all reformulation techniques have identical lower bounds, SP has slightly larger LP gaps, but this difference is insignificant.

Interestingly, we find the number of integer solutions provided by $(\ell, S) - like$

Table 3.4.1: Mean percentage improvement of lower bounds for ELSR problems

Scenario	n	Separate setups cost										Joint setups cost				
		(l,s) vs (l,s,ww)					FL vs (l,s)					(l,s) vs (l,s,ww)				
		125	250	500	1000	Avg.	125	250	500	1000	Avg.	125	250	500	1000	Avg.
Low return	25	26.55	23.14	21.57	20.65	22.98	0.02 (6)	0.06 (3)	0.17 (1)	0.16 (2)	0.10	4.28	5.43	6.90	9.24	6.46
	50	25.72	22.20	21.33	19.73	22.25	0.04 (3)	0.19	0.22	0.28	0.18	2.06	2.62	3.59	4.74	3.25
	75	25.34	22.26	20.37	19.34	21.83	0.03 (2)	0.12	0.20	0.28	0.16	1.39	1.81	2.37	3.16	2.19
Medium return	25	16.77	13.99	13.62	13.01	14.35	0.21	0.85	1.28	1.56	0.98	1.75	3.07	4.94	6.77	4.13
	50	16.10	13.17	11.46	10.90	12.91	0.46	1.13	1.70	2.25	1.39	0.84	1.62	2.43	3.44	2.08
	75	16.23	12.72	11.38	10.42	12.68	0.28	1.04	1.70	2.13	1.29	0.92	1.33	1.85	2.56	1.67
High return	25	3.62	4.09	5.25	7.53	5.12	0.03 (7)	0.27 (2)	0.67 (1)	0.76	0.43	0.74	1.54	2.59	4.96	2.46
	50	4.01	3.59	3.93	4.78	4.08	0.06 (3)	0.37 (1)	0.77	0.86	0.52	0.37	0.95	1.75	2.80	1.47
	75	3.55	3.15	3.11	3.32	3.28	0.04 (1)	0.37	0.68	0.97	0.51	0.10	0.46	0.86	1.43	0.71

*(ℓ, S) bound indicates $(\ell, S) - like$ bound.
*(ℓ, S, WW) bound indicates (ℓ, S, WW) bound provided by Retel Helmrich et al. (2013).
* ( ) indicates the number of equivalent lower bounds between FL and $(\ell, S) - like$- out of 10 iterations.

inequalities and the reformulation techniques in the case of low returns, a short period and high setup costs. When we look at the computation time, all proposed formulations require a longer time to find an optimal solution if we consider long planning periods and large number of returns. This means that all formulations can solve the test instances faster when the return rate is low, where remanufacturing becomes almost negligible and the planning horizon is short since the problem size is small (i.e., $n = 25$), especially for SP and the $(\ell, S) - like$ inequalities.

We can conclude that (1) the $(\ell, S) - like$ bound is better than the (ℓ, S, WW) inequalities provided by Retel Helmrich et al. (2013); and (2) our proposed reformulation techniques considerably outperform the $(\ell, S) - like$ inequalities, the (ℓ, S, WW) inequalities and the original formulations for ELSRs problem in terms of stronger and lower bounds; the smallest LP gaps and computation times; and the highest number (out of ten replications) of optimal solutions found. Figure 3.4.1 - 3.4.3 illustrates an easy-to-read graphical representation to visualize some important results. Note that we exclude the computational results for the original formulation as it is known to be inefficient.

Regarding the performance analysis of all formulations for the joint setups case, $(\ell, S) - like$ inequalities, FL, MC and SP are equivalent and have the best LP relaxation since they have the smallest percentage of LP gap, as shown in Tables 3.4.3 and 3.4.4. Almost (all) of the test instances are solved to optimality and are often found to be an integer. This is due to the fact that the setup variables considered in the separate setups case are twice than in the joint setups case; therefore, we would expect that optimal solutions can be possibly obtained within an allocated time. These integer solutions are mostly found when the low returns scenario is considered. We observe that the number of integer solutions decreases to more than 20% in the case of medium returns and depletes to zero when the amount of returns gets larger. Regarding average solution time, all proposed formulations obtain the optimal solution very quickly, even for the longer period of 75, which is less than 105 seconds. This shows that our proposed formulations are computationally efficient for solving the test instances in a very short time. To help with understanding of the results obtained, we illustrate the graphical representation of the solution times (s) of all formulations in Figure 3.4.4.

Finally, in contrast to the computational results of the separate setups case, we conclude that our efficient separation algorithm of $(\ell, S) - like$ inequalities for the ELSRj problem, which has less variables, demonstrates better performance than reformulation techniques in terms of saving computation time.

In conclusion, the $(\ell, S) - like$ bound is better than the (ℓ, S, WW) bound by Retel Helmrich et al. (2013), and all reformulation techniques have identical lower bounds for both ELSRs and ELSRj problems. Furthermore, for the ELSRs problem, the lower bounds provided by $(\ell, S) - like$ inequalities are at least as strong as reformulation techniques. Meanwhile, the $(\ell, S) - like$ bound is equivalent to reformulation techniques in the case of joint setups. Comparing the performance level of all formulations for ELSRs, reformulation techniques provide better lower

Table 3.4.2: [**Separate setups**] Performance analysis of all formulations

Total average	Scenario	Setup cost	n = 25 O	n = 25 (l,s)	n = 25 (l,s,ww)	n = 25 FL	n = 25 MC	n = 25 SP	n = 50 O	n = 50 (l,s)	n = 50 (l,s,ww)	n = 50 FL	n = 50 MC	n = 50 SP	n = 75 O	n = 75 (l,s)	n = 75 (l,s,ww)	n = 75 FL	n = 75 MC	n = 75 SP
LP gap (%)	Low return	125	85.84	1.01	21.77	**0.99**	**0.99**	**0.99**	92.23	1.71	22.37	**1.67**	**1.67**	**1.67**	94.55	1.36	22.55	**1.34**	**1.34**	**1.34**
		250	82.35	0.93	19.54	**0.88**	**0.88**	**0.88**	90.42	1.20	20.00	**1.02**	**1.02**	**1.02**	93.49	1.31	20.64	**1.19**	**1.19**	**1.19**
		500	77.33	1.01	18.57	**0.84**	**0.84**	**0.84**	87.99	1.27	19.41	**1.06**	**1.06**	**1.06**	91.84	1.17	19.99	**0.97**	**0.97**	**0.97**
		1000	71.57	0.30	17.36	**0.14**	**0.14**	**0.14**	84.55	0.94	17.53	**0.67**	**0.67**	**0.67**	89.46	1.02	19.83	**0.74**	**0.74**	**0.74**
	Medium return	125	82.81	6.12	19.59	**5.92**	**5.92**	**5.92**	89.12	7.60	21.30	6.98	7.02	**6.96**	89.81	8.81	22.76	7.93	**7.78**	7.99
		250	82.04	6.28	17.77	**5.48**	**5.48**	**5.48**	89.20	7.60	19.19	**6.31**	6.32	6.39	91.02	8.00	20.02	6.72	**6.62**	6.82
		500	79.19	5.40	16.72	**4.20**	**4.20**	**4.20**	87.99	6.51	16.70	**4.70**	**4.70**	**4.70**	90.77	7.67	19.00	5.35	**5.29**	5.44
		1000	74.64	5.11	16.03	**3.63**	**3.63**	**3.63**	85.47	5.92	15.35	**3.79**	**3.79**	**3.79**	89.19	7.18	17.63	**4.41**	4.46	4.54
	High return	125	46.54	9.61	12.73	**9.58**	**9.58**	**9.58**	46.86	7.74	11.35	7.35	**7.34**	7.36	49.42	8.49	12.65	8.53	**7.90**	7.96
		250	56.14	9.27	12.83	**9.03**	**9.03**	**9.03**	57.89	8.54	11.66	**7.86**	7.88	7.94	59.66	9.18	12.33	8.46	**8.42**	8.39
		500	62.64	8.32	12.88	**7.71**	**7.71**	**7.71**	66.16	8.85	12.19	**7.84**	7.89	7.86	66.00	9.92	13.31	8.72	**8.71**	9.24
		1000	64.94	6.83	13.34	**6.12**	**6.12**	**6.12**	71.65	7.16	11.34	6.25	6.29	**6.16**	74.38	8.28	11.53	**6.89**	6.92	7.15
# of optimal solution	Low return	125	0	**10**	**10**	**10**	**10**	**10**	0	**10**	0	**10**	**10**	**10**	0	**10**	0	**10**	**10**	**10**
		250	0	**10**	**10**	**10**	**10**	**10**	0	**10**	0	**10**	**10**	**10**	0	**10**	0	**10**	**10**	**10**
		500	8	**10**	**10**	**10**	**10**	**10**	0	**10**	0	**10**	**10**	**10**	0	**10**	0	**10**	**10**	**10**
		1000	0	**10** **(1)**	**10**	**10** **(2)**	**10** **(2)**	**10** **(2)**	0	**10**	1	**10**	**10**	**10**	0	**10**	0	**10**	**10**	**10**
	Medium return	125	0	**10**	**10**	**10**	**10**	**10**	0	4	0	**5**	4	**5**	0	0	0	0	0	0
		250	0	**10**	**10**	**10**	**10**	**10**	0	1	0	**3**	**3**	2	0	0	0	0	0	0
		500	0	**10**	**10**	**10**	**10**	**10**	0	4	0	**10**	9	9	0	0	0	0	0	0
		1000	1	**10**	**10**	**10**	**10**	**10**	0	8	5	**10**	**10**	**10**	0	0	0	**1**	**1**	**1**
	High return	125	9	**10**	**10**	**10**	**10**	**10**	0	5	4	**6**	5	5	0	0	0	0	0	0
		250	8	**10**	**10**	**10**	**10**	**10**	0	4	4	4	4	4	0	**1**	0	**1**	**1**	0
		500	**10**	**10**	**10**	**10**	**10**	**10**	0	4	4	**5**	**5**	4	0	0	0	0	0	0
		1000	**10**	**10**	**10**	**10**	**10**	**10**	0	6	**9**	8	8	8	0	**1**	0	0	0	0
Solution times (s)	Low return	125	600	0.09	1.44	0.08	0.11	**0.07**	600	**0.43**	600	0.82	0.86	0.62	600	**0.84**	600	2.22	2.52	1.61
		250	600	0.10	0.76	0.12	**0.09**	0.10	600	0.70	600	0.85	1.03	**0.67**	600	3.52	600	4.54	5.86	**2.8**
		500	302.8	0.12	0.54	0.09	0.09	**0.08**	600	1.30	600	1.17	0.92	**0.82**	600	6.42	600	**3.5**	5.52	4.87
		1000	0.85	0.10	0.24	0.09	**0.08**	0.10	600	1.46	579.31	0.74	0.70	**0.44**	600	10.5	600	2.61	3.02	**1.69**
	Medium return	125	600	**0.16**	1.2	0.32	0.36	0.21	600	391.8	600	351.2	372.5	**342**	600	600	600	600	600	600
		250	600	0.32	0.9	0.43	0.45	**0.31**	600	563.1	600	470.5	**462**	514	600	600	600	600	600	600
		500	600	0.34	0.58	0.38	0.41	**0.26**	600	399.7	600	**72.69**	98.58	131.8	600	600	600	600	600	600
		1000	563.7	0.33	0.42	0.36	0.36	**0.25**	600	172.7	396.8	24.58	30.74	**21.1**	600	600	600	**557.1**	571.8	600
	High return	125	132.8	**0.48**	1.1	0.75	0.73	0.71	600	321.3	389	**279.1**	318.9	318.3	600	600	600	600	600	600
		250	204.4	**0.69**	0.71	0.79	0.80	0.84	600	**367.5**	392.1	380.8	377.3	378.2	600	589.1	600	577.2	**556**	600
		500	56.5	**0.32**	0.36	0.56	0.54	0.41	600	371.6	391.8	339.7	**336.7**	382.3	600	600	600	600	600	600
		1000	10.1	**0.26**	0.29	0.50	0.51	0.36	600	264.4	149.5	179.9	**148.4**	189.5	600	**565.4**	600	600	600	600

*(ℓ, S) bound indicates $(\ell, S) - like$ bound.
*(ℓ, S, WW) bound indicates (ℓ, S, WW) bound provided by Retel Helmrich et al. (2013).
* ( ) indicates the number of integer solutions by FL and $(\ell, S) - like$- out of 10 iterations.

Table 3.4.3: [**Joint setups**] Performance analysis of all formulations

Total average	Scenario	Setup cost	$n = 25$			$n = 50$			$n = 75$		
			O	(l,s)*	(l,s,ww)	O	(l,s)*	(l,s,ww)	O	(l,s)*	(l,s,ww)
LP gap (%)	Low return	125	81.52	**0**	4.11	88.79	**0.01**	2.03	92.54	**0**	1.37
		250	77.73	**0**	5.15	86.56	**0**	2.55	91.25	**0**	1.78
		500	72.85	**0**	6.45	83.51	**0**	3.46	89.22	**0**	2.32
		1000	67.09	**0**	8.46	79.63	**0**	4.52	86.59	**0**	3.07
	Medium return	125	77.19	**0.27**	1.97	84.61	**1.03**	1.85	85.58	**1.04**	1.93
		250	76.93	**0.26**	3.21	84.87	**0.48**	2.06	87.47	**0.42**	1.73
		500	74.17	**0.03**	4.73	83.33	**0.11**	2.48	87.22	**0.22**	2.03
		1000	69.89	**0.04**	6.38	80.51	**0.01**	3.33	85.83	**0.08**	2.58
	High return	125	42.08	**4.53**	5.22	41.73	**3.26**	3.62	43.19	**3.42**	3.52
		250	52.33	**3.64**	5.09	52.65	**3.51**	4.41	54.98	**3.32**	3.76
		500	59.47	**3.16**	5.58	61.14	**3.07**	4.75	64.35	**3.65**	4.48
		1000	61.92	**2.04**	6.66	66.33	**1.95**	4.62	70.70	**2.31**	3.69
# of optimal solution	Low return	125	**10**	**10**	**10**	0	**10**	**10**	0	**10**	**10**
		250	**10**	**10**	**10**	0	**10**	**10**	0	**10**	**10**
		500	**10**	**10**	**10**	0	**10**	**10**	0	**10**	**10**
		1000	**10**	**10**	**10**	0	**10**	**10**	0	**10**	**10**
	Medium return	125	**10**	**10**	**10**	0	**10**	**10**	0	**10**	**10**
		250	**10**	**10**	**10**	0	**10**	**10**	0	**10**	**10**
		500	**10**	**10**	**10**	0	**10**	**10**	0	**10**	**10**
		1000	**10**	**10**	**10**	0	**10**	**10**	0	**10**	**10**
	High return	125	**10**	**10**	**10**	2	**10**	**10**	0	**10**	**10**
		250	**10**	**10**	**10**	1	**10**	**10**	0	**10**	**10**
		500	**10**	**10**	**10**	0	**10**	**10**	0	**9**	**9**
		1000	**10**	**10**	**10**	0	**10**	**10**	0	**10**	**10**
# of integer solutions	Low return	125	0	**10**	0	0	**9**	0	0	**10**	0
		250	0	**10**	0	0	**10**	0	0	**10**	0
		500	0	**10**	0	0	**10**	0	0	**10**	0
		1000	0	**10**	0	0	**10**	0	0	**10**	0
	Medium return	125	0	**1**	0	0	0	0	0	0	0
		250	0	**3**	0	0	**1**	0	0	0	0
		500	0	**8**	0	0	**6**	0	0	0	0
		1000	0	**8**	0	0	**7**	0	0	**5**	0
	High return	125	0	0	0	0	0	0	0	0	0
		250	0	0	0	0	0	0	0	0	0
		500	0	0	0	0	0	0	0	0	0
		1000	0	0	0	0	0	0	0	0	0

*(ℓ, S) bound indicates $(\ell, S) - like$ bound.
*(ℓ, S, WW) bound indicates (ℓ, S, WW) bound provided by Retel Helmrich et al. (2013).
*$(\ell, S)^*$ represents all formulation techniques.

Table 3.4.4: [**Joint setups**] Performance analysis of all formulations (cont.)

Total average	Scenario	Setup cost	$n = 25$						$n = 50$						$n = 75$					
			O	(l,s)	(l,s,ww)	FL	MC	SP	O	(l,s)	(l,s,ww)	FL	MC	SP	O	(l,s)	(l,s,ww)	FL	MC	SP
Solution times (s)	Low return	125	68.30	**0**	**0**	0.01	0.02	**0**	600	**0.01**	0.11	0.29	0.07	0.29	600	**0**	0.51	0.51	0.56	0.20
		250	48.88	**0**	**0**	0.01	0.02	**0**	600	**0.01**	0.10	0.27	0.09	0.28	600	**0**	0.51	0.58	0.60	0.20
		500	6.77	**0**	**0**	0.02	0.03	**0**	600	**0**	0.10	0.25	0.06	0.29	600	**0.02**	0.50	0.63	0.53	0.20
		1000	0.65	**0**	**0**	0.01	0.05	**0**	600	**0.04**	0.10	0.26	0.06	0.27	600	**0.10**	0.50	0.57	0.68	0.20
	Medium return	125	39	0.01	0.04	0.08	0.07	**0**	600	**0.07**	0.20	0.38	0.42	0.42	600	**0.13**	0.87	0.93	1.07	0.34
		250	92.7	**0.02**	0.04	0.08	0.09	**0.02**	600	**0.11**	0.22	0.36	0.41	0.41	600	**0.23**	1.21	0.82	0.89	0.31
		500	41.9	**0**	0.01	0.05	0.06	**0**	600	**0.06**	0.17	0.31	0.32	0.32	600	**0.24**	1.02	0.84	0.67	0.33
		1000	3.8	**0.01**	0.01	0.06	0.06	0.02	600	**0.08**	0.13	0.27	0.29	0.26	600	**0.23**	0.76	0.78	0.89	0.27
	High return	125	0.17	**0.06**	0.09	0.51	0.20	0.12	496.6	**1.02**	1.44	3.55	1.07	6.24	600	**5.29**	8.22	24.61	38.76	13.64
		250	0.38	**0.05**	0.10	0.58	0.18	0.10	542.1	**1.11**	1.36	4.12	0.89	5.48	600	**3.49**	10.86	35.42	29.99	12.84
		500	0.67	**0.09**	0.10	0.63	0.21	0.10	600	**0.68**	1.50	3.25	0.67	6.03	600	75.14	**74.51**	102.76	104.66	79.41
		1000	0.42	**0.08**	0.09	0.57	0.21	0.07	600	**0.48**	0.87	1.63	0.89	2.12	600	**9.99**	12.76	44.36	40.80	12.39

*(ℓ, S) bound indicates $(\ell, S) - like$ bound.
*(ℓ, S, WW) bound indicates (ℓ, S, WW) bound provided by Retel Helmrich et al. (2013).

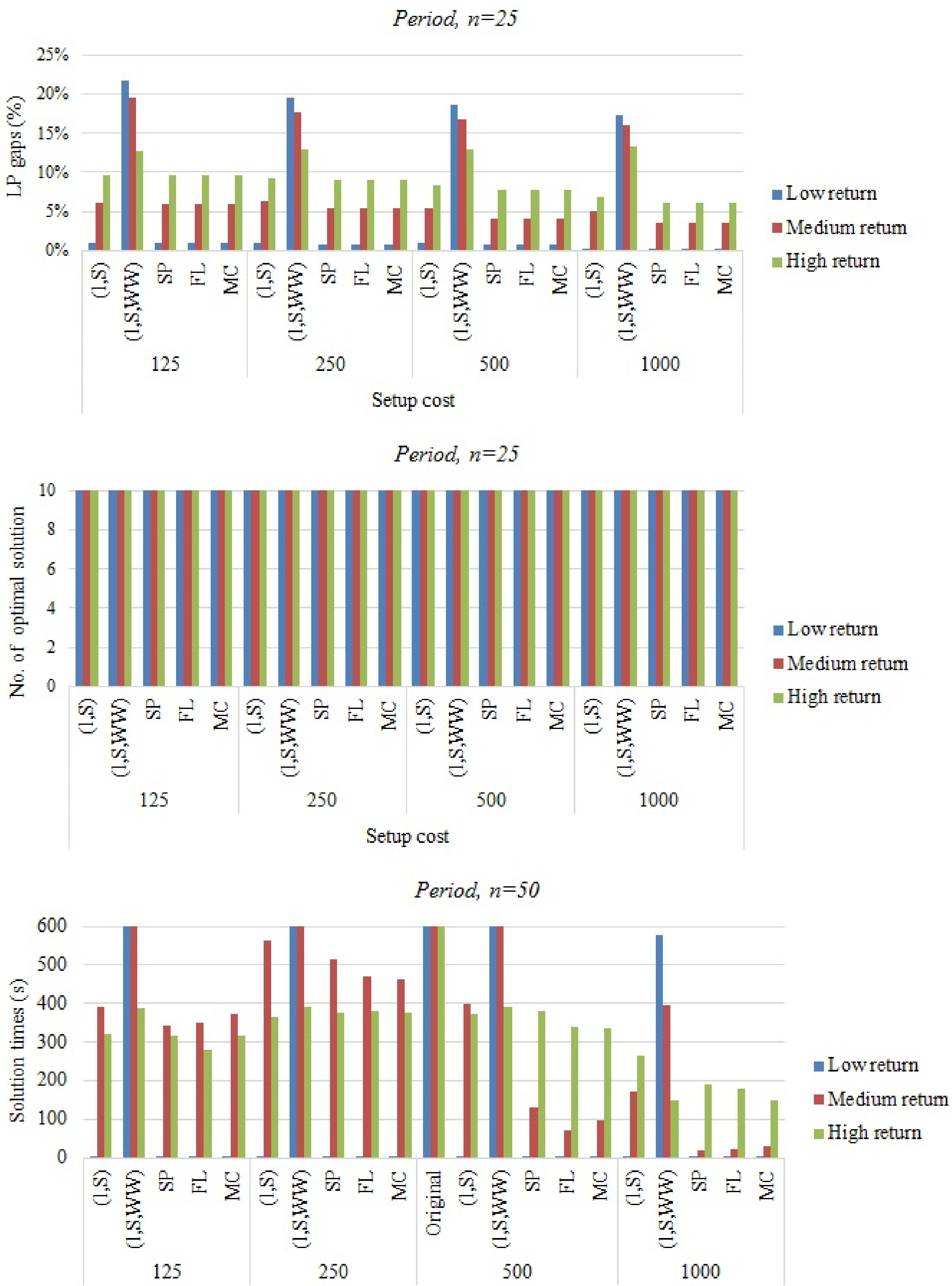

Figure 3.4.1: Separate setups, 25 periods

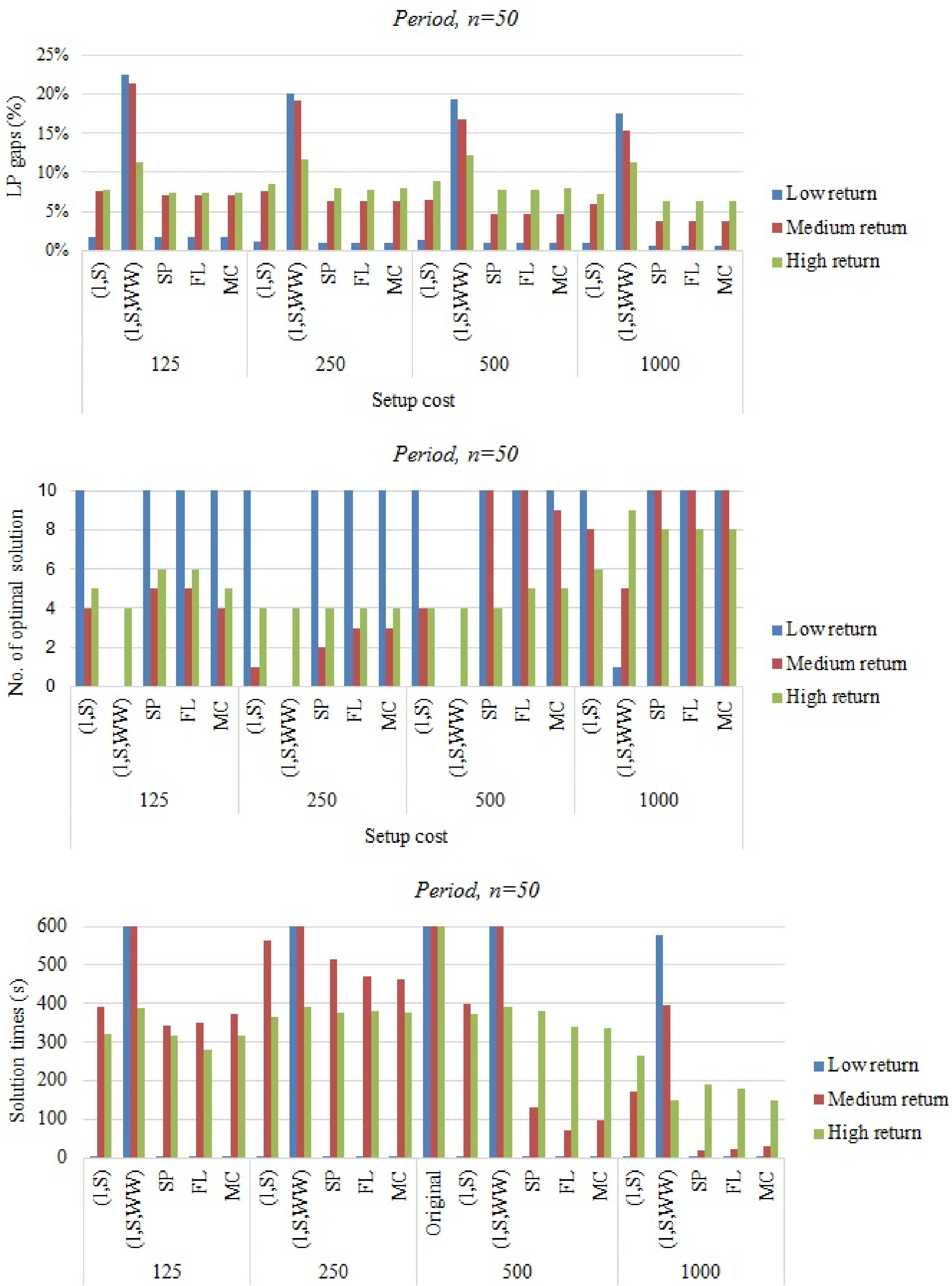

Figure 3.4.2: Separate setups, 50 periods

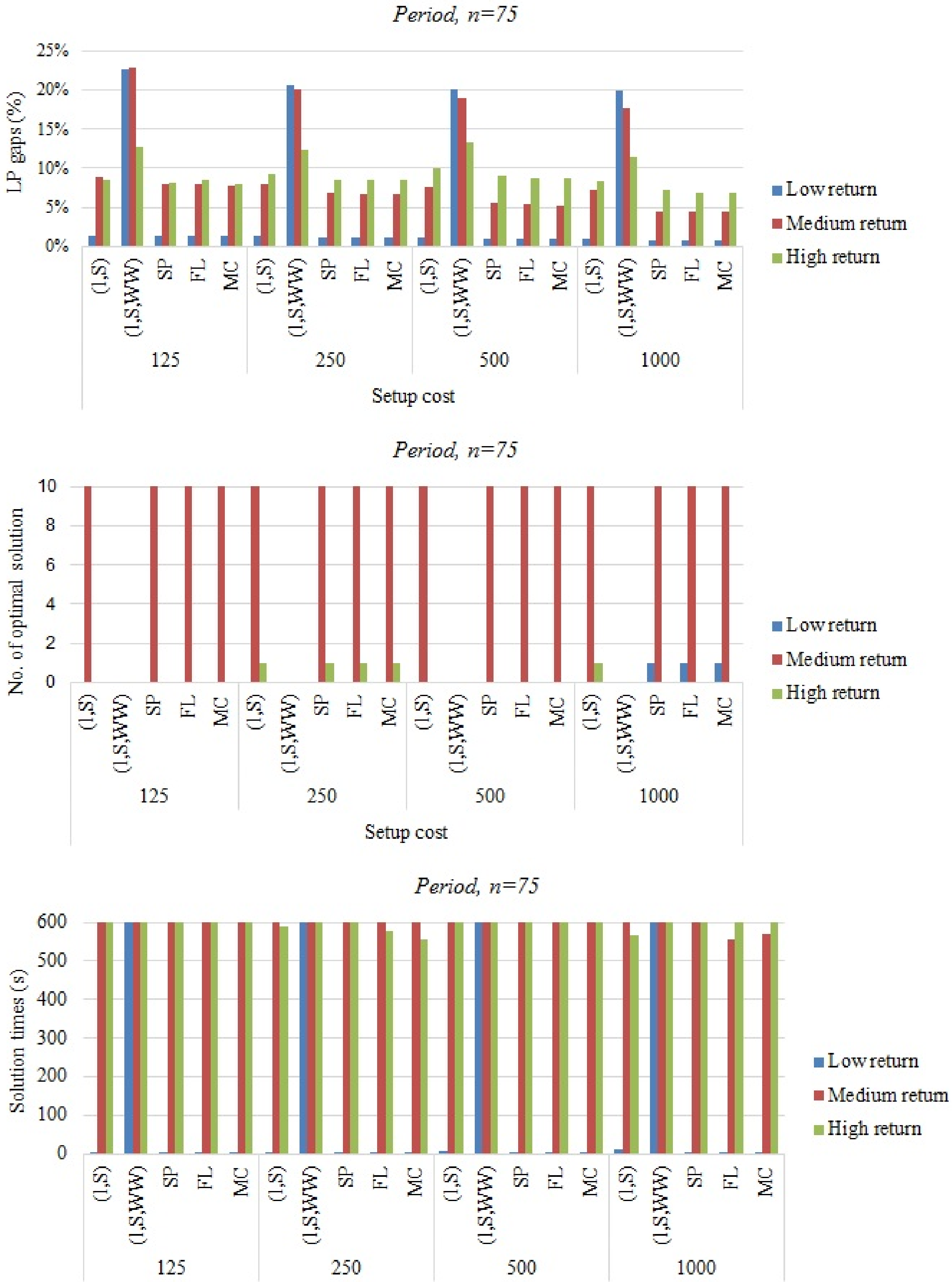

Figure 3.4.3: Separate setups, 75 periods

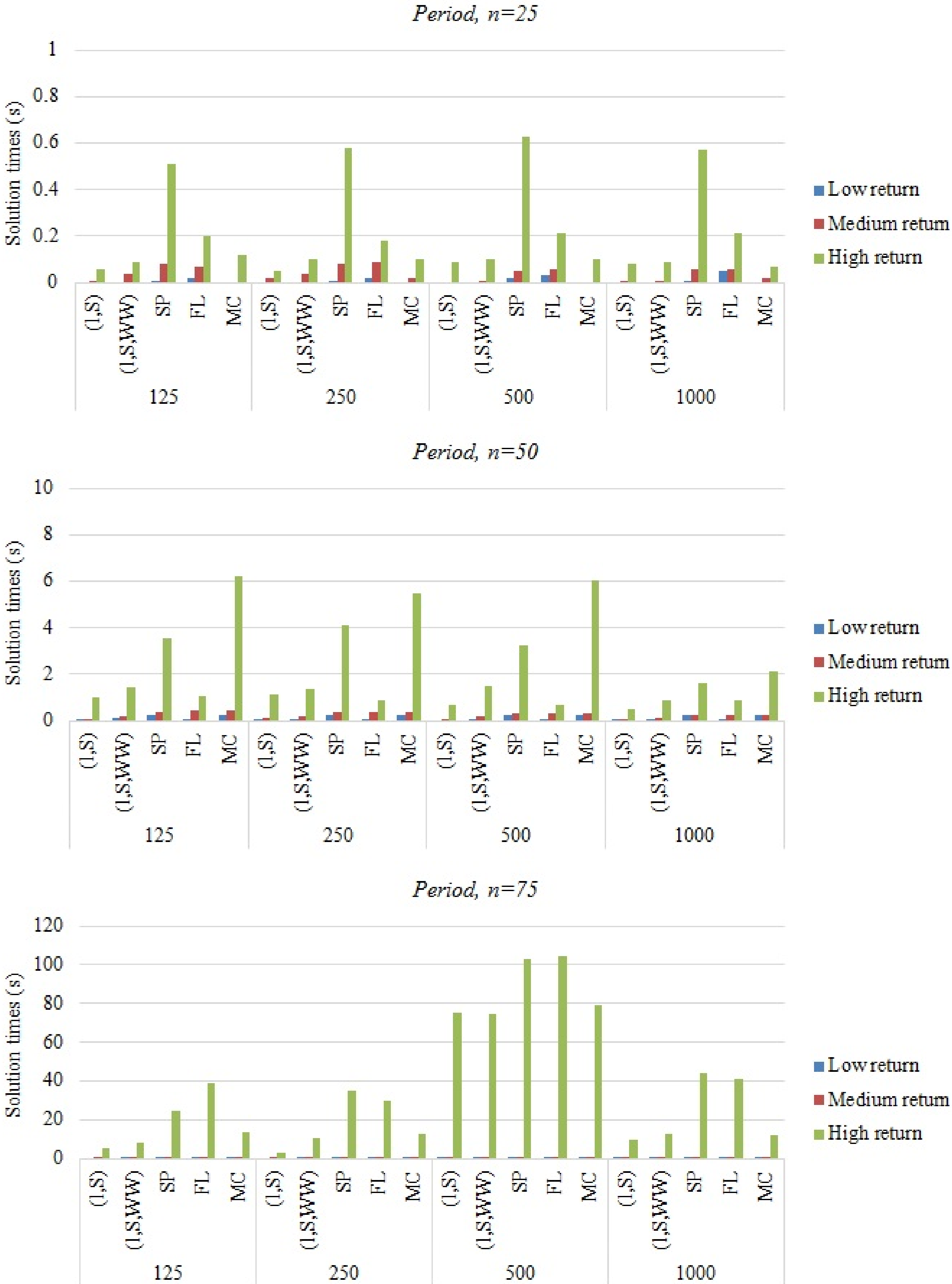

Figure 3.4.4: Joint setups, solution times (s) for all periods

bounds, the best LP gaps, a higher number of optimal solutions found (out of ten replications) and the shortest solution time in most tested instances. Meanwhile, for the ELSRj problem, $(\ell, S) - like$ inequalities show more promising results than other formulations in terms of solving instances faster than other formulations.

3.5 Concluding Remarks

In this section, we evaluate different mathematical approaches such as $(\ell, S) - like$ inequalities, FL reformulation, MC reformulation and SP reformulation to obtain lower bounds for the economic lot-sizing problem for remanufacturing and separate setups (ELSRs) and joint setups (ELSRj) problems. The findings show that the lower bounds provided by $(\ell, S) - like$ inequalities are better than (ℓ, S, WW) inequalities by Retel Helmrich et al. (2013). Further, all reformulation techniques, FL, MC and SP provide identical lower bounds for both problems, which is proven theoretically and observed from the computational results. The fact that (ℓ, S) inequalities and all reformulation techniques provide equivalent lower bounds in the classical single-item uncapacitated lot-sizing problem (see Barany et al. (1984a), Rardin and Wolsey (1993), Krarup and Bilde (1977) and Eppen and Martin (1987)), only applies to the case of joint setups as only a single setup is considered in the formulation. The ELSRj problem, which more closely resembles the structure of the classical uncapacitated lot-sizing problem is efficient for quickly solving the tested data instances. However, in the case of separate setups, the lower bounds obtained by all reformulation techniques slightly outperform $(\ell, S) - like$ inequalities in terms of lower bounds, LP gaps, the number of optimal solutions found and computation times in almost all instances tested.

Chapter 4

Valid Inequalities for Economic Lot-Sizing Problems with Remanufacturing: Separate Setups Case

4.1 Introduction

This chapter investigates the polyhedral structure of a mixed integer set arising from the feasible set of original formulation economic lot-sizing solutions with remanufacturing and separate setups, which considers two knapsack sets simultaneously based on the well-known single node fixed-charge network (SNFCN). Before explaining this further, we first define this mixed integer set formally in the feasible region:

$$X^s = \{(x^r, x^m, y^r, y^m) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times \mathbb{B}^n \times \mathbb{B}^n | \sum_{t \in N} x^r_t \leq R,\ \sum_{t \in N}(x^r_t + x^m_t) \geq D,$$
$$x^r_t \leq m^r_t y^r_t,\ x^m_t \leq m^m_t y^m_t,\ \forall t \in N\}, \quad (4.1)$$

where $R = \sum_{t=1}^n r_t$ denotes the total amount of returns and $D = \sum_{t=1}^n d_t$ is the total amount of demands. Note that the big-M constraints can be structured based on the initial formulation, using $m^r_t = \min\{r_{1,t}, d_{t,n}\}$ and $m^m_t = d_{t,n}$ for any $t \in N$.

In order to investigate the polyhedral set $conv(X^s)$, we first refer to Padberg et al. (1985) and the SNFCN set defined as follows:

$$X_\nabla = \{(x, y) \in \mathbb{R}^n_+ \times \mathbb{B}^n | \sum_{t \in N} x_t \nabla d,\ x_t \leq m_t y_t,\ \forall t \in N\}, \quad (4.2)$$

where $\nabla \in \{\leq, =, \geq\}$. Note that $X_\leq$ and $X_\geq$ are relaxations of set X^s. Firstly, they derive a class of "surrogate knapsack" facets for $conv(X_\geq)$. The "surrogate knapsack" problem as follows:

$$\sum_{t\in N} m_t y_t \geq d,\ y_t \in \{0,1\},\ \forall t \in N$$

and the associated knapsack polytope $K = \text{conv}\{y \in \mathbb{R}^n | \sum_{t\in N} m_t y_t \geq d,\ y_t \in \{0,1\},\ \forall t \in N\}$ is a relaxation of $conv(X_{\geq})$ and $conv(X_{=})$. They show that almost all facets of K are facets for $conv(X_{\geq})$. Secondly, a class of "flow cover" facets for $conv(X_{=})$ is described from a large class of valid inequalities for $conv(X_{\leq})$ is stated as follows.

Proposition 13 (**Flow cover inequalities (Padberg et al., 1985)**). *Let S be a cover such that $\sum_{t\in S} m_t = d + \lambda$, where $\lambda > 0$ and $\overline{m} = \max_{t\in S} m_t > \lambda$, then the simple flow cover inequalities*

$$\sum_{t\in S} x_t - \sum_{t\in S}(m_t - \lambda)^+ y_t \leq d - \sum_{t\in S}(m_t - \lambda)^+, \tag{4.3}$$

is valid and defines a facet of $conv(X_{=})$. Moreover, for $L \subseteq N \setminus S$ and $\overline{m_t} = \max\{m_t, \overline{m}\}$, the extended flow cover inequalities defined as

$$\sum_{t\in S\cup L} x_t - \sum_{t\in S}(m_t - \lambda)^+ y_t - \sum_{t\in L}(\overline{m_t} - \lambda) y_t \leq d - \sum_{t\in S}(m_t - \lambda)^+ \tag{4.4}$$

is valid and defines a facet of $conv(X_{\leq})$ if $0 < \overline{m} - \lambda < m_t \leq \overline{m}$ holds for all $t \in L$.

From these two classes of facets, the surrogate knapsack facets for $conv(X_{\geq})$ and the flow cover facets $conv(X_{=})$, they suggest that the surrogate knapsack facets are all valid inequalities for $conv(X_{=})$ and the flow cover facets $conv(X_{=})$ are valid for $conv(X_{\leq})$. Moreover, for every facet of $conv(X_{\geq})$ corresponds to the facet of $conv(X_{\leq})$, and vice versa. Then, they further examine the basic properties relating $conv(X_{\geq})$, $conv(X_{\leq})$ and $conv(X_{=})$ in order to construct facets of $conv(X_{\geq})$ and $conv(X_{\leq})$ from facets of $conv(X_{=})$. They obtain the flow cover facets for $conv(X_{\leq})$ as mentioned earlier in (4.4) and the following flow cover facets for $conv(X_{\geq})$ is given by:

Proposition 14 (**Extended flow cover inequalities (Padberg et al., 1985)**). *Let S be a cover such that $\sum_{t\in S} m_t = d + \lambda$, where $\lambda > 0$ and $\overline{m} = \max_{t\in S} m_t > \lambda$ and for $L \subseteq N \setminus S$ with $0 < \overline{m} - \lambda < m_t \leq \overline{m}$ for all $t \in L$, then*

$$\sum_{t\in N\setminus(S\cup L)} x_t + \sum_{t\in S}(m_t - \lambda)^+ y_t + \sum_{t\in L}(\overline{m} - \lambda) y_t \geq \sum_{t\in S}(m_t - \lambda)^+ \tag{4.5}$$

is valid and defines a facet of $conv(X_{\geq})$.

Their findings provide us better insight into polyhedral study of our mixed integer set X^s. Our main aim for this chapter is to adapt the well-known polyhedral results for the SNFCN set to the set X^s in order to further improve the lower bounds for ELSRs problem. Note that we are not interested to study the relaxations of the set X^s individually as they have been extensively studied by Padberg et al. (1985).

This chapter is structured as follows. In the Section 4.2, we establish basic polyhedral properties of $conv(X^s)$ and present some general results on the trivial facet-defining inequalities. Next, Section 4.3 discusses several known flow cover inequalities for $conv(X^s)$ and identify their facet-defining conditions. In Section 4.4, we discuss an exact separation algorithm for $conv(X^s)$. Then, in Section 4.5, we provide the preliminary computational results to test the effectiveness of these inequalities. Finally, we summarize this chapter in Section 4.6.

4.2 Properties of $conv(X^s)$

In this section, we examine basic properties and some general results on the trivial facet-defining inequalities for $conv(X^s)$. Without loss of generality, we assume the following assumptions for the remainder of the chapter.

(i) $D > R$ because if $D \leq R$, manufacturing is no longer necessary, $x_t^m = 0$ then, $x_t^r > 0$, $\forall t$,

(ii) $\sum\limits_{t \in N \setminus \{k\}} m_t^m \geq D$ for each $k \in N$,

(iii) $D = m_1^m > m_2^m > m_3^m ... > m_n^m > 0$,

(iv) $\sum\limits_{t \in N} m_t^r > R$.

Note that the second assumption allows manufacturing to satisfy all demands even when it is set to zero in any chosen period, the third assumption simply uses the structure of ELSRs used to define big-M parameters and the last assumption ensures that the total amount of returns is used for remanufacturing. We prove the full-dimensionality of $conv(X^s)$ next.

Proposition 15. *$dim(conv(X^s)) = 4n$.*

Proof. First, we note $dim(conv(X^s)) \leq 4n$ since $(x^r, x^m, y^r, y^m) \in \mathbb{R}_+^{4n}$. In order to show $dim(conv(X^s)) \geq 4n$, we present the following $4n + 1$ affinely independent points from $conv(X^s)$:

1. v_0 : Set $x_t^r = 0$ and $y_t^r = 0$ and set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in N$. **(1 point)**
2. $v_1, ..., v_n$: For each $k \in N$, set $x_k^r = 0$ and $y_k^r = 1$; set $x_t^r = 0$ and $y_t^r = 0$, $\forall t \in N \setminus \{k\}$ and set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in N$. **(n points)**
3. $v_{n+1}, ..., v_{2n}$: For each $k \in N$, set $x_k^r = m_k^r$ and $y_k^r = 1$; set $x_t^r = 0$ and $y_t^r = 0$, $\forall t \in N \setminus \{k\}$ and set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in N$. **(n points)**

4. $v_{2n+1}, ..., v_{3n}$: For each $k \in N$, set $x_k^m = 0$ and $y_k^m = 0$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in N \setminus \{k\}$ and set $x_t^r = 0$ and $y_t^r = 0$, $\forall t \in N$. **(*n* points)**

5. $v_{3n+1}, ..., v_{4n}$: For each $k \in N$, set $x_k^m = 0$ and $y_k^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in N \setminus \{k\}$ and set $x_t^r = 0$ and $y_t^r = 0$, $\forall t \in N$. **(*n* points)**

In order to show affine independence, we note that the vectors $v_0, v_1, ..., v_{4n}$ are affinely independent if the vectors $(v_i - v_0)$, $i = 1, ..., 4n$ are linearly independent or equivalently if $\sum_{i=1}^{4n} \lambda_i(v_i - v_0) = \mathbf{0}$ has the single solution $\lambda_1 = \lambda_2 = ... = \lambda_{4n} = 0$. Hence, we have the following system of equations:

$$\begin{cases} \lambda_i + \lambda_{i+n} = 0, & i = 1, ..., n \\ m_{i-n}^r(\lambda_i) = 0, & i = n+1, ..., 2n \\ m_{i-2n}^m(\lambda_i + \lambda_{i+n}) = 0, & i = 2n+1, ..., 3n \\ \lambda_i = 0 & i = 3n+1, ..., 4n \end{cases} \tag{4.6}$$

It is obvious that the only solution for second and fourth set of equations are $\lambda_i = 0$, for $i = n+1, ..., 2n$ and $i = 3n+1, ..., 4n$, and substituting these into other two equations result in $\lambda = 0$. □

Next, we note trivial facet-defining inequalities for $conv(X^s)$ in the following proposition.

Proposition 16. *The trivial facet-defining inequalities for $conv(X^s)$ (and their facet-defining conditions if applicable) are:*

(i) $x_i^r \geq 0$, $\forall i \in N$,

(ii) $x_i^r \leq m_i^r y_i^r$, $\forall i \in N$,

(iii) $x_i^m \leq m_i^m y_i^m$, $\forall i \in N$,

(iv) $y_i^m \leq 1$, $\forall i \in N$,

(v) $y_i^r \leq 1$, $\forall i \in N$,

(vi) $\sum_{t \in N} x_t^r \leq R$ *(when* $\sum_{t \in N \setminus \{k\}} m_t^r > R$ *for each* $k \in N$ *holds),*

(vii) $\sum_{t \in N} x_t^r + \sum_{t \in N} x_t^m \geq D$,

(viii) $x_i^m \geq 0$, $\forall i \in N$ *(when* $\forall k \in N \setminus \{i\}$, $\sum_{t \in N \setminus \{i,k\}} m_t^m + \sum_{t \in N} m_t^r \geq D$ *holds).*

Proof. First, we note $4n$ affinely independent points are necessary when each of these inequalities is enforced as an equation. In order to construct these points, we will use the $4n+1$ affinely independent points presented in the proof of Proposition 15. For (i), (ii), (iii) and (iv), the proof is straightforward, as dropping exactly one of the $4n+1$ points, i.e., v_{n+i}, v_i, v_{3n+i} and v_{2n+i}, respectively, provides us the necessary $4n$ points. For (v), we can set $y_i^r = 1$ and all the points except v_0 will remain valid. For (vi), let $H^r \subset N$ such that $\sum_{t\in H^r} m_t^r > R$, $\exists k \in H^r$ satisfying $\sum_{t\in H^r\setminus\{k\}} m_t^r < R$ and $\exists \ell \notin H^r$ satisfying $m_\ell^r \geq m_t^r$, $\forall t \in H^r$. For all v_i (except for $v_1, ..., v_n$ such that $i \in H^r$), set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r \setminus \{k\}$ and set $x_k^r = R - \sum_{t\in H^r\setminus\{k\}} m_t^r$ and $y_k^r = 1$ (for $v_1, ..., v_n$ such that $i \notin H^r$, in addition to that, set $x_i^r = 0$ and $y_i^r = 1$). For $v_1, ..., v_n$ such that $i \in H^r \setminus \{k\}$, set $x_i^r = 0$ and $y_i^r = 1$; set $x_\ell^r = m_\ell^r$ and $y_\ell^r = 1$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r \setminus \{i, k\}$; set $x_k^r = R - \sum_{t\in H^r\setminus\{k\}} m_t^r$ and $y_k^r = 1$. For (vii), we set $x_1^m = D$ and $y_1^m = 1$ (and also $x_i^m = 0$, $\forall i \in N \setminus \{1\}$) in all points, except setting $x_1^m = D - m_k^r$ and $y_1^m = 1$ in $v_{n+1}, ..., v_{2n}$ and $x_1^m = D - m_k^m$ and $y_1^m = 1$ (and also $x_k^m = m_k^m$ and $y_k^m = 1$) in $v_{2n+2}, ..., v_{3n}$, while removing points v_{2n+1} and v_{3n+1}; therefore, we also add a new point in the form of $x_1^m = 0$ and $y_1^m = 1$, $x_t^m = \left(D/\sum_{t\in N\setminus\{1\}} m_t^m\right) m_t^m$ and $y_t^m = 1$, $\forall t \in N \setminus \{1\}$, and $x_t^r = 0$ and $y_t^r = 0$, $\forall t \in N$. Finally, for (viii), we set $x_i^m = 0$ for all points, remove point v_{3n+i} and for any point in the set $v_{2n+1}, ..., v_{4n}$ such that $x_k^m = 0$ and $\sum_{t\in N\setminus\{i,k\}} m_t^m < D$ holds, we distribute the remaining demand $D - \sum_{t\in N\setminus\{i,k\}} m_t^m$ into remanufacturing in the lexographic order. □

4.3 Polyhedral Analysis of $conv(X^s)$

First, we provide some definitions used throughout the chapter.

Definition 27. The definitions of flow cover inequalities for $conv(X^s)$ are given as follows:

- A set $S^r \subseteq N$ is a cover for R if $\lambda_1 = \sum_{t\in S^r} m_t^r > R$.
- A set $S^m \subseteq N$ is a cover for $D - R$ if $\lambda_2 = \sum_{t\in S^m} m_t^m > (D - R)$.
- For $S^r, S^m \subseteq N$ such that $S^r \cap S^m = \phi$, pair (S^r, S^m) is a cover for D if $\lambda_3 = \sum_{t\in S^r} m_t^r + \sum_{t\in S^m} m_t^m > D$.

We also define $(x)^+ = \max\{x, 0\}$.

It can be readily seen that set $X_\leq$ is a relaxation of set X^s by removing one of the knapsack sets involving demand. Thus, any valid inequality for $X_\leq$ is also valid for X^s. Our theoretical contribution for this chapter comes from the fact that, under certain and general conditions, these inequalities are facet-defining for $conv(X^s)$.

First, we will present two well-known facet-defining inequalities for $conv(X^s)$ in the case of $\leq$. The validity proofs of these inequalities can be referred to Padberg et al. (1985).

Corollary 4 (Flow cover inequalities (Padberg et al., 1985)). *Let $S^r \subseteq N$ be a cover for R, with $\overline{m^r} = \max_{t\in S^r} m_t^r > \lambda_1$. Then, the following inequality (called* ***returns cover inequality****) is valid for X^s.*

$$\sum_{t\in S^r} x_t^r - \sum_{t\in S^r} (m_t^r - \lambda_1)^+ y_t^r \leq R - \sum_{t\in S^r} (m_t^r - \lambda_1)^+ \tag{4.7}$$

Proposition 17. *Let* $S^{r+} = \{t \in S^r | m_t^r - \lambda_1 > 0\}$. *If* $|S^{r+}| \geq 2$, *then* (4.7) *defines a facet of* $conv(X^s)$.

Proof. Suppose we consider i_1 and i_2 are any two members of S^{r+} and let $\epsilon > 0$ is an arbitrary small number. We demonstrate $4n$ affinely independent points, belonging to X^s, that satisfy

$$\sum_{t \in S^r} x_t^r - \sum_{t \in S^{r+}} (m_t^r - \lambda_1)(1 - y_t^r) = R.$$

1. For every $t' \in S^{r+}$, set $x_{t'}^r = 0$ and $y_{t'}^r = 0$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{t'\}$; set $x_1^m = m_1^m$ and $y_1^m = 1$ and set other variables to zero. **(**$|S^{r+}|$ **points)**

2. For every $t' \in S^{r+}$, set $x_{t'}^r = m_{t'}^r - \lambda_1$ and $y_{t'}^r = 1$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{t'\}$; set $x_1^m = m_1^m$ and $y_1^m = 1$ and set other variables to zero. **(**$|S^{r+}|$ **points)**

3. For every $t' \in S^r \setminus S^{r+}$, set $x_{t'}^r = 0$ and $y_{t'}^r = 0$; set $x_{i_1}^r = m_{i_1}^r - \lambda_1 + m_{t'}^r$ and $y_{i_1}^r = 1$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{t', i_1\}$; set $x_1^m = m_1^m$ and $y_1^m = 1$ and set other variables to zero. **(**$|S^r \setminus S^{r+}|$ **points)**

4. For every $t' \in S^r \setminus S^{r+}$, set $x_{t'}^r = 0$ and $y_{t'}^r = 1$; set $x_{i_2}^r = m_{i_2}^r - \lambda_1 + m_{t'}^r$ and $y_{i_2}^r = 1$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{t', i_2\}$; set $x_1^m = m_1^m$ and $y_1^m = 1$ and set other variables to zero. **(**$|S^r \setminus S^{r+}|$ **points)**

5. For every $t' \in N \setminus S^r$, set $x_{t'}^r = 0$ and $y_{t'}^r = 1$; set $x_{i_1}^r = 0$ and $y_{i_1}^r = 0$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{i_1\}$; set $x_1^m = m_1^m$ and $y_1^m = 1$ and set other variables to zero. **(**$n - |S^r|$ **points)**

6. For every $t' \in N \setminus S^r$, set $x_{t'}^r = \epsilon$ and $y_{t'}^r = 1$; set $x_{i_1}^r = 0$ and $y_{i_1}^r = 0$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{i_1\}$; set $x_1^m = m_1^m$ and $y_1^m = 1$ and set other variables to zero. **(**$n - |S^r|$ **points)**

7. For every $t' \in N \setminus \{1\}$, set $x_{i_1}^r = 0$ and $y_{i_1}^r = 0$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{i_1\}$; set $x_{t'}^m = 0$ and $y_{t'}^m = 1$; set $x_1^m = m_1^m$ and $y_1^m = 1$ and set other variables to zero. **(**$n - 1$ **points)**

8. For every $t' \in N \setminus \{1\}$, set $x_{i_1}^r = 0$ and $y_{i_1}^r = 0$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{i_1\}$; set $x_{t'}^m = \epsilon$ and $y_{t'}^m = 1$; set $x_1^m = m_1^m$ and $y_1^m = 1$ and set other variables to zero. **(**$n - 1$ **points)**

9. Set $x_{i_1}^r = 0$ and $y_{i_1}^r = 0$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{i_1\}$; set $x_1^m = 0$ and $y_1^m = 0$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in N \setminus \{1\}$ and set other variables to zero. **(**1 **point)**

10. Set $x_{i_1}^r = 0$ and $y_{i_1}^r = 0$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{i_1\}$; set $x_1^m = 0$ and $y_1^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in N \setminus \{1\}$ and set other variables to zero. **(**1 **point)**

We note that the affine independence of these $4n$ points is straightforward and; therefore, omitted here for the sake of brevity.

□

It is natural to extend inequality (4.7) as follows.

Corollary 5 (Extended flow cover inequalities (Padberg et al., 1985)). *Let $S^r \subseteq N$ be a cover for R with $\overline{m^r} = \max_{t \in S^r} m_t^r$ and $L^r \subseteq N \setminus S^r$. Assume $\overline{m_t^r} = \max(\overline{m^r}, m_t^r)$ for all $t \in L^r$. Then the following inequality (called* ***returns-extended cover inequality****) is valid for X^s.*

$$\sum_{t \in S^r \cup L^r} x_t^r - \sum_{t \in S^r} (m_t^r - \lambda_1)^+ y_t^r - \sum_{t \in L^r} (\overline{m_t^r} - \lambda_1) y_t^r \leq R - \sum_{t \in S^r} (m_t^r - \lambda_1)^+ \quad (4.8)$$

Proposition 18. *The inequality (4.8) is facet-defining for $conv(X^s)$ if both $0 < \overline{m^r} - \lambda_1 < m_t^r \leq \overline{m^r}$ for any $t \in L^r$ and the conditions of Proposition 5 hold.*

Proof. Condition $0 < \overline{m^r} - \lambda_1 < m_t^r \leq \overline{m^r}$ implies $\overline{m_t^r} = \overline{m^r}$, $\forall t \in L^r$. Let $\epsilon > 0$ is a sufficiently small number. We present $4N$ affinely independent points in X^s that satisfy (4.8) as an equation. Note that the first four and the last four valid sets listed in the proof of Proposition 5 satisfy (4.8) as an equation. Therefore, we identify the remaining sets as follows.

1. For every $t' \in L^r$, set $x_{t'}^r = \overline{m^r} - \lambda_1$ and $y_{t'}^r = 1$; set $x_{i_1}^r = 0$ and $y_{i_1}^r = 0$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{i_1\}$; set $x_1^m = m_1^m$ and $y_1^m = 1$ and set other variables to zero. **($|L^r|$ points)**

2. For every $t' \in L^r$, set $x_{t'}^r = \overline{m^r} - \lambda_1 + \epsilon$ and $y_{t'}^r = 1$; set $x_{i_1}^r = 0$ and $y_{i_1}^r = 0$; set $x_{i_2}^r = m_{i_2}^r - \epsilon$ and $y_{i_2}^r = 1$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{i_1, i_2\}$; set $x_1^m = m_1^m$ and $y_1^m = 1$ and set other variables to zero. **($|L^r|$ points)**

3. For every $t' \in N \setminus (S^r \cup L^r)$, set $x_{t'}^r = 0$ and $y_{t'}^r = 1$; set $x_{i_1}^r = 0$ and $y_{i_1}^r = 0$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{i_1\}$; set $x_1^m = m_1^m$ and $y_1^m = 1$ and set other variables to zero. **($n - |S^r| - |L^r|$ points)**

4. For every $t' \in N \setminus (S^r \cup L^r)$, set $x_{t'}^r = \epsilon$ and $y_{t'}^r = 1$; set $x_{i_1}^r = 0$ and $y_{i_1}^r = 0$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{i_1\}$; set $x_1^m = m_1^m$ and $y_1^m = 1$ and set other variables to zero. **($n - |S^r| - |L^r|$ points)**

Note that the affine independence of these $4n$ points is straightforward.

□

Next, we investigate some well-known inequalities originally proposed for $X_{\geq}$, which is again an obvious relaxation of set X^s. We obtain this relaxation of set X^s by eliminating one knapsack set involving returns. Our theoretical contribution remains with these inequalities being facet-defining for $conv(X^s)$ under certain and general conditions.

Corollary 6 (Flow cover inequalities (Padberg et al., 1985)). *Let $S^m \subseteq N$ be a cover for $D - R$. Then, the following inequality (called* ***demands cover inequality****) is valid for X^s.*

$$\sum_{t \in N \setminus S^m} x_t^m \geq \sum_{t \in S^m} (m_t^m - \lambda_2)^+ (1 - y_t^m) \quad (4.9)$$

Proof. First, we rearrange and rewrite the inequality (4.9) using the definition of S^{m+} as:

$$\sum_{t \in N \setminus S^m} x_t^m + \sum_{t \in S^{m+}} (m_t^m - \lambda_2) y_t^m \geq \sum_{t \in S^{m+}} (m_t^m - \lambda_2)$$

Consider (x^r, x^m, y^r, y^m) be a point of X^s with $T^m = \{t \in N | y_t^m = 1\}$. We identify two cases for this inequality:

Case 1. $|S^{m+} \setminus T^m| = 0$. This implies that $y_t^m = 1$ for any $t \in S^{m+}$. Then, the validity of this inequality is followed by $\sum_{t \in N} x_t^m \geq \sum_{t \in S^m} x_t^m \geq D - R \geq 0$.

Case 2. $|S^{m+} \setminus T^m| \geq 1$.

$$\begin{aligned}
&\sum_{t \in N \setminus S^m} x_t^m + \sum_{t \in S^{m+}} (m_t^m - \lambda_2) y_t^m \\
&= \sum_{t \in (N \setminus S^m) \cap T^m} x_t^m + \sum_{t \in S^{m+} \cap T^m} (m_t^m - \lambda_2) \\
&= \sum_{t \in N \cap T^m} x_t^m - \sum_{t \in S^m \cap T^m} x_t^m + \sum_{t \in S^{m+}} (m_t^m - \lambda_2) - \sum_{t \in S^{m+} \setminus T^m} (m_t^m - \lambda_2) \\
&\geq \sum_{t \in N \cap T^m} x_t^m - \sum_{t \in S^m \cap T^m} m_t^m + \sum_{t \in S^{m+}} (m_t^m - \lambda_2) - \sum_{t \in S^{m+} \setminus T^m} (m_t^m - \lambda_2) \\
&\geq (D - R) - \sum_{t \in S^m} m_t^m + \sum_{t \in S^m \setminus T^m} m_t^m + \sum_{t \in S^{m+}} (m_t^m - \lambda_2) \\
&\qquad\qquad - \sum_{t \in S^{m+} \setminus T^m} (m_t^m - \lambda_2) \\
&\geq -\lambda_2 + \sum_{t \in S^{m+} \setminus T^m} m_t^m + \sum_{t \in S^{m+}} (m_t^m - \lambda_2) - \sum_{t \in S^{m+} \setminus T^m} (m_t^m - \lambda_2) \\
&= \sum_{t \in S^{m+}} (m_t^m - \lambda_2) - \lambda_2 + \lambda_2 |S^{m+} \setminus T^m| \\
&= \sum_{t \in S^{m+}} (m_t^m - \lambda_2) + \lambda_2 \left(|S^{m+} \setminus T^m| - 1\right) \\
&\geq \sum_{t \in S^{m+}} (m_t^m - \lambda_2)
\end{aligned}$$

where the first and second inequalities use the properties of $y_t^m = 1$,

$\forall t \in T^m$, $S^m \cap T^m = S^m \setminus (S^m \setminus T^m)$, $x_t^m \leq m_t^m y_t^m$ and the fact that $\sum_{t \in N \cap T^m} x_t^m \geq D - R$. Next, the third and last inequalities use the definition of λ_2 and the properties $S^{m+} \subseteq S^m$, $|S^{m+} \setminus T^m| - 1 \geq 0$ and $\lambda_2 > 0$.

□

The facet-defining conditions for this inequality are described in the following proposition.

Proposition 19. *Let* $S^{m+} = \{t \in S^m | m_t^m - \lambda_2 > 0\}$. *If* $|S^{m+}| \geq 1$, $\sum_{t \in N \setminus S^m} m_t^m > \max_{t \in S^m} m_t^m - \lambda_2$ *and* $\sum_{t \in N} m_t^r > R + \max_{t \in N} m_t^r$ *then, the inequality* (4.9) *defines a facet for* $conv(X^s)$.

Proof. Let $H^r \subset N$ such that $\sum_{t \in H^r} m_t^r > R$, $\exists k \in H^r$ satisfying $\sum_{t \in H^r \setminus \{k\}} m_t^r < R$ and $\exists \ell \notin H^r$ satisfying $m_\ell^r \geq m_t^r$, $\forall t \in H^r$. Let i_1 be any member in the set S^{m+} and $\epsilon > 0$ be a sufficiently small number. We also define $\hat{m}_t^m = m_t^m \;/ \sum_{t \in N \setminus S^m} m_t^m$. In order to prove that this inequality is facet-defining, we will present the following $4n$ affinely independent points that satisfy it as an equation.

1. For every $t' \in S^{m+}$, set $x_{t'}^m = 0$ and $y_{t'}^m = 0$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{t'\}$; set $x_t^m = \hat{m}_t^m (m_{t'}^m - \lambda_2)$ and $y_t^m = 1$, $\forall t \in N \setminus S^m$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t^r$ and $y_k^r = 1$ and set other variables to zero. **($|S^{m+}|$ points)**

2. For every $t' \in S^{m+}$, set $x_{t'}^m = m_{t'}^m - \lambda_2$ and $y_{t'}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{t'\}$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t^r$ and $y_k^r = 1$ and set other variables to zero. **($|S^{m+}|$ points)**

3. For every $t' \in S^m \setminus S^{m+}$, set $x_{t'}^m = 0$ and $y_{t'}^m = 0$; set $x_{i_1}^m = m_{i_1}^m - \lambda_2 + m_{t'}^m$ and $y_{i_1}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{t', i_1\}$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t^r$ and $y_k^r = 1$ and set other variables to zero. **($|S^m \setminus S^{m+}|$ points)**

4. For every $t' \in S^m \setminus S^{m+}$, set $x_{t'}^m = 0$ and $y_{t'}^m = 1$; set $x_{i_1}^m = m_{i_1}^m - \lambda_2 + m_{t'}^m$ and $y_{i_1}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{t', i_1\}$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t^r$ and $y_k^r = 1$ and set other variables to zero. **($|S^m \setminus S^{m+}|$ points)**

5. For every $t' \in N \setminus S^m$, set $x_{t'}^m = 0$ and $y_{t'}^m = 1$; set $x_{i_1}^m = m_{i_1}^m - \lambda_2$ and $y_{i_1}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{i_1\}$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t^r$ and $y_k^r = 1$ and set other variables to zero. **($n - |S^m|$ points)**

6. For every $t' \in N \setminus S^m$, set $x_{t'}^m = \epsilon$ and $y_{t'}^m = 1$; set $x_{i_1}^m = m_{i_1}^m - \lambda_2 + \epsilon$ and $y_{i_1}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{i_1\}$; set $x_t^r = m_t^r$ and $y_t^r = 1$,

$\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t\in H^r\setminus\{k\}} m_t^r$ and $y_k^r = 1$ and set other variables to zero. **($n - |S^m|$ points)**

7. For every $t' \in H^r \setminus \{k\}$, set $x_{t'}^r = 0$ and $y_{t'}^r = 0$; set $x_\ell^r = m_{t'}^r$ and $y_\ell^r = 1$; set $x_{i_1}^m = m_{i_1}^m - \lambda_2$ and $y_{i_1}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{i_1\}$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r \setminus \{t', k\}$; set $x_k^r = R - \sum_{t\in H^r\setminus\{k\}} m_t^r$ and $y_k^r = 1$ and set other variables to zero. **($|H^r| - 1$ points)**

8. Set $x_k^r = 0$ and $y_k^r = 0$; set $x_\ell^r = R - \sum_{t\in H^r\setminus\{k\}} m_t^r$ and $y_\ell^r = 1$; set $x_{i_1}^m = m_{i_1}^m - \lambda_2$ and $y_{i_1}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{i_1\}$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r \setminus \{\ell, k\}$ and set other variables to zero. **(1 point)**

9. For every $t' \in H^r \setminus \{k\}$, set $x_{t'}^r = 0$ and $y_{t'}^r = 1$; set $x_\ell^r = m_{t'}^r$ and $y_\ell^r = 1$; set $x_{i_1}^m = m_{i_1}^m - \lambda_2$ and $y_{i_1}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{i_1\}$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r \setminus \{t', k\}$; set $x_k^r = R - \sum_{t\in H^r\setminus\{k\}} m_t^r$ and $y_k^r = 1$ and set other variables to zero. **($|H^r| - 1$ points)**

10. Set $x_k^r = 0$ and $y_k^r = 1$; set $x_\ell^r = R - \sum_{t\in H^r\setminus\{k\}} m_t^r$ and $y_\ell^r = 1$; set $x_{i_1}^m = m_{i_1}^m - \lambda_2$ and $y_{i_1}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{i_1\}$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r \setminus \{\ell, k\}$ and set other variables to zero. **(1 point)**

11. For every $t' \in N \setminus H^r$, set $x_{t'}^r = 0$ and $y_{t'}^r = 1$; set $x_{i_1}^m = m_{i_1}^m - \lambda_2$ and $y_{i_1}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{i_1\}$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t\in H^r\setminus\{k\}} m_t^r$ and $y_k^r = 1$ and set other variables to zero. **($n - |H^r|$ points)**

12. For every $t' \in N \setminus H^r$, set $x_{t'}^r = \epsilon$ and $y_{t'}^r = 1$; set $x_{i_1}^m = m_{i_1}^m - \lambda_2$ and $y_{i_1}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{i_1\}$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t\in H^r\setminus\{k\}} m_t^r - \epsilon$ and $y_k^r = 1$ and set other variables to zero. **($n - |H^r|$ points)**

□

We omit the affine independence proof for the sake of brevity and as it is straightforward due to its similarity to previous proofs. Next, we discuss the extended version of these inequalities in the following corollary.

Corollary 7 (Extended flow cover inequalities, Padberg et al. (1985)). *Let $S^m \subseteq N$ be a cover for $D - R$ and $L^m \subseteq N \setminus S^m$ such that $\overline{m^m} = \max_{t\in S^m} m_t^m > \lambda_2$ and $\overline{m_t^m} = \max\{m_t^m, \overline{m^m}\}$, $\forall t \in L^m$. Then, the following inequality (called* ***demands-extended cover inequality****) is valid for X^s.*

$$\sum_{t\in N\setminus(S^m\cup L^m)} x_t^m + \sum_{t\in L^m} (\overline{m_t^m} - \lambda_2) y_t^m \geq \sum_{t\in S^m} (m_t^m - \lambda_2)^+ (1 - y_t^m) \tag{4.10}$$

Proof. First, we rearrange and rewrite the inequality (4.10) using the definition of

$S^{m+} = \{t \in S^m | m_t^m - \lambda_2 > 0\}$ as:

$$\sum_{t \in N \setminus (S^m \cup L^m)} x_t^m + \sum_{t \in S^{m+}} (m_t^m - \lambda_2) y_t^m + \sum_{t \in L^m} (\overline{m_t^m} - \lambda_2) y_t^m \geq \sum_{t \in S^{m+}} (m_t^m - \lambda_2)$$

Consider (x^r, x^m, y^r, y^m) be a point of X^s with $T^m = \{t \in N | y_t^m = 1\}$. We identify two cases for this inequality:

Case 1. $|S^{m+} \setminus T^m| \leq |L^m \cap T^m|$

$$\begin{aligned}
&\sum_{t \in N \setminus (S^m \cup L^m)} x_t^m + \sum_{t \in S^{m+}} (m_t^m - \lambda_2) y_t^m + \sum_{t \in L^m} (\overline{m_t^m} - \lambda_2) y_t^m \\
&= \sum_{t \in (N \setminus (S^m \cup L^m)) \cap T^m} x_t^m + \sum_{t \in S^{m+} \cap T^m} (m_t^m - \lambda_2) + \sum_{t \in L^m \cap T^m} (\overline{m_t^m} - \lambda_2) \\
&\geq \sum_{t \in S^{m+}} (m_t^m - \lambda_2) - \sum_{t \in S^{m+} \setminus T^m} (m_t^m - \lambda_2) + \sum_{t \in L^m \cap T^m} (\overline{m^m} - \lambda_2) \\
&\geq \sum_{t \in S^{m+}} (m_t^m - \lambda_2) - \sum_{t \in S^{m+} \setminus T^m} (\overline{m^m} - \lambda_2) + \sum_{t \in L^m \cap T^m} (\overline{m^m} - \lambda_2) \\
&= \sum_{t \in S^{m+}} (m_t^m - \lambda_2) + (\overline{m^m} - \lambda_2) \big(|L^m \cap T^m| - |S^{m+} \setminus T^m|\big) \\
&\geq \sum_{t \in S^{m+}} (m_t^m - \lambda_2)
\end{aligned}$$

where the first inequality follows the properties of $y_t^m = 1, \forall t \in T^m$ and $S^{m+} \cap T^m = S^{m+} \setminus (S^{m+} \setminus T^m)$. Then, the second inequality uses the fact that $m_t^m \leq \overline{m^m} \leq \overline{m_t^m}$ and the last inequality obtained as a result of the properties $|L^m \cap T^m| - |S^{m+} \setminus T^m| \geq 0$ and $\overline{m^m} \geq \lambda_2$.

Case 2. $|S^{m+} \setminus T^m| \geq |L^m \cap T^m| + 1$

$$\begin{aligned}
&\sum_{t \in N \setminus (S^m \cup L^m)} x_t^m + \sum_{t \in S^{m+}} (m_t^m - \lambda_2) y_t^m + \sum_{t \in L^m} (\overline{m_t^m} - \lambda_2) y_t^m \\
&= \sum_{t \in (N \setminus (S^m \cup L^m)) \cap T^m} x_t^m + \sum_{t \in S^{m+} \cap T^m} (m_t^m - \lambda_2) + \sum_{t \in L^m \cap T^m} (\overline{m_t^m} - \lambda_2) \\
&= \sum_{t \in N \cap T^m} x_t^m - \sum_{t \in S^m \cap T^m} x_t^m - \sum_{t \in L^m \cap T^m} x_t^m + \sum_{t \in S^{m+}} (m_t^m - \lambda_2) \\
&\qquad - \sum_{t \in S^{m+} \setminus T^m} (m_t^m - \lambda_2) + \sum_{t \in L^m \cap T^m} (\overline{m_t^m} - \lambda_2) \\
&\geq \sum_{t \in N \cap T^m} x_t^m - \sum_{t \in S^m \cap T^m} m_t^m - \sum_{t \in L^m \cap T^m} m_t^m + \sum_{t \in S^{m+}} (m_t^m - \lambda_2) \\
&\qquad - \sum_{t \in S^{m+} \setminus T^m} (m_t^m - \lambda_2) + \sum_{t \in L^m \cap T^m} (\overline{m_t^m} - \lambda_2)
\end{aligned}$$

$$
\begin{aligned}
\geq &(D-R) - \sum_{t\in S^m} m_t^m + \sum_{t\in S^m\setminus T^m} m_t^m - \sum_{t\in L^m\cap T^m} m_t^m + \sum_{t\in S^{m+}} (m_t^m - \lambda_2) \\
&\qquad - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_2) + \sum_{t\in L^m\cap T^m} (\overline{m_t^m} - \lambda_2) \\
\geq & -\lambda_2 + \sum_{t\in S^{m+}\setminus T^m} m_t^m - \sum_{t\in L^m\cap T^m} \overline{m^m} + \sum_{t\in S^{m+}} (m_t^m - \lambda_2) \\
&\qquad - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_2) + \sum_{t\in L^m\cap T^m} (\overline{m^m} - \lambda_2) \\
= &\sum_{t\in S^{m+}} (m_t^m - \lambda_2) - \lambda_2 + \lambda_2 |S^{m+}\setminus T^m| - \lambda_2 |L^m \cap T^m| \\
= &\sum_{t\in S^{m+}} (m_t^m - \lambda_2) + \lambda_2 \left(|S^{m+}\setminus T^m| - |L^m\cap T^m| - 1 \right) \\
\geq &\sum_{t\in S^{m+}} (m_t^m - \lambda_2)
\end{aligned}
$$

where the first and second inequalities use the properties of $y_t^m = 1$, $t \in T^m$, $S^m \cap T^m = S^m \setminus (S^m \setminus T^m)$, $x_t^m \leq m_t^m y_t^m$ and the fact that $\sum_{t\in N\cap T^m} x_t^m \geq D - R$. Next, the third and the last inequalities use the definition of λ_2 and the properties $m_t^m \leq \overline{m^m} \leq \overline{m_t^m}$, $S^{m+} \subseteq S^m$, $|S^{m+}\setminus T^m| - |L^m \cap T^m| - 1 \geq 0$ and $\lambda_2 > 0$.

□

Now, we establish their facet-defining conditions in the next proposition.

Proposition 20. *Let* $S^{m+} = \{t \in S^m | m_t^m - \lambda_2 > 0\}$*. If* $0 < \overline{m^m} - \lambda_2 < m_t^m \leq \overline{m^m}$ *for any* $t \in L^m$, $\sum_{t\in N\setminus(S^m\cup L^m)} m_t^m > \max_{t\in S^m} m_t^m - \lambda_2$ *and* $\sum_{t\in N} m_t^r > R + \max_{t\in N} m_t^r$, *then the inequality* (4.10) *defines a facet for* $conv(X^s)$.

Proof. Similar to the proof of Proposition 19, we let $H^r \subset N$ such that $\sum_{t\in H^r} m_t^r > R$, $\exists k \in H^r$ satisfying $\sum_{t\in H^r\setminus\{k\}} m_t^r < R$ and $\exists \ell \notin H^r$ satisfying $m_\ell^r \geq m_t^r$, $\forall t \in H^r$. Let $i_1 \in S^{m+}$such that $m_{i_1}^m = \overline{m^m}$ and $\epsilon > 0$ be a sufficiently small number. We also define $\hat{m}_t^m = m_t^m \,/ \sum_{t\in N\setminus(S^m\cup L^m)} m_t^m$. Then, we note that all the affinely independent points from the proof of Proposition 19 are also valid for this case, except that for set 2 of these points, the values are set for $t \in N\setminus(S^m\cup L^m)$ rather than $t \in N\setminus S^m$, and for sets 5 and 6 of points, the points are valid only for $t \in N\setminus(S^m \cup L^m)$. Therefore, we need to define $2|L^m|$ new points in order to obtain $4n$ points in total, which we present as follows:

1. For every $t' \in L^m$, set $x_{t'}^m = \overline{m^m} - \lambda_2$ and $y_{t'}^m = 1$; set $x_{i_1}^m = 0$ and $y_{i_1}^m = 0$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{i_1\}$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in H^r\setminus\{k\}$; set $x_k^r = R - \sum_{t\in H^r\setminus\{k\}} m_t^r$ and $y_k^r = 1$ and set other variables to zero. ($|L^m|$ **points)**

2. For every $t' \in L^m$, set $x^m_{t'} = \overline{m^m} - \lambda_2 + \epsilon$ and $y^m_{t'} = 1$; set $x^m_{i_1} = 0$ and $y^m_{i_1} = 0$; set $x^m_t = m^m_t$ and $y^m_t = 1$, $\forall t \in S^m \setminus \{i_1\}$; set $x^r_t = m^r_t$ and $y^r_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x^r_k = R - \sum_{t \in H^r \setminus \{k\}} m^r_t$ and $y^r_k = 1$ and set other variables to zero. **($|L^m|$ points)**

□

Lastly, we present the remaining two valid inequalities for $conv(X^s)$ in the case of $\geq$.

Corollary 8 (**Flow cover inequalities (Padberg et al., 1985)**). *For $S^r, S^m \subseteq N$, let (S^r, S^m) be a pair cover for D. Then, the inequality (called **returns-demands cover inequality**) is valid for X^s.*

$$\sum_{t \in N \setminus S^r} x^r_t + \sum_{t \in N \setminus S^m} x^m_t \geq \sum_{t \in S^r} (m^r_t - \lambda_3)^+ (1 - y^r_t) + \sum_{t \in S^m} (m^m_t - \lambda_3)^+ (1 - y^m_t) \tag{4.11}$$

Proof. Firstly, this inequality (4.11) is rearranged and rewritten as:

$$\sum_{t \in N \setminus S^r} x^r_t + \sum_{t \in N \setminus S^m} x^m_t + \sum_{t \in S^{r+}} (m^r_t - \lambda_3) y^r_t + \sum_{t \in S^{m+}} (m^m_t - \lambda_3) y^m_t + \\ \geq \sum_{t \in S^{r+}} (m^r_t - \lambda_3) + \sum_{t \in S^{m+}} (m^m_t - \lambda_3)$$

where, $S^{r+} = \{t \in S^r | m^r_t - \lambda_3 > 0\}$ and $S^{m+} = \{t \in S^m | m^m_t - \lambda_3 > 0\}$. Suppose (x^r, x^m, y^r, y^m) be a point of X^s with $T^r = \{t \in N | y^r_t = 1\}$ and $T^m = \{t \in N | y^m_t = 1\}$. We consider four following cases:

Case 1. $|S^{r+} \setminus T^r| + |S^{m+} \setminus T^m| = 0$. This implies that both $y^r_t = 1$ for any $t \in S^{r+}$ and $y^m_t = 1$ for any $t \in S^{m+}$. This shows that $\sum_{t \in N}(x^r_t + x^m_t) \geq \sum_{t \in S^r} x^r_t + \sum_{t \in S^m} x^m_t \geq D$.

Case 2. $|S^{r+} \setminus T^r| = 0$ and $|S^{m+} \setminus T^m| \geq 1$.

$$\sum_{t \in N \setminus S^r} x^r_t + \sum_{t \in N \setminus S^m} x^m_t + \sum_{t \in S^{r+}} (m^r_t - \lambda_3) y^r_t + \sum_{t \in S^{m+}} (m^m_t - \lambda_3) y^m_t$$
$$= \sum_{t \in (N \setminus S^r) \cap T^r} x^r_t + \sum_{t \in (N \setminus S^m) \cap T^m} x^m_t + \sum_{t \in S^{r+} \cap T^r} (m^r_t - \lambda_3) \\ + \sum_{t \in S^{m+} \cap T^m} (m^m_t - \lambda_3)$$
$$= \sum_{t \in N \cap T^r} x^r_t + \sum_{t \in N \cap T^m} x^m_t - \sum_{t \in S^r \cap T^r} x^r_t - \sum_{t \in S^m \cap T^m} x^m_t + \sum_{t \in S^{r+}} (m^r_t - \lambda_3) \\ + \sum_{t \in S^{m+}} (m^m_t - \lambda_3) - \sum_{t \in S^{r+} \setminus T^r} (m^r_t - \lambda_3) - \sum_{t \in S^{m+} \setminus T^m} (m^m_t - \lambda_3)$$

$$
\begin{aligned}
&\geq \sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m - \sum_{t\in S^r\cap T^r} m_t^r - \sum_{t\in S^m\cap T^m} m_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) \\
&\qquad + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3) - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3) \\
&\geq D - \sum_{t\in S^r} m_t^r - \sum_{t\in S^m} m_t^m + \sum_{t\in S^r\setminus T^r} m_t^r + \sum_{t\in S^m\setminus T^m} m_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) \\
&\qquad + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3) - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3) \\
&\geq -\lambda_3 + \sum_{t\in S^{m+}\setminus T^m} m_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) \\
&\qquad - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3) \\
&= \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \lambda_3 + \lambda_3(|S^{m+} \setminus T^m|) \\
&= \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) + \lambda_3 \big(|S^{m+} \setminus T^m| - 1\big) \\
&\geq \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3)
\end{aligned}
$$

where the first inequality uses the properties of $y_t^r = 1$, $\forall t \in T^r$, $y_t^m = 1$, $\forall t \in T^m$, $x_t^r \leq m_t^r y_t^r$ and $x_t^m \leq m_t^m y_t^m$. The second inequality follows the fact that $S^r \cap T^r = S^r \setminus (S^r \setminus T^r)$, $S^m \cap T^m = S^m \setminus (S^m \setminus T^m)$ and $\sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m \geq D$. The third inequality uses the definition of λ_3, the properties $S^{r+} \subseteq S^r$ and $S^{m+} \subseteq S^m$ and $|S^{r+} \setminus T^r| = 0$. Lastly, the inequality holds the properties $|S^{m+} \setminus T^m| - 1 \geq 0$ and $\lambda_3 > 0$.

Case 3. $|S^{r+} \setminus T^r| \geq 1$ and $|S^{m+} \setminus T^m| = 0$.

$$
\begin{aligned}
&\sum_{t\in N\setminus S^r} x_t^r + \sum_{t\in N\setminus S^m} x_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) y_t^r + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) y_t^m \\
&= \sum_{t\in (N\setminus S^r)\cap T^r} x_t^r + \sum_{t\in (N\setminus S^m)\cap T^m} x_t^m + \sum_{t\in S^{r+}\cap T^r} (m_t^r - \lambda_3) \\
&\qquad + \sum_{t\in S^{m+}\cap T^m} (m_t^m - \lambda_3) \\
&= \sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m - \sum_{t\in S^r\cap T^r} x_t^r - \sum_{t\in S^m\cap T^m} x_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) \\
&\qquad + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3) - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3)
\end{aligned}
$$

$$
\begin{aligned}
&\geq \sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m - \sum_{t\in S^r\cap T^r} m_t^r - \sum_{t\in S^m\cap T^m} m_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) \\
&\qquad + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3) - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3) \\
&\geq D - \sum_{t\in S^r} m_t^r - \sum_{t\in S^m} m_t^m + \sum_{t\in S^r\setminus T^r} m_t^r + \sum_{t\in S^m\setminus T^m} m_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) \\
&\qquad + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3) - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3) \\
&\geq -\lambda_3 + \sum_{t\in S^{r+}\setminus T^r} m_t^r + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) \\
&\qquad - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3) \\
&= \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \lambda_3 + \lambda_3(|S^{r+} \setminus T^r|) \\
&= \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) + \lambda_3 \big(|S^{r+} \setminus T^r| - 1\big) \\
&\geq \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3)
\end{aligned}
$$

where the first inequality follows the properties of $y_t^r = 1, \forall t \in T^r$, $y_t^m = 1, \forall t \in T^m$, $x_t^r \leq m_t^r y_t^r$ and $x_t^m \leq m_t^m y_t^m$. The second inequality uses the fact that $S^r \cap T^r = S^r \setminus (S^r \setminus T^r)$ and $S^m \cap T^m = S^m \setminus (S^m \setminus T^m)$ and $\sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m \geq D$. Next, the third inequality follows the definition of λ_3, the properties $S^{r+} \subseteq S^r$ and $S^{m+} \subseteq S^m$ and $|S^{m+} \setminus T^m| = 0$. Finally, the last inequality makes use of the properties $|S^{r+} \setminus T^r| - 1 \geq 0$ and $\lambda_3 > 0$.

Case 4. $|S^{r+} \setminus T^r| + |S^{m+} \setminus T^m| \geq 1$.

$$
\begin{aligned}
&\sum_{t\in N\setminus S^r} x_t^r + \sum_{t\in N\setminus S^m} x_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) y_t^r + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) y_t^m \\
&= \sum_{t\in (N\setminus S^r)\cap T^r} x_t^r + \sum_{t\in (N\setminus S^m)\cap T^m} x_t^m + \sum_{t\in S^{r+}\cap T^r} (m_t^r - \lambda_3) \\
&\qquad + \sum_{t\in S^{m+}\cap T^m} (m_t^m - \lambda_3) \\
&= \sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m - \sum_{t\in S^r\cap T^r} x_t^r - \sum_{t\in S^m\cap T^m} x_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) \\
&\qquad + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3) - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3)
\end{aligned}
$$

$$
\begin{aligned}
&\geq \sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m - \sum_{t\in S^r\cap T^r} m_t^r - \sum_{t\in S^m\cap T^m} m_t^m + \sum_{t\in S^{r+}} (m_t^r-\lambda_3)\\
&\qquad + \sum_{t\in S^{m+}} (m_t^m-\lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r-\lambda_3) - \sum_{t\in S^{m+}\setminus T^m} (m_t^m-\lambda_3)\\
&\geq D - \sum_{t\in S^r} m_t^r - \sum_{t\in S^m} m_t^m + \sum_{t\in S^r\setminus T^r} m_t^r + \sum_{t\in S^m\setminus T^m} m_t^m + \sum_{t\in S^{r+}} (m_t^r-\lambda_3)\\
&\qquad + \sum_{t\in S^{m+}} (m_t^m-\lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r-\lambda_3) - \sum_{t\in S^{m+}\setminus T^m} (m_t^m-\lambda_3)\\
&\geq -\lambda_3 + \sum_{t\in S^{r+}\setminus T^r} m_t^r + \sum_{t\in S^{m+}\setminus T^m} m_t^m + \sum_{t\in S^{r+}} (m_t^r-\lambda_3) + \sum_{t\in S^{m+}} (m_t^m-\lambda_3)\\
&\qquad - \sum_{t\in S^{r+}\setminus T^r} (m_t^r-\lambda_3) - \sum_{t\in S^{m+}\setminus T^m} (m_t^m-\lambda_3)\\
&= \sum_{t\in S^{r+}} (m_t^r-\lambda_3) + \sum_{t\in S^{m+}} (m_t^m-\lambda_3) - \lambda_3\\
&\qquad + \lambda_3(|S^{r+}\setminus T^r| + |S^{m+}\setminus T^m| - 1)\\
&= \sum_{t\in S^{r+}} (m_t^r-\lambda_3) + \sum_{t\in S^{m+}} (m_t^m-\lambda_3) + \lambda_3\big(|S^{r+}\setminus T^r| + |S^{m+}\setminus T^m| - 1\big)\\
&\geq \sum_{t\in S^{r+}} (m_t^r-\lambda_3) + \sum_{t\in S^{m+}} (m_t^m-\lambda_3)
\end{aligned}
$$

where the first inequality follows the properties of $y_t^r = 1$, $\forall t \in T^r$, $y_t^m = 1$, $\forall t \in T^m$, $x_t^r \leq m_t^r y_t^r$ and $x_t^m \leq m_t^m y_t^m$. Next, by using the fact $S^r \cap T^r = S^r \setminus (S^r \setminus T^r)$, $S^m \cap T^m = S^m \setminus (S^m \setminus T^m)$ and $\sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m \geq D$, we obtain the second inequality. The third inequality follows the definition of λ_3, the properties $S^{r+} \subseteq S^r$ and $S^{m+} \subseteq S^m$. The properties $|S^{r+}\setminus T^r| - |S^{m+}\setminus T^m| - 1 \geq 0$ and $\lambda_3 > 0$ are used to generate the last inequality.

□

Proposition 21. *If* $|S^{r+}| + |S^{m+}| \geq 2$ *and* $\sum_{t\in N\setminus S^r} m_t^r + \sum_{t\in N\setminus S^m} m_t^m > \max\{\max_{t\in S^r} m_t^r, \max_{t\in S^m} m_t^m\} - \lambda_3$, *then the inequality* (4.11) *is facet-defining for* $conv(X^s)$.

Proof. First, we define $S^{r+} = \{t \in S^r | m_t^r - \lambda_3 > 0\}$ and $S^{m+} = \{t \in S^m | m_t^m - \lambda_3 > 0\}$. Let $i_1 \in S^{r+} \cup S^{m+}$ be any member and $\epsilon > 0$ be a sufficiently small number. We also define $\hat{m}_t^r = m_t^r / (\sum_{t\in N\setminus S^r} m_t^r + \sum_{t\in N\setminus S^m} m_t^m)$ for all $t \in N \setminus S^r$ and $\hat{m}_t^m = m_t^m / (\sum_{t\in N\setminus S^r} m_t^r + \sum_{t\in N\setminus S^m} m_t^m)$ for all $t \in N \setminus S^m$. We next present $4n$ affinely independent points that satisfy (4.11) as an equation.

1. For every $t' \in S^{r+}$, set $x_{t'}^r = 0$ and $y_{t'}^r = 0$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{t'\}$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m$; set $x_t^r = \hat{m}_t^r(m_{t'}^r - \lambda_3)$ and

$y_t^r = 1$, $\forall t \in N \setminus S^r$; set $x_t^m = \hat{m}_t^m(m_{t'}^r - \lambda_3)$ and $y_t^m = 1$, $\forall t \in N \setminus S^m$ and set other variables to zero. **($|S^{r+}|$ points)**

2. For every $t' \in S^{r+}$, set $x_{t'}^r = m_{t'}^r - \lambda_3$ and $y_{t'}^r = 1$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{t'\}$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m$ and set other variables to zero. **($|S^{r+}|$ points)**

3. For every $t' \in S^r \setminus S^{r+}$, set $x_{t'}^r = 0$ and $y_{t'}^r = 0$; set $x_{i_1}^r = m_{i_1}^r - \lambda_3 + m_{t'}^r$ and $y_{i_1}^r = 1$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{t', i_1\}$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m$ and set other variables to zero. **($|S^r \setminus S^{r+}|$ points)**

4. For every $t' \in S^r \setminus S^{r+}$, set $x_{t'}^r = 0$ and $y_{t'}^r = 1$; set $x_{i_1}^r = m_{i_1}^r - \lambda_3 + m_{t'}^r$ and $y_{i_1}^r = 1$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r \setminus \{t', i_1\}$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m$ and set other variables to zero. **($|S^r \setminus S^{r+}|$ points)**

5. For every $t' \in S^{m+}$, set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r$; set $x_{t'}^m = 0$ and $y_{t'}^m = 0$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{t'\}$; set $x_t^r = \hat{m}_t^r(m_{t'}^m - \lambda_3)$ and $y_t^r = 1$, $\forall t \in N \setminus S^r$; set $x_t^m = \hat{m}_t^m(m_{t'}^m - \lambda_3)$ and $y_t^m = 1$, $\forall t \in N \setminus S^m$ and set other variables to zero. **($|S^{m+}|$ points)**

6. For every $t' \in S^{m+}$, set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r$; set $x_{t'}^m = m_{t'}^m - \lambda_3$ and $y_{t'}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{t'\}$ and set other variables to zero. **($|S^{m+}|$ points)**

7. For every $t' \in S^m \setminus S^{m+}$, set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r$; set $x_{t'}^m = 0$ and $y_{t'}^m = 0$; set $x_{i_1}^m = m_{i_1}^m - \lambda_3 + m_{t'}^m$ and $y_{i_1}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{t', i_1\}$ and set other variables to zero. **($|S^m \setminus S^{m+}|$ points)**

8. For every $t' \in S^m \setminus S^{m+}$, set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r$; set $x_{t'}^m = 0$ and $y_{t'}^m = 1$; set $x_{i_2}^m = m_{i_2}^m - \lambda_3 + m_{t'}^m$ and $y_{i_2}^m = 1$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m \setminus \{t', i_2\}$ and set other variables to zero. **($|S^m \setminus S^{m+}|$ points)**

9. For every $t' \in N \setminus S^r$, set $x_{t'}^r = 0$ and $y_{t'}^r = 1$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m$ and set other variables to zero. **($n - |S^r|$ points)**

10. For every $t' \in N \setminus S^r$, set $x_{t'}^r = \epsilon$ and $y_{t'}^r = 1$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m$ and set other variables to zero. **($n - |S^r|$ points)**

11. For every $t' \in N \setminus S^m$, set $x_{t'}^m = 0$ and $y_{t'}^m = 1$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m$ and set other variables to zero. **($n - |S^m|$ points)**

12. For every $t' \in N \setminus S^m$, set $x_{t'}^m = \epsilon$ and $y_{t'}^m = 1$; set $x_t^r = m_t^r$ and $y_t^r = 1$, $\forall t \in S^r$; set $x_t^m = m_t^m$ and $y_t^m = 1$, $\forall t \in S^m$ and set other variables to zero. **($n - |S^m|$ points)**

The affine independence proof for this inequality is also omitted for the sake of brevity.

□

Corollary 9 (**Extended flow cover inequalities (Padberg et al., 1985)**). *For $S^r, S^m \subseteq N$, let (S^r, S^m) be a pair cover for D. Also, for $L^r, L^m \subseteq N \setminus (S^r \cup S^m)$ such that $\overline{m^r} = \max_{t\in S^r} m_t^r$, $\overline{m^m} = \max_{t\in S^m} m_t^m$ then $\overline{m^c} = \max\{\overline{m^r}, \overline{m^m}\}$ and $\overline{m_t^c} = \max\{m_t^r, m_t^m, \overline{m^c}\}$ for any $t \in L^c$. Therefore, the inequality (called **returns-and-demands-extended cover inequality**) is valid for X^s.*

$$\sum_{t\in N\setminus(S^r\cup L^r)} x_t^r + \sum_{t\in N\setminus(S^m\cup L^m)} x_t^m + \sum_{t\in L^r}(\overline{m_t^c} - \lambda_3)y_t^r + \sum_{t\in L^m}(\overline{m_t^c} - \lambda_3)y_t^m \geq \sum_{t\in S^r}(m_t^r - \lambda_3)^+(1 - y_t^r) + \sum_{t\in S^m}(m_t^m - \lambda_3)^+(1 - y_t^m) \quad (4.12)$$

Proof. Firstly, this inequality (4.12) is rearranged and rewritten as:

$$\sum_{t\in N\setminus(S^r\cup L^r)} x_t^r + \sum_{t\in N\setminus(S^m\cup L^m)} x_t^m + \sum_{t\in S^{r+}}(m_t^r - \lambda_3)y_t^r + \sum_{t\in S^{m+}}(m_t^m - \lambda_3)y_t^m + \sum_{t\in L^r}(\overline{m_t^c} - \lambda_3)y_t^r + \sum_{t\in L^m}(\overline{m_t^c} - \lambda_3)y_t^m \geq \sum_{t\in S^{r+}}(m_t^r - \lambda_3) + \sum_{t\in S^{m+}}(m_t^m - \lambda_3)$$

where, $S^{r+} = \{t \in S^r | m_t^r - \lambda_3 > 0\}$ and $S^{m+} = \{t \in S^m | m_t^m - \lambda_3 > 0\}$. Suppose (x^r, x^m, y^r, y^m) be a point of X^s with $T^r = \{t \in N | y_t^r = 1\}$ and $T^m = \{t \in N | y_t^m = 1\}$. We consider four cases:

Case 1. $|S^{r+} \setminus T^r| + |S^{m+} \setminus T^m| \leq |L^r \cap T^r| + |L^m \cap T^m|$

$$\begin{aligned}
&\sum_{t\in N\setminus(S^r\cup L^r)} x_t^r + \sum_{t\in N\setminus(S^m\cup L^m)} x_t^m + \sum_{t\in S^{r+}}(m_t^r - \lambda_3)y_t^r + \sum_{t\in S^{m+}}(m_t^m - \lambda_3)y_t^m \\
&\qquad + \sum_{t\in L^r}(\overline{m_t^c} - \lambda_3)y_t^r + \sum_{t\in L^m}(\overline{m_t^c} - \lambda_3)y_t^m \\
&= \sum_{t\in(N\setminus(S^r\cup L^r))\cap T^r} x_t^r + \sum_{t\in(N\setminus(S^m\cup L^m))\cap T^m} x_t^m + \sum_{t\in S^{r+}\cap T^r}(m_t^r - \lambda_3) \\
&\qquad + \sum_{t\in S^{m+}\cap T^m}(m_t^m - \lambda_3) + \sum_{t\in L^r\cap T^r}(\overline{m_t^c} - \lambda_3) + \sum_{t\in L^m\cap T^m}(\overline{m_t^c} - \lambda_3) \\
&\geq \sum_{t\in S^{r+}}(m_t^r - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r}(m_t^r - \lambda_3) + \sum_{t\in S^{m+}}(m_t^m - \lambda_3) \\
&\qquad - \sum_{t\in S^{m+}\setminus T^m}(m_t^m - \lambda_3) + \sum_{t\in L^r\cap T^r}(\overline{m_t^c} - \lambda_3) + \sum_{t\in L^m\cap T^m}(\overline{m_t^c} - \lambda_3)
\end{aligned}$$

$$\geq \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (\overline{m^c} - \lambda_3)$$
$$- \sum_{t\in S^{m+}\setminus T^m} (\overline{m^c} - \lambda_3) + \sum_{t\in L^r\cap T^r} (\overline{m^c} - \lambda_3) + \sum_{t\in L^m\cap T^m} (\overline{m^c} - \lambda_3)$$
$$= \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3)$$
$$+ (\overline{m^c} - \lambda_3)\big(|L^r \cap T^r| + |L^m \cap T^m| - |S^{r+} \setminus T^r| - |S^{m+} \setminus T^m|\big)$$
$$\geq \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3)$$

where the first inequality uses the property of $y_t^r = 1, \forall t \in T^r$ and $y_t^m = 1, \forall t \in T^m$ as well as the simple properties of $S^{r+}\cap T^r = S^{r+}\setminus(S^{r+}\setminus T^r)$ and $S^{m+}\cap T^m = S^{m+}\setminus(S^{m+}\setminus T^m)$. Next, the second inequality considers the fact that $m_t^r < \overline{m^c} \leq \overline{m_t^c}$ and $m_t^m < \overline{m^c} \leq \overline{m_t^c}$. The last inequality takes the properties $|L^r \cap T^r| + |L^m \cap T^m| - |S^{r+} \setminus T^r| - |S^{m+} \setminus T^m|$ and $\overline{m^c} \geq \lambda_3$.

Case 2. $|S^{r+} \setminus T^r| + |S^{m+} \setminus T^m| \geq |L^r \cap T^r| + |L^m \cap T^m| + 1$

$$\sum_{t\in N\setminus(S^r\cup L^r)} x_t^r + \sum_{t\in N\setminus(S^m\cup L^m)} x_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3)y_t^r + \sum_{t\in S^{m+}} (m_t^m - \lambda_3)y_t^m$$
$$+ \sum_{t\in L^r} (\overline{m_t^c} - \lambda_3)y_t^r + \sum_{t\in L^m} (\overline{m_t^c} - \lambda_3)y_t^m$$
$$= \sum_{t\in(N\setminus(S^r\cup L^r))\cap T^r} x_t^r + \sum_{t\in(N\setminus(S^m\cup L^m))\cap T^m} x_t^m + \sum_{t\in S^{r+}\cap T^r} (m_t^r - \lambda_3)$$
$$+ \sum_{t\in S^{m+}\cap T^m} (m_t^m - \lambda_3) + \sum_{t\in L^r\cap T^r} (\overline{m_t^c} - \lambda_3) + \sum_{t\in L^m\cap T^m} (\overline{m_t^c} - \lambda_3)$$
$$= \sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m - \sum_{t\in S^r\cap T^r} x_t^r - \sum_{t\in L^r\cap T^r} x_t^r - \sum_{t\in S^m\cap T^m} x_t^m$$
$$- \sum_{t\in L^m\cap T^m} x_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3)$$
$$- \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3) + \sum_{t\in L^r\cap T^r} (\overline{m_t^c} - \lambda_3) + \sum_{t\in L^m\cap T^m} (\overline{m_t^c} - \lambda_3)$$
$$\geq \sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m - \sum_{t\in S^r\cap T^r} m_t^r - \sum_{t\in S^m\cap T^m} m_t^m - \sum_{t\in L^r\cap T^r} m_t^r$$
$$- \sum_{t\in L^m\cap T^m} m_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3)$$
$$- \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3) + \sum_{t\in L^r\cap T^r} (\overline{m_t^c} - \lambda_3) + \sum_{t\in L^m\cap T^m} (\overline{m_t^c} - \lambda_3)$$

$$
\begin{aligned}
&\geq D - \sum_{t\in S^r} m_t^r - \sum_{t\in S^m} m_t^m + \sum_{t\in S^r\setminus T^r} m_t^r + \sum_{t\in S^m\setminus T^m} m_t^m - \sum_{t\in L^r\cap T^r} m_t^r \\
&\quad - \sum_{t\in L^m\cap T^m} m_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3) \\
&\qquad - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3) + \sum_{t\in L^r\cap T^r} (\overline{m_t^c} - \lambda_3) + \sum_{t\in L^m\cap T^m} (\overline{m_t^c} - \lambda_3) \\
&\geq -\lambda_3 + \sum_{t\in S^{r+}\setminus T^r} m_t^r + \sum_{t\in S^{m+}\setminus T^m} m_t^m - \sum_{t\in L^r\cap T^r} \overline{m^c} - \sum_{t\in L^m\cap T^m} \overline{m^c} \\
&\qquad + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3) \\
&\qquad - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3) + \sum_{t\in L^r\cap T^r} (\overline{m^c} - \lambda_3) + \sum_{t\in L^m\cap T^m} (\overline{m^c} - \lambda_3) \\
&= \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) \\
&\qquad - \lambda_3 + \lambda_3(|S^{r+}\setminus T^r| + |S^{m+}\setminus T^m|) - \lambda_3(|L^r\cap T^r| + |L^m\cap T^m|) \\
&= \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) \\
&\qquad + \lambda_3 \left(|S^{r+}\setminus T^r| + |S^{m+}\setminus T^m| - |L^r\cap T^r| - |L^m\cap T^m| - 1\right) \\
&\geq \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3)
\end{aligned}
$$

where the first inequality uses the properties of $y_t^r = 1, \forall t \in T^r$, $y_t^m = 1, \forall t \in T^m$, $x_t^r \leq m_t^r y_t^r$ and $x_t^m \leq m_t^m y_t^m$. The second inequality follows that $S^r \cap T^r = S^r \setminus (S^r \setminus T^r)$, $S^m \cap T^m = S^m \setminus (S^m \setminus T^m)$ and $\sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m \geq D$. The third inequality uses the definition of λ_3, the fact that $m_t^r < \overline{m^c} \leq \overline{m_t^c}$ and $m_t^m < \overline{m^c} \leq \overline{m_t^c}$ and the properties $S^{r+} \subseteq S^r$ and $S^{m+} \subseteq S^m$. Lastly, the inequality holds the properties $|S^{r+}\setminus T^r| + |S^{m+}\setminus T^m| - |L^r\cap T^r| - |L^m\cap T^m| - 1 \geq 0$ and $\lambda_3 > 0$.

Case 3. $|S^{r+}\setminus T^r| \leq |L^r\cap T^r|$ and $|S^{m+}\setminus T^m| \geq |L^m\cap T^m| + 1$

$$
\begin{aligned}
&\sum_{t\in N\setminus(S^r\cup L^r)} x_t^r + \sum_{t\in N\setminus(S^m\cup L^m)} x_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) y_t^r + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) y_t^m \\
&\qquad + \sum_{t\in L^r} (\overline{m_t^c} - \lambda_3) y_t^r + \sum_{t\in L^m} (\overline{m_t^c} - \lambda_3) y_t^m \\
&= \sum_{t\in (N\setminus(S^r\cup L^r))\cap T^r} x_t^r + \sum_{t\in (N\setminus(S^m\cup L^m))\cap T^m} x_t^m + \sum_{t\in S^{r+}\cap T^r} (m_t^r - \lambda_3) \\
&\qquad + \sum_{t\in S^{m+}\cap T^m} (m_t^m - \lambda_3) + \sum_{t\in L^r\cap T^r} (\overline{m_t^c} - \lambda_3) + \sum_{t\in L^m\cap T^m} (\overline{m_t^c} - \lambda_3)
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m - \sum_{t\in S^r\cap T^r} x_t^r - \sum_{t\in L^r\cap T^r} x_t^r - \sum_{t\in S^m\cap T^m} x_t^m \\
&\quad - \sum_{t\in L^m\cap T^m} x_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3) \\
&\qquad - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3) + \sum_{t\in L^r\cap T^r} (\overline{m_t^c} - \lambda_3) + \sum_{t\in L^m\cap T^m} (\overline{m_t^c} - \lambda_3) \\
&\ge \sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m - \sum_{t\in S^r\cap T^r} m_t^r - \sum_{t\in S^m\cap T^m} m_t^m - \sum_{t\in L^r\cap T^r} m_t^r \\
&\quad - \sum_{t\in L^m\cap T^m} m_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3) \\
&\qquad - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3) + \sum_{t\in L^r\cap T^r} (\overline{m_t^c} - \lambda_3) + \sum_{t\in L^m\cap T^m} (\overline{m_t^c} - \lambda_3) \\
&\ge D - \sum_{t\in S^r} m_t^r - \sum_{t\in S^m} m_t^m + \sum_{t\in S^r\setminus T^r} m_t^r + \sum_{t\in S^m\setminus T^m} m_t^m - \sum_{t\in L^r\cap T^r} m_t^r \\
&\quad - \sum_{t\in L^m\cap T^m} m_t^m + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (m_t^r - \lambda_3) \\
&\qquad - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3) + \sum_{t\in L^r\cap T^r} (\overline{m_t^c} - \lambda_3) + \sum_{t\in L^m\cap T^m} (\overline{m_t^c} - \lambda_3) \\
&\ge -\lambda_3 + \sum_{t\in S^{r+}\setminus T^r} m_t^r + \sum_{t\in S^{m+}\setminus T^m} m_t^m - \sum_{t\in L^r\cap T^r} m_t^r - \sum_{t\in L^m\cap T^m} \overline{m^c} \\
&\qquad + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in S^{r+}\setminus T^r} (\overline{m^c} - \lambda_3) \\
&\qquad - \sum_{t\in S^{m+}\setminus T^m} (m_t^m - \lambda_3) + \sum_{t\in L^r\cap T^r} (\overline{m^c} - \lambda_3) + \sum_{t\in L^m\cap T^m} (\overline{m^c} - \lambda_3) \\
&= \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) + (\overline{m^c} - m_t^r - \lambda_3)\,|L^r \cap T^r| \\
&\qquad - (\overline{m^c} - m_t^r - \lambda_3)\,|S^{r+} \setminus T^r| - \lambda_3 + \lambda_3 |S^{m+} \setminus T^m| - \lambda_3 |L^m \cap T^m| \\
&= \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) \\
&\qquad + (\overline{m^c} - m_t^r - \lambda_3)\left(|L^r \cap T^r| - |S^{r+} \setminus T^r|\right) \\
&\qquad + \lambda_3 \left(|S^{m+} \setminus T^m| - |L^m \cap T^m| - 1\right) \\
&\ge \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3)
\end{aligned}
$$

where the first inequality follows the properties of $y_t^r = 1$, $\forall t \in T^r$, $y_t^m = 1$, $\forall t \in T^m$, $x_t^r \le m_t^r y_t^r$ and $x_t^m \le m_t^m y_t^m$. The second inequality uses $S^r \cap T^r = S^r \setminus (S^r \setminus T^r)$, $S^m \cap T^m = S^m \setminus (S^m \setminus T^m)$ and $\sum_{t\in N\cap T^r} x_t^r + \sum_{t\in N\cap T^m} x_t^m \ge D$. Next, the third inequality follows the definition of λ_3 and the fact that $m_t^r < \overline{m^c} \le \overline{m_t^c}$ and $m_t^m < \overline{m^c} \le \overline{m_t^c}$ and the properties $S^{r+} \subseteq S^r$ and $S^{m+} \subseteq S^m$. Finally, the last inequality makes use of the properties $|L^r \cap T^r| - |S^{r+} \setminus T^r| \geq 0$, $|S^{m+} \setminus T^m| - |L^m \cap T^m| - 1 \geq 0$, $\overline{m^c} - m_t^r \geq \lambda_3$ and $\lambda_3 > 0$ hold true.

Case 4. $|S^{r+} \setminus T^r| \geq |L^r \cap T^r| + 1$ and $|S^{m+} \setminus T^m| \leq |L^m \cap T^m|$

$$\sum_{t \in N \setminus (S^r \cup L^r)} x_t^r + \sum_{t \in N \setminus (S^m \cup L^m)} x_t^m + \sum_{t \in S^{r+}} (m_t^r - \lambda_3) y_t^r + \sum_{t \in S^{m+}} (m_t^m - \lambda_3) y_t^m$$
$$+ \sum_{t \in L^r} (\overline{m_t^c} - \lambda_3) y_t^r + \sum_{t \in L^m} (\overline{m_t^c} - \lambda_3) y_t^m$$

$$\sum_{t \in (N \setminus (S^r \cup L^r)) \cap T^r} x_t^r + \sum_{t \in (N \setminus (S^m \cup L^m)) \cap T^m} x_t^m + \sum_{t \in S^{r+} \cap T^r} (m_t^r - \lambda_3)$$
$$+ \sum_{t \in S^{m+} \cap T^m} (m_t^m - \lambda_3) + \sum_{t \in L^r \cap T^r} (\overline{m_t^c} - \lambda_3) + \sum_{t \in L^m \cap T^m} (\overline{m_t^c} - \lambda_3)$$
$$= \sum_{t \in N \cap T^r} x_t^r + \sum_{t \in N \cap T^m} x_t^m - \sum_{t \in S^r \cap T^r} x_t^r - \sum_{t \in L^r \cap T^r} x_t^r - \sum_{t \in S^m \cap T^m} x_t^m$$
$$- \sum_{t \in L^m \cap T^m} x_t^m + \sum_{t \in S^{r+}} (m_t^r - \lambda_3) + \sum_{t \in S^{m+}} (m_t^m - \lambda_3) - \sum_{t \in S^{r+} \setminus T^r} (m_t^r - \lambda_3)$$
$$- \sum_{t \in S^{m+} \setminus T^m} (m_t^m - \lambda_3) + \sum_{t \in L^r \cap T^r} (\overline{m_t^c} - \lambda_3) + \sum_{t \in L^m \cap T^m} (\overline{m_t^c} - \lambda_3)$$
$$\geq \sum_{t \in N \cap T^r} x_t^r + \sum_{t \in N \cap T^m} x_t^m - \sum_{t \in S^r \cap T^r} m_t^r - \sum_{t \in S^m \cap T^m} m_t^m - \sum_{t \in L^r \cap T^r} m_t^r$$
$$- \sum_{t \in L^m \cap T^m} m_t^m + \sum_{t \in S^{r+}} (m_t^r - \lambda_3) + \sum_{t \in S^{m+}} (m_t^m - \lambda_3) - \sum_{t \in S^{r+} \setminus T^r} (m_t^r - \lambda_3)$$
$$- \sum_{t \in S^{m+} \setminus T^m} (m_t^m - \lambda_3) + \sum_{t \in L^r \cap T^r} (\overline{m_t^c} - \lambda_3) + \sum_{t \in L^m \cap T^m} (\overline{m_t^c} - \lambda_3)$$
$$\geq D - \sum_{t \in S^r} m_t^r - \sum_{t \in S^m} m_t^m + \sum_{t \in S^r \setminus T^r} m_t^r + \sum_{t \in S^m \setminus T^m} m_t^m - \sum_{t \in L^r \cap T^r} m_t^r$$
$$- \sum_{t \in L^m \cap T^m} m_t^m + \sum_{t \in S^{r+}} (m_t^r - \lambda_3) + \sum_{t \in S^{m+}} (m_t^m - \lambda_3) - \sum_{t \in S^{r+} \setminus T^r} (m_t^r - \lambda_3)$$
$$- \sum_{t \in S^{m+} \setminus T^m} (m_t^m - \lambda_3) + \sum_{t \in L^r \cap T^r} (\overline{m_t^c} - \lambda_3) + \sum_{t \in L^m \cap T^m} (\overline{m_t^c} - \lambda_3)$$
$$\geq -\lambda_3 + \sum_{t \in S^{r+} \setminus T^r} m_t^r + \sum_{t \in S^{m+} \setminus T^m} m_t^m - \sum_{t \in L^r \cap T^r} \overline{m^c} - \sum_{t \in L^m \cap T^m} m_t^m$$
$$+ \sum_{t \in S^{r+}} (m_t^r - \lambda_3) + \sum_{t \in S^{m+}} (m_t^m - \lambda_3) - \sum_{t \in S^{r+} \setminus T^r} (m_t^r - \lambda_3)$$
$$- \sum_{t \in S^{m+} \setminus T^m} (\overline{m^c} - \lambda_3) + \sum_{t \in L^r \cap T^r} (\overline{m^c} - \lambda_3) + \sum_{t \in L^m \cap T^m} (\overline{m^c} - \lambda_3)$$
$$= \sum_{t \in S^{r+}} (m_t^r - \lambda_3) + \sum_{t \in S^{m+}} (m_t^m - \lambda_3) - \lambda_3 + \lambda_3 |S^{r+} \setminus T^r| - \lambda_3 |L^r \cap T^r|$$
$$+ (\overline{m^c} - m_t^m - \lambda_3) |L^m \cap T^m| - (\overline{m^c} - m_t^m - \lambda_3) |S^{m+} \setminus T^m|$$

$$= \sum_{t \in S^{r+}} (m_t^r - \lambda_3) + \sum_{t \in S^{m+}} (m_t^m - \lambda_3) + \lambda_3 \left(|S^{r+} \setminus T^r| - |L^r \cap T^r| - 1\right)$$
$$+ (\overline{m^c} - m_t^m - \lambda_3) \left(|L^m \cap T^m| - |S^{m+} \setminus T^m|\right)$$
$$\geq \sum_{t \in S^{r+}} (m_t^r - \lambda_3) + \sum_{t \in S^{m+}} (m_t^m - \lambda_3)$$

where the first inequality follows the properties of $y_t^r = 1$, $\forall t \in T^r$, $y_t^m = 1$, $\forall t \in T^m$, $x_t^r \leq m_t^r y_t^r$ and $x_t^m \leq m_t^m y_t^m$. Next, by using $S^r \cap T^r = S^r \setminus (S^r \setminus T^r)$, $S^m \cap T^m = S^m \setminus (S^m \setminus T^m)$ and $\sum_{t \in N \cap T^r} x_t^r + \sum_{t \in N \cap T^m} x_t^m \geq D$, we obtain the second inequality. The third inequality follows the definition of λ_3 and and the fact that $m_t^r < \overline{m^c} \leq \overline{m_t^c}$ and $m_t^m < \overline{m^c} \leq \overline{m_t^c}$ and the properties $S^{r+} \subseteq S^r$ and $S^{m+} \subseteq S^m$. The properties $|S^{r+} \setminus T^r| - |L^r \cap T^r| - 1 \geq 0$, $|L^m \cap T^m| - |S^{m+} \setminus T^m| \geq 0$, $\overline{m^c} - m_t^m \geq \lambda$ and $\lambda_3 > 0$ are used to generate the last inequality.

□

Note that the extended version of the following inequalities is valid for $conv(X^s)$ however, it is not facet-defining. To conclude this section, we note that the feasible region of the basic formulation for ELSRs is now updated with additional flow cover inequalities and hence can be written as:

$$X_{fc}^{ss} = \{(x^r, x^m, y^r, y^m, I^r, I^s) | (1.8), (1.9), (1.11) - (1.15), (3.1) - (3.4), (4.7) - (4.12)\}$$

with the objective function $Z_{fc}^{ss} = \min\{(1.7) | (x^r, x^m, y^r, y^m, I^r, I^s) \in X^s\}$. In the next section, we will discuss the separation procedures for all proposed valid inequalities.

4.4 The Separation Problems for $conv(X^s)$

In order to use class of valid inequalities in a cutting plane algorithm, one needs a separation algorithm. Given a solution to the linear relaxation of $(\ell, S) - like$ inequalities, $(x^{r*}, x^{m*}, y^{r*}, y^{m*}, I^{r*}, I^{s*}) \in X_{LS}^{ss}$, we can either finding an inequality from the class violated by the solution or proving that all inequalities from the class are satisfied by the given solution.

This section provides the exact separation algorithms for valid inequalities described in the previous sections. These separation algorithms are then computationally tested to examine the strength of the violated cuts generated by each inequality rather than their computational efficiency. We note that without loss of generality, all problem parameters are assumed to be integer valued.

Firstly, we discuss the separation algorithms of the flow cover inequalities for the case $\leq$ described by (4.7). Generally, there are two ways of generating the most violated inequalities; either define the objective function as a minimization problem as studied by Padberg et al. (1985) or alternatively as a maximization problem discussed by Doostmohammadi (2014).

In this study, we rewrite the inequality (4.7) as a maximization problem.

$$\sum_{t \in S^r} \left(x_t^r + (m_t^r - \lambda_1)^+ (1 - y_t^r) \right) \leq R,$$

where S^r is a cover with $\lambda_1 > 0$. Then, we solve the following knapsack problem in order to find the most violated inequalities that cuts off the fractional points $(x^{r*}, x^{m*}, y^{r*}, y^{m*}, I^{r*}, I^{s*})$,

$$f^r = \max \left\{ \sum_{t \in N} \varphi_t(\lambda_1) u_t^r | \sum_{t \in N} m_t^r u_t^r = R + \lambda_1;\ u_t^r \in \{0, 1\},\ \forall t \in N \right\},$$

where $\varphi_t(\lambda_1) = x_t^{r*} + (m_t^r - \lambda_1)^+ (1 - y_t^{r*})$, the u_t^r variable ensures the set $S^r \neq \emptyset$ such that

$$u_t^r = \begin{cases} 1, & \text{the period, } t \text{ belongs to } S^r \\ 0, & \text{otherwise} \end{cases}$$

and $\lambda_1 \in [1, \sum_{t \in S^r} m_t^r - R]$. From this, we test whether $f^r > R$ as to find the most violated inequality.

Next, the inequality (4.8) can be rewritten as:

$$\sum_{t \in S^r} \left(x_t^r + (m_t^r - \lambda_1)^+ (1 - y_t^r) \right) + \sum_{t \in L^r} \left(x_t^r - (\overline{m_t^r} - \lambda_1) y_t^r \right) \leq R,$$

which is the extension of the flow cover inequalities (4.7). In order to find the most violated (S^r, L^r) flow cover facet, one can define the set L^r as:

$$L^r = \left\{ t \in N \setminus S^r | x_t^{r*} - (\overline{m_t^r} - \lambda_1) y_t^{r*} > 0 \right\}$$

such that $\overline{m_t^r} \geq \lambda_1$.

The similar approach as discussed previously can be applied for the case of $\geq$, described by (4.9). This inequality can be rewriten as follows:

$$\sum_{t \in S^m} \left(x_t^m + (m_t^m - \lambda_2)^+ (1 - y_t^m) \right) \leq \sum_{t \in N} x_t^m,$$

where S^m is a cover with $\lambda_2 > 0$. For a given value λ_2, the most violated inequalities that cuts off the fractional solutions $(x^{r*}, x^{m*}, y^{r*}, y^{m*}, I^{r*}, I^{s*})$ can be obtained by solving the following knapsack problem:

$$f^m = \max \left\{ \sum_{t \in S^m} \tau_t(\lambda_2) u_t^m | \sum_{t \in N} m_t^m u_t^m = (D - R) + \lambda_2;\ u_t^m \in \{0, 1\},\ \forall t \in N \right\},$$

where $\tau_t(\lambda_2) = x_t^{m*} + (m_t^m - \lambda_2)^+(1 - y_t^{m*})$, the u_t^m is the decision variable that determine the number of elements in the set S^m.

The first constraint indicates that the cover set S^m must be at least $(D - R)$. Then, the most violated inequality can be found if and only if $f^m > \sum_{t \in N} x_t^{m*}$ such that $\lambda_2 \in [1, \sum_{t \in S^m} m_t^m - (D - R)]$. Next, we rewrite its extended of flow cover inequality (4.10) as:

$$\sum_{t \in S^m} \left(x_t^m + (m_t^m - \lambda_2)^+(1 - y_t^m)\right) + \sum_{t \in L^m} \left(x_t^m - (\overline{m_t^m} - \lambda_2)y_t^m\right) \leq \sum_{t \in N} x_t^m$$

The most violated (S^m, L^m) flow cover facet can be found by defining the set L^m as:

$$L^m = \left\{t \in N \setminus S^m | x_t^{m*} - (\overline{m_t^m} - \lambda_2)y_t^{m*} > 0\right\}$$

such that $\overline{m_t^m} \geq \lambda_2$.

For the separation algorithms for $\geq$, defined by (4.11), we can rewrite it as:

$$\sum_{t \in S^r} x_t^r + \sum_{t \in S^m} x_t^m - \sum_{t \in S^{r+}} (m_t^r - \lambda_3)y_t^r - \sum_{t \in S^{m+}} (m_t^m - \lambda_3)y_t^m$$
$$+ \sum_{t \in S^{r+}} (m_t^r - \lambda_3) + \sum_{t \in S^{m+}} (m_t^m - \lambda_3) \leq \sum_{t \in N} (x_t^r + x_t^m),$$

where S^r and S^m are a pair cover with $\lambda_3 > 0$. For a given value of λ_3, we can solve the following knapsack problem to obtain the most violated inequalities that cuts off the fractional solutions $(x^{r*}, x^{m*}, y^{r*}, y^{m*}, I^{r*}, I^{s*})$.

$$f^c = \max \left\{ \sum_{t \in S^r} \tau_t'(\lambda_3)u_t^r + \sum_{t \in S^m} \tau_t''(\lambda_3)u_t^m | \sum_{t \in N} (m_t^r u_t^r + m_t^m u_t^m) = D + \lambda_3; \right.$$
$$\left. u_t^r + u_t^m = 1;\ u_t^r \in \{0, 1\},\ u_t^m \in \{0, 1\},\ \forall t \in N \right\},$$

where $\tau_t'(\lambda_3) = x_t^{r*} + (m_t^r - \lambda_3)^+(1 - y_t^{r*})$ and $\tau_t''(\lambda_3) = x_t^{m*} + (m_t^m - \lambda_3)^+(1 - y_t^{m*})$ and $\lambda_3 \in [1, \sum_{t \in S^r} m_t^r + \sum_{t \in S^m} m_t^m - D]$. The decision variables, u_t^r and u_t^m determine the number of elements in the sets S^r and S^m, respectively.

The first constraint denotes that the pair cover set, S^r and S^m must be at least D. The second constraint ensures only a single production will take place at a particular period t. From this, if $f^c > \sum_{t \in N}(x_t^{r*} + x_t^{m*})$, then we can obtain the most violated inequalities. Lastly, as for the extended of flow cover inequality

(4.12), we can rewrite it as:

$$\begin{aligned}\sum_{t\in S^r} x_t^r + \sum_{t\in S^m} x_t^m - \sum_{t\in S^{r+}} (m_t^r - \lambda_3) y_t^r - \sum_{t\in S^{m+}} (m_t^m - \lambda_3) y_t^m \\ + \sum_{t\in S^{r+}} (m_t^r - \lambda_3) + \sum_{t\in S^{m+}} (m_t^m - \lambda_3) - \sum_{t\in L^r} \left(x_t^r - (\overline{m_t^c} - \lambda_3) y_t^r\right) \\ - \sum_{t\in L^m} \left(x_t^m - (\overline{m_t^c} - \lambda_3) y_t^m\right) \leq \sum_{t\in N} (x_t^r + x_t^m)\end{aligned}$$

In order to find the most violated flow cover facet, the sets L^r and L^m are defined as follows.

$$L^r = \left\{t \in N \setminus S^r | x_t^{r*} - (\overline{m_t^c} - \lambda_3) y_t^{r*} > 0\right\}$$
$$L^m = \left\{t \in N \setminus S^m | x_t^{m*} - (\overline{m_t^c} - \lambda_3) y_t^{m*} > 0\right\}$$

such that $\overline{m_t^c} \geq \lambda_3$.

In the next section, we will run computational experiments for testing their effectiveness as cutting planes when incorporated in a Branch-and-Cut algorithm and compare with other MIP formulations proposed in Chapter 3.

4.5 Preliminary Computational Results

This section provides computational comparisons of the strength of the various cuts proposed in this chapter (i.e., flow cover inequalities with embedded $(\ell, S) - like$ inequalities) and the MIP formulations (i.e., FL reformulation and the $(\ell, S) - like$ inequalities addressed in Chapter 3). All the separation algorithms and mathematical models are implemented and solved using the Mosel modelling language version 7.7 of FICO (R) Xpress Optimization Suite on a PC with Intel (R) Core(TM) i7-4500U CPU 2.40 GHz processor and 8 GB RAM with no solver cuts.

To test the effectiveness of the cuts proposed, a of 540 random test instances are generated. In this study, we consider low, medium and high return variabilities. The return parameters, r_t, are generated randomly between the intervals of $[5, 15]$, $[5, 35]$ and $[5, 50]$ and the demand parameters, d_t , take values between $[10, 60]$. This results in 15 demand-returns data sets, where three possible parameter settings are replicated 5 times.

We note that the exact separation algorithms can be excessively time-consuming when the problem size gets larger; therefore, we consider small planning horizons of $n = 2$, 4, 6, 8 and 12. We also consider all test instances with a large period of 24 in order to observe the effectiveness of the cuts generated with the short periods. In contrast to Chapter 3, we assume that the setup costs for remanufacturing are at the most equal to the setup costs for manufacturing, $K_t^r \leq K_t^m$, $\forall t \in N$. This assumption is also stated in Piñeyro and Viera (2012) as in practice, the remanufacturing of used products is economically preferred over the production of new products due to the energy and raw materials savings from remanufacturing activity. Additionally, when the setup costs for remanufacturing are kept as low as possible compared to those for manufacturing, the chances of remanufacturing to occur is potentially high. In this study, the setup costs for remanufacturing range from 10, 30, 50, 90 and 200, up to a maximum of 500, which is equal to the setup costs for manufacturing. The holding costs for both product returns, h_t^r, and serviceable products, h_t^s, take values between $[0.5, 2]$ and no production costs for either remanufacturing and manufacturing processes are considered. The variation of n and setup costs results in 36 different combinations.

Tables 4.5.1 - 4.5.3 present the computational results for low, medium and high returns variabilities. The details of the tables are as follows.

- The first column lists the six tested periods, $\boldsymbol{n}$.
- The second colum provides the variants of setup costs for **R**emanufacturing.
- The next column represents the average percentage of the initial integrality gap of the LP relaxation at the root node. If all test instances are solved to optimality by all methods, the rows where the initial integrality gap is zero are omitted.
- This is followed by the average percentage of gap closed by **F**acility **L**ocation reformulation and $(\ell, S) - like$ inequalities. Note that the percentage of gap closed for all test instances provided by FL is the same for MC and SP reformulation techniques.
- Then, we present the average percentage of gap closed of $(\ell, S) - like$ with the addition of the **F**low **C**over inequalities defined earlier. The average total number of cuts generated by flow cover inequalities are also included and arranged in the following order: eturns cover (4.7), **R**eturns-**E**xtended cover (4.8), **D**emands cover (4.9), **D**emands-**E**xtended cover (4.10), **R**eturns-and-**D**emands cover (4.11) and **R**eturns-and-**D**emands-**E**xtended cover (4.12).
- The last two columns denote the pairwise comparison of the average percentage of gap closed between the $(\ell, S) - like$ + FC inequalities vs $(\ell, S) - like$ inequalities and between the $(\ell, S) - like$ + FC inequalities vs FL. Similar to Chapter 3, the "$(\ell, S) - like$ + FC vs $(\ell, S) - like$" represents how much improvement of the average percentage of gap closed provided by flow cover inequalities improves the average percentage of gap closed of $(\ell, S) - like$ inequalities. The average improvement of gap closed (%) can be defined as:

$$\text{AI (\%)} = \frac{(\ell, S) - like\text{+ FC gap closed} - (\ell, S) - like \text{ gap closed}}{(\ell, S) - like + \text{FC gap closed}} \times 100$$

The "$(\ell, S) - like$ + FC vs FL" is interpreted in a similar manner.

According to Tables 4.5.1 - 4.5.3, the average percentage of gap closed for all formulations deteriorates gradually from low returns to high returns. The cuts

Table 4.5.1: **[Low return]** Computational comparisons of the strength of different solution techniques for ELSRs problem

n	R	Root node (%)	Average of gap closed (%)									Pairwise comparisons of average gap closed (%)	
			FL	(l,s)-like	(l,s)-like + FC	Average # of cuts generated						(l,s)-like+FC vs (l,s)	(l,s)-like+FC vs FL
						R	RE	D	DE	RD	RDE		
2	10	28.8541	31.4328	31.4328	99.8529	0	0	2	0	1	0	68.5072	68.5072
	30	26.5228	40.6872	40.6872	100	0	0	2	0	1	0	59.3128	59.3128
	50	22.7890	50.0475	50.0475	100	0	0	2	0	2	0	49.9525	49.9525
	90	14.1078	66.7907	66.7907	100	0	0	2	0	2	0	33.2093	33.2093
	200	0.9273	100	100	100	0	0	0	0	0	0	0	0
4	10	23.9931	39.0948	39.0948	89.6031	0	0	3	0	1	0	56.5679	56.5679
	30	23.5822	59.2905	59.1957	98.6087	0	0	3	0	1	0	39.8882	39.7904
	50	21.8994	75.8580	75.6050	99.5070	0	0	3	0	1	0	24.0848	23.8318
	90	19.3709	94.4144	94.1655	99.8686	0	0	2	0	0	0	5.7230	5.4741
	200	12.2554	100	100	100	0	0	0	0	0	0	0	0
	500	6.3232	100	100	100	0	0	0	0	0	0	0	0
6	10	35.0957	70.9881	70.8825	76.6853	0	0	3	4	0	0	7.7267	7.5911
	30	34.9737	79.5150	79.0457	83.9535	0	1	3	4	0	0	5.9741	5.4238
	50	34.7577	85.1482	84.3085	88.3908	1	1	2	2	0	0	4.6522	3.7090
	90	33.7086	90.1658	89.0848	93.2182	2	3	0	2	0	0	4.4595	3.3214
	200	28.8887	98.8462	98.5530	99	0	1	0	0	0	0	0.6460	0.3401
	500	21.5066	100	100	100	0	0	0	0	0	0	0	0
8	10	39.8131	70.8516	70.8050	72.5591	0	0	3	6	0	0	2.9461	2.8802
	30	38.6172	77.0986	76.9250	78.2567	0	0	3	6	0	0	1.9593	1.7400
	50	37.0794	82.2884	81.9714	83.0450	1	2	5	5	0	0	1.3873	1.0082
	90	35.3551	86.9766	86.7058	87.5638	2	3	1	4	0	0	1.0277	0.7226
	200	32.8643	90.4093	90.3414	91	1	6	1	4	0	0	0.6781	0.6035
	500	24.7705	97.0644	96.9772	97	0	0	0	0	0	0	0.0785	-0.0131
12	10	53.6100	69.2926	69.1886	69.4444	0	0	1	2	0	0	0.4321	0.2653
	30	52.3521	75.3886	74.9986	75.1378	0	0	0	2	0	0	0.2139	-0.3476
	50	50.9684	79.5795	78.8544	78.9228	0	0	0	2	0	0	0.1010	-0.9031
	90	48.4052	84.4135	83.1754	83.1792	0	0	0	0	0	0	0.0053	-1.6194
	200	42.8525	90.6798	89.1121	89.1232	0	1	0	0	0	0	0.0116	-1.8462
	500	34.2976	94.4163	93.2686	93.2686	0	0	0	0	0	0	0	-1.2693
24	10	64.8175	82.1393	82.0653	82.0653	0	0	1	0	0	0	0	-0.0915
	30	64.4994	86.5893	86.2009	86.2009	0	0	1	1	0	0	0	-0.4580
	50	63.9900	88.8336	88.2429	88.2429	0	0	0	1	0	0	0	-0.6787
	90	62.4372	91.6915	90.9197	90.9197	0	0	0	0	0	0	0	-0.8532
	200	59.4003	93.5604	92.2280	92.2280	0	0	0	0	0	0	0	-1.4575
	500	52.1829	94.7717	92.6187	92.6187	0	0	0	0	0	0	0	-2.3308
Average		**29.9727**	**77.9510**	**77.6336**	**90.9123**	**0**	**0**	**1**	**1**	**0**	**0**	**13.7247**	**13.3483**

Table 4.5.2: [**Medium return**] Computational comparisons of the strength of different solution techniques for ELSRs problem

n	R	Root node (%)	Average of gap closed (%)									Pairwise comparisons of average gap closed (%)	
			FL	(l,s)-like	(l,s)-like + FC	Average # of cuts generated						(l,s)-like+FC vs (l,s)-like	(l,s)-like+FC vs FL
						R	RE	D	DE	RD	RDE		
2	10	45.0950	26.9075	26.9075	94.4611	0	0	2	0	0	0	72.5786	72.5786
	30	42.8864	31.1835	31.1835	95.2767	0	0	2	0	0	0	68.3412	68.3412
	50	40.2386	34.7207	34.7207	96.4270	0	0	2	0	0	0	64.7389	64.7389
	90	32.8733	40.3537	40.3537	98.5051	0	0	2	0	0	0	59.2821	59.2821
	200	15.9223	78.7215	78.7215	100	0	0	1	0	0	0	21.2786	21.2786
4	10	49.7403	35.3968	35.3968	68.0173	0	0	2	1	0	0	52.1367	52.1367
	30	47.6494	46.0011	46.0011	74.8182	0	0	2	1	0	0	40.7644	40.7644
	50	45.2408	53.6317	53.6267	80.5717	0	0	2	1	0	0	34.9550	34.9492
	90	40.3614	58.9151	58.3087	90.0060	1	1	2	1	0	0	27.1524	26.3695
	200	29.4189	91.5988	90.6750	97	0	0	1	0	0	0	6.9608	5.8726
	500	3.2088	100	100	100	0	0	0	0	0	0	0	0
6	10	49.3599	49.9758	49.3480	60.1883	0	0	2	5	0	0	19.6737	18.4488
	30	47.9154	59.4096	57.2193	67.6241	0	0	2	6	0	0	16.8429	13.3061
	50	46.4717	66.3107	63.0358	72.9706	0	0	2	6	0	0	14.7892	9.9277
	90	43.8035	77.6025	74.6435	82.3302	0	0	2	5	0	0	10.0493	6.1446
	200	36.0323	89.7151	89.0028	91.4156	0	1	1	0	0	0	3.0128	2.1343
	500	24.3197	98.7498	98.7498	100	1	1	0	0	0	0	1.2502	1.2502
8	10	54.8327	56.2942	55.9605	67.5875	0	0	5	6	0	0	19.9872	19.2394
	30	52.2178	64.0220	63.1207	72.3168	0	0	4	5	0	0	14.8934	13.2511
	50	49.9530	69.0474	68.2947	76.2635	0	0	5	5	0	0	11.8003	10.3687
	90	46.1356	76.6449	75.2341	77.4577	1	2	3	3	0	0	3.2150	0.7057
	200	38.4081	83.8108	81.8518	83.2445	1	1	1	2	0	0	1.6415	-1.1520
	500	26.3794	91.7442	90.9589	92.2815	1	0	0	1	0	0	1.3542	0.4788
12	10	59.2999	52.0590	50.4911	51.8342	0	0	2	6	0	0	3.7069	1.2009
	30	57.5021	59.1354	58.4847	60.3081	0	0	2	8	0	0	3.6539	2.6055
	50	56.0083	64.7985	63.6031	64.9259	0	0	2	8	0	0	2.4422	0.5619
	90	53.6279	71.6369	68.8374	69.5996	0	0	1	7	0	0	1.2388	-3.0779
	200	46.4377	83.4929	78.5031	79.4322	0	0	0	5	0	0	1.1836	-5.6437
	500	34.2895	90.1208	87.3760	87.6771	0	1	0	2	0	0	0.3339	-2.8061
24	10	66.9426	61.7085	61.5330	61.7012	0	0	0	5	0	0	0.3102	0.0072
	30	66.8369	68.2630	67.7371	67.8608	0	0	0	5	0	0	0.2006	-0.6099
	50	66.4118	72.7964	71.4222	71.5121	0	0	0	7	0	0	0.1383	-1.8828
	90	64.8153	79.3885	76.8173	76.8516	0	0	0	5	0	0	0.0472	-3.3833
	200	61.4938	85.6780	82.2381	82.2381	0	0	0	0	0	0	0	-4.2394
	500	53.9197	90.1691	87.9051	87.9123	0	0	0	1	0	0	0.0084	-2.5931
Average		**41.7011**	**64.8126**	**63.7663**	**81.6484**	**0**	**0**	**1**	**3**	**0**	**0**	**21.2167**	**19.6833**

Table 4.5.3: **[High return]** Computational comparisons of the strength of different solution techniques for ELSRs problem

n	R	Root node (%)	Average of gap closed (%)									Pairwise comparisons of average gap closed (%)	
			FL	(l,s)-like	(l,s)-like + FC	Average # of cuts generated						(l,s)-like+FC vs (l,s)-like	(l,s)-like+FC vs FL
						R	RE	D	DE	RD	RDE		
2	10	57.5876	24.5649	24.5649	79.4025	0	0	1	0	0	0	71.2885	71.2885
	30	54.3917	27.2093	27.2093	80.5020	0	0	1	0	0	0	69.1753	69.1753
	50	51.1973	29.2590	29.2590	80.5231	0	0	1	0	0	0	66.8088	66.8088
	90	44.7598	31.1094	31.1094	80.7856	0	0	1	0	0	0	64.2208	64.2208
	200	22.2754	37.5051	37.5051	89.1584	0	0	1	0	0	0	39.0290	59.0290
4	10	55.3992	32.8398	32.2483	55.2681	0	0	2	1	0	0	48.4337	47.3823
	30	52.6569	41.0820	41.0024	62.1798	0	0	2	1	0	0	38.0768	37.9533
	50	50.3678	46.6797	46.6370	67.0364	0	0	2	1	0	0	32.9390	32.8480
	90	45.3242	51.0317	51.0317	73.3504	0	0	1	1	0	0	24.7479	23.8992
	200	31.5619	81.5030	80.4126	90.3511	0	0	1	0	0	0	11.5722	10.4204
	500	13.7853	100	100	100	0	0	0	0	0	0	0	0
6	10	55.6137	45.5415	44.9531	56.9709	0	0	2	3	0	0	21.4013	20.1430
	30	52.9243	53.6717	51.9857	61.9214	0	0	3	3	0	0	16.2147	12.9995
	50	50.7867	59.5644	57.2936	67.9446	0	0	3	3	0	0	15.5320	11.4530
	90	47.5418	67.9697	65.2407	73.8230	0	0	3	4	0	0	11.6275	6.9820
	200	37.4267	82.9882	81.7459	86.8705	1	2	2	2	0	0	6.0435	4.4484
	500	23.4955	97.7132	97.7132	97.9943	0	0	0	0	0	0	0.2914	0.2914
8	10	56.9692	58.4371	58.2127	62.8906	0	0	3	3	0	0	7.0414	6.7093
	30	54.5931	64.2582	63.4942	67.4552	0	0	3	3	0	0	5.8945	4.6992
	50	51.6575	68.7684	68.1761	71.4358	0	0	3	1	1	0	4.4487	3.5303
	90	47.5161	75.2219	74.0519	76.6774	0	0	2	1	0	0	3.4867	1.8891
	200	39.3639	87.0346	85.8170	86.4874	0	0	2	0	0	0	0.8083	-0.7299
	500	25.7692	93.9996	93.7480	94.9315	0	1	0	0	0	0	1.2646	0.9840
12	10	63.0503	48.3811	47.6177	53.9802	0	0	2	6	0	0	14.9078	13.6969
	30	60.8534	57.1965	55.9344	60.7815	0	0	2	5	0	0	9.7401	7.1225
	50	59.0818	62.7979	60.8937	65.0870	0	0	1	6	0	0	7.6809	4.3014
	90	55.7265	70.2096	67.4279	70.7575	0	0	1	6	0	0	5.4825	1.1469
	200	48.3331	82.5790	79.5792	81.1046	0	0	1	5	0	0	1.9588	-1.7651
	500	34.6776	88.4368	87.4606	87.5205	0	0	1	0	0	0	0.0673	-1.0873
24	10	63.3546	59.5230	59.0445	60.9084	0	0	1	8	0	0	3.0590	2.3387
	30	63.9191	67.1785	66.3262	67.5560	0	0	1	8	0	0	1.7824	0.5511
	50	64.2062	71.4966	70.1913	71.1044	0	0	1	8	0	0	1.2637	-0.5887
	90	63.9053	77.2983	74.9314	75.8265	0	0	1	5	0	0	1.1862	-2.0085
	200	61.8837	83.5224	80.6524	81.3234	0	0	0	6	0	0	0.8464	-2.7801
	500	54.0915	91.1676	88.8176	88.9633	0	0	0	4	0	0	0.1705	-2.4821
Average		**46.3577**	**59.9161**	**59.8752**	**75.5088**	**0**	**0**	**2**	**3**	**0**	**0**	**22.0763**	**21.5315**

close the gap on average more than 59% and 75% of the initial gap for extended reformulation cuts and $(\ell, S) - like$ with the addition of flow cover cuts, respectively. In contrast, the average gap closed by the cuts of all methods, FL, $(\ell, S) - like$ and $(\ell, S) - like$ with FC cuts increases when either the setup costs for remanufacturing approach the setup costs for manufacturing or the average initial gap deteriorates. When the setup costs for remanufacturing increases to the setup costs for manufacturing, remanufacturing process becomes negligible, especially in the case of low returns. In this situation, manufacturing normally dominates the entire production. We observe that this problem more closely resembles the structure of the classical uncapacitated problem; the test instances can be effectively described by FL and $(\ell, S) - like$ inequalities and hence there is little room for improvement (i.e., a small, initial gap that could be made by other cuts).

When looking at the average gap closed for each method, $(\ell, S) - like$ with the addition of flow cover cuts shows significant results in closing overall gaps when either a small number of periods or low return variability is considered compared to FL cuts and $(\ell, S) - like$ cuts. In general, the number of cover cuts generated by all various cuts proposed in this chapter is quite small. However, the number of cuts generated does not determine the effectiveness or strength of the cuts yet is more inherent for problems of different sizes. Specifically, R and RE cuts become less effective when either the number of periods or return variability increases. Further, D and DE cuts consistently make cuts in most data instances and are the most often generated inequalities in our framework. RD cuts seem to be the least violated, performs considerably better in closing the gaps in the case of low returns with a short planning horizon. This is because remanufacturing and manufacturing processes can never occur at the same time; therefore, when a small number of used products is retrieved for the production system, the decision can be made to produce new products only to satisfy demand. Finally, RDE cuts are never violated for any of the 540 instances due to the fact that they are not facet-defining.

In regard to pairwise comparison of the average percentage of\textbf{ }gap closed between the "$(\ell, S) - like$ + FC cuts vs $(\ell, S) - like$ cuts" and between "$(\ell, S) - like$ + FC cuts vs FL cuts", the $(\ell, S) - like$ with the addition of the flow cover gap closed further improves both the $(\ell, S) - like$ and FL gap closed on average more than 13\% of the initial gap. Interestingly, we find that the average gap closed of $(\ell, S) - like$ and FL are identical in some test instances. Specifically, the average improvement of gap closed for both cases increases as return variability is increased. This is because the remanufacturing process occurs more frequently if a large amount of returns is put back into the system; therefore, the flow cover cuts become significant when making effective cuts. As expected, the average improvement of gap closed between the "$(\ell, S) - like$ + FC cuts vs $(\ell, S) - like$ cuts" and between "$(\ell, S) - like$ + FC cuts vs FL cuts" declines drastically when the number of periods increases. As the number of periods increases, these cuts are difficult to generate since the structure of the problem becomes more complex.

Note that the negative value of gap closed by "$(\ell, S) - like$ + FC cuts vs FL cuts" indicates that the gap closed by FL cuts is better than the gap closed provided by $(\ell, S) - like$ + FC cuts. We observe that negative values begin to appear if a large number of periods is considered. This can be consistently observed from the average improvements of gap closed for all problems with periods of 12 and 24. The results show that the gap closed by FL cuts slightly outperforms the gap closed by $(\ell, S) - like$ + FC cuts.

Under the condition that setup costs for remanufacturing are at most equal to the setup costs for manufacturing, we conclude that the flow cover cuts with embedded $(\ell, S) - like$ inequalities outperform $(\ell, S) - like$ inequalities and reformulation techniques in almost all test instances when either a low return variability or a short-term planning horizon is considered. For a large number of periods (i.e., period of 24 and high setup costs) the reformulation technique seems to provide better gap closure since the gap closed from $(\ell, S) - like$ inequalities + FC cuts to $(\ell, S) - like$ cuts or FL cuts is decreases as the number of periods increases.

4.6 Concluding Remarks

This chapter investigates the polyhedral structure of the mixed integer set X^s arising from the case of a feasible set of ELSR with separate setups. This mixed integer set is a combination of two knapsack sets and is a variant of the well-known single-node fixed-charge set. This chapter aims to examine the strength of several families of flow cover inequalities with added $(\ell, S) - like$ inequalities introduced in this chapter and other formulations discussed in Chapter 3. In this study, we describe six families of flow cover inequalities and identify their facet-defining conditions. Then, we present comparisons of preliminary computational results between different solution techniques in order to examine their effectiveness. By assuming the setup costs for remanufacturing are at most equal to the setup costs for manufacturing, the results show that adding this combination of valid inequalities, $(\ell, S) - like$ inequalities and flow cover inequalities notably tightens the lower bounds for randomly generated instances when either a low return scenario or a short-term planning horizon is taken into account when compared to other formulations. As for future research directions corresponding to set X^s, it would be interesting to study fast separation heuristics for this mix of inequalities, to include inventory variables and capacity constraints in the formulation, and to investigate the remaining types of facet-defining inequalities generated by the PORTA software.

Chapter 5

Valid Inequalities for Economic Lot-Sizing Problems with Remanufacturing: Joint Setups Case

5.1 Introduction

In contrast to Chapter 4, this chapter investigates the polyhedral structure of a general mixed integer set arising from the feasible set of original formulation of economic lot-sizing problems with remanufacturing and joint setups addressed by Teunter et al. (2006) and Retel Helmrich et al. (2013), where remanufacturing and manufacturing operations share one production line. This general mixed integer set is also a variant of the well-known single node fixed-charge network (SNFCN) set that examines the intersection of two knapsack sets as follows:

$$X^j = \{(x^r, x^m, y) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times \mathbb{B}^n | \sum_{t \in N} x^r_t \leq R, \sum_{t \in N}(x^r_t + x^m_t) \geq D, \\ x^r_t + x^m_t \leq m_t y_t, \forall t \in N\} \quad (5.1)$$

where $R = \sum_{t=1}^n r_t$ denotes the total amount of returns, $D = \sum_{t=1}^n d_t$ is the total amount of demands and the big-M constraint is given by $m_t = d_{t,n}$ for any $t \in N$. As stated in the Propositions 13 and 14 of the flow cover inequalities and the extended flow cover inequalities for the SNFCN sets in Chapter 4, we aim to extend their well-known polyhedral results to the set X^j.

This chapter is organized as follows. First, in Section 5.2, we study the basic polyhedral properties of $conv(X^j)$ and present trivial facet-defining inequalities. Next, we discuss polyhedral analysis of $conv(X^j)$ by deriving several families of valid inequalities for $conv(X^j)$ and establish their facet-defining conditions in Section 5.3. Then, in Section 5.4, the exact separation algorithms for $conv(X^j)$ are discussed. In Section 5.5, the preliminary computational experiments are carried out to test the effectiveness of these inequalities and compare with other formulations proposed in Chapter 3. Lastly, we conclude this chapter in Section 5.6.

5.2 Properties of $conv(X^j)$

In this section, we examine the basic properties and discuss some general results on the trivial facet-defining inequalities for $conv(X^j)$. Similar to Chapter 4, without loss of generality, we make the following assumptions:

(i) $D > R$,

(ii) $\sum_{t \in N \setminus \{k\}} m_t \geq D$ for each $k \in N$,

(iii) $D = m_1 > m_2 > m_3 ... > m_n > 0$,

(iii) $\sum_{t \in N} m_t > R$.

Similarly as in Chapter 4, we note that the second assumption indicates that only manufacturing (except in a single period) will satisfy all demands and the third assumption uses the big-M parameter of ELSRj. The last assumption ensures total amount of returns is sufficient for remanufacturing. Next, we prove the full-dimensionality of $conv(X^j)$.

Proposition 22. *$dim(conv(X^j)) = 3n$.*

Proof. In order to show $dim(conv(X^j)) = 3n$, we present the following $3n + 1$ affinely independent points from $conv(X^j)$. Suppose that ϵ is a relatively small number.

1. v_0 : Set $x_t^r = 0$, $x_t^m = m_t$ and $y_t = 1$, $\forall t \in N$. **(1 point)**
2. $v_1, ..., v_n$: For every $k \in N$, set $x_k^r = \epsilon$, $x_k^m = 0$, $y_k = 1$ and set $x_t^r = 0$, $x_t^m = m_t$, $y_t = 1, \forall t \in N \setminus \{k\}$. **($n$ points)**
3. $v_{n+1}, ..., v_{2n}$: For every $k \in N$, set $x_k^r = x_k^m = 0$, $y_k = 0$ and set $x_t^r = 0$, $x_t^m = m_t$, $y_t = 1$, $\forall t \in N \setminus \{k\}$. **($n$ points)**
4. $v_{2n+1}, ..., v_{3n}$: For every $k \in N$, set $x_k^r = x_k^m = 0$, $y_k = 1$ and set $x_t^r = 0$, $x_t^m = m_t$, $y_t = 1$, $\forall t \in N \setminus \{k\}$. **($n$ points)**

The vectors, $v_0, v_1, ..., v_{3n}$ are affinely independent if the vectors $(v_i - v_0)$, $i = 1, ..., 3n$ are linearly independent or equivalently if $\sum_{i=1}^{3n} \lambda_i (v_i - v_0) = \mathbf{0}$ implies that $\lambda_1 = \lambda_2 = ... = \lambda_{3n} = 0$, where λ_i, $i = 1, ..., 3n$ are scalars. Then, we obtain

$$\begin{cases} \epsilon\,(\lambda_i) = 0, & i = 1, ..., n \\ \lambda_i = 0, & i = n+1, ..., 2n \\ m_{i-2n}(\lambda_{i-2n} + \lambda_{i-n} + \lambda_i) = 0, & i = 2n+1, ..., 3n \end{cases} \tag{5.2}$$

From these equations (5.2), the first and second equations imply that $\lambda_i = 0$, for $i = 1, ..., n$ and for $i = n+1, ..., 2n$, respectively which these solutions are substituted into third equation provides $\lambda_{2n+1} = ... = \lambda_{3n} = 0$. □

The following proposition presents the trivial facet-defining inequalities for $conv(X^j)$.

Proposition 23. *The trivial facet-defining inequalities for $conv(X^j)$ (and their facet-defining conditions if applicable) are :*

(i) $x_i^r \geq 0$, $\forall i \in N$,

(ii) $x_i^r + x_i^m \leq m_i y_i$, $\forall i \in N$,

(iii) $y_i \leq 1$, $\forall i \in N$,

(iv) $\sum_{t \in N} x_t^r \leq R$ *(when* $\sum_{t \in N \setminus \{k\}} m_t > R$ *for each* $k \in N$ *holds),*

(v) $\sum_{t \in N} x_t^r + \sum_{t \in N} x_t^m \geq D$,

(vi) $x_i^m \geq 0$, $\forall i \in N$ *(when* $\forall k \in N \setminus \{i\}$, $\sum_{t \in N \setminus \{i,k\}} m_t \geq D$ *holds).*

Proof. By using the $3n+1$ affinely independent points presented in the proof of Proposition 22, we demonstrate $3n$ affinely independent points, where each of these inequalities is enforced as an equation. For (i) and (iii), the proof is straightforward, as we remove exactly one of the $3n+1$ points, i.e., v_i and v_{n+i}, respectively gives us the necessary $3n$ points. For (ii), we remove two points, v_i and v_{2n+i} and add a new point in the form of $x_i^r = \epsilon$, $x_i^m = m_i - \epsilon$ and $y_i = 1$, $x_t^r = 0$, $x_t^m = m_t$ and $y_t = 1$, $\forall t \in N \setminus \{i\}$. For (iv), let $H^r \subset N$ such that $\sum_{t \in H^r} m_t > R$, $\exists k \in H^r$ satisfying $\sum_{t \in H^r \setminus \{k\}} m_t < R$ and $\exists \ell \notin H^r$ satisfying $m_\ell \geq m_t$, $\forall t \in H^r$. For $v_1, ..., v_n$ (except for $v_{n+1}, ..., v_{2n}$ and $v_{2n+1}, ..., v_{3n}$ such that $i \in H^r$), set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$ and set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t$, $x_k^m = 0$ and $y_k = 1$ (for $v_{n+1}, ..., v_{2n}$ and $v_{2n+1}, ..., v_{3n}$ such that $i \notin H^r$, in addition to that, set $x_i^r = x_i^m = 0$ and $y_i = 0$ and set $x_i^r = x_i^m = 0$ and $y_i = 1$, respectively). For $v_{n+1}, ..., v_{2n}$ such that $i \in H^r \setminus \{k\}$, set $x_i^r = x_i^m = 0$ and $y_i = 0$; set $x_\ell^r = m_\ell$, $x_\ell^m = 0$ and $y_\ell = 1$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{i, k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t$, $x_k^m = 0$ and $y_k = 1$. For $v_{2n+1}, ..., v_{3n}$ such that $i \in H^r \setminus \{k\}$, set $x_i^r = x_i^m = 0$ and $y_i = 1$; set $x_\ell^r = m_\ell$, $x_\ell^m = 0$ and $y_\ell = 1$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{i, k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t$, $x_k^m = 0$ and $y_k = 1$. For (v), we set $x_1^r = 0$, $x_1^m = D$ and $y_1 = 1$ (and also $x_i^r = 0$, $x_i^m = 0$, $\forall i \in N \setminus \{1\}$) in all points, except setting $x_1^r = 0$, $x_1^m = D - \epsilon$ and $y_1 = 1$ in $v_1, ..., v_n$ and $x_1^r = 0$, $x_1^m = D - m_k$ and $y_1 = 1$ (and also $x_k^r = 0$, $x_k^m = m_k$ and $y_k = 1$) in $v_{n+2}, ..., v_{2n}$, while removing points v_{n+1} and v_{2n+1}; therefore, we also add a new point in the form of $x_1^r = 0$, $x_1^m = 0$ and $y_1 = 1$, $x_t^r = 0$, $x_t^m = \left(D / \sum_{t \in N \setminus \{1\}} m_t \right) m_t$ and $y_t = 1$, $\forall t \in N \setminus \{1\}$. Lastly, for (vi), we set $x_i^r = \epsilon$ and $x_i^m = 0$ for all points, eliminate point v_{2n+i} and for any point in the set $v_{n+1}, ..., v_{3n}$ such that $x_k^r = x_k^m = 0$ and $\sum_{t \in N \setminus \{i,k\}} m_t \geq D$ holds true and then add a new point $x_i^r = x_i^m = 0$ and $y_i = 1$ and $x_t^r = 0$, $x_t^m = m_t$ and $y_t = 1$, $\forall t \in N \setminus \{i\}$. □

Next, in the following section, we study the polyhedral structure of $conv(X^j)$ by defining several families of valid inequalities.

5.3 Polyhedral Analysis of $conv(X^j)$

This section discusses several families of valid inequalities for $conv(X^j)$ with their facet-defining conditions. We firstly provide some definitions used throughout the chapter as follows.

Definition 28. The definitions of flow cover inequalities for $conv(X^j)$ are:

- A set $S \subseteq N$ is a cover for R if $\lambda_1 = \sum_{t \in S} m_t - R$.
- A set $S \subseteq N$ is a cover for $D - R$ if $\lambda_2 = \sum_{t \in S} m_t - (D - R)$.
- A set $S \subseteq N$ is a cover for D if $\lambda_3 = \sum_{t \in S} m_t - D$.

We denote $(x)^+ = \max\{x, 0\}$. The main contribution in this chapter relies upon on establishing the facet-defining conditions of several existing and new inequalities.

First, we will describe several families of valid inequalities for $conv(X^j)$ in the case of $\leq$ along with their facet-defining conditions.

Corollary 10 (**Flow cover inequalities (Padberg et al., 1985)**). *Let $S \subseteq N$ be a cover for R with $\overline{m} = \max_{t \in S} m_t > \lambda_1$. Then, the following inequality (called **returns cover inequality**) is valid for X^j.*

$$\sum_{t \in S} x_t^r + \sum_{t \in S} (m_t - \lambda_1)^+ (1 - y_t) \leq R \tag{5.3}$$

Note that the validity proof for this valid inequality can be clearly seen in Padberg et al. (1985). Now, we establish facet-defining conditions for this simple inequality.

Proposition 24. *Let $S^+ = \{t \in S | m_t - \lambda_1 > 0\}$. If $|S^+| \geq 1$, then* (5.3) *defines a facet of $conv(X^j)$.*

Proof. Suppose i_1 be any member in the set S^+ and let $\epsilon > 0$, where ϵ is a relatively small number. Now, we present $3n$ affinely independent points that satisfy $\sum_{t \in S} x_t^r + \sum_{t \in S^+} (m_t - \lambda_1)(1 - y_t) = R$.

1. For every $t' \in S^+$, set $x_{t'}^r = 0$, $x_{t'}^m = 0$ and $y_{t'} = 0$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t \in S \setminus \{t'\}$; set $x_t^r = 0$, $x_t^m = m_t$ and $y_t = 1$, $\forall t \in N \setminus S$ and set other variables to zero. **($|S^+|$ points)**
2. For every $t' \in S^+$, set $x_{t'}^r = m_{t'} - \lambda_1$, $x_{t'}^m = 0$ and $y_{t'} = 1$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t \in S \setminus \{t'\}$; set $x_t^r = 0$, $x_t^m = m_t$ and $y_t = 1$, $\forall t \in N \setminus S$ and set other variables to zero. **($|S^+|$ points)**

3. For every $t' \in S^+$, set $x^r_{t'} = m_{t'} - \lambda_1$, $x^m_{t'} = \epsilon$ and $y_{t'} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{t'\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus S$ and set other variables to zero. **($|S^+|$ points)**

4. For every $t' \in S \setminus S^+$, set $x^r_{t'} = 0$, $x^m_{t'} = \epsilon$ and $y_{t'} = 1$; set $x^r_{i_1} = m_{i_1} - \lambda_1 + m_{t'}$, $x^m_{i_1} = 0$ and $y_{i_1} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{t', i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus S$ and set other variables to zero. **($|S \setminus S^+|$ points)**

5. For every $t' \in S \setminus S^+$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_{i_1} = m_{i_1} - \lambda_1 + m_{t'}$, $x^m_{i_1} = 0$ and $y_{i_1} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{t', i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus S$ and set other variables to zero. **($|S \setminus S^+|$ points)**

6. For every $t' \in S \setminus S^+$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 0$; set $x^r_{i_1} = m_{i_1} - \lambda_1 + m_{t'}$, $x^m_{i_1} = 0$ and $y_{i_1} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{t', i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus S$ and set other variables to zero. **($|S \setminus S^+|$ points)**

7. For every $t' \in N \setminus S$, set $x^r_{t'} = \epsilon$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_{i_1} = m_{i_1} - \lambda_1$, $x^m_{i_1} = \epsilon$ such that $\epsilon \leq \lambda_1$ and $y_{i_1} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup \{t'\})$ and set other variables to zero. **($n - |S|$) points)**

8. For every $t' \in N \setminus S$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 0$; set $x^r_{i_1} = m_{i_1} - \lambda_1$, $x^m_{i_1} = \epsilon$ such that $\epsilon \leq \lambda_1$ and $y_{i_1} = 1$; set $x^r_t = m^m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup \{t'\})$ and set other variables to zero. **($n - |S|$) points)**

9. For every $t' \in N \setminus S$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_{i_1} = m_{i_1} - \lambda_1$, $x^m_{i_1} = \epsilon$ such that $\epsilon \leq \lambda_1$ and $y_{i_1} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup \{t'\})$ and set other variables to zero. **($n - |S|$) points)**

□

The extended version of this type of inequalities is discussed as follows.

Corollary 11 (**Extended flow cover inequalities (Padberg et al., 1985)**). *Let $S \subseteq N$ be a cover for R with $\overline{m} = \max_{t \in S} m_t > \lambda_1$ and $L \subseteq N \setminus S$. Then, suppose that $\overline{m_t} = \max\{m_t, \overline{m}\}$ for all $t \in L$, then the extended flow cover inequality (called* ***returns-extended cover inequality****) is valid for X^j.*

$$\sum_{t \in S \cup L} x^r_t + \sum_{t \in S} (m_t - \lambda_1)^+ (1 - y_t) - \sum_{t \in L} (\overline{m_t} - \lambda_1) y_t \leq R \tag{5.4}$$

The facet-defining conditions for this inequality (5.4) are discussed in the following proposition.

Proposition 25. *Let* $S^+ = \{t \in S | m_t - \lambda_1 > 0\}$*. If* $|S^+| \geq 1$, $\sum_{t \in N \setminus (S \cup L)} m_t + \sum_{t \in S} m_t - \overline{m} > D$ *and* $0 < \overline{m} - \lambda_1 < m_t \leq \overline{m}$ *for any* $t \in L$ *then the inequality* (5.4) *is facet-defining for* $conv(X^j)$.

Proof. This proof requires the condition that $i_1 \in S^+$ such that $\max_{t \in S} m_t = m_{i_1}$ and assume ϵ is an arbitrary small number. Next, we demonstrate $3N$ affinely independent points as follows.

1. For every $t' \in S^+$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 0$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{t'\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup L)$ and set other variables to zero. (**$|S^+|$ points**)

2. For every $t' \in S^+$, set $x^r_{t'} = m_{t'} - \lambda_1$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{t'\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup L)$ and set other variables to zero. (**$|S^+|$ points**)

3. For every $t' \in S^+$, set $x^r_{t'} = m_{t'} - \lambda_1$, $x^m_{t'} = \epsilon$ and $y_{t'} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{t'\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup L)$ and set other variables to zero. (**$|S^+|$ points**)

4. For every $t' \in S \setminus S^+$, set $x^r_{t'} = 0$, $x^m_{t'} = \epsilon$ and $y_{t'} = 1$; set $x^r_{i_1} = m_{i_1} - \lambda_1 + m_{t'}$, $x^m_{i_1} = 0$ and $y_{i_1} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{t', i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup L)$ and set other variables to zero. (**$|S \setminus S^+|$ points**)

5. For every $t' \in S \setminus S^+$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_{i_1} = m_{i_1} - \lambda_1 + m_{t'}$, $x^m_{i_1} = 0$ and $y_{i_1} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{t', i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup L)$ and set other variables to zero. (**$|S \setminus S^+|$ points**)

6. For every $t' \in S \setminus S^+$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 0$; set $x^r_{i_1} = m_{i_1} - \lambda_1 + m_{t'}$, $x^m_{i_1} = 0$ and $y_{i_1} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{t', i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup L)$ and set other variables to zero. (**$|S \setminus S^+|$ points**)

7. For every $t' \in L$, set $x^r_{t'} = \overline{m} - \lambda_1$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = 0$ and $y_{i_1} = 0$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup L)$ and set other variables to zero. (**$|L|$ points**)

8. For every $t' \in L$, set $x^r_{t'} = \overline{m} - \lambda_1$, $x^m_{t'} = \epsilon$ and $y_{t'} = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = 0$ and $y_{i_1} = 0$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup L)$ and set other variables to zero. (**$|L|$ points**)

9. For every $t' \in L$, set $x^r_{t'} = \overline{m} - \lambda_1$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = m_{i_1}$ and $y_{i_1} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup L)$ and set other variables to zero. (**$|L|$ points**)

10. For every $t' \in N \setminus (S \cup L)$, set $x^r_{t'} = \epsilon$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = 0$ and $y_{i_1} = 0$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup L \cup \{t'\})$ and set other variables to zero. **($n - |S| - |L|$ points)**
11. For every $t' \in N \setminus (S \cup L)$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 0$; set $x^r_{i_1} = m_{i_1} - \lambda_1$, $x^m_{i_1} = \epsilon$ such that $\epsilon \leq \lambda_1$ and $y_{i_1} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup L \cup \{t'\})$ and set other variables to zero. **($n - |S| - |L|$ points)**
12. For every $t' \in N \setminus (S \cup L)$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_{i_1} = m_{i_1} - \lambda_1$, $x^m_{i_1} = \epsilon$ such that $\epsilon \leq \lambda_1$ and $y_{i_1} = 1$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in N \setminus (S \cup L \cup \{t'\})$ and set other variables to zero. **($n - |S| - |L|$ points)**

□

Then, we will discuss the remaining four families of valid inequalities $conv(X^j)$ in the case of $\geq$ and identify their facet-defining conditions.

Corollary 12 (**Flow cover inequalities (Padberg et al., 1985)**). *Let $S \subseteq N$ be a cover for $D - R$ with $\overline{m} = \max_{t \in S} m_t > \lambda_2$, then the flow cover inequality (called **demands cover inequality**) is valid for X^j.*

$$\sum_{t \in N \setminus S} x^m_t \geq \sum_{t \in S} (m_t - \lambda_2)^+ (1 - y_t) \tag{5.5}$$

Proof. Using the definition of $S^+ = \{t \in S | m_t - \lambda_2 > 0\}$, this inequality can be rearranged and rewritten as:

$$\sum_{t \in N \setminus S} x^m_t + \sum_{t \in S^+} (m_t - \lambda_2) y_t \geq \sum_{t \in S^+} (m_t - \lambda_2)$$

Let (x^r, x^m, y) be a point of X^j with $T = \{t \in N | y_t = 1\}$. We consider two cases:

Case 1. $|S^+ \setminus T| = 0$. This shows that $y_t = 1$ for any $t \in S^+$. Then, we get $\sum_{t \in N} x^m_t \geq \sum_{t \in S} x^m_t \geq D - R \geq 0$.

Case 2. $|S^+ \setminus T| \geq 1$.

$$\begin{aligned}
&\sum_{t \in N \setminus S} x^m_t + \sum_{t \in S^+} (m_t - \lambda_2) y_t \\
=&\sum_{t \in N \cap T} x^m_t - \sum_{t \in S \cap T} x^m_t + \sum_{t \in S^+ \cap T} (m_t - \lambda_2) \\
\geq&\sum_{t \in N \cap T} x^m_t - \sum_{t \in S \cap T} m_t + \sum_{t \in S^+ \cap T} (m_t - \lambda_2) \\
\geq&(D - R) - \sum_{t \in S \cap T} m_t + \sum_{t \in S^+} (m_t - \lambda_2) - \sum_{t \in S^+ \setminus T} (m_t - \lambda_2)
\end{aligned}$$

$$
\begin{aligned}
&=(D-R)-\sum_{t\in S} m_t+\sum_{t\in S\setminus T} m_t+\sum_{t\in S^+}(m_t-\lambda_2)-\sum_{t\in S^+\setminus T}(m_t-\lambda_2)\\
&\geq(D-R)-\sum_{t\in S} m_t+\sum_{t\in S^+\setminus T} m_t^m+\sum_{t\in S^+}(m_t-\lambda_2)-\sum_{t\in S^+\setminus T}(m_t-\lambda_2)\\
&=(D-R)-\sum_{t\in S} m_t+\lambda_2|S^+\setminus T|+\sum_{t\in S^+}(m_t-\lambda_2)\\
&=-\lambda_2+\lambda_2|S^+\setminus T|+\sum_{t\in S^+}(m_t-\lambda_2)\\
&=\sum_{t\in S^+}(m_t-\lambda_2)+\lambda_2(|S^+\setminus T|-1)\geq\sum_{t\in S^+}(m_t-\lambda_2)
\end{aligned}
$$

where the first inequality is obtained by using the property $y_t = 1, \forall t \in T$ and the defining inequality $x_t^m \leq m_t y_t$. Next, we consider the properties $S^+ \cap T = S^+ \setminus (S^+ \setminus T)$ and the definition $\sum_{t\in N\cap T} x_t^m \geq D-R$ to generate the second inequality. The third inequality follows the property of $S^+ \subseteq S$. The last inequality uses the definition of λ_2 and $|S^+ \setminus T| - 1 \geq 0$.

□

The facet-defining conditions for this simple inequality is discussed in the next proposition.

Proposition 26. *Let $S^+ = \{t \in S | m_t - \lambda_1 > 0\}$. If $|S^+| \geq 1$, $\sum_{t\in N\setminus S} m_t > \max_{i\epsilon S} m_i - \lambda_2$ and $\sum_{t\in N} m_t > R + \max_{t\epsilon N} m_t$, then the inequality* (5.5) *is facet-defining for $conv(X^j)$.*

Proof. Let $H^r \subset N$ such that $\sum_{t\in H^r} m_t > R$, $\exists k \in H^r$ satisfying $\sum_{t\in H^r\setminus\{k\}} m_t < R$ and $\exists \ell \notin H^r$ satisfying $m_\ell \geq m_t, \forall t \in H^r$. Let i_1 be any member in the set S^+ and ϵ is an arbitrary small number. We also define $\hat{m}_t = m_t / \sum_{t\in N\setminus S} m_t$. Now, we will present $3n$ affinely independent points that satisfy this inequality as an equation.

1. For every $t' \in S^+$, set $x_{t'}^r = 0$, $x_{t'}^m = 0$ and $y_{t'} = 0$; set $x_t^r = 0$, $x_t^m = m_t$ and $y_t = 1, \forall t \in S \setminus \{t'\}$; set $x_t^r = 0$, $x_t^m = \hat{m}_t(m_{t'} - \lambda_2)$ and $y_t = 1, \forall t \in N \setminus S$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1, \forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t\in H^r\setminus\{k\}} m_t$, $x_k^m = 0$ and $y_k = 1$ and set other variables to zero. **($|S^+|$ points)**

2. For every $t' \in S^+$, set $x_{t'}^r = 0$, $x_{t'}^m = m_{t'} - \lambda_2$ and $y_{t'} = 1$; set $x_t^r = 0$, $x_t^m = m_t$ and $y_t = 1, \forall t \in S \setminus \{t'\}$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1, \forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t\in H^r\setminus\{k\}} m_t$, $x_k^m = 0$ and $y_k = 1$ and set other variables to zero. **($|S^+|$ points)**

3. For every $t' \in S^+$, set $x_{t'}^r = \epsilon$, $x_{t'}^m = m_{t'} - \lambda_2$ and $y_{t'} = 1$; set $x_t^r = 0$, $x_t^m = m_t^m$ and $y_t = 1, \forall t \in S \setminus \{t'\}$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1, \forall t \in H^r \setminus \{k\}$; set

$x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t - \epsilon$, $x_k^m = 0$ and $y_k = 1$ and set other variables to zero. **($|S^+|$ points)**

4. For every $t' \in S \setminus S^+$, set $x_{t'}^r = 0$, $x_{t'}^m = 0$ and $y_{t'} = 0$; set $x_{i_1}^r = 0$, $x_{i_1}^m = m_{i_1} - \lambda_2 + m_{t'}$ and $y_{i_1} = 1$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t$, $x_k^m = 0$ and $y_k = 1$ and set other variables to zero. **($|S \setminus S^+|$ points)**

5. For every $t' \in S \setminus S^+$, set $x_{t'}^r = 0$, $x_{t'}^m = 0$ and $y_{t'} = 1$; set $x_{i_1}^r = 0$, $x_{i_1}^m = m_{i_1} - \lambda_2 + m_{t'}$ and $y_{i_1} = 1$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t$, $x_k^m = 0$ and $y_k = 1$ and set other variables to zero. **($|S \setminus S^+|$ points)**

6. For every $t' \in S \setminus S^+$, set $x_{t'}^r = \epsilon$, $x_{t'}^m = 0$ and $y_{t'} = 1$; set $x_{i_1}^r = 0$, $x_{i_1}^m = m_{i_1} - \lambda_2 + m_{t'}$ and $y_{i_1} = 1$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t - \epsilon$, $x_k^m = 0$ and $y_k = 1$ and set other variables to zero. **($|S \setminus S^+|$ points)**

7. For every $t' \in N \setminus S$, set $x_{t'}^r = 0$, $x_{t'}^m = 0$ and $y_{t'} = 1$; set $x_{i_1}^r = 0$, $x_{i_1}^m = m_{i_1} - \lambda_2$ and $y_{i_1} = 1$; set $x_t^r = 0$, $x_t^m = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t$, $x_k^m = 0$ and $y_k = 1$ and set other variables to zero. **($n - |S|$ points)**

8. For every $t' \in N \setminus S$, set $x_{t'}^r = \epsilon$, $x_{t'}^m = 0$ and $y_{t'} = 1$; set $x_{i_1}^r = 0$, $x_{i_1}^m = m_{i_1} - \lambda_2$ and $y_{i_1} = 1$; set $x_t^r = 0$, $x_t^m = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t - \epsilon$, $x_k^m = 0$ and $y_k = 1$ and set other variables to zero. **($n - |S|$ points)**

9. For every $t' \in N \setminus S$, set $x_{t'}^r = 0$, $x_{t'}^m = \epsilon$ and $y_{t'} = 1$; set $x_{i_1}^r = 0$, $x_{i_1}^m = m_{i_1} - \lambda_2 + \epsilon$ and $y_{i_1} = 1$; set $x_t^r = 0$, $x_t^m = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t$, $x_k^m = 0$ and $y_k = 1$ and set other variables to zero. **($n - |S|$ points)**

10. For every $t' \in H^r \setminus \{k\}$, set $x_{t'}^r = 0$, $x_{t'}^m = 0$ and $y_{t'} = 0$; set $x_\ell^r = m_{t'}^r$, $x_\ell^m = 0$ and $y_\ell = 1$; set $x_{i_1}^r = 0$, $x_{i_1}^m = m_{i_1} - \lambda_2$ and $y_{i_1} = 1$; set $x_{t'}^r = 0$, $x_t^m = m_t^m$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x_k^r = R - \sum_{t \in H^r \setminus \{k\}} m_t$, $x_k^m = 0$ and $y_k = 1$ and set other variables to zero. **($|H^r| - 1$ points)**

11. Set $x_k^r = 0$, $x_k^m = 0$ and $y_k = 0$; set $x_\ell^r = R - \sum_{t \in H^r \setminus \{k\}} m_t^r$, $x_\ell^m = 0$ and $y_\ell = 1$; set $x_{i_1}^r = 0$, $x_{i_1}^m = m_{i_1} - \lambda_2$ and $y_{i_1} = 1$; set $x_t^r = 0$, $x_t^m = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{\ell, k\}$ and set other variables to zero. **(1 point)**

12. For every $t' \in H^r \setminus \{k\}$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_\ell = m^r_{t'}$, $x^m_\ell = 0$ and $y_\ell = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = m_{i_1} - \lambda_2$ and $y_{i_1} = 1$; set $x^r_{t'} = 0$, $x^m_t = m^m_t$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x^r_k = R - \sum_{t \in H^r \setminus \{k\}} m_t$, $x^m_k = 0$ and $y_k = 1$ and set other variables to zero. **($|H^r| - 1$ points)**

13. Set $x^r_k = 0$, $x^m_k = 0$ and $y_k = 1$; set $x^r_\ell = R - \sum_{t \in H^r \setminus \{k\}} m^r_t$, $x^m_\ell = 0$ and $y_\ell = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = m_{i_1} - \lambda_2$ and $y_{i_1} = 1$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{\ell, k\}$ and set other variables to zero. **(1 point)**

14. For every $t' \in H^r \setminus \{k\}$, set $x^r_{t'} = 0$, $x^m_{t'} = \epsilon$ and $y_{t'} = 1$; set $x^r_\ell = m^r_{t'}$, $x^m_\ell = 0$ and $y_\ell = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = m_{i_1} - \lambda_2 - \epsilon$ and $y_{i_1} = 1$; set $x^r_{t'} = 0$, $x^m_t = m^m_t$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x^r_k = R - \sum_{t \in H^r \setminus \{k\}} m_t$, $x^m_k = 0$ and $y_k = 1$ and set other variables to zero. **($|H^r| - 1$ points)**

15. Set $x^r_k = 0$, $x^m_k = \epsilon$ and $y_k = 1$; set $x^r_\ell = R - \sum_{t \in H^r \setminus \{k\}} m^r_t$, $x^m_\ell = 0$ and $y_\ell = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = m_{i_1} - \lambda_2 - \epsilon$ and $y_{i_1} = 1$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{\ell, k\}$ and set other variables to zero. **(1 point)**

16. For every $t' \in N \setminus H^r$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = m_{i_1} - \lambda_2$ and $y_{i_1} = 1$; set $x^r_t = 0$, $x^m_t = m^m_t$ and $y^m_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x^r_k = R - \sum_{t \in H^r \setminus \{k\}} m^r_t$, $x^m_k = 0$ and $y_k = 1$ and set other variables to zero. **($n - |H^r|$ points)**

17. For every $t' \in N \setminus H^r$, set $x^r_{t'} = \epsilon$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = m_{i_1} - \lambda_2$ and $y_{i_1} = 1$; set $x^r_t = 0$, $x^m_t = m^m_t$ and $y^m_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x^r_k = R - \sum_{t \in H^r \setminus \{k\}} m^r_t - \epsilon$, $x^m_k = 0$ and $y_k = 1$ and set other variables to zero. **($n - |H^r|$ points)**

18. For every $t' \in N \setminus H^r$, set $x^r_{t'} = 0$, $x^m_{t'} = \epsilon$ and $y_{t'} = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = m_{i_1} - \lambda_2 - \epsilon$ and $y_{i_1} = 1$; set $x^r_t = 0$, $x^m_t = m^m_t$ and $y^m_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x^r_k = R - \sum_{t \in H^r \setminus \{k\}} m^r_t$, $x^m_k = 0$ and $y_k = 1$ and set other variables to zero. **($n - |H^r|$ points)**

□

Then, the extended flow cover inequalities are derived next along with their facet-defining conditions.

Corollary 13 (Extended Flow cover inequalities (Padberg et al., 1985)). *Let $S \subseteq N$ be a cover for $D - R$ with $\overline{m} = \max_{t \in S} m_t > \lambda_2$ and $L \subseteq N \setminus S$. Assume that $\overline{m_t} = \max(\overline{m}, m_t)$ for all $t \in L$. Then, the extended flow cover inequality* *(called **demands-extended cover inequality**) is valid for X^j.*

$$\sum_{t\in N\setminus(S\cup L)} x_t^m + \sum_{t\in L}(\overline{m_t} - \lambda_2)y_t \geq \sum_{t\in S}(m_t - \lambda_2)^+(1 - y_t) \quad (5.6)$$

Proof. By using the definition of $S^+ = \{t \in S | m_t - \lambda_2 > 0\}$, we rearrange and rewrite the inequality (5.6) as:

$$\sum_{t\in N\setminus(S\cup L)} x_t^m + \sum_{t\in S^+}(m_t - \lambda_2)y_t + \sum_{t\in L}(\overline{m_t} - \lambda_2)y_t \geq \sum_{t\in S^+}(m_t - \lambda_2)$$

Suppose that (x^r, x^m, y) be a point of X^j with $T = \{t \in N | y_t = 1\}$. We show the validity of this inequality as follows:

Case 1. $|S^+ \setminus T| \leq |L \cap T|$

$$\begin{aligned}
&\sum_{t\in N\setminus(S\cup L)} x_t^m + \sum_{t\in S^+}(m_t - \lambda_2)y_t + \sum_{t\in L}(\overline{m_t} - \lambda_2)y_t \\
&= \sum_{t\in N\setminus(S\cup L)} x_t^m + \sum_{t\in S^+\cap T}(m_t - \lambda_2) + \sum_{t\in L\cap T}(\overline{m_t} - \lambda_2) \\
&\geq \sum_{t\in S^+}(m_t - \lambda_2) - \sum_{t\in S^+\setminus T}(m_t - \lambda_2) + \sum_{t\in L\cap T}(\overline{m_t} - \lambda_2) \\
&\geq \sum_{t\in S^+}(m_t - \lambda_2) - \sum_{t\in S^+\setminus T}(\overline{m} - \lambda_2) + \sum_{t\in L\cap T}(\overline{m} - \lambda_2) \\
&= \sum_{t\in S^+}(m_t - \lambda_2) + (\overline{m} - \lambda_2)\left(|L \cap T| - |S^+ \setminus T|\right) \geq \sum_{t\in S^+}(m_t - \lambda_2)
\end{aligned}$$

where the first inequality uses the properties of $y_t = 1$, $\forall t \in T$ and $S^+ \cap T = S^+ \setminus (S^+ \setminus T)$. Next, the second inequality considers the fact that $m_t \leq \overline{m} \leq \overline{m_t}$ and the last inequality obtained as a result of the properties $|L \cap T| - |S^+ \setminus T| \geq 0$ and $\overline{m} \geq \lambda_2$.

Case 2. $|S^+ \setminus T| \geq |L \cap T| + 1$

$$\begin{aligned}
&\sum_{t\in N\setminus(S\cup L)} x_t^m + \sum_{t\in S^+}(m_t - \lambda_2)y_t + \sum_{t\in L}(\overline{m_t} - \lambda_2)y_t \\
&= \sum_{t\in (N\setminus(S\cup L))\cap T} x_t^m + \sum_{t\in S^+\cap T}(m_t - \lambda_2) + \sum_{t\in L\cap T}(\overline{m_t} - \lambda_2) \\
&= \sum_{t\in N\cap T} x_t^m - \sum_{t\in S\cap T} x_t^m - \sum_{t\in L\cap T} x_t^m + \sum_{t\in S^+}(m_t - \lambda_2) \\
&\qquad - \sum_{t\in S^+\setminus T}(m_t - \lambda_2) + \sum_{t\in L\cap T}(\overline{m_t} - \lambda_2)
\end{aligned}$$

$$\geq \sum_{t\in N\cap T} x_t^m - \sum_{t\in S\cap T} m_t - \sum_{t\in L\cap T} m_t + \sum_{t\in S^+}(m_t-\lambda_2)$$
$$- \sum_{t\in S^+\setminus T}(m_t-\lambda_2) + \sum_{t\in L\cap T}(\overline{m}-\lambda_2)$$
$$\geq -\lambda_2 + \sum_{t\in S^+\setminus T} m_t + \sum_{t\in S^+}(m_t-\lambda_2) - \sum_{t\in S^+\setminus T}(m_t-\lambda_2) - \lambda_2|L\cap T|$$
$$= \sum_{t\in S^+}(m_t-\lambda_2) - \lambda_2 + \lambda_2|S^+\setminus T| - \lambda_2|L\cap T|$$
$$= \sum_{t\in S^+}(m_t-\lambda_2) + \lambda_2\left(|S^+\setminus T| - |L\cap T| - 1\right)$$
$$\geq \sum_{t\in S^+}(m_t-\lambda_2)$$

where the first and second inequalities follow the properties of $y_t = 1$, $\forall t \in T$, $S\cap T = S\setminus(S\setminus T)$, $x_t^m \leq m_t y_t$ and the fact that $m_t \leq \overline{m} \leq \overline{m_t}$ and $\sum_{t\in N\cap T} x_t^m \geq D - R$. Then, we obtain the third and last inequalities by using the definition of λ_2 and the properties $S^+ \subseteq S$, $|S^+\setminus T| - |L\cap T| - 1 \geq 0$ and $\lambda_2 > 0$.

□

Proposition 27. *Let $S^+ = \{t\in S | m_t - \lambda_1 > 0\}$. Suppose that $0 < \overline{m} - \lambda_2 < m_t \leq \overline{m}$ for any $t\in L$, $\sum\limits_{t\in N\setminus(S\cup L)} m_t > \max_{i\epsilon S} m_i - \lambda_2$ and $\sum\limits_{t\in N} m_t > R + \max_{t\epsilon N} m_t$ then the inequality* (5.6) *defines a facet for $conv(X^j)$.*

Proof. As similar to the proof of Proposition 26, let $H^r \subset N$ such that $\sum\limits_{t\in H^r} m_t > R$, $\exists k\in H^r$ satisfying $\sum\limits_{t\in H^r\setminus\{k\}} m_t < R$ and $\exists \ell \notin H^r$ satisfying $m_\ell \geq m_t$, $\forall t\in H^r$. Then, we let $i_1 \in S^+$ such that $\overline{m} = m_{i_1}$ and ϵ is an arbitrary small number. We also define $\hat{m_t} = m_t / \sum\limits_{t\in N\setminus S} m_t$ for all $t\in N\setminus S$. Note that all the affinely independent points from the proof of Proposition 26 are also valid for this case, except that set 1 of these points, the values are set for $t\in N\setminus(S\cup L)$ instead of $t\in N\setminus S$ and for set points 7, 8 and 9, the points are valid only for $t\in N\setminus(S\cup L)$. From this, we need to define $3|L^m|$ new affinely independent points in order to obtain $3n$ points as follows.

1. For every $t'\in L$, set $x_{t'}^r = 0$, $x_{t'}^m = \overline{m} - \lambda_2$ and $y_{t'} = 1$; set $x_{i_1}^r = 0$, $x_{i_1}^m = 0$ and $y_{i_1} = 0$; set $x_t^r = 0$, $x_t^m = m_t$ and $y_t = 1$, $\forall t\in S\setminus\{i_1\}$; set $x_t^r = m_t$, $x_t^m = 0$ and $y_t = 1$, $\forall t\in H^r\setminus\{k\}$; set $x_k^r = R - \sum\limits_{t\in H^r\setminus\{k\}} m_t$, $x_k^m = 0$ and $y_k = 1$ and set other variables to zero. **($|L|$ points)**

2. For every $t'\in L$, set $x_{t'}^r = \epsilon$, $x_{t'}^m = \overline{m} - \lambda_2$ and $y_{t'} = 1$; set $x_{i_1}^r = 0$, $x_{i_1}^m = 0$ and $y_{i_1} = 0$; set $x_t^r = 0$, $x_t^m = m_t$ and $y_t = 1$, $\forall t\in S\setminus\{i_1\}$; set $x_t^r = m_t$, $x_t^m = 0$

and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x^r_k = R - \sum_{t \in H^r \setminus \{k\}} m_t - \epsilon$, $x^m_k = 0$ and $y_k = 1$ and set other variables to zero. **($|L|$ points)**

3. For every $t' \in L$, set $x^r_{t'} = 0$, $x^m_{t'} = \overline{m} - \lambda_2 + \epsilon$ and $y_{t'} = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = 0$ and $y_{i_1} = 0$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$; set $x^r_t = m_t$, $x^m_t = 0$ and $y_t = 1$, $\forall t \in H^r \setminus \{k\}$; set $x^r_k = R - \sum_{t \in H^r \setminus \{k\}} m_t$, $x^m_k = 0$ and $y_k = 1$ and set other variables to zero. **($|L|$ points)**

□

Now, we derive two new flow cover inequalities and then show their validity proofs and facet-defining conditions.

Proposition 28. *Let $S \subseteq N$ be a cover for D with $\overline{m} = \max_{t \in S} m_t > \lambda_3$, then we have the following inequality (called **returns-demands cover inequality**) that is valid for X^j.*

$$\sum_{t \in N \setminus S} (x^r_t + x^m_t) \geq \sum_{t \in S} (m_t - \lambda_3)^+ (1 - y_t) \quad (5.7)$$

Proof. Likewise, we rearrange and rewrite this inequality using the definition of $S^+ = \{t \in S | m_t - \lambda_3 > 0\}$.

$$\sum_{t \in N \setminus S} (x^r_t + x^m_t) + \sum_{t \in S^+} (m_t - \lambda_3) y_t \geq \sum_{t \in S^+} (m_t - \lambda_3)$$

Given that (x^r, x^m, y) be a point of X^j with $T = \{t \in N | y_t = 1\}$. For this inequality, we consider two cases:

Case 1. $|S^+ \setminus T| = 0$. Since $y_t = 1$ for any $t \in S^+$. Then, we get $\sum_{t \in N} (x^r_t + x^m_t) \geq \sum_{t \in S} (x^r_t + x^m_t) \geq D \geq 0$.

Case 2. $|S^+ \setminus T| \geq 1$.

$$\begin{aligned}
&\sum_{t \in N \setminus S} (x^r_t + x^m_t) + \sum_{t \in S^+} (m_t - \lambda_3) y_t \\
=& \sum_{t \in N \cap T} (x^r_t + x^m_t) - \sum_{t \in S \cap T} (x^r_t + x^m_t) + \sum_{t \in S^+ \cap T} (m_t - \lambda_3) \\
\geq& \sum_{t \in N \cap T} (x^r_t + x^m_t) - \sum_{t \in S \cap T} m_t + \sum_{t \in S^+ \cap T} (m_t - \lambda_3) \\
\geq& D - \sum_{t \in S \cap T} m_t + \sum_{t \in S^+} (m_t - \lambda_3) - \sum_{t \in S^+ \setminus T} (m_t - \lambda_3) \\
=& D - \sum_{t \in S} m_t + \sum_{t \in S \setminus T} m_t + \sum_{t \in S^+} (m_t - \lambda_3) - \sum_{t \in S^+ \setminus T} (m_t - \lambda_3)
\end{aligned}$$

$$
\begin{aligned}
&\geq D - \sum_{t\in S} m_t + \sum_{t\in S^+\setminus T} m_t + \sum_{t\in S^+}(m_t-\lambda_3) - \sum_{t\in S^+\setminus T}(m_t-\lambda_3)\\
&= -\lambda_3 + \lambda_3|S^+\setminus T| + \sum_{t\in S^+}(m_t-\lambda_3)\\
&= \sum_{t\in S^+}(m_t-\lambda_3) + \lambda_3(|S^+\setminus T|-1) \geq \sum_{t\in S^+}(m_t-\lambda_3)
\end{aligned}
$$

where the property $y_t = 1$, $\forall t \in T$ and the defining inequality $x^r_t + x^m_t \leq m_t y_t$ are used to obtain the first inequality. Then, the second inequality follows the property of $S^+ \cap T = S^+ \setminus (S^+ \setminus T)$ and the definition $\sum_{t\in N\cap T}(x^r_t + x^m_t) \geq D$ and the third inequality consider the property of $S^+ \subseteq S$. The last inequality derived from the definition of λ_3 and $|S^+ \setminus T| - 1 \geq 0$.

□

Proposition 29 provides the facet-defining conditions for this inequality.

Proposition 29. *Let $S^+ = \{t \in S | m_t - \lambda_3 > 0\}$. Then, let $|S^+| \geq 1$ and $\sum_{t\in N\setminus S} m_t > \max_{t\in S} m_t - \lambda_3$, then the inequality* (5.7) *is facet-defining for conv*(X^j).

Proof. Suppose i_1 be any member in the set S^+ and assume that ϵ be a sufficiently small number. We also define $\hat{m_t} = m_t / \sum_{t\in N\setminus S} m_t$ for all $t \in N \setminus S$. Then, we present $3n$ affinely independent points that satisfy (5.7) as an equation.

1. For every $t' \in S^+$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 0$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{t'\}$; set $x^r_t = 0$, $x^m_t = \hat{m}_t(m_{t'} - \lambda_3)$ and $y_t = 1$, $\forall t \in N \setminus S$ and set other variables to zero. ($|S^+|$ **points**)

2. For every $t' \in S^+$, set $x^r_{t'} = 0$, $x^m_{t'} = m_{t'} - \lambda_3$ and $y_{t'} = 1$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{t'\}$ and set other variables to zero. ($|S^+|$ **points**)

3. For every $t' \in S^+$, set $x^r_{t'} = \epsilon$, $x^m_{t'} = m_{t'} - \lambda_3 - \epsilon$ and $y_{t'} = 1$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{t'\}$ and set other variables to zero. ($|S^+|$ **points**)

4. For every $t' \in S \setminus S^+$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 0$; set $x^r_{i_1} = 0$, $x^m_{i_1} = m_{i_1} - \lambda_3 + m_{t'}$ and $y_{i_1} = 1$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{t', i_1\}$ and set other variables to zero. ($|S \setminus S^+|$ **points**)

5. For every $t' \in S \setminus S^+$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = m_{i_1} - \lambda_3 + m_{t'}$ and $y_{i_1} = 1$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{t', i_1\}$ and set other variables to zero. ($|S \setminus S^+|$ **points**)

6. For every $t' \in S \setminus S^+$, set $x^r_{t'} = \epsilon$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = m_{i_1} - \lambda_3 + m_{t'}$ and $y_{i_1} = 1$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{t', i_1\}$ and set other variables to zero. ($|S \setminus S^+|$ **points**)

7. For every $t' \in N \setminus S$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S$ and set other variables to zero. **($n - |S|$ points)**
8. For every $t' \in N \setminus S$, set $x^r_{t'} = \epsilon$, $x^m_{t'} = 0$ and $y_{t'} = 1$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S$ and set other variables to zero. **($n - |S|$ points)**
9. For every $t' \in N \setminus S$, set $x^r_{t'} = 0$, $x^m_{t'} = 0$ and $y_{t'} = 0$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S$ and set other variables to zero. **($n - |S|$ points)**

□

Lastly, we prove the validity of the extended version of the previous inequalities and identify conditions under which these inequalities are facet defining.

Proposition 30. *Suppose that $S \subseteq N$ be a cover for D with $\overline{m} = \max_{t \in S} m_t > \lambda_3$. Also, $L \subseteq N \setminus S$, then suppose that $\overline{m_t} = \max(\overline{m}, m_t)$ for all $t \in L$. Then, the inequality (called **returns-demands-extended cover inequality**) is valid for X^j.*

$$\sum_{t \in N \setminus (S \cup L)} (x^r_t + x^m_t) + \sum_{t \in L} (\overline{m_t} - \lambda_3) y_t \geq \sum_{t \in S} (m_t - \lambda_3)^+ (1 - y_t) \quad (5.8)$$

Proof. We rearrange and rewrite the inequality (5.8) as follows using the definition of $S^+ = \{t \in S | m_t - \lambda_3 > 0\}$.

$$\sum_{t \in N \setminus (S \cup L)} (x^r_t + x^m_t) + \sum_{t \in S^+} (m_t - \lambda_3) y_t + \sum_{t \in L} (\overline{m_t} - \lambda_3) y_t \geq \sum_{t \in S^+} (m_t - \lambda_3)$$

Given that (x^r, x^m, y) be a point of X^j with $T = \{t \in N | y_t = 1\}$. Firstly, we provide the validity of this inequality.

Case 1. $|S^+ \setminus T| \leq |L \cap T|$

$$\begin{aligned}
&\sum_{t \in N \setminus (S \cup L)} (x^r_t + x^m_t) \sum_{t \in S^+} (m_t - \lambda_3) y_t + \sum_{t \in L} (\overline{m_t} - \lambda_3) y_t \\
&= \sum_{t \in N \setminus (S \cup L)} (x^r_t + x^m_t) + \sum_{t \in S^+ \cap T} (m^m_t - \lambda_3) + \sum_{t \in L \cap T} (\overline{m_t} - \lambda_3) \\
&\geq \sum_{t \in S^+} (m_t - \lambda_3) - \sum_{t \in S^+ \setminus T} (m^m_t - \lambda_3) + \sum_{t \in L \cap T} (\overline{m_t} - \lambda_3) \\
&\geq \sum_{t \in S^+} (m_t - \lambda_3) - \sum_{t \in S^+ \setminus T} (\overline{m} - \lambda_3) + \sum_{t \in L \cap T} (\overline{m} - \lambda_3) \\
&= \sum_{t \in S^+} (m_t - \lambda_3) + (\overline{m} - \lambda_3) \left(|L \cap T| - |S^+ \setminus T| \right) \geq \sum_{t \in S^+} (m_t - \lambda_3)
\end{aligned}$$

where we obtain the first inequality by using the properties $y_t = 1$, $\forall t \in T$ and $S^+ \cap T = S^+ \setminus (S^+ \setminus T)$. The second inequality follows the fact that $m_t \leq \overline{m} \leq \overline{m_t}$ and the last inequality uses the properties $|L \cap T| - |S^+ \setminus T| \geq 0$ and $\overline{m} \geq \lambda_3$.

Case 2. $|S^+ \setminus T| \geq |L \cap T| + 1$

$$
\begin{aligned}
&\sum_{t\in N\setminus(S\cup L)} (x_t^r + x_t^m) + \sum_{t\in S^+} (m_t - \lambda_3)y_t + \sum_{t\in L} (\overline{m_t} - \lambda_3)y_t \\
&= \sum_{t\in (N\setminus(S\cup L))\cap T} (x_t^r + x_t^m) + \sum_{t\in S^+\cap T} (m_t - \lambda_3) + \sum_{t\in L\cap T} (\overline{m_t} - \lambda_3) \\
&= \sum_{t\in N\cap T} (x_t^r + x_t^m) - \sum_{t\in S\cap T} (x_t^r + x_t^m) - \sum_{t\in L\cap T} (x_t^r + x_t^m) + \sum_{t\in S^+} (m_t - \lambda_3) \\
&\qquad - \sum_{t\in S^+\setminus T} (m_t - \lambda_3) + \sum_{t\in L\cap T} (\overline{m_t} - \lambda_3) \\
&\geq \sum_{t\in N\cap T} (x_t^r + x_t^m) - \sum_{t\in S\cap T} m_t - \sum_{t\in L\cap T} m_t + \sum_{t\in S^+} (m_t - \lambda_3) \\
&\qquad - \sum_{t\in S^+\setminus T} (m_t - \lambda_3) + \sum_{t\in L\cap T} (\overline{m_t} - \lambda_3) \\
&\geq D - \sum_{t\in S} m_t + \sum_{t\in S\setminus T} m_t - \sum_{t\in L\cap T} m_t + \sum_{t\in S^+} (m_t - \lambda_3) \\
&\qquad - \sum_{t\in S^+\setminus T} (m_t - \lambda_3) + \sum_{t\in L\cap T} (\overline{m_t} - \lambda_3) \\
&\geq -\lambda_3 + \sum_{t\in S^+\setminus T} m_t - \sum_{t\in L\cap T} \overline{m} + \sum_{t\in S^+} (m_t - \lambda_3) \\
&\qquad - \sum_{t\in S^+\setminus T} (m_t - \lambda_3) + \sum_{t\in L\cap T} (\overline{m} - \lambda_3) \\
&= \sum_{t\in S^+} (m_t - \lambda_3) - \lambda_3 + \lambda_3 |S^+ \setminus T| - \lambda_3 |L \cap T| \\
&= \sum_{t\in S^+} (m_t - \lambda_3) + \lambda_3 \big(|S^+ \setminus T| - |L \cap T| - 1\big) \\
&\geq \sum_{t\in S^+} (m_t - \lambda_3)
\end{aligned}
$$

where the first inequalities uses the properties $y_t = 1$, $\forall t \in T$, $S^+ \cap T = S^+ \setminus (S^+ \setminus T)$, $x_t^r + x_t^m \leq m_t y_t$. The second inequality considers the fact that $\sum_{t\in N\cap T}(x_t^r + x_t^m) \geq D$ and $S \cap T = S \setminus (S \setminus T)$. Finally, the definition of λ_3, $m_t \leq \overline{m} \leq \overline{m_t}$ and the properties $S^+ \subseteq S$, $|S^+ \setminus T| - |L \cap T| - 1 \geq 0$ and $\lambda_3 > 0$ are taking into an account to get the last two inequalities.

□

Finally, we discuss the facet-defining conditions for this inequality in the following proposition.

Proposition 31. *Let* $S^+ = \{t \in S | m_t - \lambda_3 > 0\}$. *If* $0 < \overline{m} - \lambda_3 < m_t \leq \overline{m}$ *for any* $t \in L$ *and* $\sum_{t\in N\setminus(S\cup L)} m_t + \sum_{t\in S} m_t - \overline{m} > D$ *hold true, then the inequality* (5.8) *defines* *a facet for $conv(X^j)$.*

Proof. In order to prove this inequality is a facet, we require the condition of $i_1 \in S^+$ such that $\overline{m}^m = m^m_{i_1}$ and let $\epsilon > 0$ is an arbitrary small number. We also define $\hat{m}_t = m_t / \sum_{t \in N \setminus S} m_t$. Now, we present $3n$ sets, including the valid sets 2 - 6 of Proposition 29, set of point 1, the values are set for $t \in N \setminus (S \cup L)$ instead of $t \in N \setminus S$ and for set points 7, 8 and 9, the points are valid only for $t \in N \setminus (S \cup L)$. Then, we present the remaining sets of $3|L^m|$ new affinely independent points as follows.

1. For every $t' \in L$, set $x^r_{t'} = 0$, $x^m_{t'} = \overline{m} - \lambda_3$ and $y_{t'} = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = 0$ and $y_{i_1} = 0$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$ and set other variables to zero. **($|L|$ points)**

2. For every $t' \in L$, set $x^r_{t'} = \epsilon$, $x^m_{t'} = \overline{m} - \lambda_3$ and $y_{t'} = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = 0$ and $y_{i_1} = 0$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$ and set other variables to zero. **($|L|$ points)**

3. For every $t' \in L$, set $x^r_{t'} = 0$, $x^m_{t'} = \overline{m} - \lambda_3 + \epsilon$ and $y_{t'} = 1$; set $x^r_{i_1} = 0$, $x^m_{i_1} = 0$ and $y_{i_1} = 0$; set $x^r_t = 0$, $x^m_t = m_t$ and $y_t = 1$, $\forall t \in S \setminus \{i_1\}$ and set other variables to zero. **($|L|$ points)**

□

This completes the proof of the proposition. Note that in this study, since we assume that there is no initial stock of serviceable product at the beginning of period 1, $I^s_0 = 0$ and the demand is nonnegative and nonzero in all periods, $d_t > 0$ for all $t \in N$; therefore, we have also included $y_1 = 1$ in this formulation. From this, the feasible region of the basic formulation for ELSRj along with flow cover inequalities and setup production at a first period can be described as:

$$X^{js}_{fc} = \{(x^r, x^m, y, I^r, I^s) | (1.8), (1.9), (1.13), (1.14), (1.17), (1.18), \\ (5.3) - (5.8), y_1 = 1\}$$

and the objective function of $Z^{js}_{fc} = \min \left\{ (1.16) | (x^r, x^m, y, I^r, I^s) \in X^j_{fc} \right\}$. In the next section, we will discuss the separation algorithms for the inequalities discussed previously.

5.4 The Separation Problems for $conv(X^j)$

With the aim to investigate the effectiveness of the cuts generated by each inequality discussed in the previous section, we provide the exact separation algorithms that cuts off the fractional linear relaxation points $(x^{r*}, x^{m*}, y^*, I^{r*}, I^{s*})$. Firstly, we discuss separation algorithm for the case $\leq$ and then followed by the case for $\geq$.

Note that these separation algorithms are similar to the ones presented in the Chapter 4.

Next, we discuss the separation algorithms of the flow cover inequalities for the case $\leq$. We use the similar procedures as in Chapter 4 to obtain the most violated inequalities. Suppose we consider the first inequality (5.3). We rewrite this inequality as:

$$\sum_{t\in S}\left(x_t^r + (m_t^m - \lambda_1)^+(1-y_t)\right) \leq R$$

where S is a cover with $\lambda_1 > 0$. By solving the following knapsack problem, we obtain the most violated inequalities that cut off the fractional points if and only if $f^r > R$.

$$f^r = \max\left\{\sum_{t\in N}\varphi_t(\lambda_1)u_t^r|\sum_{t\in N}m_t^m u_t^r = R+\lambda_1;\ u_t^r \in \{0,1\}\,,\ \forall t\in N\right\}$$

where $\varphi_t(\lambda_1) = x_t^{r*} + (m_t^m - \lambda_1)^+(1-y_t^*)$. We define u_t^r variables as to ensure the set $S \neq \emptyset$ such that

$$u_t^r = \begin{cases} 1, & \text{the period, } t \text{ belongs to } S \\ 0, & \text{otherwise} \end{cases}$$

and $\lambda_1 \in [1, \sum_{t\in S} m_t^m - R]$. Then, the inequality (5.4) is the extended flow cover inequalities can be rewritten as:

$$\sum_{t\in S}\left(x_t^r + (m_t^m - \lambda_1)^+(1-y_t)\right) + \sum_{t\in L}\left(x_t^r - (\overline{m_t^m} - \lambda_1)y_t\right) \leq R$$

From this, we define the set L as:

$$L = \left\{t \in N \setminus S | x_t^{r*} - (\overline{m_t^m} - \lambda_1)y_t^* > 0\right\}$$

in order to find the most violated (S, L) flow cover facet, where $\overline{m_t^m} \geq \lambda_1$.

This follows by the separation algorithms for $\geq$, defined by (5.5), we can rewrite the inequality as:

$$\sum_{t\in S}\left(x_t^m + (m_t^m - \lambda_2)^+(1-y_t)\right) \leq \sum_{t\in N} x_t^m$$

where S is a cover with $\lambda_2 > 0$. Given that the value of λ_2, we can find the most violated inequalities that cuts off the fractional solutions $(x^{r*}, x^{m*}, y^*, I^{r*}, I^{s*})$ by solving the knapsack problem as stated below:

$$f^m = \max\left\{\sum_{t\in S}\tau_t(\lambda_2)u_t^m|\sum_{t\in N}m_t^m u_t^m = (D-R)+\lambda_2;\ u_t^m \in \{0,1\},\ \forall t\in N\right\}$$

where, $\tau_t(\lambda_2) = x_t^{m*} + (m_t^m - \lambda_2)^+(1-y_t^*)$. The first constraints shows that the cover set S^m must be at least $D-R$ ($\lambda_2 = 0$) and second constraint determines the number of elements (period) in the set S^m. Then, the most violated inequality can be found if and only if $f^m > \sum_{t\in N} x_t^{m*}$ such that $\lambda_2 \in [1, \sum_{t\in S} m_t^m - (D-R)]$. Next, the extended of the flow cover inequality (5.6) can be rewritten as :

$$\sum_{t\in S}\left(x_t^m + (m_t^m-\lambda_2)^+(1-y_t)\right) + \sum_{t\in L}\left(x_t^m - (\overline{m_t^m}-\lambda_2)y_t\right) \le \sum_{t\in N}x_t^m$$

This is followed by defining the set L as:

$$L = \left\{t\in N\setminus S|x_t^{m*} - (\overline{m_t^m}-\lambda_2)y_t^* > 0\right\}$$

Then, we obtain the most violated (S,L) flow cover facet given that $\overline{m_t^m} \ge \lambda_2$. Lastly, the similar procedures of exact separation algorithms can be applied to the inequalities (5.7) and (5.8).

5.5 Preliminary Computational Results

In this section, we present the computational analysis of the strength of flow cover inequalities with added setup production during the first period and $(\ell, S)-like$ inequalities. Note that $(\ell, S)-like$ inequalities are equivalent to all reformulation techniques and provide better performance in almost all cases as discussed in Chapter 3. We implement and execute all the separation algorithms and mathematical models using Mosel language version 7.7 of FICO (R) Xpress Optimization Suite on a PC with Intel (R) Core(TM) i7-4500U CPU 2.40 GHz processor and 8 GB RAM with no solver cuts.

In this study, a total of 375 random test Instances are generated. As in Chapter 4, we also consider small planning horizons of 3, 4, 6, 8 and 12 periods since the exact separation algorithms are computationally expensive. Note that the results of Period 2 is omitted from this study because all instances tested close the initial gaps to 100% by all proposed methods. We use the same returns and demands parameter settings discussed in Chapter 4. Three return parameter settings: low, medium and high return variabilities are generated randomly between the intervals of $[5, 15]$, $[5, 35]$ and $[5, 50]$, respectively, and demand parameter is set between the values of $[10, 60]$. Further, we also use the same setup costs for both remanufacturing and manufacturing addressed in Chapter 3, namely 125, 250, 500, 1000 and 5000. This provides a total of 75 possible combinations, where five test instances are iterated for each combination. Lastly, we assume that the holding costs for both product returns and serviceable products take values between $[0.5, 2]$ and zero production costs for both the remanufacturing and manufacturing processes.

We present the computational results for different returns variabilities in Tables 5.5.1 - 5.5.3. In each table:

- The first column lists the time periods, $\boldsymbol{n}$.
- The second column indicates the variation of setups costs for remanufacturing and manufacturing.
- This is followed by the average percentage of initial integrality gap of all combinations, which is based on the LP relaxation at a root node. If all instances tested are solved to optimality by both $(\ell, S) - like$ inequalities and **F**low **C**over inequalities with added setup production during the first period, where the initial integrality gap is found to be zero, we omit these rows from the table.
- The next two columns represent the average percentage of gap closed after adding $(\ell, S) - like$ inequalities cuts and FC, respectively. In the next few columns, the average total number of cuts generated by flow cover inequalities are arranged in the following order: **R**eturns cover (5.3), **R**eturns-**E**xtended cover (5.4), **D**emands cover (5.5), **D**emands-**E**xtended cover (5.6), **R**eturns-and-**D**emands cover (5.7) and **R**eturns-and-**D**emands-**E**xtended cover (5.8).
- The last column denotes the pairwise comparison of the average percentage of gap closed of $(\ell, S) - like$ inequalities vs FC.

Based on Tables 5.5.1 - 5.5.3, we observe that the FC cuts close the gap on average more than 71% of the initial gap compared to 99% of the initial gap by $(\ell, S) - like$ cuts. Unlike the results obtained for the case of separate setups, the average percentage of gap closed by FC cuts decreases gradually when return variability increases. This is because when the amount of returns is overly low, the problem more closely resembles the structure of a single uncapacitated problem in which the remanufacturing process is almost negligible and the production of new products is carried out to fulfil the entire demand. In addition to this, we observe that the average percentage of initial integrality gap (root node) has a negative relationship with the average percentage of gap closed by FC cuts as setup costs increase. When the average percentage of initial integrality gap deteriorates steadily with the increase of setup costs, the average percentage of gap closed by FC cuts increases accordingly.

Furthermore, FC cuts close 100% of the gap in most instances if the setup costs are somewhat higher. In regard to the number of cuts generated by FC cuts for all test instances, the average number of cuts generated decreases when return variability increases and increases as the problem size gets larger; however, this does not guarantee their effectiveness in closing gaps. In general, R cuts consistently generate violated cuts in almost all instances tested. However, in terms of frequency

Table 5.5.1: [**Low return**] Computational comparisons of the strength of different solution techniques for ELSRj problem

n	Setup cost	Root node (%)	Average of gap closed (%)								Pairwise comparisons of average gap closed (%)
			(l,s)-like	FC	Average # of cuts generated						
					R	RE	D	DE	RD	RDE	(l,s)-like vs FC
3	125	32.4402	100	84.5326	1	0	1	1	2	1	15.4674
	250	16.2561	100	92.5413	1	1	1	0	1	1	7.4587
	500	5.5240	100	100	1	1	2	0	1	0	0
	1000	1.1922	100	100	1	1	1	0	1	1	0
4	125	31.7879	100	67.6264	1	1	2	1	2	3	32.3736
	250	21.6474	100	83.7137	2	1	2	1	2	3	16.2863
	500	8.5105	100	96.1494	2	2	2	0	1	3	3.8506
	1000	2.5018	100	100	1	2	3	0	1	2	0
6	125	47.2611	100	47.7071	1	1	1	4	2	6	52.2929
	250	37.9584	100	72.0611	2	2	1	3	2	6	27.9389
	500	30.8949	100	84.2385	3	4	1	3	2	6	15.7615
	1000	16.3920	100	91.5925	2	4	1	3	2	6	8.4075
	5000	1.1206	100	100	1	3	4	0	1	3	0
8	125	52.3991	99.8362	34.4299	1	2	0	4	2	9	65.5136
	250	44.6845	100	49.2247	1	2	0	4	2	10	50.7753
	500	33.3081	100	80.4398	2	3	0	5	3	12	19.5602
	1000	22.4797	100	81.7586	3	6	1	3	3	10	18.2414
	5000	2.7085	100	100	2	5	9	2	1	5	0
12	125	62.4527	99.8916	20.0548	2	4	1	2	1	3	79.9235
	250	55.8246	100	28.3092	2	4	0	3	1	7	71.6908
	500	46.6866	100	44.2365	2	5	0	6	2	12	55.7635
	1000	36.2895	100	72.1880	4	7	0	7	2	20	27.8120
	5000	9.1400	100	99.3299	5	9	9	3	0	11	0.6701
Average		**25.9770**	**99.9891**	**76.4510**	**2**	**3**	**2**	**2**	**2**	**6**	**23.5407**

Table 5.5.2: [**Medium return**] Computational comparisons of the strength of different solution techniques for ELSRj problem

n	Setup cost	Root node (%)	Average of gap closed (%)								Pairwise comparisons of average gap closed (%)
			(l,s)-like	FC	Average # of cuts generated						
					R	RE	D	DE	RD	RDE	(l,s)-like vs FC
3	125	34.7730	99.8476	80.7552	1	1	1	0	2	1	19.1215
	250	22.1414	100	88.9902	1	1	1	0	1	1	11.0098
	500	8.4589	100	100	1	0	1	0	1	0	0
	1000	2.1096	100	100	1	0	1	0	1	1	0
4	125	36.5061	99.7961	64.6165	1	0	1	1	2	3	35.2514
	250	26.1682	100	79.5254	1	0	1	1	2	3	20.4746
	500	14.7566	100	94.7395	1	1	2	0	1	3	5.2605
	1000	4.9853	100	100	1	1	2	0	1	2	0
6	125	48.4492	100	43.2104	1	2	1	2	1	5	56.7896
	250	40.4876	100	61.6123	1	2	0	1	2	6	38.3877
	500	32.5836	100	79.2012	2	1	1	2	1	5	20.7988
	1000	21.2999	100	87.9862	2	1	1	2	1	6	12.0138
	5000	1.8254	100	100	1	1	2	0	1	4	0
8	125	53.6664	99.1058	29.4593	2	1	0	1	1	6	70.2749
	250	46.0299	100	43.4144	1	1	0	1	2	7	56.5856
	500	36.8649	98.7704	67.5439	1	1	0	1	2	10	31.6153
	1000	27.5616	100	72.7555	2	1	1	1	2	9	27.2445
	5000	4.4160	100	100	2	2	5	1	1	5	0
12	125	59.7832	100	18.6212	3	4	1	2	1	3	81.3788
	250	55.9979	100	25.0257	2	4	0	1	1	4	74.9743
	500	48.7781	100	37.6506	2	3	0	1	0	8	62.3494
	1000	39.4295	99.9507	61.6399	2	3	0	2	1	15	38.3297
	5000	14.3107	100	98.4479	3	4	4	2	0	14	1.5521
Average		**28.7543**	**99.8953**	**72.4941**	**1**	**1**	**1**	**1**	**1**	**5**	**27.4299**

Table 5.5.3: **[High return]** Computational comparisons of the strength of different solution techniques for ELSRj problem

n	Setup cost	Root node (%)	Average of gap closed (%)								Pairwise comparisons of average gap closed (%)
			(l,s)-like	FC	Average # of cuts generated						
					R	RE	D	DE	RD	RDE	(l,s)-like vs FC
3	125	31.1486	99.2280	80.1339	2	1	1	0	2	1	19.2427
	250	20.4436	100	87.0060	1	1	1	0	2	1	12.9940
	500	9.6277	100	100	1	0	1	0	1	1	0
	1000	2.2617	100	100	1	0	1	0	1	1	0
4	125	34.0886	100	76.2753	1	0	0	0	2	3	23.7247
	250	27.1481	99.9325	84.3376	1	0	0	0	2	3	15.6054
	500	17.0760	100	94.2377	1	1	2	0	1	3	5.7623
	1000	5.8930	100	100	1	1	2	0	1	2	0
6	125	45.7457	99.2994	44.9722	1	2	1	1	2	5	54.7105
	250	39.0709	99.6430	60.7372	1	2	1	2	2	6	39.0452
	500	32.1200	100	75.5181	1	1	1	2	1	6	24.4819
	1000	21.0277	100	87.1819	1	1	1	1	1	6	12.8181
	5000	2.0193	100	100	1	1	2	0	1	4	0
8	125	51.0051	99.6467	27.9725	1	0	0	1	2	6	71.9284
	250	44.4627	99.1832	41.3538	1	0	0	1	1	6	58.3056
	500	36.2084	99.1955	63.8882	1	0	0	0	1	11	35.5936
	1000	26.5688	100	72.9292	1	0	0	1	2	8	27.0708
	5000	4.3452	100	100	1	1	3	0	1	5	0
12	125	60.1438	96.4902	16.7142	1	2	1	3	0	2	82.6778
	250	56.5300	98.1801	22.5516	1	1	0	3	1	5	77.0304
	500	50.7124	98.6809	32.4056	1	2	0	2	0	3	67.1612
	1000	42.6695	99.7194	50.4580	1	2	0	1	0	9	49.4000
	5000	15.4735	100	97.7324	1	1	2	2	0	13	2.2676
Average		**28.2057**	**99.5596**	**71.8761**	**1**	**1**	**1**	**1**	**1**	**4**	**27.8059**

of cuts generated, for our new flow cover inequalities, RDE cuts often effectively generate cuts in our framework. On the other hand, other cuts, D and DE cuts are the least violated in the most test instances. In contrast, these similar types of inequalities produce better than average percentages of gap closed for the case of separate setups. In regard to $(\ell, S)-like$ inequalities and extended reformulations, these also reduce the gap up to 100% in almost all instances tested when the return variability is low.

To provide more details, we examine the pairwise comparisons of the average percentage of gap closed between $(\ell, S)-like$ cuts and FC cuts. We observe that the $(\ell, S)-like$ cuts significantly improve the average percentage of gap closed of FC when a large horizon is considered. This is generally about 23% improvement in the case of low returns and jumps moderately to 27% for high returns. Furthermore, the average improvement of gap closed declines as the amount of setup costs increases. This shows that $(\ell, S)-like$ cuts become less effective for improving the average percentage of FC cuts.

In summary, all the FC cuts are more often violated and make more of an impact when return variability is low or a small number of periods is considered. Meanwhile, in many cases, the $(\ell, S)-like$ inequalities are the best solution techniques to obtain strong lower bounds for the ELSRj problem compared to the FC cuts. As a comparison, the case of separate setups shows significant results with flow cover inequalities whereas having joint setups case weakens the effect of flow cover inequalities.

5.6 Concluding Remarks

In this chapter, we study the polyhedral structure of a new mixed integer set X^j arising from the original formulation of the economic lot-sizing problem with remanufacturing and joint setups, where two knapsack sets are considered simultaneously. This mixed integer set is also a variant of the well-known single-node fixed-charge set, which was studied previously by Padberg et al. (1985). Our main aim in this chapter is to examine the strength of several families of flow cover inequalities with additional setup production during the first period, which were addressed in the present chapter, and the $(\ell, S)-like$ inequalities, which are studied in Chapter 3, for improving the lower bounds for the ELSRj problem. This study discusses six existing and new families of flow cover inequalities, along with their facet-defining conditions and separation algorithms. Then, comparisons of preliminary computational results of different solution methods are presented. The findings indicate that adding $(\ell, S)-like$ inequalities is efficient for improving the lower bounds for these randomly generated test instances. In addition, the lower bounds provided by proposed valid inequalities with additional setup production during the first period provide comparable results for a smaller number of returns and a short planning horizon. As for future research, opportunities remain for further enhancement of this mixed integer set, such as study of the remaining unknown facet-defining inequalities generated by PORTA, the use of several alternative techniques for generating families of valid inequalities, the inclusion of a capacity constraint or inventory variables in this mixed integer set, or the investigation of separation heuristics for the inequalities derived.

Chapter 6

Conclusion and Future Research

In this section, we address the contributions obtained in this thesis and discuss some future research directions.

In this thesis, we investigate two variants of the economic lot sizing problem with remanufacturing. As previously discussed, the problem with both separate and joint setups for remanufacturing and manufacturing operation must be proven to be NP-hard. First, we present theoretical and computational analysis of different well-known lower bounding techniques of the classical lot-sizing problem. We then further investigate the polyhedral structure of two mixed integer sets that arise as a relaxation of the main problems of both separate and joint setup cases. Several classes of valid inequalities for these sets were derived to obtain better lower bounds of both problems.

In Chapter 3, we present different traditional mathematical programming approaches (i.e., $(\ell, S) - like$ inequalities) along with exact separation algorithm and the reformulation techniques such as FL reformulation, MC reformulation and SP reformulation for obtaining lower bounds. Mathematical analysis is conducted to extend existing theory for a deeper understanding of the structure of the problem addressed. Computational results on a wide variety of test instances obtained from Retel Helmrich et al. (2013) are presented in order to gain a better insight into these theoretical results. In particular, our $(\ell, S) - like$ inequalities show stronger lower bounds than the (ℓ, S, WW) inequalities of Retel Helmrich et al. (2013) for both problems. One of the main reasons why we get different results is that the inequalities derived by Retel Helmrich et al. (2013) do not consider production at the first period. It is necessary to produce the products at the beginning of the planning period if the demand is always positive and no initial stock is on hand. Apart from these results, we prove that all the reformulation techniques are identical theoretically and computationally for both problems. Interestingly, in the case of joint setups, the lower bounds provided by both $(\ell, S) - like$ inequalities and all reformulation techniques are proven to be equivalent since the ELSRj problem structure more closely resembles the structure of the simple lot-sizing problem. These equivalence results have significant implications since the same optimal solution values are obtained, regardless of which data set is tested. Due to these equivalence results, we observe the computational effort associated with these formulations in order to determine which formulation is the best choice. According to our computational results, the $(\ell, S) - like$ inequalities are the most efficient among the three well-known reformulation techniques in terms of saving computation time. However, in the case of separate setups, the $(\ell, S) - like$ inequalities appear to be the least efficient formulation and this is especially true for test instances with larger planning periods and if the setup costs for remanufacturing and manufacturing are equal. Because of these differences, we further investigate the polyhedral structure of ELSRs hereafter in Chapter 4 in order to derive several classes of valid inequalities and identify their facet-defining conditions with the hope of obtaining equivalent or better lower bounds for the problem. We are also interested in describing other families of valid inequalities for the ELSRj problem to test and compare their effectiveness with the previously proposed formulations.

In Chapter 4 and 5, we study two new mixed integer sets, X^s and X^j, which arise as a relaxation (substructures) of ELSRs and ELSRj problems, respectively. Unlike the well-known single-node fixed-charge network set, these mixed integer sets simultaneously examine two knapsack sets. The polyhedral structure of the simpler mixed integer sets is studied to derive several classes of strong valid inequalities in order to include them in the original formulations. In both chapters, our main contribution relies upon establishing the facet-defining conditions of the proposed valid inequalities. We derive several existing and new classes of valid inequalities that generalize the well-known flow cover inequalities. Then, we report comparisons of the computational results of different combinations of solution techniques to test the impact of the inclusion of these inequalities in improving the lower bounds. The results indicate significant potential for improving the lower bounds on a set of randomly generated instances by adding these valid inequalities with embedded $(\ell, S) - like$ inequalities for the case of separate setups under which the setup cost for remanufacturing is at the most equal to the setup cost for manufacturing. However, $(\ell, S) - like$ inequalities remain to be the best formulation for the case of joint setup.

To summarize, we have used different traditional MIP approaches, with a focus on polyhedral analysis, to tackle these two production planning problems with remanufacturing options. In this study, we believe our contribution to these problems, which currently have limited results, can provide valuable insight and motivation for other researchers, especially in the areas of production planning and MIP, to further investigate hybrid remanufacturing-manufacturing production systems. Many intriguing and difficult questions remain unsolved and need to be addressed in the future.

One immediate future research topic is the identification of the remaining unknown facet-defining inequalities corresponding to both mixed integer sets X^s and X^j, generated by PORTA, which is used to analyse polytopes and polyhedra sets. Note that the number of unknown facet-defining inequalities generated by PORTA increases as the number of periods increases. Other extensions for this research include the study of fast separation heuristics as the exact separation algorithm for these flow cover inequalities is computationally expensive in terms of time and memory when the problem size increases. Next, one could also examine the following mixed integer set.

$$X^P = \{(x^r, x^m, y^m) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times \mathbb{B}^n | \sum_{t \in N} (x^r_t + x^m_t) \geq D,$$
$$0 < x^r_t \leq C,\ x^m_t \leq m^m_t y^m_t,\ \forall t \in N\}$$

where C denotes the capacity or resources of remanufactured products. This mixed integer set indicates that there is no remanufacturing activity occurring in the production facility. This is because OEMs outsource (or contract out) their remanufacturing activities to third-party remanufacturers, which are commonly referred to as contract remanufacturers. Normally, OEMs have a contract agreement with contract remanufacturers to remanufacture products on their behalf to minimize the risk and uncertainty issues associated with product returns. These remanufactured products are then stored in a limited storage space, C.

Other than that, it would be interesting to use a mixed integer programming (MIP)-based heuristic method to improve both computation time and lower bounds for ELSRs problem since the computation times to find an optimal solution (out of ten replications) increases as the problem size increases. This MIP-based heuristic method offers the best trade-off between quality and run time. In regard to the ELSRj problem, the overall run time required to solve big data instances are acceptable for our study.

Setup operations play a significant role in production planning in many production environments. These setup activities, which involve cost and take time can disrupt the production/service processes. Therefore, reduction in setup cost and time is necessary for continuous improvement of the production system. For further research, we can extend our original formulation for ELSRs by considering machine/labour capacity constraints since as follows:

$$a^r_k x^r_t + st^r_k y^r_t \leq C^k_t$$
$$a^m_k x^m_t + st^m_k y^m_t \leq D^k_t$$

where the setup times of machine/labour are part of the capacity constraints, the parameters a^r_k and a^m_k represent the variable processing time to produce one unit of remanufactured product and new product, respectively. The parameters st^r_k and st^m_k, respectively, the setup time to remanufacture a used product and manufacture a new product using machine/labour k, which has a capacity of C^k_t and D^k_t, respectively. The inclusion of these capacity constraints is important because the time required to prepare the necessary resource (e.g., machines, people) to perform both remanufacturing and manufacturing activities is highly variable compared to classical production. With respect to the joint setup case, we only consider a single setup variable, y_t; a_k^h is the variable processing time to produce one unit of remanufactured or new product and st_k^h indicates the setup time for the remanufacture of a used product or manufacture of a new product using machine/labour k, which has a capacity of E_t^k.

Since our models are deterministic models, there are still opportunities available to investigate uncertainty issues with regard to the amount of used products retrieved by the system, R, and the demand for both remanufactured and new products, D. Further, the used products returned to the production system are not guaranteed to be remanufactured due to greater uncertainty about quality. Finally, both problems will be more realistic if we incorporate the three sub-systems of remanufacturing (i.e. disassembly, remanufacturing and assembly processes) into the original formulation. Firstly, used products are collected from core suppliers such as a core broker, retailer/dealer or end customers. Potential used products are then cleaned, sorted and disassembled into parts (items). A visual inspection process will be performed to eliminate non-remanufacturable parts and defective or failed parts and replace them with new parts. Subsequently, remanufacturable parts are reprocessed to a like-new condition and then integrated with new parts to produce a remanufactured product. This problem scenario considers multi-item and multi-level problems, where these multiple parts (items) compete for the same resources. This general production planning for remanufacturing is complex and becomes more challenging when taking into account the manufacturing process.

References

Absi, N., Dauzère-Pérès, S., Kedad-Sidhoum, S., Penz, B., Rapine, C., 2013. Lot sizing with carbon emission constraints. European Journal of Operational Research 227 (1), 55–61.

Absi, N., Kedad-Sidhoum, S., 2007. Mip-based heuristics for multi-item capacitated lot-sizing problem with setup times and shortage costs. RAIRO-Operations Research 41 (2), 171–192.

Absi, N., Kedad-Sidhoum, S., 2009. The multi-item capacitated lot-sizing problem with safety stocks and demand shortage costs. Computers & Operations Research 36 (11), 2926–2936.

Aggarwal, A., Park, J. K. P., 1993. Improved algorithms for economic lot size problems. Operations Research 41 (3), 549–571.

Ahmadi, S. A., Zargaran, A., Mehdizadeh, A., Mortazavi, S. M. J., 2013. Remanufacturing and evaluation of al zahrawi surgical instruments, al mokhdea as scalpel handle. Galen Medical Journal 2 (1), 22–25.

Akartunalı, K., Miller, A. J., 2009. A heuristic approach for big bucket multi-level production planning problems. European Journal of Operational Research 193 (2), 396–411.

Akartunalı, K., Miller, A. J., 2012. A computational analysis of lower bounds for big bucket production planning problems. Computational Optimization and Applications 53 (3), 729–753.

Akbalik, A., Pochet, Y., 2009. Valid inequalities for the single-item capacitated lot sizing problem with step-wise costs. European Journal of Operational Research 198 (2), 412–434.

Almada-Lobo, B., Clark, A., Guimarães, L., Figueira, G., Amorim, P., 2015. Industrial insights into lot sizing and scheduling modeling. Pesquisa Operacional 35 (3), 439–464.

Armentano, V. A., França, P. M., de Toledo, F. M., 1999. A network flow model for the capacitated lot-sizing problem. Omega 27 (2), 275–284.

Atamtürk, A., Küçükyavuz, S., 2005. Lot sizing with inventory bounds and fixed costs: Polyhedral study and computation. Operations Research 53 (4), 711–730.

Atasu, A., Guide, V. D. R., Van Wassenhove, L. N., 2010. So what if remanufacturing cannibalizes my new product sales? California Management Review 52 (2), 56–76.

Baki, M. F., Chaouch, B. A., Abdul-Kader, W., 2014. A heuristic solution procedure for the dynamic lot sizing problem with remanufacturing and product recovery. Computers & Operations Research 43, 225–236.

Barany, I., Van Roy, T., Wolsey, L. A., 1984b. Uncapacitated lot-sizing: The convex hull of solutions. Springer.

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

Beraldi, P., Ghiani, G., Grieco, A., Guerriero, E., 2008. Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. Computers & Operations Research 35 (11), 3644–3656.

Bitran, G. R., Yanasse, H. H., 1982. Computational complexity of the capacitated lot size problem. Management Science 28 (10), 1174–1186.

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

Buschkühl, L., Sahling, F., Helber, S., Tempelmeier, H., 2010. Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. Or Spectrum 32 (2), 231–261.

Cao, H., Li, H., Cheng, H., Luo, Y., Yin, R., Chen, Y., 2012. A carbon efficiency approach for life-cycle carbon emission characteristics of machine tools. Journal of Cleaner Production 37, 19–28.

Chandrasekaran, C., Rajendran, C., Krishnaiah Chetty, O., Hanumanna, D., 2009. A two-phase metaheuristic approach for solving economic lot scheduling problems. International Journal of Operational Research 4 (3), 296–322.

Chang, J.-f., Zhong, Y.-x., Han, Z.-d., 2006. Optimal method of capacitated lot-sizing planning in manufacturing systems. Frontiers of Mechanical Engineering in China 1 (1), 67–70.

Chapman, A., Bartlett, C., McGill, I., Parker, D., Walsh, B., 2010. Remanufacturing in the uk: 2009 survey.
URL `http://www2.wrap.org.uk/downloads/2009REman1.d51435ff.8948.pdf`

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

Chu, C., Chu, F., Zhong, J., Yang, S., 2013. A polynomial algorithm for a lot-sizing problem with backlogging, outsourcing and limited inventory. Computers & Industrial Engineering 64 (1), 200–210.

Chu, F., Chu, C., 2007. Polynomial algorithms for single-item lot-sizing models with bounded inventory and backlogging or outsourcing. Automation Science and Engineering, IEEE Transactions on 4 (2), 233–251.

Chung, C.-S., Flynn, J., Lin, C.-H. M., 1994. An effective algorithm for the capacitated single item lot size problem. European Journal of Operational Research 75 (2), 427–440.

Chung, C.-S., Lin, C.-H. M., 1988. An o (t2) algorithm for the ni/g/ni/nd capacitated lot size problem. Management Science 34 (3), 420–426.

Clark, A., Almada-Lobo, B., Almeder, C., 2011. Lot sizing and scheduling: industrial extensions and research opportunities. International Journal of Production Research 49 (9), 2457–2461.

Danna, E., Rothberg, E., Le Pape, C., 2005. Exploring relaxation induced neighborhoods to improve mip solutions. Mathematical Programming 102 (1), 71–90.

Diaby, M., Bahl, H., Karwan, M., Zionts, S., 1992. Capacitated lot-sizing and scheduling by lagrangean relaxation. European Journal of Operational Research 59 (3), 444–458.

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

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

Doostmohammadi, M., 2014. Polyhedral study of mixed integer sets arising from inventory problems. Ph.D. thesis.

Drexl, A., Kimms, A., 1997. Lot sizing and scheduling-survey and extensions. European Journal of Operational Research 99 (2), 221–235.

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

Escalante, M. S., Marenco, J. L., del Carmen Varaldo, M., 2011. A polyhedral study of the single-item lot-sizing problem with continuous start-up costs. Electronic Notes in Discrete Mathematics 37, 261–266.

Federgruen, A., Tzur, M., 1991. A simple forward algorithm to solve general dynamic lot sizing models with n periods in 0 (n log n) or 0 (n) time. Management Science 37 (8), 909–925.

Ferguson, M., 2010. Strategic and tactical aspects of closed-loop supply chains. Vol. 8. Now Publishers Inc.

Ferrer, G., 1997. The economics of personal computer remanufacturing. Resources, Conservation and Recycling 21 (2), 79–108.

Fischetti, M., Lodi, A., 2003. Local branching. Mathematical programming 98 (1-3), 23–47.

Fleischmann, B., 1990. The discrete lot-sizing and scheduling problem. European Journal of Operational Research 44 (3), 337–348.

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

Gaafar, L., 2006. Applying genetic algorithms to dynamic lot sizing with batch ordering. Computers & Industrial Engineering 51 (3), 433–444.

Gaafar, L. K., Nassef, A. O., Aly, A. I., 2009. Fixed-quantity dynamic lot sizing using simulated annealing. The International Journal of Advanced Manufacturing Technology 41 (1-2), 122–131.

Gelders, L. F., Maes, J., van Wassenhove, L. N., 1986. A branch and bound algorithm for the multi item single level capacitated dynamic lotsizing problem. In: Multi-stage production planning and inventory control. Springer, pp. 92–108.

Gicquel, C., Minoux, M., 2014. Multi-product valid inequalities for the discrete lot-sizing and scheduling problem. Computers & Operations Research.

Gicquel, C., Minoux, M., Dallery, Y., 2008. Capacitated lot sizing models: a literature review.

Golany, B., Yang, J., Yu, G., 2001. Economic lot-sizing with remanufacturing options. IIE Transactions 33 (11), 995–1003.

Guide Jr, V., Srivastava, R., Spencer, M. S., 1997. An evaluation of capacity planning techniques in a remanufacturing environment. International Journal of Production Research 35 (1), 67–82.

Guide Jr, V. D. R., 2000. Production planning and control for remanufacturing: industry practice and research needs. Journal of Operations Management 18 (4), 467–483.

Gutiérrez, J., Sedeño-Noda, A., Colebrook, M., Sicilia, J., 2003. A new characterization for the dynamic lot size problem with bounded inventory. Computers & Operations Research 30 (3), 383–395.

Gutiérrez, J., Sedeño-Noda, A., Colebrook, M., Sicilia, J., 2008. An efficient approach for solving the lot-sizing problem with time-varying storage capacities. European Journal of Operational Research 189 (3), 682–693.

Haugen, K. K., Olstad, A., Pettersen, B. I., 2007a. The profit maximizing capacitated lot-size (pclsp) problem. European Journal of Operational Research 176 (1), 165–176.

Haugen, K. K., Olstad, A., Pettersen, B. I., 2007b. Solving large-scale profit maximization capacitated lot-size problems by heuristic methods. Journal of Mathematical Modelling and Algorithms 6 (1), 135–149.

Heuvel, W., 2004. On the complexity of the economic lot-sizing problem with remanufacturing options. Tech. rep., Erasmus School of Economics (ESE).

Hindi, K., 1995. Computationally efficient solution of the multi-item, capacitated lot-sizing problem. Computers & industrial engineering 28 (4), 709–719.

Hwang, H.-C., 2007. An efficient procedure for dynamic lot-sizing model with demand time windows. Journal of Global Optimization 37 (1), 11–26.

Ijomah, W. L., 2009. Addressing decision making for remanufacturing operations and design-for-remanufacture. International Journal of Sustainable Engineering 2 (2), 91–102.

Karimi, B., Ghomi, S. F., Wilson, J., 2003. The capacitated lot sizing problem: a review of models and algorithms. Omega 31 (5), 365–378.

Kirca, Ö., 1990. An efficient algorithm for the capacitated single item dynamic lot size problem. European Journal of Operational Research 45 (1), 15–24.

Krarup, J., Bilde, O., 1977. Plant location, set covering and economic lot size: An 0 (mn)-algorithm for structured problems. In: Numerische Methoden bei Optimierungsaufgaben Band 3. Springer, pp. 155–180.

Küçükyavuz, S., Pochet, Y., 2009. Uncapacitated lot sizing with backlogging: the convex hull. Mathematical Programming 118 (1), 151–175.

Lebreton, B., Tuma, A., 2006. A quantitative approach to assessing the profitability of car and truck tire remanufacturing. International Journal of production economics 104 (2), 639–652.

Lee, C.-Y., Çetinkaya, S., Wagelmans, A. P., 2001. A dynamic lot-sizing model with demand time windows. Management Science 47 (10), 1384–1395.

Leung, J. M., Magnanti, T. L., Vachani, R., 1989. Facets and algorithms for capacitated lot sizing. Mathematical programming 45 (1-3), 331–359.

Li, J., González, M., Zhu, Y., 2009. A hybrid simulation optimization method for production planning of dedicated remanufacturing. International Journal of Production Economics 117 (2), 286–301.

Li, X., Baki, F., Tian, P., Chaouch, B. A., 2013. A robust block-chain based tabu search algorithm for the dynamic lot sizing problem with product returns and remanufacturing. Omega.

Liu, T., 2008. Economic lot sizing problem with inventory bounds. European Journal of Operational Research 185 (1), 204–215.

Lotfi, V., Yoon, Y.-S., 1994. An algorithm for the single-item capacitated lot-sizing problem with concave production and holding costs. Journal of the Operational Research Society, 934–941.

Love, S. F., 1973. Bounded production and inventory models with piecewise concave costs. Management Science 20 (3), 313–318.

Maes, J., Van Wassenhove, L., 1988. Multi-item single-level capacitated dynamic lot-sizing heuristics: A general review. Journal of the Operational Research Society, 991–1004.

Matsumoto, M., Umeda, Y., 2011. An analysis of remanufacturing practices in japan. Journal of Remanufacturing 1 (1), 1–11.

Melo, R. A., Wolsey, L. A., 2010. Uncapacitated two-level lot-sizing. Operations Research Letters 38 (4), 241–245.

Mercé, C., Fontan, G., 2003. Mip-based heuristics for capacitated lotsizing problems. International Journal of Production Economics 85 (1), 97–111.

Okhrin, I., Richter, K., 2009. The single item dynamic lot sizing problem with minimum lot size restriction. In: Operations Research Proceedings 2008. Springer, pp. 91–96.

Okhrin, I., Richter, K., 2011. An o (t3) algorithm for the capacitated lot sizing problem with minimum order quantities. European Journal of Operational Research 211 (3), 507–514.

Önal, M., Van den Heuvel, W., Liu, T., 2012. A note on 'the economic lot sizing problem with inventory bounds'. European Journal of Operational Research 223 (1), 290–294.

Ortega, F., Wolsey, L. A., 2003. A branch-and-cut algorithm for the single-commodity, uncapacitated, fixed-charge network flow problem. Networks 41 (3), 143–158.

Özdamar, L., Bozyel, M. A., 2000. The capacitated lot sizing problem with overtime decisions and setup times. IIE transactions 32 (11), 1043–1057.

Padberg, M. W., Van Roy, T. J., Wolsey, L. A., 1985. Valid linear inequalities for fixed charge problems. Operations Research 33 (4), 842–861.

Pan, Z., Tang, J., Liu, O., 2009. Capacitated dynamic lot sizing problems in closed-loop supply chain. European Journal of Operational Research 198 (3), 810–821.

Parker, D., Riley, K., Robinson, S., Symington, H., Tewson, J., Jansson, K., Ramkumar, S., Peck, D., October 2015. Remanufacturing market study.
URL https://www.remanufacturing.eu/wp-content/uploads/2016/01/study.pdf

Parsopoulos, K. E., Konstantaras, I., Skouri, K., 2015. Metaheuristic optimization for the single-item dynamic lot sizing problem with returns and remanufacturing. Computers & Industrial Engineering 83, 307–315.

Patel, G., 2006. A Stochastic Production Cost Model for Remanufacturing Systems. The University of Texas - Pan American.
URL http://books.google.co.uk/books?id=YFXb0BI1CxEC

Piñeyro, P., Viera, O., 2012. Analysis of the quantities of the remanufacturing plan of perfect cost. Journal of Remanufacturing 2 (1), 1–8.

Pochet, Y., 1988. Valid inequalities and separation for capacitated economic lot sizing. Operations Research Letters 7 (3), 109–115.

Pochet, Y., Wolsey, L. A., 1988. Lot-size models with backlogging: Strong reformulations and cutting planes. Mathematical Programming 40 (1-3), 317–335.

Pochet, Y., Wolsey, L. A., 1994. Polyhedra for lot-sizing with wagner-whitin costs. Mathematical Programming 67 (1-3), 297–323.

Pochet, Y., Wolsey, L. A., 2006. Production planning by mixed integer programming. Springer Science & Business Media.

Pochet, Y., Wolsey, L. A., 2010. Single item lot-sizing with non-decreasing capacities. Mathematical programming 121 (1), 123–143.

Pochet, Y., Wolsey, L. A., et al., 1995. Algorithms and reformulations for lot sizing problems. DIMACS Series in Discrete Mathematics and Theoretical Computer Science 20, 245–293.

Quadt, D., Kuhn, H., 2008. Capacitated lot-sizing with extensions: a review. 4OR 6 (1), 61–83.

Rardin, R. L., Wolsey, L. A., 1993. Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems. European Journal of Operational Research 71 (1), 95–109.

Retel Helmrich, M. J., Jans, R., van den Heuvel, W., Wagelmans, A. P., 2013. Economic lot-sizing with remanufacturing: complexity and efficient formulations. IIE Transactions (just-accepted).

Richter, K., Sombrutzki, M., 2000. Remanufacturing planning for the reverse wagner/whitin models. European Journal of Operational Research 121 (2), 304–315.

Richter, K., Weber, J., 2001. The reverse wagner/whitin model with variable manufacturing and remanufacturing cost. International Journal of Production Economics 71 (1), 447–456.

Rizk, N., Martel, A., Ramudhin, A., 2006. A lagrangean relaxation algorithm for multi-item lot-sizing problems with joint piecewise linear resource costs. International Journal of Production Economics 102 (2), 344–357.

Schulz, T., 2011. A new silver–meal based heuristic for the single-item dynamic lot sizing problem with returns and remanufacturing. International Journal of Production Research 49 (9), 2519–2533.

Senyiğit, E., 2009. New heuristics to stochastic dynamic lot sizing problem. Gazi University Journal of Science 22 (2), 97–106.

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

Shi, J., Zhang, G., Sha, J., 2011. Optimal production planning for a multi-product closed loop system with uncertain demand and return. Computers & Operations Research 38 (3), 641–650.

Sifaleras, A., Konstantaras, I., Mladenović, N., 2015. Variable neighborhood search for the economic lot sizing problem with product returns and recovery. International Journal of Production Economics 160, 133–143.

Souza, G. C., 2012. Product disposition decisions on closed-loop supply chains. In: Sustainable Supply Chains. Springer, pp. 149–164.

Staggemeier, A. T., Clark, A. R., 2001. A survey of lot-sizing and scheduling models. In: 23rd annual symposium of the Brazilian operational research society (SOBRAPO). Citeseer, pp. 938–947.

Süer, G. A., Badurdeen, F., Dissanayake, N., 2008. Capacitated lot sizing by using multi-chromosome crossover strategy. Journal of Intelligent Manufacturing 19 (3), 273–282.

Tang, O., Teunter, R., 2006. Economic lot scheduling problem with returns. Production and Operations Management 15 (4), 488–497.

Taşgetiren, M. F., Liang, Y.-C., 2003. A binary particle swarm optimization algorithm for lot sizing problem. Journal of Economic and Social Research 5 (2), 1–20.

Teunter, R., Kaparis, K., Tang, O., 2008. Multi-product economic lot scheduling problem with separate production lines for manufacturing and remanufacturing. European Journal of Operational Research 191 (3), 1241–1253.

Teunter, R. H., Bayindir, Z. P., Den Heuvel, W. V., 2006. Dynamic lot sizing with product returns and remanufacturing. International Journal of Production Research 44 (20), 4377–4400.

The All-Party Parliamentary Sustainable Resource Group, March 2014. Remanufacturing: Towards a resource efficient economy.
URL http://www.policyconnect.org.uk/apsrg/research/report-remanufacturing-towards-resource-efficient-economy-0

Thierry, M., Salomon, M., Van Nunen, J., Van Wassenhove, L., 1995. Strategic issues in product recovery management. California management review 37 (2).

Toledo, F. M. B., Armentano, V. A., 2006. A lagrangian-based heuristic for the capacitated lot-sizing problem in parallel machines. European Journal of Operational Research 175 (2), 1070–1083.

Ullah, H., Parveen, S., 2010. A literature review on inventory lot sizing problems. Global Journal of Researches in Engineering 10 (5).

Van den Heuvel, W., Wagelmans, A. P., 2006. An efficient dynamic programming algorithm for a special case of the capacitated lot-sizing problem. Computers & operations research 33 (12), 3583–3599.

Van Hoesel, C., Wagelmans, A. P., Wolsey, L. A., 1994a. Polyhedral characterization of the economic lot-sizing problem with start-up costs. SIAM Journal on Discrete Mathematics 7 (1), 141–151.

Van Hoesel, C., Wagelmans, A. P. M., 1996. An o (t3) algorithm for the economic lot-sizing problem with constant capacities. Management Science 42 (1), 142–150.

Van Hoesel, S., Kuik, R., Salomon, M., Van Wassenhove, L. N., 1994b. The single-item discrete lotsizing and scheduling problem: optimization by linear and dynamic programming. Discrete Applied Mathematics 48 (3), 289–303.

Van Roy, T. J., Wolsey, L. A., 1985. Valid inequalities and separation for uncapacitated fixed charge networks. Operations Research Letters 4 (3), 105–112.

Van Vyve, M., Ortega, F., 2004. Lot-sizing with fixed charges on stocks: the convex hull. Discrete Optimization 1 (2), 189–203.

Vanderbeck, F., 1998. Lot-sizing with start-up times. Management Science 44 (10), 1409–1425.

Vyve, M., Wolsey, L. A., Yaman, H., 2014. Relaxations for two-level multi-item lot-sizing problems. Mathematical Programming: Series A and B 146 (1-2), 495–523.

Wagelmans, A., Van Hoesel, S., Kolen, A., 1992. Economic lot sizing: an o (nlogn) algorithm that runs in linear time in the wagner-whitin case. Operations Research 40 (1-supplement-1), S145–S156.

Wagner, H. M., Whitin, T. M., 1958. Dynamic version of the economic lot size model. Management science 5 (1), 89–96.

Wang, N., He, Z., Sun, J., Xie, H., Shi, W., 2011. A single-item uncapacitated lot-sizing problem with remanufacturing and outsourcing. Procedia Engineering 15, 5170–5178.

Xia, X., Govindan, K., Zhu, Q., 2015. Analyzing internal barriers for automotive parts remanufacturers in china using grey-dematel approach. Journal of Cleaner Production 87, 811–825.

Yaman, H., 2009. Polyhedral analysis for the two-item uncapacitated lot-sizing problem with one-way substitution. Discrete Applied Mathematics 157 (14), 3133–3151.

Yang, J., Golany, B., Yu, G., 2005. A concave-cost production planning problem with remanufacturing options. Naval Research Logistics (NRL) 52 (5), 443–458.

Zangwill, W. I., 1969. A backlogging model and a multi-echelon model of a dynamic economic lot size production system-a network approach. Management Science 15 (9), 506–527.

Zhao, Y., Klabjan, D., 2012. A polyhedral study of lot-sizing with supplier selection. Discrete Optimization 9 (2), 65–76.

Appendix A

$(\ell, S) - like$ Inequalities in Mosel - Separate Setups

```
model '(l,S) Inequalities for ELSR with Separate Setups'
uses 'mmxprs'
uses 'mmsystem'

declarations
 NT=25                        !number of time periods
 period=1..NT
 p_r:array(period) of real    !production cost for remanufactured products
 p_m:array(period) of real    !production cost for new products
 k_r:array(period) of real    !setup cost for remanufacturing
 k_m:array(period) of real    !setup cost for manufacturing
 h_r:array(period) of real    !holding cost for used products
 h_s:array(period) of real    !holding cost for serviceable products
 x_r:array(period) of mpvar   !production amount of remanufactured product
 x_m:array(period) of mpvar   !production amount of manufactured product
 y_r:array(period) of mpvar   !setup variable for remanufacturing
 y_m:array(period) of mpvar   !setup variable for manufacturing
 I_r:array(period) of mpvar   !inventory variable for product returns
 I_s:array(period) of mpvar   !inventory variable for serviceable products
 return:array(period) of real !amount of used products returned
 demand:array(period) of real !amount of demand for serviceable products
 totdem:array(period) of real !total demand from period t until NT
 totret:array(period) of real !total return from period 1 until t

!================================================================================
!(l,S) INEQUALITIES
!================================================================================
 !maximum number of iterations
 maxiter=100
 iter=1..maxiter

 !set S (1 if t in S, 0 otherwise) for each iteration + period l
 setS:array(iter, period, period) of integer

 ret:array(period, period) of real !total return from period t until period l
 dem:array(period, period) of real !total demand from period t until period l

 !counter for number of violations of constraints at each iteration
 countviol_1:integer
 countviol_2:integer
 countviol_3:integer
 countviol_4:integer
```

```
end-declarations

!===============================================================================
!DATA INPUT
!===============================================================================
starttime:= gettime

!read the data from the file
fopen('LaHM100(125)_11.txt', F_INPUT)
forall(t in period)
readln(t, demand(t), return(t), k_m(t), k_r(t), h_s(t), h_r(t), p_m(t), p_r(t))
fclose(F_INPUT)

!===============================================================================
!PARAMETERS CALCULATION
!===============================================================================

!calculate the bigM-constraints
forall (t in period)do
 totdem(t):=sum(tt in t..NT)demand(tt)
 totret(t):=sum(tt in 1..t)return(tt)
end-do

!===============================================================================
!CONSTRAINTS
!===============================================================================

!total costs function
costpro:=sum(t in period)(p_r(t)*x_r(t) + p_m(t)*x_m(t))
costfixed:=sum (t in period)(k_r(t)*y_r(t) + k_m(t)*y_m(t))
costinv:=sum(t in period)(h_r(t)*I_r(t) + h_s(t)*I_s(t))
cost:= costpro + costfixed + costinv

!flow balance for remanufacturing and manufacturing
forall(t in period)do
 const_1(t):= if(t>1, I_r(t-1), 0)- x_r(t) + return(t) = I_r(t)
 const_2(t):= if(t>1, I_s(t-1), 0)+ x_m(t) + x_r(t)- demand(t) = I_s(t)
end-do

!production variable-binary variable relations
forall(t in period)do
 const_3(t):=x_r(t)<= minlist(totret(t),totdem(t))*y_r(t)
 const_4(t):=x_m(t)<= totdem(t)*y_m(t)
end-do

!relax the setup variables
forall(t in period)do
 y_r(t)<= 1
 y_m(t)<= 1
end-do

setparam('XPRS_CPUTIME',1)
setparam('XPRS_MAXTIME', -600)
setparam('XPRS_CUTSTRATEGY', 0)
setparam('XPRS_GOMCUTS', 0)

!===============================================================================
!ADDS (l,S)INEQUALITIES TO THE ORIGINAL FORMULATION
!===============================================================================

starttime := gettime

!calculate the returns and demands from period t to l
forall(l in 1..NT)do
```

```
 forall(t in 1..l)do
  ret(t,l):= 0          !(1)set initial value of ret(t,l) and dem(t,l) as zero
  ret(l,l):= return(l)
  dem(t,l):= 0          !(2)calculate other ret(l,t) and dem (t,l) quantities
  dem(l,l):= demand(l)

  if(l>=2) then
   forall(tt in 1..(l-1))do
    ret(l-tt,l):= ret(l-tt+1,l) + return(l-tt)
    dem(l-tt,l):= dem(l-tt+1,l) + demand(l-tt)
   end-do
  end-if
 end-do
end-do

!first, solve LP relaxation
minimize(cost)
writeln(!===========================================================================)
writeln('LP RELAXATION SOLUTION')
writeln(!===========================================================================)
writeln('The total cost for LP relaxation: ', getobjval)
writeln(' ') ls_soln:= getobjval

!count the total number of inequalities added to the original problem formulation
 total_viol_1:=0
 total_viol_2:=0
 total_viol_3:=0
 total_viol_4:=0

!separation algorithm for (l,s)inequalities
forall(iteration in iter) do
     !initialize the counter
     countviol_1:=0
     countviol_2:=0
     countviol_3:=0
     countviol_4:=0

 forall(l in period) do
  forall(k in 1..l) do
   !initialize the set S
    forall(t in k..l)
     setS(iteration,t,l):=0

   forall(t in k..l)do
        if(getsol(x_r(t))>ret(k,t)*getsol(y_r(t)) or
        getsol(x_r(t))>dem(t,l)*getsol(y_r(t))or
        getsol(x_m(t))>dem(t,l)*getsol(y_m(t))or
        getsol(x_r(t))+ getsol(x_m(t))>dem(t,l)*(getsol(y_r(t))+ getsol(y_m(t))))then
         setS(iteration,t,l):=1
        end-if
   end-do

   if(sum(u in k..l)setS(iteration,u,l)*(getsol(x_m(u))+getsol(x_r(u)))>
   getsol(I_s(l))+ sum(u in k..l)setS(iteration,u,l)*dem(u,l)
   *(getsol(y_m(u))+getsol(y_r(u)))+0.00001) then
    addcons_1(iteration, l):=sum(u in k..l)setS(iteration,u,l)*(x_m(u)+x_r(u))
    <= I_s(l)+ sum(u in k..l)setS(iteration,u,l)*dem(u,l)*(y_m(u)+y_r(u))
    countviol_1:= countviol_1 + 1
   end-if

   if(sum(u in k..l)setS(iteration,u,l)*getsol(x_m(u))>
   getsol(I_s(l))+ sum(u in k..l)setS(iteration,u,l)
   *dem(u,l)*getsol(y_m(u))+0.00001) then
    addcons_2(iteration, l):=sum(u in k..l)setS(iteration,u,l)*x_m(u)<= I_s(l)
```

```
     + sum(u in k..l)setS(iteration,u,l)*dem(u,l)*(y_m(u))
     countviol_2:= countviol_2 + 1
    end-if

    if(sum(u in k..l)setS(iteration,u,l)*getsol(x_r(u))>
    getsol(I_s(l))+ sum(u in k..l)setS(iteration,u,l)
    *dem(u,l)*getsol(y_r(u))+0.00001)then
     addcons_3(iteration, l):=sum(u in k..l)setS(iteration,u,l)*x_r(u)<= I_s(l)
     + sum(u in k..l)setS(iteration,u,l)*dem(u,l)*(y_r(u))
     countviol_3:= countviol_3 + 1
    end-if

    if(sum(u in k..l)setS(iteration,u,l)*getsol(x_r(u))> if(k>1,getsol(I_r(k-1)),0)
    + sum(u in k..l)setS(iteration,u,l)*ret(k,u)*getsol(y_r(u))+0.00001)then
     addcons_4(iteration,l):=sum(u in k..l)setS(iteration,u,l)*x_r(u)
     <= if(k>1,I_r(k-1),0)+ sum(u in k..l)setS(iteration,u,l)*ret(k,u)*(y_r(u))
     countviol_4:= countviol_4 + 1
    end-if
   end-do
  end-do

  if(countviol_1=0 and countviol_2=0 and countviol_3=0 and countviol_4=0) then break
   else
     !solve the strengthened LP relaxation with added maximum violated (l,s)cuts
     minimize(cost)

     ls_soln:=getobjval

     total_viol_1:= total_viol_1 + countviol_1
     total_viol_2:= total_viol_2 + countviol_2
     total_viol_3:= total_viol_3 + countviol_3
     total_viol_4:= total_viol_4 + countviol_4
     num_iter:= iteration

     writeln(!==================================================================)
     writeln('LP RELAXATION SOLUTION WITH ADDED INEQUALITIES')
     writeln(!==================================================================)
     writeln('Iteration: ',iteration)
     writeln('Number of constraints added for constraint I: ',countviol_1)
     writeln('Number of constraints added for constraint II: ',countviol_2)
     writeln('Number of constraints added for constraint III: ',countviol_3)
     writeln('Number of constraints added for constraint III: ',countviol_4)
     writeln('Total cost for LP relaxation with added inequalities: ',getobjval)
     writeln(' ')
   end-if
end-do

!=====================================================================================
!GENERAL STATISTICS FOR(l,S) INEQUALITIES
!=====================================================================================

writeln(!==================================================================)
writeln('(l,S)INEQUALITIES STATISTICS')
writeln(!==================================================================)
writeln('Total cost for LP relaxation      : ',getobjval)
writeln(' ')
writeln('Number of valid inequalities I    : ',total_viol_1)
writeln('Number of valid inequalities II   : ',total_viol_2)
writeln('Number of valid inequalities III  : ',total_viol_3)
writeln('Number of valid inequalities III  : ',total_viol_4)
writeln(' ')

ls_time:= gettime-starttime
writeln('(l,s) time spent    : ', ls_time)
```

```
writeln('Number of iterations: ', num_iter)
writeln(' ')

!==============================================================================
!IP SOLUTION OF THE MODEL
!==============================================================================

forall (t in period)do
 y_r(t) is_binary
 y_m(t) is_binary
end-do

!solve IP
starttime := gettime
minimize(cost)
writeln(!==============================================================================)
writeln('IP SOLUTION')
writeln(!==============================================================================)
writeln('Best solution - the total cost for IP: ', getobjval)
writeln(' ')

mip_time:=gettime-starttime
writeln('MIP time spent: ', mip_time)
writeln(' ')

!==============================================================================
!EXIT
!==============================================================================
exit(0)
end-model
```

Appendix B

Shortest Path Reformulation in Mosel - Separate Setups

```
model 'Shortest Path Reformulation for ELSR with Separate Setups'
uses 'mmxprs'
uses 'mmsystem'

declarations
 NT=25                              !number of time periods
 period=1..NT
 p_r:array(period) of real          !production cost for remanufactured products
 p_m:array(period) of real          !production cost for new products
 k_r:array(period) of real          !setup cost for remanufacturing
 k_m:array(period) of real          !setup cost for manufacturing
 h_r:array(period) of real          !holding cost for used products
 h_s:array(period) of real          !holding cost for serviceable products
 x_r:array(period) of mpvar         !production amount of remanufactured product
 x_m:array(period) of mpvar         !production amount of manufactured product
 y_r:array(period) of mpvar         !setup variable for remanufacturing
 y_m:array(period) of mpvar         !setup variable for manufacturing
 f:array(period) of mpvar           !final serviceable inventory variable
 return:array(period) of real       !amount of used products returned
 demand:array(period) of real       !amount of demand for serviceable products
 ret:array(period, period) of real !total return from period t until l
 dem:array(period, period) of real !total demand from period t until l
 c_s:array(period, period) of real !total cost for remanufacturing process
 c_r:array(period, period) of real !total holding cost for used products
 c_f:array(period) of real          !total cost of final inventory of returns

 !the fraction of demand in each of the periods i until j that is fulfilled
 !by remanufactured products in period i
 z_sr:dynamic array(period, period) of mpvar
 !the fraction of demand in each of the periods i until j that is fulfilled
 !by newly produced products in period i
 z_sm:dynamic array(period, period) of mpvar
 !the fraction of returns in each of the periods i until j that is remanufactured
 !in period j
 z_r:dynamic array(period, period) of mpvar
end-declarations

!================================================================================
!DATA INPUT
!================================================================================
starttime:= gettime
```

```
!read the data from the file
fopen('LaHM100(125)_11.txt', F_INPUT)
forall(t in period)
readln(t, demand(t), return(t), k_m(t), k_r(t), h_s(t), h_r(t), p_m(t), p_r(t))
fclose(F_INPUT)

!================================================================================
!PARAMETERS CALCULATION
!================================================================================
starttime := gettime

!calculate the bigM-constraints
forall(t in period, l in t..NT)do
 ret(t,l):=sum(u in t..l)return(u)
 dem(t,l):=sum(u in t..l)demand(u)
end-do

!calculate the total costs from period t to l
forall(l in period)do
 if(l>=2) then
  forall(u in 1..(l-1))do
   c_s(u,l):=sum(i in u..(l-1))h_s(i)*dem(i+1,l)
   c_r(u,l):=sum(i in u..(l-1))h_r(i)*ret(u,i)
  end-do
 end-if
end-do

forall(t in period) c_f(t):=sum(j in t..NT)h_r(j)*ret(t,j)

forall(l in 1..NT, t in 1..NT)do
 create(z_sr(l,t))
 create(z_sm(l,t))
 create(z_r(l,t))
end-do

!================================================================================
!CONSTRAINTS
!================================================================================

!total costs function
costpro:=sum(t in period)(p_r(t)*x_r(t) + p_m(t)*x_m(t))
costfixed:=sum (t in period)(k_r(t)*y_r(t) + k_m(t)*y_m(t))
costinv:=sum(t in period, l in t..NT)(c_r(t,l)*z_r(t,l)
+ c_s(t,l)*(z_sr(t,l)+z_sm(t,l)))+ sum(t in period)c_f(t)*f(t)
cost:= costpro + costfixed + costinv

!constraints-nodes
const_1:=sum(l in period)(z_sr(l,NT) + z_sm(l,NT))=1
const_2:=-sum(l in period)(z_sr(1,l) + z_sm(1,l))=-1
forall(t in 1..NT-1) const_3(t):=sum(l in 1..t)(z_sr(l,t) + z_sm(l,t))
=sum(l in t+1..NT)(z_sr(t+1,l) + z_sm(t+1,l))
const_4:=sum(l in period)(z_r(l,NT)+f(l))=1
const_5:=-sum(l in period)(z_r(1,l)+f(1))=-1
forall(t in 1..NT-1) const_6(t):=sum(l in 1..t)z_r(l,t)=sum(l in t+1..NT)z_r(t+1,l)
+f(t+1)

!relationship old and new variables
forall(t in period)do
 const_7(t):=x_r(t)=sum(l in 1..t)ret(l,t)*z_r(l,t)
 const_8(t):=x_m(t)=sum(l in t..NT)dem(t,l)*z_sm(t,l)
end-do

!production variable-binary variable relations
forall(t in period)do
```

```
 const_9(t):=sum(l in t..NT|dem(t,l)>0)z_sr(t,l)<= y_r(t)
 const_10(t):=sum(l in t..NT|dem(t,l)>0)z_sm(t,l)<= y_m(t)
 const_11(t):=sum(l in 1..t|ret(t,l)>=0)z_r(l,t)<= y_r(t)
end-do

!link constraint between z_r and z_sr
forall(t in period) const_12(t):=sum(tt in 1..t)ret(tt,t)*z_r(tt,t)
=sum(tt in t..NT)dem(t,tt)*z_sr(t,tt)

!relax the setup variables
forall(t in period)do
 y_r(t)<=1
 y_m(t)<=1
end-do

setparam('XPRS_CPUTIME',1)
setparam('XPRS_MAXTIME', -600)
setparam('XPRS_CUTSTRATEGY', 0)

!==========================================================================
!LP RELAXATION SOLUTION OF THE MODEL
!==========================================================================

!first, solve LP relaxation
minimize(cost)
writeln(!=======================================================================)
writeln('LP RELAXATION SOLUTION')
writeln(!=======================================================================)
writeln('The total cost for LP relaxation: ', getobjval)
writeln(' ')

!==========================================================================
!IP SOLUTION OF THE MODEL
!==========================================================================
forall (t in period)do
 y_r(t) is_binary
 y_m(t) is_binary
end-do

!solve IP
minimize(cost)
writeln(!=======================================================================)
writeln('IP SOLUTION')
writeln(!=======================================================================)
writeln('The total cost for IP: ', getobjval)
writeln(' ')

ls_time:= gettime-starttime
writeln('MIP time spent: ', ls_time)
writeln(' ')

!==========================================================================
 !EXIT
!==========================================================================
exit(0)
end-model
```

Appendix C

Detailed Results of Lower Bounds - Separate Setups

C.1 Low Return ($n = 25$)

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$	
		LB	UB	LB	UB	LB	UB	LB	UB
125	1	464.35	3359.62*	3189.72	3236.15	2532.83	3236.15	3189.91	3236.15
	2	506.64	3570.12*	3425.37	3467.05	2688.53	3467.05	3425.37	3467.05
	3	474.44	3447.61*	3334.67	3349.75	2663.53	3349.75	3334.67	3349.75
	4	426.11	3333.23*	3209.77	3234.96	2523.74	3234.96	3212.56	3234.96
	5	524.69	3458.44*	3278.08	3314.37	2523.62	3314.37	3278.16	3314.37
	6	453.10	3409.5*	3209.43	3238.29	2506.58	3238.29	3212.62	3238.29
	7	448.33	3237.66*	3099.32	3141.92	2467.76	3141.92	3099.32	3141.92
	8	496.14	3625.71*	3469.15	3511.13	2745.44	3511.13	3469.15	3511.13
	9	510.52	3238.12*	3188.34	3199.49	2546.26	3199.49	3188.34	3199.49
	10	520.00	3407.27*	3244.27	3288.93	2600.61	3288.93	3244.27	3288.93
Average		482.43	3408.73	3264.81	3298.20	2579.89	3298.20	3265.44	3298.20
250	1	914.32	5101.95*	4927.87	4964.39	4062.02	4964.39	4932.71	4964.39
	2	997.12	5462.89*	5263.13	5290.72	4312.43	5290.72	5266.82	5290.72
	3	948.88	5353.13*	5215.07	5299.84	4245.51	5299.84	5215.41	5299.84
	4	847.59	5315.92*	5017.91	5105.49	4100.77	5105.49	5017.91	5105.49
	5	988.68	5539.79*	5200.18	5294.24	4138.32	5294.24	5205.54	5294.24
	6	906.19	5385.55*	5166.32	5217	4097.32	5217	5174.58	5217
	7	888.35	5007.02*	4785.83	4812.14	3946.5	4812.14	4791.04	4812.14
	8	934.31	5572.55*	5358.38	5360.62	4376.94	5360.62	5358.38	5360.62
	9	993.29	5112.98*	5009.75	5049.38	4046.37	5049.38	5009.75	5049.38
	10	977.07	5417.75*	5127.79	5161.67	4147.82	5161.67	5129.01	5161.67
Average		939.58	5326.95	5107.22	5155.55	4147.4	5155.55	5110.12	5155.505

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$	
		LB	UB	LB	UB	LB	UB	LB	UB
500	1	1724.64	7381.85	7304.22	7381.85	6018.52	7381.85	7304.22	7381.85
	2	1831.34	7873.84	7763.39	7873.84	6325.37	7873.84	7773.45	7873.84
	3	1836.61	8019.16	7980.17	8019.16	6598.65	8019.16	7986.64	8019.16
	4	1643.97	7644.34	7539.59	7644.34	6193.77	7644.34	7549.81	7644.34
	5	1828.59	8036.64	7931.33	8036.64	6499.84	8036.64	7966.75	8036.64
	6	1789.93	7975.8	7871.45	7975.8	6382.43	7975.8	7875.27	7975.8
	7	1658.5	7321.54	7254.53	7321.54	5997.39	7321.54	7277.36	7321.54
	8	1684.38	8262.92*	8066.97	8099.03	6644.42	8099.03	8083.54	8099.03
	9	1945.94	7791.65	7720.64	7791.65	6414.56	7791.65	7732.5	7791.65
	10	1787.06	7967.69*	7764.19	7837.54	6424.2	7837.54	7775.05	7837.54
Average		1773.10	7827.54	7719.65	7798.14	6349.92	7798.14	7732.46	7798.14
1000	1	3274.45	11047.6	11008.5	11047.6	8895.41	11047.6	11031.5	11047.6
	2	3269.02	11610.2	11605.2	11610.2	9484.89	11610.2	11605.2	11610.2
	3	3405.95	11686.8	11640.3	11686.8	9739.97	11686.8	11666.2	11686.8
	4	3108.61	11336	11250.8	11336	9483.84	11336	11299.3	11336
	5	3354.71	11875.1	11825.4	11875.1	9775.96	11875.1	11852.9	11875.1
	6	3306.88	11746.7	11729.7	11746.7	9756.82	11746.7	11742.1	11746.7
	7	3036.81	10624	10601.1	10624	8886.71	10624	10624	10624
	8	3127.85	12208.6	12137.9	12208.6	9950.63	12208.6	12159.2	12208.6
	9	3466.87	11282.1	11266.7	11282.1	9302.88	11282.1	11267.5	11282.1
	10	3213.96	11241.4	11241.4	11241.4	9469.48	11241.4	11241.4	11241.4
Average		3256.51	11465.85	11430.7	11465.85	9474.66	11465.85	11448.93	11465.85

* indicates solution is not optimal-allocated computation time exceeded 600s.

C.2 Low Return $(n = 50)$

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$	
		LB	UB	LB	UB	LB	UB	LB	UB
125	1	565.36	7361.2*	6638.24	6733.5	5295.6	6774.64*	6651.97	6733.5
	2	570.40	7436.05*	6430.19	6584.36	5099.2	6638.49*	6434.68	6584.36
	3	553.55	7571.91*	6588.09	6690.34	5213.66	6728.79*	6589.05	6690.34
	4	561.28	7558.84*	6608.57	6744.22	5228.78	6761.62*	6609.03	6744.22
	5	608.52	7009.21*	5974.02	6119.18	4795.26	6166.71*	5974.73	6119.18
	6	598.26	7289.03*	6368.27	6475.72	5028.27	6508.96*	6376.44	6475.72
	7	601.34	7638.84*	6750.22	6843.73	5434.13	6955.53*	6750.22	6843.73
	8	566.69	7490.95*	6657.33	6771.58	5289.75	6813.44*	6657.33	6771.58
	9	583.84	7410.4*	6466.38	6582.94	5138.76	6608.4*	6467.07	6582.94
	10	556.45	7487.08*	6313.83	6374.63	5015.24	6431.78*	6313.83	6374.63
Average		576.57	7425.35	6479.51	6592.07	5153.87	6638.84	6482.44	6592.07
250	1	1065.16	11597.6*	10239.3	10370.2	8527.6	10498.7*	10280.6	10370.2
	2	1118.48	11562.2*	10040.5	10168.2	8083.49	10183.9*	10054.5	10168.2
	3	1095.2	11520.1*	10336	10421.8	8495.36	10526.8*	10348.8	10421.8
	4	1110.99	12170.8*	10321.2	10494	8471.64	10718.1*	10341.7	10494
	5	1157.45	10996.5*	9372.01	9470.86	7665.58	9609.91*	9402.45	9470.86
	6	1110.25	11619.3*	9928.3	10013.6	8097.82	10133.5*	9952.62	10013.6
	7	1173.83	11607.4*	10589.6	10712.9	8731.04	10743.1*	10590.2	10712.9
	8	1093.83	11983.4*	10433.8	10596.2	8498.37	10735.4*	10461	10596.2
	9	1121.51	11534.9*	10044.7	10170.7	8160.24	10213*	10056.8	10170.7
	10	1080.81	11634.1*	9759.75	9876.83	7989.38	10033.4*	9763.86	9876.83
Average		1112.75	11622.63	10106.52	10229.53	8272.05	10339.58	10125.25	10229.53

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$	
		LB	UB	LB	UB	LB	UB	LB	UB
500	1	2056.98	17364.1*	15595.7	15892.5	12933.5	16018.1*	15637.7	15892.5
	2	2128.89	18166.6*	15197.7	15353.6	12481.4	15572.6*	15235.2	15353.6
	3	2075.36	17690.6*	15649.9	15874.2	12878	15902.8*	15681.5	15874.2
	4	2144.6	17381.7*	15716.3	15947.4	12802.5	16132.5*	15767.2	15947.4
	5	2159.19	16361.6*	14364.2	14560.6	11817.3	14646.9*	14396.4	14560.6
	6	2120.27	17560.6*	15255	15414.8	12472.1	15524.1*	15261.3	15414.8
	7	2188.55	17251.3*	15927.2	15990.1	13334.6	16270.4*	15929	15990.1
	8	2042.05	18366.5*	15689.2	15917.4	13009.8	16119.7*	15734	15917.4
	9	2105.41	17705.9*	15256.2	15387.9	12600.5	15588.5*	15293.5	15387.9
	10	2006.89	17402.8*	14864.8	15157.4	12220	15237*	14917.5	15157.4
Average		2102.82	17525.17	15351.62	15549.59	12654.97	15701.26	15385.33	15549.59
1000	1	4000.77	25152.3*	23205	23461.1	19514.4	23539*	23337.4	23461.1
	2	3939.26	25512.6*	22673.2	22798.7	18897.6	22900.3*	22681.7	22798.7
	3	3825.42	27041.4*	23445.9	23670.2	19328.1	23670.2*	23485.1	23670.2
	4	4077.8	25865.4*	23161.3	23241.6	19389.4	23241.6*	23224.8	23241.6
	5	4073.09	24085.6*	21256.8	21473.7	17846.1	21473.7	21296.3	21473.7
	6	3975.92	25658.2*	22785.5	22840.5	19063.6	22840.5*	22797.7	22840.5
	7	3979.09	26060.7*	23372.4	23632.4	19563	23878.6*	23430.4	23632.4
	8	3911.06	26003.6*	23188.8	23493.9	19295.8	23702.8*	23277.7	23493.9
	9	3942.07	25701.8*	22512.9	22916.6	18852.2	23022.7*	22594	22916.6
	10	3674.01	24147.1*	22487.5	22730.7	18739.7	22730.7*	22597.7	22730.7
Average		3939.85	25522.87	22808.93	23025.94	19048.99	23096.63	22872.28	23025.94

* indicates solution is not optimal-allocated computation time exceeded 600s.

C.3 Low Return $(n = 75)$

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$	
		LB	UB	LB	UB	LB	UB	LB	UB
125	1	666.69	11483.7*	9693.04	9855.47	7791.05	9969.51*	9699.29	9855.47
	2	583.38	11222.1*	9269.78	9376.61	7413.97	9583.57*	9270.76	9376.61
	3	640.09	11529.8*	9639.32	9805.77	7677.52	9921.27*	9641.48	9805.77
	4	635.57	11410.3*	9929.34	10075	7946.94	10242.4*	9933.02	10075
	5	601.41	11127.9*	9968.45	10086	8013.99	10333.9*	9969.24	10086
	6	597.97	11252.3*	9922.69	10057.8	7909.06	10236.2*	9923.57	10057.8
	7	592.33	11460.3*	10038.7	10184.1	7987.67	10330.5*	10038.7	10184.1
	8	641.58	11188.3*	10018.2	10120.5	8087.1	10260.7*	10018.2	10120.5
	9	624.73	11613.8*	9986.93	10151.8	7863.22	10390.8*	9998.2	10151.8
	10	618.33	11602.8*	10322.1	10438.4	8125.69	10503.9*	10322.2	10438.4
Average		620.21	11389.13	9878.86	10015.15	7881.62	10177.28	9881.47	10015.15
250	1	1237.59	18068.2*	15198.5	15446.2	12575.7	15677.6*	15220.3	15446.2
	2	1141.33	17431.4*	14570.4	14701.3	11981.6	14938.7*	14585.6	14701.3
	3	1201.04	18084.6*	14924.7	15139.9	12065.4	15445.6*	14940.7	15139.9
	4	1220.39	18114.4*	15456.6	15682.1	12698.6	15964.4*	15489	15682.1
	5	1202.83	18097.2*	15681	15832.4	12838.9	15905.6*	15713.2	15832.4
	6	1133.72	18462.3*	15460	15725.6	12527.1	15929.5*	15467	15725.6
	7	1165.38	17999.7*	15642.4	15846.5	12772.5	16232.2*	15656.1	15846.5
	8	1219.04	18474.7*	15544.6	15719	12860.8	16138*	15558.2	15719
	9	1197.6	19008.2*	15632.8	15919.1	12673	16127.9*	15651.8	15919.1
	10	1167.3	18924.9*	16174.9	16317	13208.8	16666.5*	16189.7	16317
Average		1188.62	18266.56	15428.59	15632.91	12620.24	15902.6	15447.16	15632.91

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$	
		LB	UB	LB	UB	LB	UB	LB	UB
500	1	2375.9	27889.3*	23152.5	23406.8	19266.5	24416.3*	23195.4	23406.8
	2	2257.22	26481.9*	22110.8	22448.3	18455.4	23050.7*	22196.9	22448.3
	3	2271.86	28197.6*	22536.6	22802.5	18697.8	23208.6*	22575.6	22802.5
	4	2321.98	28181.2*	23202.8	23631.9	19129.4	24566.6*	23306.2	23631.9
	5	2306.48	27433.6*	23810.2	24026.1	19906.7	24525.9*	23866	24026.1
	6	2186.97	27956.4*	23546.8	23699.2	19652.3	24435.2*	23559.3	23699.2
	7	2261.02	27996.7*	23508.2	23682.2	19639.4	24252.3*	23536.5	23682.2
	8	2271.76	28103.5*	23962.4	24162.2	19944.1	24486.1*	24008.9	24162.2
	9	2289.29	27793.8*	23748.1	24036.3	19595.3	24846.7*	23771.1	24036.3
	10	2214.72	28905*	24697.2	25150.8	20333.6	25466.2*	24729.5	25150.8
Average		2275.72	27893.90	23427.56	23704.63	19462.05	24325.46	23474.54	23704.63
1000	1	4608.09	40496.1*	34752.9	35115.7	29293.1	35868.7*	34838.5	35115.7
	2	4413.49	39456.1*	32866.6	33118.7	27668.9	34887.1*	32923	33118.7
	3	4247.08	41868.6*	33739.1	33954.8	28252.2	35111.5*	33820.7	33954.8
	4	4357.92	41466.2*	34699.8	35036.1	28708.8	36525.8*	34817.8	35036.1
	5	4339.77	41467.1*	35546.7	35957.2	29888.7	37226*	35638.8	35957.2
	6	4264.32	40216.5*	34702.7	35103.1	29148.3	35852*	34797.4	35103.1
	7	4337.33	41308.3*	34841	35229.3	29060.9	36420.8*	34937.8	35229.3
	8	4362.27	41501.4*	35837.2	36203.7	30007.3	37447.2*	35976.2	36203.7
	9	4321.8	40553.6*	35413.1	35868.4	29655.8	37004.9*	35533	35868.4
	10	4266.52	45366.2*	36925.9	37345	31046.8	38812.8*	37031.7	37345
Average		4351.85	41370.01	34932.50	35293.20	29273.08	36515.68	35031.49	35293.20

* indicates solution is not optimal-allocated computation time exceeded 600s.

C.4 Medium Return ($n = 25$)

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$	
		LB	UB	LB	UB	LB	UB	LB	UB
125	1	624.87	3963.53*	3626.25	3741.25	3026.34	3741.25	3638.39	3741.25
	2	1133.14	4261.01*	3807.91	4037.24	3280.21	4037.24	3811.45	4037.24
	3	606.06	3975.73*	3445.4	3720.15	2983.85	3720.15	3449.63	3720.15
	4	588.90	3816.39*	3394.38	3605.47	2874.67	3605.47	3402.27	3605.47
	5	686.57	3898.61*	3479.23	3703.68	2927.56	3703.68	3493.82	3703.68
	6	529.38	3869.51*	3554.82	3777.79	2988.95	3777.79	3559.28	3777.79
	7	740.54	4146.84*	3744.36	3934.48	3219.15	3934.48	3745.52	3934.48
	8	587.14	4141.29*	3523.67	3835.62	3052.57	3835.62	3539.28	3835.62
	9	702.99	3625.78*	3261.17	3512.26	2912.56	3512.26	3261.21	3512.26
	10	619.82	3838.76*	3432.8	3697.65	2940.17	3697.65	3443.86	3697.65
Average		681.94	3953.75	3527	3756.56	3020.60	3756.56	3534.47	3756.56
250	1	1063.65	6147.76*	5612.74	5903.2	4938.03	5903.2	5655.84	5903.2
	2	1553.97	6351.62*	5957.21	6284.12	5268.54	6284.12	5965.45	6284.12
	3	1043.79	6599.35*	5616.64	6102.84	5042.18	6102.84	5661.33	6102.84
	4	1001.96	6237.01*	5523.3	5851.62	4774.91	5851.62	5552.48	5851.62
	5	1177.17	6171.08*	5552.67	5963.37	4856.86	5963.37	5621.37	5963.37
	6	1037.18	6319.34*	5597.25	5973.87	4856.51	5973.87	5640.29	5973.87
	7	1157.84	6457.29*	5876.59	6203.47	5103.88	6203.47	5950.95	6203.47
	8	1021.21	6370.71*	5465.56	5958.21	4802.16	5958.21	5559.2	5958.21
	9	1168.14	5968.66*	5255.71	5675.93	4739.82	5675.93	5296.84	5675.93
	10	1045.72	6176.23*	5517.09	5804.16	4729.68	5804.16	5546.06	5804.16
Average		1127.06	6279.91	5597.48	5972.08	4911.26	5972.08	5644.98	5972.08

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$	
		LB	UB	LB	UB	LB	UB	LB	UB
500	1	1896.95	9525.66*	8597.21	9232.07	7532.97	9232.07	8734.41	9232.07
	2	2395.63	9741.7*	9089.86	9474.87	8045.5	9474.87	9148.22	9474.87
	3	1881.98	9951.85*	8674.05	9174.45	7849.86	9174.45	8812.12	9174.45
	4	1760.84	9250.32*	8633.97	9171.27	7488.44	9171.27	8700.19	9171.27
	5	2022.81	9090.98*	8377.99	8888.91	7420.47	8888.91	8491.81	8888.91
	6	1881.8	9369.86*	8626.56	8999.86	7627.77	8999.86	8738.27	8999.86
	7	1992.46	9802.4*	9098.31	9513	8011.59	9513	9207.47	9513
	8	1813.12	9352.64*	8489.37	9153.01	7465.46	9153.01	8704.76	9153.01
	9	2058.79	8784.92*	8193.68	8627.31	7243.81	8627.31	8230.65	8627.31
	10	1826.66	9079.95*	8461.48	8929.45	7240.4	8929.45	8572.6	8929.45
Average		1953.10	9395.03	8624.25	9116.42	7592.63	9116.42	8734.05	9116.42
1000	1	3432.15	13719.9*	12847.9	13492.9	11392.4	13492.9	13142.2	13492.9
	2	3887.36	14193.3*	13573.2	14193.3	11942.9	14193.3	13716.9	14193.3
	3	3435.36	14221.2*	13016.9	13979.4	11766.8	13979.4	13243.9	13979.4
	4	3798.89	13913.1*	12905.8	13770.3	11583	13770.3	13106.7	13770.3
	5	3551.44	12799.8	12396	12799.8	10950.5	12799.8	12540.2	12799.8
	6	3436.12	13589.8*	12827.4	13584.7	11340	13584.7	13036.3	13584.7
	7	3580.56	14437.3*	13610.5	14437.3	11976.1	14437.3	13852.9	14437.3
	8	3310.31	13462.6*	12649.5	13310.9	11263	13310.9	12862.6	13310.9
	9	3531.99	13097.1*	12373.8	13073.9	10800.5	13073.9	12555.9	13073.9
	10	3246.52	13174.2*	12648.5	13174.2	11015	13174.2	12804.5	13174.2
Average		3461.07	13660.83	12884.95	13581.67	11403.02	13581.67	13089.21	13581.67

* indicates solution is not optimal-allocated computation time exceeded 600s.

C.5 Medium Return ($n = 50$)

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$			
		LB	UB	LB	UB	LB	UB	LB	UB-FL	UB-MC	UB-SP
125	1	832.32	8208.58*	7011.1	7674.76*	6056.75	7766.81*	7043.34	7624.46*	7629.62*	7605.62*
	2	1003.86	7959.2*	6667.87	7160.75	5751.82	7219.8*	6699.21	7160.75	7160.75	7160.75
	3	1079.88	8015.92*	6662.66	7343.37*	5794.86	7404.8*	6708.24	7262*	7281.45*	7266.8*
	4	1072.93	8968.17*	7039.22	7491.85	5993.83	7616.99*	7069.73	7491.85	7491.85	7491.85
	5	721.19	8636.47*	6989.94	7419.35	6011.62	7494.8*	6998.27	7419.35	7419.35	7419.35
	6	730.43	8307.33*	6830.14	7225.31	5785.86	7266.52*	6861.54	7225.31	7225.31	7225.31
	7	843.40	8660.86*	6965.85	7691.34*	6019.99	7756.52*	6993.38	7672.13*	7667.51*	7667.51*
	8	806.40	8397.65*	7062.54	7572.22*	6021.95	7719.33*	7102.76	7572.22	7572.22*	7572.22
	9	1229.96	8249.9*	6573.39	7306.03*	5809.35	7405.24*	6602.53	7306.03*	7325.95*	7306.03*
	10	778.77	8388.77*	7035.03	7623.24*	6042.44	7696.78*	7076.62	7619.75*	7325.95*	7619.75*.
Average		909.91	7433.04	6883.77	7450.82	5928.85	7534.76	6915.56	7435.39	7436.98	7433.52
250	1	1390.9	12913.1*	11043.9	11846.8*	9726.77	12004.8*	11222.9	11851.5*	11383.9*	11851.5*
	2	1536.32	13103.9*	10632.8	11489.2*	9416.77	11712*	10738.1	11473.8*	11489.2*	11500*
	3	1634.25	13106.1*	10382.6	11413*	9286.76	11490.9*	10506	11402.1*	11413*	11422.5*
	4	1595.93	13688.9*	10977.9	11682	9621.71	12016.4*	11074.7	11682	11682	11682
	5	1274.82	13905.1*	11242.9	11950.8*	9916.44	12025.7*	11323	11938.1	11938.1	11938.1
	6	1290.31	13530.3*	10771.4	11578.5*	9441.8	11665.2*	10904.7	11570.8*	11570.8*	11582.3*
	7	1372.2	14177*	11146.7	12322.6*	9907.72	12355.7*	11270.4	12235.2*	12235.2*	12246*
	8	1378.42	13458.5*	11178.8	12103.3*	9720.72	12135.4*	11316.6	12023.2*	12028.6*	12023.2*
	9	1760.91	13571.9*	10442.3	11292.2*	9453.68	11345.5*	10533.5	11253.1	11253.1	11287.1*
	10	1275.93	13459.4*	11041.2	12146.5*	9691.21	12300.9*	11201.6	12089.6*	12073*	12073*
Average		1451	13491.42	10886.05	11782.49	9618.36	11905.25	11009.15	11750.38	11752.19	11760.57

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$			
		LB	UB	LB	UB	LB	UB	LB	UB-FL	UB-MC	UB-SP
	1	2428.55	20348.1*	16870.8	17914.7*	15086.3	18040.9*	17137.1	17914.7	17914.7	17914.7
	2	2535.3	20438.2*	16185.4	17266.6	14565.9	17351.8*	16517.7	17266.6	17266.6	17266.6
	3	2694.29	20642.2*	15715.1	16772.6	14141.3	16772.6*	16054.6	16772.6	16772.6	16772.6
	4	2641.93	21090.1*	16833.9	17926.5*	14951.9	18015.6*	17151	17920.6	17920.6	17920.6
500	5	2328.04	20657.5*	17163.2	18094.8	15351.2	18366.2*	17331.9	18094.8	18094.8	18094.8
	6	2380.53	20447.2*	16513.5	17516	14807.2	17531.5*	16763.5	17516	17516	17516
	7	2380.5	21043.1*	17085.5	18362.4*	15325.5	18720.1*	17362.9	18348.1	18348.1	18348.1
	8	2414.17	20914.7*	16988.2	18074.2*	15129	18223.8*	17277.2	18074.2	18074.2	18074.2
	9	2693.33	20253.2*	15949.6	17425.8*	14724.9	17528.2*	16185.6	17273.6	17273.6	17273.6
	10	2270.26	20429.7*	16999.3*	18534.7*	15108.5	18613.9*	17337.3	18275.3	18275.3*	18275.3
Average		2476.69	20626.4	16630.45	17788.83	14919.17	17916.46	16911.88	17745.65	17745.65	17745.65
	1	4269.82	29850.3*	25054	26657.4	22555	26758.1*	25655.8	26657.4	26657.4	26657.4
	2	4297.05	30132.3*	24292.4	26137.8*	22131.9	26137.6*	24896	26111.7	26111.7	26111.7
	3	4680.59	28716.2*	23377.4	24891.4	21372.9	24891.4	23936.3	24891.4	24891.4	24891.4
	4	4569.69	29026.1*	25087.5	26387.3	22505.2	26387.3	25611.3	26387.3	26387.3	26387.3
1000	5	4293.51	30214.2*	25789	27025.2	23103.4	27025.2	26211.8	27025.2	27025.2	27025.2
	6	4349.62	29369.3*	24719.2	26212.5	22178	26367.9*	25345.5	26212.5	26212.5	26212.5
	7	4271.79	31736.1*	25656.5	27259.8	23316.7	27259.8	26134.9	27259.8	27259.8	27259.8
	8	4305.19	31084.9*	25627.6	27258.5	22577.8	27370.2*	26103.7	27258.5	27258.5	27258.5
	9	4467.93	30097.1*	23958.6	25846.1	21874.7	25846.1	24627.1	25846.1	25846.1	25846.1
	10	4133.11	30520.5*	25360.7	27267.4*	22806.1	27525.4*	25992.7	27264.5	27264.5	27264.5
Average		4363.83	30074.7	24892.29	26458.34	22442.17	26517.90	25451.51	26491.44	26455.44	26491.44

* indicates solution is not optimal-allocated computation time exceeded 600s.

C.6 Medium Return ($n = 75$)

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$			
		LB	UB	LB	UB	LB	UB	LB	UB-FL	UB-MC	UB-SP
	1	1789.47	12875.2*	10362.5	11440*	9123.6	11598.1*	10378.4	11297.3*	11320.6*	11289.2*
	2	1404.45	12497.4*	10165.9	11072.4*	8856.31	11115.1*	10195.3	11024.6*	11024.6*	11024.6*
	3	1492.5	12786.4*	10018.8	11251.4*	8652.07	11361.2*	10060.8	11139.5*	11094.5*	11223.5*
	4	1535.4	13159.9*	10496.7	11459.2*	9050.89	11482*	10513.4	11355.4*	11422.4*	11404.4*
125	5	960.12	12994.4*	10214	11121.6*	8698.99	11309*	10265.8	11152.1*	11051.5*	11150.6*
	6	1575.52	13008.4*	10039.6	11139.8*	8686.32	11222.2*	10050.1	11020.4*	11020.4*	11159.2*
	7	1327.46	12474.6*	10053	11052.7*	8647.38	11354.8*	10085.6	11039.6*	10974.6*	10980.7*
	8	887.23	13005.4*	10222.3	11281.4*	8646.93	11563.5*	10242.4	11130.2*	11060.9*	11044.1*
	9	919.22	13185.7*	10686.9	11554.3*	9016.33	12069.1*	10706.3	11530.4*	11489.3*	11459.9*
	10	1156.55	12169.2*	10203.5	10985.5*	8786.49	11109*	10249.6	10912*	10957.1*	10933*
Average		1304.79	12815.66	10246.32	11235.83	8816.53	11418.40	10274.77	11160.15	11141.55	11166.92
	1	2366.15	20759.6*	16201.5	17790.6*	14378.9	18363.1*	16323.6	17621.6*	17627.5*	17652.4*
	2	1959.81	21635*	16125.4	17579.3*	14547.7	17853.9*	16178.3	17556.3*	17457.9*	17563.7*
	3	2074.14	20545.5*	15763.9	17325.9*	14264.9	17509.5*	16012.2	17355.1*	17282.2*	17333.1*
	4	2129.03	21113.1*	16394.7	17809.7*	14534.5	18255.5*	16505.8	17671.3*	17695.9*	17625.8*
250	5	1519.67	20992.3*	16124.1	17594*	14189.4	17804.3*	16402.6	17427.2*	17447.6*	17431.3*
	6	2119.96	20736.6*	15729.7	17272.8*	13870.9	17745.7*	15871.6	17210.8*	17188.3*	17127.9*
	7	1940.6	21181*	15824.7	17072.5*	13985	17341.4*	15977.9	16950.2*	17021*	17008.9*
	8	1471.24	20934.8*	16023.6	17208.8*	14289.5	17576.4*	16166.5	17243*	17199*	17238.9*
	9	1496.52	21721.1*	16969.3	18283.5*	14855.5	18930.2*	17213.1	18316.1*	18284.4*	18413.2*
	10	1729	20089.8*	15954.9	17195*	14023	17377.4*	16131.6	17157.9*	17120.9*	17295.4*
Average		1880.61	20970.88	16111.18	17513.21	14293.93	17875.74	16278.32	17450.95	17432.47	17469.06

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$			
		LB	UB	LB	UB	LB	UB	LB	UB-FL	UB-MC	UB-SP
500	1	3519.5	33336.5*	24523.6	26931.8*	22278.2	27821.5*	24932.3	26802.2*	26808.5*	26737.6*
	2	3070.52	31808.3*	24641	26944.5*	22349.2	27584.3*	25064.6	26575*	26630.1*	26839.5*
	3	3237.42	31852.2*	24260.4	26577.8*	21966.6	26991.9*	24681.7	26285.5*	26146.9*	26336.6*
	4	3265.48	32419*	25219.3	26682.5*	22306.8	27286*	25677.4	26675.3*	26682.5*	26661.5*
	5	2638.78	33062.9*	24666.5	26785.3*	22137.3	27046.8*	25067.1	26598.2*	26651.7*	26589.9*
	6	3208.83	32943.5*	24055.9	26200.9*	21526.6	26987.1*	24547.2	25816.3*	25871.3*	25814.5*
	7	3135.7	32846.7*	24201.2	26072.1*	21775.5	26566.5*	24553.2	26014.3*	25952.5*	26066.6*
	8	2612.17	33469*	24688.5	26444.4*	21901	26992.7*	25063.1	26040*	26040*	26076.5*
	9	2608.5	33065.3*	26043.2	28389.3*	23289.9	29303.3*	26590.1	28207.6*	28003.1*	28031.5*
	10	2873.9	32210.6*	24467.2	26246.8*	22031	26998.5*	24794.2	26163.9*	26203.7*	26253.9*
Average		3017.08	32704.99	24676.68	26727.54	22156.21	27357.86	25097.09	26517.83	26499.03	26540.81
1000	1	5690.22	46917.6*	36647.2	39849.8*	33566.6	40328.8*	37518.8	39376.7*	39352.1*	39334.7*
	2	5124.63	48046.1*	36889	40108.2*	33784.6	39966.7*	37701.7	39791.5*	39892.1*	39829.5*
	3	5554.98	46926.1*	36197.3	39250.3*	33258.6	39681.9*	37087.3	38875.5*	38875.5*	39219.9*
	4	5347.57	49598.3*	37544.2	40439.5*	33509.5	41813*	38243.6	40009.8*	40073.7*	40073.7*
	5	4713.05	48104.7*	37000.2	39653.8*	33519.1	40383.7*	37760.4	39315.7*	39369*	39398*
	6	5370.18	46802.8*	35917.1	38766.8*	32842.1	40184*	36723	38469*	38469*	38439.1*
	7	5292.38	47305*	36102.5	38580.1*	32719.5	39407.6*	36665.7	38128.1	38128.1	38128.1
	8	4747.85	48211.3*	37242.5	40287.1*	33204.3	40937.5*	38097.5	39968.5*	40019.9*	40072*
	9	4665.38	49824.7*	38916.4	41829.9*	35079.9	43504*	39880.3	41654.1*	41655.4*	41641.5*
	10	4996.74	45482.8*	36747.3	38997.3*	32891.9	39882.5*	37409.8	38886.9*	38886.9*	38878.5*
Average		5150.30	47721.94	36920.37	39776.28	33437.61	40608.97	37708.81	39447.58	39472.17	39501.5

* indicates solution is not optimal-allocated computation time exceeded 600s.

C.7 High Return $(n = 25)$

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$			
		LB	UB	LB	UB	LB	UB	LB	UB-FL	UB-MC	UB-SP
125	1	2312.49	4812.87	4393.96	4812.87	4170.14	4812.87	4393.96	4812.87	4812.87	4812.87
	2	2360.86	4635.89	4057	4635.89	3889.31	4635.89	4057	4635.89	4635.89	4635.89
	3	2861.92	4845.44	4494.49	4845.44	4384.53	4845.44	4494.49	4845.44	4845.44	4845.44
	4	1675.61	4414.15*	3769.82	4394.41	3601.64	4394.41	3775.64	4394.41	4394.41	4394.41
	5	1866.07	4408.13	3815.58	4408.13	3657.24	4408.13	3815.58	4408.13	4408.13	4408.13
	6	5158.93	6494.15	6278.86	6494.15	6142	6494.15	6278.86	6494.15	6494.15	6494.15
	7	1181.54	4083.36	3496.93	4083.36	3367.51	4083.36	3497.17	4083.36	4083.36	4083.36
	8	2811.91	4698.35	4352.37	4698.35	4153.69	4698.35	4352.37	4698.35	4698.35	4698.35
	9	3153.98	5187.34	4741.92	5187.34	4625.87	5187.34	4747.06	5187.34	5187.34	5187.34
	10	3578.91	5321.16	4983.03	5321.16	4901.87	5321.16	4983.03	5321.16	5321.16	5321.16
Average		2696.22	4890.08	4438.40	4888.11	4289.38	4888.11	4439.52	4888.11	4888.11	4888.11
250	1	2643.76	7004.25*	6357.51	6977.06	5942.21	6977.06	6365.33	6977.06	6977.06	6977.06
	2	2693.36	6555.65	5855.03	6555.65	5629.4	6555.65	5857.8	6555.65	6555.65	6555.65
	3	3217.78	6965.34	6281.02	6965.34	6126.03	6965.34	6281.02	6965.34	6965.34	6965.34
	4	2124.98	6624.74*	5789.58	6601.94	5532.86	6601.94	5818.08	6601.94	6601.94	6601.94
	5	2259.52	6320.74	5769.11	6320.74	5475.52	6320.74	5788.33	6320.74	6320.74	6320.74
	6	5429.99	7994.15	7637.99	7994.15	7362.48	7994.15	7637.99	7994.15	7994.15	7994.15
	7	1617.24	6474.63	5632.19	6474.63	5507.84	6474.63	5675.93	6474.63	6474.63	6474.63
	8	3134.05	6530.19	6008.17	6530.19	5661.8	6530.19	6024.71	6530.19	6530.19	6530.19
	9	3547.59	7040.44	6415.06	7040.44	6220.34	7040.44	6423.73	7040.44	7040.44	7040.44
	10	3896	7159.58	6589.7	7159.58	6450.22	7159.58	6623.38	7159.58	7159.58	7159.58
Average		3056.43	6866.97	6233.54	6861.97	5990.87	6861.97	6249.63	6861.97	6861.97	6861.97

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$			
		LB	UB	LB	UB	LB	UB	LB	UB-FL	UB-MC	UB-SP
500	1	3306.28	10032.4	9390.38	10032.4	8627.78	10032.4	9480.03	10032.4	10032.4	10032.4
	2	3358.35	10055.6	8859.75	10055.6	8410.73	10055.6	8893.69	10055.6	10055.6	10055.6
	3	3857.98	10077.2	9195.9	10077.2	8886.98	10077.2	9358.51	10077.2	10077.2	10077.2
	4	3012.63	9718.81	8690.33	9718.81	8293.4	9718.81	8805.62	9718.81	9718.81	9718.81
	5	3044.44	9805.69	8913.91	9805.69	8516.64	9805.69	8949.77	9805.69	9805.69	9805.69
	6	5972.12	10994.2	10223.2	10994.2	9803.44	10994.2	10223.2	10994.2	10994.2	10994.2
	7	2340.17	9526.4	8719.98	9526.4	8370.6	9526.4	8819.36	9526.4	9526.4	9526.4
	8	3778.33	9476.37	8856.2	9476.37	8242.19	9476.37	8870.57	9476.37	9476.38	9476.37
	9	4334.81	9775.09	9171.25	9775.09	8826.94	9775.09	9214.97	9775.09	9775.09	9775.09
	10	4530.18	10328.4	9469.74	10328.4	8968.35	10328.4	9478.02	10328.4	10328.4	10328.4
Average		3753.53	9979.02	9149.06	9979.02	8694.71	9979.02	9209.37	9979.02	9979.02	9979.02
1000	1	4631.34	14891.9	13833.5	14891.9	12666.7	14891.9	14049.9	14891.9	14891.9	14891.9
	2	4688.32	14845.1	13599.5	14845.1	12669.3	14845.1	13639.1	14845.1	14845.1	14845.1
	3	5137.67	14697	13611.3	14697	12719.7	14697	13810	14697	14697	14697
	4	4599.63	14615.4	13465.1	14615.4	12605.5	14615.4	13650.8	14615.4	14615.4	14615.4
	5	4559.02	14481.3	13492.3	14481.3	12394.8	14481.3	13563	14481.3	14481.3	14481.3
	6	7056.36	15689.6	14680.2	15689.6	14051.2	15689.6	14757.5	15689.6	15689.6	15689.6
	7	3746.73	14099.2	13130.2	14099.2	12199.8	14099.2	13271.1	14099.2	14099.2	14099.2
	8	5066.88	13751.2	12862.1	13751.2	11986.3	13751.2	12950	13751.2	13751.2	13751.2
	9	5866.04	14274	13293.7	14274	12296.1	14274	13312.5	14274	14274	14274
	10	5798.55	14299.8	13712.7	14299.8	12636.6	14299.8	13712.7	14299.8	14299.8	14299.8
Average		5115.05	14564.45	13568.06	14564.45	12622.6	14564.45	13671.66	14564.45	14564.45	14564.45

* indicates solution is not optimal-allocated computation time exceeded 600s.

C.8 High Return $(n = 50)$

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$			
		LB	UB	LB	UB	LB	UB	LB	UB-FL	UB-MC	UB-SP
125	1	10577.8	13424.3*	12864.5	13424.3	12731.1	13424.3	12864.5	13424.3	13424.3	13424.3
	2	6952.03	10763.4*	9714.8	10437.1	9542.93	10437.1	9716.61	10437.1	10437.1	10437.1
	3	8196.99	11047.9*	10259.6	10979.3*	10201.9	10979.3*	10259.6	10979.3	10979.3*	10979.3*
	4	3475.91	9143.18*	7921.49	8690.49*	7455.64	8655.55*	7925.02	8645.26*	8628.98*	8628.98*
	5	11880.1	14410.3*	13863.7	14346.3	13776.7	14346.3	13863.7	14346.3	14346.3	14346.3
	6	2470.71	9419.94*	7396.8	8510.44*	6917.19	8470.61*	7404.91	8318.79*	8318.79*	8321.39*
	7	5584.89	10444	9313.42	10002.8	9048.19	10002.8*	9318.6	10002.8	10002.8	10002.8
	8	1613.56	9146.57*	7331.45	8133.51*	6534.91	8456.09*	7336.29	8106.54*	8106.54*	8106.54*
	9	6107.42	10128.2*	9176.94	9915.61	9029.89	9915.61	9188.2	9915.61	9915.61	9915.61
	10	3247.05	9102.86*	7724.99	8602.96*	7277.42	8594.32*	7741.71	8559.84*	8559.84*	8576.16*
Average		6010.65	10703.07	9556.77	10304.28	9251.59	10328.20	9561.91	10273.58	10271.96	10273.85
250	1	10879.2	16668*	15355.9	16426.3	15093.8	16426.3	15362.1	16426.3	16426.3	16426.3
	2	7409.73	14348.5*	12593	13625.1	12309.8	13625.1	12645.8	13625.1	13625.1	13625.1
	3	8600.62	14183.1*	12718.9	14029.5*	12608*	14029.5*	12718.9	14029.5*	14033*	14029.5*
	4	3941.83	14485.4*	11580.1	13060.2*	11033.1*	13041.7*	11653.6	12874.6*	12903.2*	12937*
	5	12408.2	17721.3*	16269.7	17221.3	16129.7	17221.3	16273.2	17221.3	17221.3	17221.3
	6	2989.98	14266.5*	11577.9	12753.2*	10994.4	12706.2*	11644.8	12674.2*	12674.2*	12706.2*
	7	6026.01	15521.6*	12966.6	14124*	12537*	14150.5*	12991.2	14129.2*	14124*	14124*
	8	2150.96	14502.6*	11435	12803*	10588.3*	12836.4*	11547	12593.6*	12593.6*	12613.5*
	9	6535.24	14557.5*	12491.8	13606.6	12183.5	13606.6	12513	13606.6	13606.6	13606.6
	10	3700.95	13821*	11732	12863.3*	11076.5*	12900.1*	11813.3	12855.6*	12855.6*	12855.6*
Average		6464.27	15007.55	12872.09	14051.25	12455.41	14054.37	12916.29	14003.60	14006.29	14014.51

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$			
		LB	UB	LB	UB	LB	UB	LB	UB-FL	UB-MC	UB-SP
500	1	11482	21915.2*	19934.7	21533.7	19539.9	21533.7	19947.5	21533.7	21533.7	21533.7
	2	8289.69	21186.4*	17597.1	18829.8	17035	18829.8	17676.3	18829.8	18829.8	18828.8
	3	9407.89	20021.3*	17559.7	19386.2*	17342.9	19386.2*	17578.5	19386.2	19386.2	19386.2*
	4	4841.27	20547.3*	17178	19457.7*	16095.2	19379.4*	17440.5	19282.7*	19283.2*	19193.7*
	5	13346.4	23426.7*	20620.7	22533.2	20419.9	22533.2	20644.5	22533.2	22533.2	22533.2
	6	4028.53	22265.7*	17816.2	19770.9*	17015.2	19630.4*	17978.3	19566.3*	19566.3*	19630.4*
	7	6899	22834.5*	18845.3	20631.6*	18181.6	20631.1*	18991.9	20631.6*	20825.4*	20601.3*
	8	3103.61	22167.6*	17445.1	19629.4*	16210.4	19833*	17835.7	19416.2*	19302.8*	19413.8*
	9	7390.89	21022.6*	17916.5	19171.6	17294.5	19171.6	17984	19171.6	19978.26	19171.6
	10	4608.74	21034.9*	17712.6	19401.1*	16799.8	19302.8*	17904.6	19302.8*	19406.7*	19401.1*
Average		7339.80	21642.22	18262.59	20034.52	17593.44	20023.12	18398.18	19965.41	19945.51	19969.48
1000	1	12687.6	30554.4*	27455.7	29347.3	26587.4	29347.3	27528.2	29347.3	29347.3	29347.3
	2	10017.2	31446.1*	25600.8	27988*	24531.3	27988	25748.7	27988*	27988*	27643.5*
	3	11022.4	30217.1*	25876.5	27875.5	25370.9	27875.5	25960.5	27875.5	27875.5	27875.5
	4	6593.32	31364.5*	25541.2	27616.9*	23964	27405.8	25950.8	27405.8	27405.8	27405.8
	5	15033.9	32893.6*	28750.5	30848.8	27668.9	30848.8	28831.4	30848.8	30848.8	30848.8
	6	6049.11	33899.9*	26805.9	28788.8*	25577.5	28788.8	27137.8	28788.8	28788.8	28788.8
	7	8641.13	32621.2*	27581.4	29429.3	26223.7	29429.3	27769.1	29429.3	29429.3	29429.3
	8	4892.94	32698.9*	26413.6	29425.9*	24564.4	29549.8*	27001.5	29294.6*	29321.8*	29378.8*
	9	9102.2	32037.3*	26614.9	28449.1	25496	28449.1	26738.9	28449.1	28449.1	28449.1
	10	6413.04	33468.8*	26451.2	27927.8	24993.8	27927.8	26708.3	27927.8	27927.8	27927.8
Average		9045.28	32120.18	26709.17	28769.74	25497.79	28761.02	26937.52	28735.5	28747.28	28709.47

* indicates solution is not optimal-allocated computation time exceeded 600s.

C.9 High Return ($n = 75$)

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$			
		LB	UB	LB	UB	LB	UB	LB	UB-FL	UB-MC	UB-SP
125	1	5180.05	14693.1*	11642	13153*	11176.1	13315.7*	11642	13015.8*	13029.6*	13070.8
	2	18573.8	24426.1*	22463	23390.4*	22213.1	23553*	22463.7	23309.8*	23299.9*	23299.9*
	3	10919.4	17484.2*	15950.7	16775.6*	15495.7	16897.7*	15958.4	16707.7*	16707.7*	16707.7*
	4	4686.71	13836.5*	11628.1	12884.9*	10924.6	13223.1*	11629.9	12800*	12800*	12800*
	5	21187.8	24534.8*	23694.1	24409.8*	23595.8	24409.8*	23696	24409.8*	24409.8*	24409.8*
	6	4588.37	14586.7*	11391.2	13225*	10964.9	13648.5*	11399.6	13190.1*	13082.9*	13100.3*
	7	12063.4	17700.9*	16411.7	17301.9*	16156.6	17301.9*	16412.1	17301.9*	17301.9*	17301.9*
	8	3515.13	14306.1*	11090.6	12655.4*	10371.2	12891.9*	11101.2	13322.3*	12383.5*	12366.9*
	9	4744.27	14972.1*	11575.2	13381.7*	10963.5	13842.9*	11588.2	13289.3*	13288*	13426*
	10	9172.04	16393.2*	14552.5	15638*	14211.8	15630.2*	14552.8	15549.9*	15652.8*	15553.9*
Average		9463.10	17293.37	15039.91	16281.57	14607.33	16471.47	15044.39	16289.66	16195.61	16203.72
250	1	5745.25	21946.1*	17406.5	19484*	16932.6	19571.6*	17501.6	19472.2*	19464.1*	19494*
	2	19038.5	29421.5*	26269	27928.9*	26001.4	28006*	26284.2	27615.6*	27659*	27635.5*
	3	11339.3	23154.3*	20321.1	21550.6	19773.4	21742.8*	20382.3	21550.6	21550.6	21550.6*
	4	5290.15	21923.8*	17507.3	19578.2*	16505.1	19681*	17601.3	19478*	19378.6*	19396.3*
	5	21592.5	28159.8*	26478.4	27909.8*	26301.2	27909.8*	26482.2	27909.8*	27909.8*	27909.8*
	6	5133.43	21102*	17178.5	19689.5*	16633	20104.6*	17310.3	19649.7*	19733.1*	19618.2*
	7	12471.7	23821.1*	20754.2	22506.4*	20403.7	22506.4*	20756.2	22506.4*	22506.4*	22506.4*
	8	4076.5	22169.5*	17124.9	19351.9*	16152.8	19210.1*	17278.5	19038.2*	19006.7*	19045.9*
	9	5301.61	22797.3*	17503.4	20187.6*	16679.2	20508.1*	17593.8	20055.1*	20013.5*	19964.3*
	10	10295.1	23206.9*	19351.1	21125*	18936.4	21168.3*	19365.3	21121.7*	21095.4*	21115*
Average		10028.40	23770.23	19989.44	21931.19	19431.88	22040.87	20055.57	21839.73	21831.72	21823.6

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original		$(\ell, S) - like$		(ℓ, S, WW)		$FL = MC = SP$			
		LB	UB	LB	UB	LB	UB	LB	UB-FL	UB-MC	UB-SP
500	1	6859.69	33177.2*	26400.2	29623.8*	25661.5	29856.2*	26602.1	29609.1*	29410.9*	29390.4*
	2	19968	40275.7*	33027.6	35991.2*	32487.6	36224.4*	33162.6	35457.3*	35574.8*	35731.1*
	3	12179.3	34396.1*	27968	30899.6*	27039.2	31237.1*	28128.4	30589*	30376.6*	30992.1*
	4	6431.11	34568.2*	26521.8	29611.4*	25235.7	30470.7*	26769.8	29561.2*	29285.9*	29606.6*
	5	22401.9	37160.2*	32047	34568.7*	31692.5	34568.7*	32054.5	34568.7*	34568.7*	34568.7*
	6	6223.54	32236.8*	26076.8	29194.9*	25175.5	29615*	26381.2	28917.3*	29078.2*	29169*
	7	13288.3	33760.4*	28649.4	31795.4*	28146.1	31614.4*	28709.2	31873.6*	31873.6*	32125.1*
	8	5142.69	33718.3*	26452.1	29392.1*	25218.7	29976.6*	26815.8	28906.7*	29409.8*	29258*
	9	6390.03	35457.1*	26709.2	30129.5*	25482.8	29936.5*	27004.5	29906.2*	29785.3*	30119.8*
	10	11382.4	34007.1*	27470.9	30829.9*	26985.8	31030.5*	27539.3	30662.4*	30575.1*	30865.7*
Average		11026.70	34875.71	28132.3	31203.65	27312.54	31453.01	28316.74	31005.15	30993.89	31182.65
1000	1	8956.85	51761.6*	39303.5	42828*	38109.7	43032.2*	39712.9	42856.6*	42828*	43115.4
	2	21770.2	53787.6*	44722.1	47924.5*	43531.3	48947*	44970.7	48069.6*	48325.5*	48351.4*
	3	13859.2	49065.8*	40251.2	42758.8*	38793.9	42705.6*	40544.5	42642.1*	42642.1*	42642.1*
	4	8488.13	52583.3*	39729.3	43489.8*	37871.2	43248.6*	40290.6	43034.6*	43143.8*	42949.1*
	5	24020.7	48977.3*	42783	45434.9	41885.2	45716*	42789.5	45434.9*	45434.9*	45434.9*
	6	8403.77	50072.7*	39434.5	44341*	38394.9	44351.6*	39951.1	43545.8*	43545.8*	43691.2*
	7	14921.4	47608.4*	41031.8	44634.7*	39938.6	44265.5*	41107.6	44515.6*	44005.5*	44515.6*
	8	7115.93	51224.8*	39999.1	44445*	38355.8	44649.8*	40664.5	43845.6*	43800.7*	43866.6*
	9	8537.04	51931.3*	40362.2	44537.2*	38563.1	45034.4*	41151.4	44028.5*	44058.3*	44693.1*
	10	13428.6	50524.6*	40091.9	44082.4*	39249.3	44169.7*	40419.7	44072.7*	44429.6*	44072.7*
Average		12950.18	50753.74	40770.86	44447.63	39469.3	44612.04	41160.25	44204.6	44221.42	44333.21

* indicates solution is not optimal-allocated computation time exceeded 600s.

Appendix D

Detailed Results of Lower Bounds - Joint Setups

D.1 Low Return ($n = 25$)

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
125	1	464.35	2584.51	2584.51	2584.51	2470.55	2584.51
	2	506.64	2734.12	2734.12	2734.12	2622.2	2734.12
	3	474.44	2692.42	2692.42	2692.42	2584.34	2692.42
	4	426.11	2569.15	2569.15	2569.15	2454.32	2569.15
	5	524.69	2549.41	2549.41	2549.41	2437.75	2549.41
	6	453.10	2486.77	2486.77	2486.77	2404.32	2486.77
	7	448.33	2502.97	2502.97	2502.97	2400.82	2502.97
	8	496.14	2786.56	2786.56	2786.56	2686.4	2786.56
	9	510.52	2570.77	2570.77	2570.77	2462.07	2570.77
	10	520.00	2634.42	2634.42	2634.42	2516.2	2634.42
Average		482.43	2611.11	2611.11	2611.11	2503.90	2611.11
250	1	914.32	4075.26	4075.26	4075.26	3859.58	4075.26
	2	997.12	7356.91	7356.91	7356.91	4149.56	7356.91
	3	948.88	4354.53	4354.53	4354.53	4128.93	4354.53
	4	847.59	4176.54	4176.54	4176.54	3953.55	4176.54
	5	988.68	4171.97	4171.97	4171.97	3969.91	4171.97
	6	906.19	4172.86	4172.86	4172.86	3970.52	4172.86
	7	888.35	4060.85	4060.85	4060.85	3841.19	4060.85
	8	934.31	4466.74	4466.74	4466.74	4259.29	4466.74
	9	993.29	4144.17	4144.17	4144.17	3915.56	4144.17
	10	977.07	4228.45	4228.45	4228.45	3988.08	4228.45
Average		939.58	4220.83	4220.83	4220.83	4003.62	4220.83

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
500	1	1724.64	6163.41	6163.41	6163.41	5759.64	6163.41
	2	1831.34	6496.81	6496.81	6496.81	6075.73	6496.81
	3	1836.61	6875.62	6875.62	6875.62	6415.02	6875.62
	4	1643.97	6406.88	6406.88	6406.88	5985.96	6406.88
	5	1828.59	6641.72	6641.72	6641.72	6207.39	6641.72
	6	1789.93	6593.56	6593.56	6593.56	6191.9	6593.56
	7	1658.50	6193.53	6193.53	6193.53	5799.42	6193.53
	8	1684.38	6848.78	6848.78	6848.78	6417.41	6848.78
	9	1945.94	6566.74	6566.74	6566.74	6151.59	6566.74
	10	1787.06	6540.60	6540.60	6540.60	6109.81	6540.60
Average		1773.10	6532.77	6532.77	6532.77	6111.39	6532.77
1000	1	3274.45	9386.12	9386.12	9386.12	8559.3	9386.12
	2	3269.02	10025.1	10025.1	10025.1	9155.94	10025.1
	3	3405.95	10171.9	10171.9	10171.9	9340.3	10171.9
	4	3108.61	9922.63	9922.63	9922.63	9065.62	9922.63
	5	3354.71	10232.7	10232.7	10232.7	9413.89	10232.7
	6	3306.88	10126.5	10126.5	10126.5	9304.27	10126.5
	7	3036.81	9270.1	9270.1	9270.1	8446.8	9270.1
	8	3127.85	10417.6	10417.6	10417.6	9528.14	10417.6
	9	3466.87	9701.9	9701.9	9701.9	8849.84	9701.9
	10	3213.96	9774.78	9774.78	9774.78	8994.92	9774.78
Average		3256.51	9902.93	9902.93	9902.93	9065.90	9902.93

D.2 Low Return ($n = 50$)

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
125	1	565.36	5534.8*	5304.67	5304.67	5192.83	5304.67
	2	570.40	5439.04*	5075.8	5075.8	4979.3	5075.8
	3	553.55	5599.56*	5199.06	5199.06	5103.8	5199.06
	4	561.28	5543.74*	5262.1	5262.1	5157.49	5262.1
	5	608.52	5010.79*	4766.46	4766.46	4679.99	4766.46
	6	598.26	5192.71*	4997.99	4997.99	4892.94	4997.99
	7	601.34	5667.57*	5467.11	5471.91	5353.75	5471.91
	8	566.69	5377.71*	5277.03	5277.03	5169.48	5277.03
	9	583.84	5406.86*	5151.71	5151.71	5034.23	5151.71
	10	556.45	5340.98*	5029.13	5029.13	4926.11	5029.13
Average		576.57	5411.38	5153.11	5153.59	5048.99	5153.59
250	1	1065.16	8898.12*	8529.02	8529.02	8313.62	8529.02
	2	1118.48	8302.42*	8114.56	8114.56	7903.38	8114.56
	3	1095.2	8879.86*	8518.7	8518.7	8318.15	8518.7
	4	1110.99	9000.93*	8504.39	8504.39	8278.16	8504.39
	5	1157.45	8069.31*	7632.11	7632.11	7427.64	7632.11
	6	1110.25	8697.83*	8095.63	8095.63	7906	8095.63
	7	1173.83	9280.84*	8772.66	8772.66	8556.74	8772.66
	8	1093.83	9431.14*	8523.25	8523.25	8309.96	8523.25
	9	1121.51	8711.75*	8231.65	8231.65	8003.38	8231.65
	10	1080.81	8683.31*	8028	8028	7818.74	8028
Average		1112.75	8795.55	8295	8295	8083.58	8295

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original LB	Original UB	$(\ell, S) - like = FL = MC = SP$ LB	$(\ell, S) - like = FL = MC = SP$ UB	(ℓ, S, WW) LB	(ℓ, S, WW) UB
500	1	2056.98	13555.5*	13027.6	13027.6	12573	13027.6
	2	2128.89	13721*	12595	12595	12169.5	12595
	3	2075.36	13757.4*	12960.9	12960.9	12521.3	12960.9
	4	2144.6	13574.8*	12959.9	12959.9	12507	12959.9
	5	2159.19	12592.5*	11808	11808	11367.6	11808
	6	2120.27	13461.5*	12569.5	12569.5	12143.9	12569.5
	7	2188.55	14014.8*	13528.9	13528.9	13077.7	13528.9
	8	2042.05	14005.5*	13107.1	13107.1	12685.6	13107.1
	9	2105.41	13380*	12775.8	12775.8	12308.7	12775.8
	10	2006.89	13282*	12365.1	12365.1	11924.3	12365.1
Average		2102.82	13534.5	12769.78	12769.78	12327.86	12769.78
1000	1	4000.77	20519.4*	19749	19749	18881.2	19749
	2	3939.26	21182.7*	19152.3	19152.3	18300.7	19152.3
	3	3825.42	21304.8*	19706.2	19706.2	18788.5	19706.2
	4	4077.8	20981.8*	19714.9	19714.9	18811.4	19714.9
	5	4073.09	19339.6*	18021.1	18021.1	17191.3	18021.1
	6	3975.92	20700.3*	19373.3	19373.3	18514.9	19373.3
	7	3979.09	20083.7*	19938.6	19938.6	19077	19938.6
	8	3911.06	20630.5*	19549.7	19549.7	18664.6	19549.7
	9	3942.07	20634.3*	19208.8	19208.8	18339.2	19208.8
	10	3674.01	20375.1*	19136.6	19136.6	18232.9	19136.6
Average		3939.85	20635.22	19355.05	19355.05	18480.17	19355.05

* indicates solution is not optimal-allocated computation time exceeded 600s.

D.3 Low Return $(n = 75)$

SC	No	Original LB	Original UB	$(\ell, S) - like = FL = MC = SP$ LB	$(\ell, S) - like = FL = MC = SP$ UB	(ℓ, S, WW) LB	(ℓ, S, WW) UB
125	1	666.69	8320.16*	7781.52	7781.52	7663.72	7781.52
	2	583.38	7999.59*	7431.44	7431.44	7316.31	7431.44
	3	640.09	8379.92*	7694.35	7694.35	7591.97	7694.35
	4	635.57	8368.35*	7972.62	7972.62	7860.78	7972.62
	5	601.41	8377.84*	7988.49	7988.49	7880.03	7988.49
	6	597.97	8229.95*	7899.33	7899.33	7792.64	7899.33
	7	592.33	8437.54*	8011.76	8011.76	7903.88	8011.76
	8	641.58	8388.95*	8095.41	8095.41	7986.08	8095.41
	9	624.73	8209.51*	7858.59	7858.59	7759.65	7858.59
	10	618.33	8468.1*	8158.89	8158.89	8054.63	8158.89
Average		620.21	8317.99	7889.24	7889.24	7780.97	7889.24
250	1	1237.59	13686.6*	12579.5	12579.5	12347.7	12579.5
	2	1141.33	12965*	12033.4	12033.4	11815.8	12033.4
	3	1201.04	13172*	12130.4	12130.4	11905.5	12130.4
	4	1220.39	13625*	12771.9	12771.9	12537.5	12771.9
	5	1202.83	13999.4*	12851.9	12851.9	12638	12851.9
	6	1133.72	13654.4*	12573.3	12573.3	12357	12573.3
	7	1165.38	13592.6*	12803.3	12803.3	12570.9	12803.3
	8	1219.04	13598.1*	12920.5	12920.5	12688.3	12920.5
	9	1197.6	13647.8*	12691.7	12691.7	12470.9	12691.7
	10	1167.3	14085.4*	13274.7	13274.7	13048.2	13274.7
Average		1188.62	13602.63	12663.06	12663.06	12437.98	12663.06

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
500	1	2375.9	20961.2*	19363.6	19363.6	18903.9	19363.6
	2	2257.22	20315.4*	18567.5	18567.5	18113.3	18567.5
	3	2271.86	21038*	18828.1	18828.1	18358.2	18828.1
	4	2321.98	20507.9*	19226.5	19226.5	18769.6	19226.5
	5	2306.48	21601.8*	19937.8	19937.8	19500.7	19937.8
	6	2186.97	21111*	19852.5	19852.5	19405.4	19852.5
	7	2261.02	20999.1*	19629.4	19629.4	19213.8	19629.4
	8	2271.76	21225.7*	20111.1	20111.1	19633.2	20111.1
	9	2289.29	21486.3*	19737.8	19737.8	19290.7	19737.8
	10	2214.72	22121.1*	20508.5	20508.5	20037.5	20508.5
Average		2275.72	21136.75	19576.28	19576.28	19122.63	19576.28
1000	1	4608.03	31345.4*	29553.9	29553.9	28621.4	29553.9
	2	4413.49	30492.7*	27993.8	27993.8	27082.3	27993.8
	3	4247.08	32083.4*	28645.9	28645.9	27751.9	28645.9
	4	4357.92	31926.2*	29098.1	29098.1	28163.2	29098.1
	5	4339.77	32879.6*	30112.9	30112.9	29226.7	30112.9
	6	4264.32	32648.8*	29589.4	29589.4	28678.3	29589.4
	7	4337.33	32929.5*	29312.2	29312.2	28417	29312.2
	8	4362.27	34316.3*	30380.3	30380.3	29454.3	30380.3
	9	4321.8	33470.4*	30037.1	30037.1	29181.7	30037.1
	10	4266.52	33743.6*	31411.5	31411.5	30481.7	31411.5
Average		4351.85	32583.59	29613.51	29613.51	28705.85	29613.51

* indicates solution is not optimal-allocated computation time exceeded 600s.

D.4 Medium Return $(n = 25)$

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
125	1	624.87	3025.5	3017.02	3025.5	2932.56	3025.5
	2	1133.14	3186.39	3180.65	3186.39	3180.65	3186.39
	3	606.06	2988.24	2980.83	2988.24	2926.97	2988.24
	4	588.9	2867.73	2864.53	2867.73	2758.33	2867.73
	5	686.57	2919.99	2905.73	2919.99	2831.76	2919.99
	6	529.38	3021.69	3021.69	3021.69	2917.47	3021.69
	7	740.54	3171.73	3170.33	3171.73	3110.33	3171.73
	8	587.14	2923.78	2907.39	2923.78	2907.39	2923.78
	9	702.99	2829.66	2825.61	2829.66	2815.92	2829.66
	10	619.82	2845.67	2827.16	2845.67	2813.5	2845.67
Average		681.94	2978.04	2970.09	2978.04	2919.49	2978.04
250	1	1063.65	4941.63	4935.39	4941.63	4775.1	4941.63
	2	1553.97	5184.67	5143.98	5184.67	5093.66	5184.67
	3	1043.79	5031.94	4986.48	5031.94	4864.89	5031.94
	4	1001.96	4817.01	4817.01	4817.01	4592.78	4817.01
	5	1177.17	4846.18	4846.18	4846.18	4654.44	4846.18
	6	1037.18	4847.34	4847.34	4847.34	4647.1	4847.34
	7	1157.84	5136.24	5114.58	5136.24	4935.45	5136.24
	8	1021.21	4657.9	4654.33	4657.9	4566.55	4657.9
	9	1168.14	4663.63	4653.49	4663.63	4564.35	4663.63
	10	1045.72	4664.33	4662.78	4664.33	4527.25	4664.33
Average		1127.06	4879.09	4866.16	4879.09	4722.16	4879.09

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
500	1	1896.95	7531.13	7531.13	7531.13	7146.5	7531.13
	2	2395.63	7916.87	7896.41	7916.87	7549.38	7916.87
	3	1881.98	7917.85	7917.85	7917.85	7569.41	7917.85
	4	1760.84	7509.64	7509.64	7509.64	7049.38	7509.64
	5	2022.81	7411.49	7411.49	7411.49	7081.02	7411.49
	6	1881.80	7743.13	7743.13	7743.13	7318.09	7743.13
	7	1992.46	7942.05	7942.05	7942.05	7666.58	7942.05
	8	1813.12	7384.96	7384.96	7384.96	7058.47	7384.96
	9	2058.79	7160.78	7160.7	7160.78	6881.8	7160.78
	10	1826.66	7119.97	7119.97	7119.97	6746.18	7119.97
Average		1953.10	7563.79	7561.73	7563.79	7206.68	7563.79
1000	1	3432.15	11562.6	11562.6	11562.6	10811.8	11562.6
	2	3887.36	11982.8	11982.8	11982.8	11274.4	11982.8
	3	3435.36	12042.9	12010.7	12042.9	11324.1	12042.9
	4	3198.89	11670.9	11670.9	11670.9	10884.6	11670.9
	5	3551.44	11025.6	11025.6	11025.6	10334.1	11025.6
	6	3436.12	11487.7	11469.8	11487.7	10685.8	11487.7
	7	3580.56	12116.6	12116.6	12116.6	11402.3	12116.6
	8	3310.31	11234.7	11234.7	11234.7	10646.1	11234.7
	9	3531.99	10689.7	10689.7	10689.7	9943.79	10689.7
	10	3246.52	11265.4	11265.4	11265.4	10443.1	11265.4
Average		3461.07	11507.89	11502.88	11507.89	10775.01	11507.89

D.5 Medium Return $(n = 50)$

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
125	1	832.32	6064.39*	6001.6	6032.65	5954.03	6032.65
	2	1003.86	5983.83*	5718.44	5813.3	5650.76	5813.3
	3	1079.88	5876.04*	5720.24	5801.75	5700.11	5801.75
	4	1072.93	6094.6*	5968.15	6032.21	5873.31	6032.21
	5	721.19	5991.64*	5899	5936.16	5896.3	5936.16
	6	730.43	5859.78*	5765.69	5794.99	5693.41	5794.99
	7	843.40	6022.48*	5928.37	5990.03	5850.13	5990.03
	8	806.40	6091.62*	5931.32	5990.55	5911.01	5990.55
	9	1229.96	6047.04*	5764.35	5835.18	5683.9	5835.18
	10	778.77	6009.69*	5876.65	5955.05	5874.61	5955.05
Average		909.91	6004.11	5857.38	5918.19	5808.76	5918.19
250	1	1390.9	9991.33*	9617.78	9680.22	9500.82	9680.22
	2	1536.32	9765.17*	9388.49	9398.55	9250.52	9398.55
	3	1634.25	9515.39*	9243.94	9300.96	9119.78	9300.96
	4	1595.93	10223.4*	9653.44	9754.04	9437.96	9754.04
	5	1274.82	10122.7*	9762.66	9846.47	9688.79	9846.47
	6	1290.31	9669.62*	9424	9424	9228.62	9424
	7	1372.2	10091.8*	9861.42	9932.51	9668.42	9932.51
	8	1378.42	10097.6*	9577.05	9589.36	9433.13	9589.36
	9	1760.91	9690.7*	9474.21	9524.89	9270.47	9524.89
	10	1275.93	9851.95*	9552.48	9569.88	9438.35	9569.88
Average		1451	9901.97	9555.55	9602.09	9403.69	9602.09

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original LB	Original UB	$(\ell, S) - like = FL = MC = SP$ LB	$(\ell, S) - like = FL = MC = SP$ UB	(ℓ, S, WW) LB	(ℓ, S, WW) UB
500	1	2428.55	15242*	14886.6	14886.6	14526.2	14886.6
	2	2535.3	15243.6*	14487.7	14530.7	14151.6	14530.7
	3	2694.29	14763.2*	14054.1	14054.1	13766.7	14054.1
	4	2641.93	15648.7*	15024.8	15070.8	14568.1	15070.8
	5	2328.04	15660.1*	15253.8	15253.8	14957	15253.8
	6	2380.53	15351.3*	14790.6	14790.6	14469.6	14790.6
	7	2380.5	15798.6*	15234.7	15234.7	14895.5	15234.7
	8	2414.17	15580.9*	15113.8	15113.8	14738.6	15113.8
	9	2693.33	15337.8*	14724.7	14788.8	14331.2	14788.8
	10	2270.26	15527.2*	15028.5	15037.3	14671.9	15037.3
Average		2476.69	15415.34	14859.93	14876.12	14507.64	14876.12
1000	1	4269.82	23789.6*	22339.7	22339.7	21648.1	22339.7
	2	4297.05	22808.5*	21956.7	21964.7	21265.6	21964.7
	3	4680.59	22340.6*	21314.6	21314.6	20760	21314.6
	4	4569.69	23628.1*	22745.7	22745.7	21806.4	22745.7
	5	4293.51	23823.1*	22946.7	22946.7	22279.5	22946.7
	6	4349.62	23267.9*	22156.6	22156.6	21370.8	22156.6
	7	4271.79	24073.3*	23114.9	23123.8	22323.7	23123.8
	8	4305.19	24088.7*	22623.9	22623.9	21800.9	22623.9
	9	4467.93	23235.4*	22001	22003.1	21196.6	22003.1
	10	4133.11	24220.5*	22874.8	22874.8	22172.1	22874.8
Average		4363.83	23527.57	22407.46	22409.36	21662.37	22409.36

* indicates solution is not optimal-allocated computation time exceeded 600s.

D.6 Medium Return ($n = 75$)

SC	No	Original LB	Original UB	$(\ell, S) - like = FL = MC = SP$ LB	$(\ell, S) - like = FL = MC = SP$ UB	(ℓ, S, WW) LB	(ℓ, S, WW) UB
125	1	1789.47	9357.35*	9107.14	9299.67	9038.34	9299.67
	2	1404.45	9058.1*	8704.39	8773.46	8702.62	8773.46
	3	1492.5	8881.92*	8638.02	8771.58	8547.62	8771.58
	4	1535.4	9392.24*	9006.74	9123.16	8923.7	9123.16
	5	960.12	8814.17*	8625.49	8678	8549.67	8678
	6	1575.52	8911.88*	8691.68	8775.69	8568.65	8775.69
	7	1327.46	8906.2*	8622.63	8718.05	8519.08	8718.05
	8	887.23	8785.96*	8595.18	8647.35	8472.44	8647.35
	9	919.22	9023.07*	8918.82	8944.88	8841.21	8944.88
	10	1156.55	9145.75*	8750.68	8854.57	8705.58	8854.57
Average		1304.79	9027.66*	8766.08	8858.64	8686.89	8858.64
250	1	2366.15	14589*	14349	14410.4	14177.4	14410.4
	2	1959.81	14946.5*	14358	14463.4	14266.2	14463.4
	3	2074.14	15123.8*	14301	14441.3	14122.5	14441.3
	4	2129.03	15534.9*	14469.4	14527.1	14306.4	14527.1
	5	1519.67	15088*	14122.6	14157.3	13923.2	14157.3
	6	2119.96	14735.1*	13887.1	13915.8	13644.1	13915.8
	7	1940.6	15032.4*	13981.9	14058.6	13777.6	14058.6
	8	1471.24	15036.7*	14257.5	14282.9	14012.1	14282.9
	9	1496.52	15572.6*	14736.3	14805.8	14535	14805.8
	10	1729	14804.3*	14008.2	14012.9	13840.4	14012.9
Average		1880.61	15046.33	14247.1	14307.55	14060.49	14307.55

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
500	1	3519.5	23980.7*	22254	22283.3	21904	22283.3
	2	3070.52	23778*	22183.3	22303.8	21842.4	22303.8
	3	3237.42	23566.6*	22053.7	22101.9	21633.4	22101.9
	4	3265.48	23862.7*	22327.5	22401.8	21922.7	22401.8
	5	2638.78	23558.3*	22007.6	22016	21666.4	22016
	6	3208.83	22920.4*	21643.2	21759.1	21166.9	21759.1
	7	3135.7	23333.4*	21851.2	21885.3	21403.8	21885.3
	8	2612.17	23036.1*	21833.5	21838.1	21353.2	21838.1
	9	2608.5	24276.5*	23163.6	23207	22818.9	23207
	10	2873.9	23906.8*	22100.5	22110.9	21687	22110.9
Average		3017.08	23621.95	22141.81	22190.72	21739.87	22190.72
1000	1	5690.22	36053.2*	33717.9	33717.9	32889	33717.9
	2	5124.63	36564.6*	33633.2	33698.3	32912	33698.3
	3	5554.98	35884.5*	33303.4	33358.5	32486.5	33358.5
	4	5347.57	37687.9*	33439.2	33485.8	32694.8	33485.8
	5	4713.05	36464.1*	33229	33229	32407.5	33229
	6	5370.18	35725.8*	33125.1	33175.4	32172.4	33175.4
	7	5292.38	35505*	32900.1	32949.3	32021.7	32949.3
	8	4747.85	36628.2*	33184.6	33184.6	32223.9	33184.6
	9	4665.38	38888.7*	34988.5	34988.5	34165.4	34988.5
	10	4996.74	36380.9*	33011.3	33011.3	32208.3	33011.3
Average		5150.30	36578.29	33453.23	33479.86	32618.15	33479.86

* indicates solution is not optimal-allocated computation time exceeded 600s.

D.7 High Return ($n = 25$)

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
125	1	2312.49	4301.81	4154.2	4301.81	4154.2	4301.81
	2	2360.86	4087.5	3877.45	4087.5	3870.05	4087.5
	3	2861.92	4546.31	4386.94	4546.31	4367.3	4546.31
	4	1675.61	3787.63	3573.56	3787.63	3573.56	3787.63
	5	1866.07	3874.43	3668.92	3874.43	3613.65	3874.43
	6	5158.93	6412.11	6229.98	6412.11	6138.89	6412.11
	7	1181.54	3577.65	3310.31	3577.65	3310.31	3577.65
	8	2811.91	4468.45	4257.92	4468.45	4139.38	4468.45
	9	3153.98	4764.2	4644.19	4764.2	4604.53	4764.2
	10	3578.91	5136.82	4894.3	5136.82	4894.3	5136.82
Average		2696.22	4495.69	4299.78	4495.69	4266.62	4495.69
250	1	2643.76	6171.18	6002.86	6171.18	5910.54	6171.18
	2	2693.36	5957.93	5647.94	5957.93	5601.68	5957.93
	3	3217.78	6313.43	6150.86	6313.43	6099.82	6313.43
	4	2124.98	5656.42	5513.7	5656.42	5477.32	5656.42
	5	2259.52	5770.48	5547.05	5770.48	5427.15	5770.48
	6	5429.99	7912.11	7572.1	7912.11	7357.71	7912.11
	7	1617.24	5511.57	5373.11	5511.57	5349.72	5511.57
	8	3134.05	6115.89	5880.31	6115.89	5643.24	6115.89
	9	3547.59	6549.22	6269.61	6549.22	6185.37	6549.22
	10	3896	6769.41	6459.88	6769.41	6435.71	6769.41
Average		3056.43	6272.76	6041.74	6272.76	5948.83	6272.76

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
500	1	3306.28	8852	8779.8	8852	8530.18	8852
	2	3358.35	9025.94	8581.65	9025.94	8365.6	9025.94
	3	3857.98	9348.27	9035.36	9348.27	9035.36	9348.27
	4	3012.63	8413.64	8271.14	8413.64	8182.2	8413.64
	5	3044.44	8842.73	8601.6	8842.73	8382.44	8842.73
	6	5972.12	10912.10	10158.9	10912.10	9794.56	10912.10
	7	2340.17	8511.57	8316.32	8511.57	8145.96	8511.57
	8	3778.33	9007.90	8681.44	9007.90	8207.31	9007.90
	9	4334.81	9063.88	8896.58	9063.88	8750.15	9063.88
	10	4530.18	9584.76	9255.93	9584.76	8947.5	9584.76
Average		3753.53	9156.28	8857.87	9156.28	8634.13	9156.28
1000	1	4631.34	13440	13229.5	13440	12523.9	13440
	2	4688.32	13557.5	13179	13557.5	12570.7	13557.5
	3	5137.67	13648.1	13323.3	13648.1	12664.7	13648.1
	4	4599.63	12903	12837.6	12903	12379.7	12903
	5	4559.02	13042.3	12736	13042.3	12143.5	13042.3
	6	7056.36	15014.5	14319.7	15014.5	14003.6	15014.5
	7	3746.73	12513.9	12509.2	12513.9	11988.8	12513.9
	8	5066.88	13153.2	12794.3	13153.2	11921.3	13153.2
	9	5866.04	12897.1	12731	12897.1	12111	12897.1
	10	5798.55	13610.2	13315.8	13610.2	12529.7	13610.2
Average		5115.05	13377.98	13097.54	13377.98	12483.69	13377.98

D.8 High Return $(n = 50)$

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
125	1	10577.8	13151.4	12802.3	13151.4	12729.2	13151.4
	2	6952.03	10057.4*	9559.41	10011.7	9531.73	10011.7
	3	8196.99	10735.7*	10255	10735.7	10201.6	10735.7
	4	3475.91	7722.1*	7434.01	7702.1	7419.61	7702.1
	5	11880.1	14102*	13759	14102	13759	14102
	6	2470.71	7225.82*	6865.65	7110.02	6865.65	7110.02
	7	5584.89	9354.59*	9054.19	9285.71	9029.27	9285.71
	8	1613.56	6712.76*	6502.2	6655.84	6437.95	6655.84
	9	6107.42	9476.23	9080.91	9476.23	9023.32	9476.23
	10	3247.05	7678.81*	7262.17	7461.51	7246.78	7461.51
Average		6010.65	9621.68	9257.48	9569.22	9224.41	9569.22
250	1	10879.2	15924.3	15301.7	15924.3	15090.2	15924.3
	2	7409.73	12926.6*	12384.3	12885.9	12276	12885.9
	3	8600.62	13610.7*	12711	13532.5	12604.3	13532.5
	4	3941.83	11611.3*	11047	11356.9	11047	11356.9
	5	12408.2	16837.2*	16093	16805.1	16093	16805.1
	6	2989.98	11599*	11013.3	11197.9	10902.8	11197.9
	7	6026.01	13309.3*	12647.8	13062.6	12499.5	13062.6
	8	2150.96	11113.6*	10574	10813.3	10403.4	10813.3
	9	6535.24	13006.2*	12304.3	13006.2	12162.7	13006.2
	10	3700.95	11538.2*	11157.6	11366.4	11012.4	11366.4
Average		6464.27	13147.64	12523.4	12995.11	12409.13	12995.11

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
500	1	11482	21423.8*	19883.8	21131.2	19537.5	21131.2
	2	8289.69	18441.7*	17307.4	17750.4	16993.7	17750.4
	3	9407.89	18750.3*	17543.8	18677.7	17330.3	18677.7
	4	4841.27	17543.1*	16210.4	16508.2	15931.5	16508.2
	5	13346.4	21549*	20381.9	21438.9	20381.9	21438.9
	6	4028.53	17938.9*	17120.1	17470.9	16817	17470.9
	7	6899	19570.7*	18483.2	18957.9	18110.7	18957.9
	8	3103.61	16687.5*	16208.1	16286.2	15822.3	16286.2
	9	7390.89	19061.3*	17711.8	18250.7	17259.7	18250.7
	10	4608.74	17693.7*	17025.8	17299.5	16687.5	17299.5
Average		7339.80	18866	17787.63	18377.16	17487.21	18377.16
1000	1	12687.6	28568.1*	27357.1	28131.2	26575.1	28131.2
	2	10017.2	27281.4*	25163.7	25644.4	24429.6	25644.4
	3	11022.4	27255.9*	25794.9	26711.7	25356.3	26711.7
	4	6593.32	25450.6*	24373.4	24697.2	23661.8	24697.2
	5	15033.9	30376.7*	28223	29359.4	27456.9	29359.4
	6	6049.11	27111.2*	25700.4	25917.8	25190.2	25917.8
	7	8641.13	28208.5*	26776.9	26987.1	26051.7	26987.1
	8	4892.94	26088.3*	24845.7	24904.3	23981.3	24904.3
	9	9102.2	28587.7*	26299	27161.2	25447.5	27161.2
	10	6413.04	26926.8*	25498.3	25827.6	24813.8	25827.6
Average		9045.28	27585.52	26003.24	26534.19	25296.42	26534.19

* indicates solution is not optimal-allocated computation time exceeded 600s.

D.9 High Return ($n = 75$)

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
125	1	5180.05	11875.8*	11098.7	11586.2	11098.7	11586.2
	2	18573.8	23015.1*	22186.6	22559.1	22186.6	22559.1
	3	10919.4	16347.3*	15550.3	15919.3	15484	15919.3
	4	4686.71	11634.7*	10848.1	11344.4	10848.1	11344.4
	5	21187.8	24401.8*	23596.9	24151.8	23592.5	24151.8
	6	4588.37	11667*	10926.5	11604.6	10925.4	11604.6
	7	12063.4	16716*	16161.9	16614.6	16148.8	16614.6
	8	3515.13	10856.1*	10293.8	10578.6	10285	10578.6
	9	4744.27	11840.2*	10889.9	11505	10889.9	11505
	10	9712.04	14745.9*	14206.7	14605.7	14148.6	14605.7
Average		9517.10	15309.99	14575.94	15046.93	14560.76	15046.93
250	1	5745.25	18547.8*	16938	17694.5	16831.8	17694.5
	2	19038.5	27153.4*	25964.7	26529.4	25964.7	26529.4
	3	11339.3	21518.6*	19946.1	20663.6	19753.6	20663.6
	4	5290.15	17769.4*	16419.8	16879.9	16363.5	16879.9
	5	21592.5	27651.8*	26314.8	27401.8	26294.7	27401.8
	6	5133.43	18138.3*	16507.4	17086.8	16492	17086.8
	7	12471.7	21938.3*	20530.1	21466.6	20394	21466.6
	8	4076.5	17606.8*	16155.7	16457.7	16022.9	16457.7
	9	5301.61	18281.9*	16624.7	17258.3	16572.1	17258.3
	10	10295.1	20329.3*	18908.3	19571.7	18785	19571.7
Average		10028.40	20893.56	19430.96	20101.03	19347.43	20101.03

* indicates solution is not optimal-allocated computation time exceeded 600s.

SC	No	Original		$(\ell, S) - like = FL = MC = SP$		(ℓ, S, WW)	
		LB	UB	LB	UB	LB	UB
500	1	6859.69	28185.9*	25750.9	26549.4	25467.1	26549.4
	2	19968	34798*	32416.6	33679.9	32369.8	33679.9
	3	12179.3	30338.1*	27374.9	28413.7	27001.2	28413.7
	4	6431.11	27099.1*	25098.2	25771.3	24884.8	25771.3
	5	22401.9	34901.8*	31750.5	33809.5	31679.6	33809.5
	6	6223.54	27192.5*	25261	25831.6	25007.5	25831.6
	7	13288.3	31204.1*	28397.2	30420.9*	28123.7	30420.9*
	8	5142.69	27106.1*	25252.6	25593.6	24879.7	25593.6
	9	6390.03	27343.9*	25500.1	26043.9	25249.4	26043.9
	10	11382.4	29452.1*	26907.8	28356.9	26791.4	28356.9
Average		11026.70	29762.16	27370.98	28447.07	27145.42	28447.07
1000	1	8956.85	42407.9*	38455.1	39346.6	37858.5	39346.6
	2	21770.2	47418.5*	43832.5	45422.4	43325.8	45422.4
	3	13859.2	43166.7*	39520.6	40804.1	38726.8	40804.1
	4	8488.13	41880.3*	37880	38365.8	37337.6	38365.8
	5	24020.7	46860.4*	42452.4	43992.6	41854.1	43992.6
	6	8403.77	41682.1*	38548.3	39021.4	38157.4	39021.4
	7	14921.4	44156*	40518.3	41987.9	39893.7	41987.9
	8	7115.93	42005.4*	38508.8	38902.5	37923.7	38902.5
	9	8537.04	42569.9*	38872.5	39328.7	38221.8	39328.7
	10	13428.6	43121.8*	39137.8	40168.3	38818.6	40168.3
Average		12950.18	43526.9	39772.63	40734.03	39211.8	40734.03

* indicates solution is not optimal-allocated computation time exceeded 600s.

Appendix E

Flow Cover Inequalities in Mosel

```
model 'Original Formulation for ELSRs with Added (l,s) and Flow Cover Inequalities'
uses 'mmsystem'
uses 'mmxprs'
uses 'mmive'

forward procedure getsolution(t: integer)
forward procedure flowcover_1(lambda: integer)
forward procedure flowcover_2(lambda: integer)
forward procedure flowcover_3(lambda: integer)

setrandseed(10)

declarations
 NT=3                          !number of time periods
 period=1..NT
 p_r:array(period) of real     !production cost for remanufactured products
 p_m:array(period) of real     !production cost for newly products
 k_r:array(period) of real     !setup cost for remanufacturing
 k_m:array(period) of real     !setup cost for manufacturing
 h_r:array(period) of real     !holding cost for used products
 h_s:array(period) of real     !holding cost for serviceable products
 x_r:array(period) of mpvar    !production amount of remanufactured product
 x_m:array(period) of mpvar    !production amount of manufactured product
 y_r:array(period) of mpvar    !setup variable for remanufacturing
 y_m:array(period) of mpvar    !setup variable for manufacturing
 I_r:array(period) of mpvar    !inventory variable for product returns
 I_s:array(period) of mpvar    !inventory variable for serviceable products
 return:array(period) of real !amount of used products returned
 demand:array(period) of real !amount of demand for serviceable products
 totdem:array(period) of real !total demand from period t until NT
 totret:array(period) of real !total return from period 1 until t
 bigm_r:array(period) of real !big M constraint for remanufacturing
 bigm_m:array(period) of real !big M constraint for manufacturing

 solopt_x_r:array(period) of real !optimal solution for each variable
 solopt_x_m:array(period) of real
 solopt_y_r:array(period) of real
 solopt_y_m:array(period) of real
 solopt_I_r:array(period) of real
 solopt_I_s:array(period) of real
 optval:real                       !optimal solution for objective function
 linrelaxval:real                  !linear relaxation for objective function

 !get problem status
 status:array({XPRS_OPT,XPRS_UNF,XPRS_INF,XPRS_UNB,XPRS_OTH}) of string
```

```
 count1,count2,count3,count4,count5,count6: integer

 !(l,s) inequalities
 maxiter=100     !maximum number of iterations
 iter=1..maxiter
 !set S (1 if t in S, 0 otherwise) for each iteration + period l
 setS:array(iter, period, period) of integer

 ret:array(period, period) of real !total return from period t until period l
 dem:array(period, period) of real !total demand from period t until period l
 countviol_1:integer               !counter for number of violations of constraints
 countviol_2:integer
 countviol_3:integer
 countviol_4:integer
end-declarations

setparam('XPRS_CPUTIME',1)
setparam('XPRS_PRESOLVE',1)

!================================================================================
!DATA INPUT
!================================================================================

forall(t in period)return(t):=5+4*round(2.5*random)
forall(t in period)demand(t):=10+20*round(2.5*random)
forall(t in period)h_r(t):=0.5+round(1.5*random)
forall(t in period)h_s(t):=0.5+round(1.5*random)
forall(t in period)p_r(t):=0
forall(t in period)p_m(t):=0
forall(t in period)k_r(t):=50
forall(t in period)k_m(t):=500

!================================================================================
!GET PROBLEM STATUS
!================================================================================

status::([XPRS_OPT,XPRS_UNF,XPRS_INF,XPRS_UNB,XPRS_OTH])
['Optimum found','Unfinished','Infeasible','Unbounded','Failed']
count1:=0
count2:=0
count3:=0
count4:=0
count5:=0
count6:=0

!================================================================================
!PARAMETERS CALCULATION
!================================================================================

!calculate the bigM-constraints
forall(t in period)do
 totdem(t):=sum(tt in t..NT)demand(tt)
 totret(t):=sum(tt in 1..t)return(tt)
end-do

forall(t in period)do
 bigm_r(t):=minlist(totret(t),totdem(t))
 bigm_m(t):=totdem(t)
end-do

!================================================================================
!CONSTRAINTS
!================================================================================
```

```
!total cost function
costpro := sum(t in period)(p_r(t)*x_r(t) + p_m(t)*x_m(t))
costfixed := sum (t in period)(k_r(t)*y_r(t) + k_m(t)*y_m(t))
costinv := sum(t in period)(h_r(t)*I_r(t) + h_s(t)*I_s(t))
cost := costpro + costfixed + costinv

!flow balance for remanufactured and manufactured products
forall(t in period)do
 dem_sat_r(t):= if(t>1, I_r(t-1), 0)- x_r(t) + return(t) = I_r(t)
 dem_sat_s(t):= if(t>1, I_s(t-1), 0)+ x_m(t) + x_r(t)- demand(t) = I_s(t)
end-do

!production variable-binary variable relations
forall(t in period)do
 vub_r(t):=x_r(t)<=bigm_r(t)*y_r(t)
 vub_m(t):=x_m(t)<=bigm_m(t)*y_m(t)
end-do

!==========================================================================
!IP SOLUTION OF THE MODEL
!==========================================================================

fopen('output-1.txt',F_OUTPUT)
writeln('Period:',NT)
writeln('Setup cost-remanufacturing:',k_r(1))
writeln('Setup cost-manufacturing:',k_m(1))
writeln('')
forall(t in period)writeln('Demands:', demand(t))
forall(t in period)writeln('Returns:', return(t))
D:=sum(t in period)demand(t)
writeln('Total demands:', D)
R:=sum(t in period)return(t)
writeln('Total returns:', R)
writeln('')

forall (t in period)do
 y_r(t) is_binary
 y_m(t) is_binary
end-do

!solve IP
minimize(cost)
optval:=getobjval

forall(t in period)do
 solopt_x_r(t):=getsol(x_r(t))
 solopt_x_m(t):=getsol(x_m(t))
 solopt_y_r(t):=getsol(y_r(t))
 solopt_y_m(t):=getsol(y_m(t))
 solopt_I_r(t):=getsol(I_r(t))
 solopt_I_s(t):=getsol(I_s(t))
end-do

writeln(!=======================================================================)
writeln('IP SOLUTION')
writeln(!=======================================================================)
writeln('Total cost for IP:', getobjval)
writeln(' ')
writeln(!=======================================================================)
writeln('Optimal solutions for IP')
writeln(!=======================================================================)
forall(t in period)
writeln('x_r(',t,')=',getsol(x_r(t)),'','x_m(',t,')=',getsol(x_m(t)),
'','y_r(',t,')=',getsol(y_r(t)),'','y_m(',t,')=',getsol(y_m(t)),
```

```
'','I_r(',t,')=',getsol(I_r(t)),'','I_s(',t,')=',getsol(I_s(t)))
writeln(' ')

!==============================================================================
!LP RELAXATION SOLUTION OF THE MODEL
!==============================================================================
forall (t in period)do
 y_r(t)<=1
 y_m(t)<=1
end-do

!solve LP relaxation
minimize(XPRS_LIN,cost)
linrelaxval1:=getobjval

writeln(!=============================================================================)
writeln('LP RELAXATION SOLUTION')
writeln(!=============================================================================)
writeln('Total cost for LP relaxation: ', getobjval)
writeln(' ')
writeln(!=============================================================================)
writeln('Optimal solutions for LP relaxation')
writeln(!=============================================================================)
forall(t in period)
writeln('x_r(',t,')=',getsol(x_r(t)),'','x_m(',t,')=',getsol(x_m(t)),
'','y_r(',t,')=',getsol(y_r(t)),'','y_m(',t,')=',getsol(y_m(t)),
'','I_r(',t,')=',getsol(I_r(t)),'','I_s(',t,')=',getsol(I_s(t)))
writeln(' ')

!==============================================================================
!ADD (l,S)INEQUALITIES TO THE ORIGINAL FORMULATION
!==============================================================================
setparam('XPRS_COVERCUTS',0)
setparam('XPRS_GOMCUTS',0)
setparam('XPRS_CUTSTRATEGY',0)
setparam('XPRS_MAXTIME', -600)

starttime:=gettime

!calculate the returns and demands
forall(l in 1..NT)do
 forall(t in 1..l)do
  ret(t,l):= 0            !(1)set initial value of ret(t,l) and dem(t,l) as zero
  ret(l,l):= return(l)  !(2)calculate other ret(l,t) and dem (t,l) quantities
  dem(t,l):= 0
  dem(l,l):= demand(l)

  if(l>=2) then
   forall(tt in 1..(l-1))do
    ret(l-tt,l):= ret(l-tt+1,l) + return(l-tt)
    dem(l-tt,l):= dem(l-tt+1,l) + demand(l-tt)
   end-do
  end-if
 end-do
end-do

forall(l in period) do
  forall(k in 1..l) do
   !initialize the set S
    forall(t in k..l)
     setS(iteration,t,l):=0

   forall(t in k..l)do
    if(getsol(x_r(t))>ret(k,t)*getsol(y_r(t))
```

```
    or getsol(x_r(t))>dem(t,l)*getsol(y_r(t))
    or getsol(x_m(t))>dem(t,l)*getsol(y_m(t)))then
     setS(iteration,t,l):=1
    end-if
   end-do

   if(sum(u in k..l)setS(iteration,u,l)*(getsol(x_m(u))+getsol(x_r(u)))>
   getsol(I_s(l))+ sum(u in k..l)setS(iteration,u,l)*dem(u,l)
   *(getsol(y_m(u))+getsol(y_r(u)))+0.00001) then
    addcons_1(iteration, l):=sum(u in k..l)setS(iteration,u,l)*(x_m(u)+x_r(u))
    <= I_s(l)+ sum(u in k..l)setS(iteration,u,l)*dem(u,l)*(y_m(u)+y_r(u))
    countviol_1:= countviol_1 + 1
   end-if

   if(sum(u in k..l)setS(iteration,u,l)*getsol(x_m(u))>
   getsol(I_s(l))+ sum(u in k..l)setS(iteration,u,l)
   *dem(u,l)*getsol(y_m(u))+0.00001) then
    addcons_2(iteration, l):=sum(u in k..l)setS(iteration,u,l)*x_m(u)<= I_s(l)
    + sum(u in k..l)setS(iteration,u,l)*dem(u,l)*(y_m(u))
    countviol_2:= countviol_2 + 1
   end-if

   if(sum(u in k..l)setS(iteration,u,l)*getsol(x_r(u))>
   getsol(I_s(l))+ sum(u in k..l)setS(iteration,u,l)
   *dem(u,l)*getsol(y_r(u))+0.00001)then
    addcons_3(iteration, l):=sum(u in k..l)setS(iteration,u,l)*x_r(u)<= I_s(l)
    + sum(u in k..l)setS(iteration,u,l)*dem(u,l)*(y_r(u))
    countviol_3:= countviol_3 + 1
   end-if

   if(sum(u in k..l)setS(iteration,u,l)*getsol(x_r(u))> if(k>1,getsol(I_r(k-1)),0)
   + sum(u in k..l)setS(iteration,u,l)*ret(k,u)*getsol(y_r(u))+0.00001)then
    addcons_4(iteration,l):=sum(u in k..l)setS(iteration,u,l)*x_r(u)
    <= if(k>1,I_r(k-1),0)+ sum(u in k..l)setS(iteration,u,l)*ret(k,u)*(y_r(u))
    countviol_4:= countviol_4 + 1
   end-if
  end-do
 end-do

 if(countviol_1=0 and countviol_2=0 and countviol_3=0 and countviol_4=0) then break
 end-if
    !solve the strengthened LP relaxation with added maximum violated (l,s)cuts
    minimize(XPRS_LIN,cost)
    linrelaxval2:=getobjval
end-do

!================================================================================
!LP RELAXATION SOLUTION WITH ADDED (L,S) INEQUALITIES
!================================================================================

forall(t in period) linx_r(t):=getsol(x_r(t))
forall(t in period) linx_m(t):=getsol(x_m(t))
forall(t in period) liny_r(t):=getsol(y_r(t))
forall(t in period) liny_m(t):=getsol(y_m(t))
forall(t in period) linI_r(t):=getsol(I_r(t))
forall(t in period) linI_s(t):=getsol(I_s(t))

writeln(!=========================================================================)
writeln('LP RELAXATION SOLUTION WITH ADDED (L,S) INEQUALITIES')
writeln(!=========================================================================)
writeln('Total cost for LP Relaxation with added (l,s) Inequalities: ', getobjval)
writeln('')
writeln(!=========================================================================)
writeln('Optimal Solutions for LP relaxation with added (l,s) Inequalities')
```

```
writeln(!========================================================================)
forall(t in period)
writeln('x_r(',t,')=',linx_r(t),'','x_m(',t,')=',linx_m(t),'','y_r(',t,')=',liny_r(t),
'','y_m(',t,')=',liny_m(t),'','I_r(',t,')=',linI_r(t),'','I_s(',t,')=',linI_s(t))
writeln('')
writeln('Initial integrality gap: ',((optval-linrelaxval1)/optval)*100)
writeln('Integrality gap after adding (l,S) cuts: ',((optval-getobjval)/optval)*100)
writeln('Closed gap by (l,S) cuts: ',(getobjval-linrelaxval1)
/(optval-linrelaxval1)*100)
c_time:=gettime-starttime writeln('') fclose(F_OUTPUT)

if (((optval-getobjval)/optval)*100=0) then
 writeln(" !!! STOP !!! ")
 exit(1)
end-if

!==========================================================================
!LP RELAXATION WITH ADDED (L,S) AND FLOW COVER INEQUALITIES
!==========================================================================

declarations
 linearx_r:array(period) of real !linear relaxation solutions for each variable
 linearx_m:array(period) of real
 lineary_r:array(period) of real
 lineary_m:array(period) of real
 maxbigm_r:real                 !taking the maximum value of bigm on t
 maxbigm_m:real
 maxbigm_c:real
end-declarations

maxbigm_r:=0
maxbigm_m:=0

forall(t in period)do
 if(bigm_r(t)>=maxbigm_r)then
  maxbigm_r:=bigm_r(t)
 end-if
end-do

forall(t in period)do
 if(bigm_m(t)>=maxbigm_m)then
  maxbigm_m:=bigm_m(t)
 end-if
end-do

maxbigm_c:=maxlist(maxbigm_r, maxbigm_m)

forall(t in period) getsolution(t)

fopen("output-2.txt",F_OUTPUT)
writeln('Period:',NT)
writeln('Setup cost-remanufacturing:',k_r(1))
writeln('Setup cost-manufacturing:',k_m(1))
writeln('')

starttime:=gettime
forall(lambda in 1..round(maxbigm_r))
 flowcover_1(lambda)

writeln(!========================================================================)
writeln('(1)Flow cover inequalities (<=)')
writeln(!========================================================================)
writeln('Number of flow cover inequalities 1 added(<=): ',count1)
writeln('Number of extended flow cover inequalities 1 added (<=): ',count2)
```

```
writeln('')
fc1_time:=gettime-starttime

starttime:=gettime
forall(lambda in 1..round(maxbigm_m))
 flowcover_2(lambda)

writeln(!=====================================================================)
writeln('(2)Flow cover inequalities (>=)')
writeln(!=====================================================================)
writeln('Number of flow cover inequalities 2 added (>=): ',count3)
writeln('Number of extended flow cover inequalities 2 added (>=): ',count4)
writeln('')
fc2_time:=gettime-starttime

starttime:=gettime
forall(lambda in 1..round(maxbigm_c))
 flowcover_3(lambda)

writeln(!=====================================================================)
writeln('(3)Flow cover inequalities (>=)')
writeln(!=====================================================================)
writeln('Number of flow cover inequalities 3 added (>=): ',count5)
writeln('Number of extended flow cover inequalities 3 added (>=): ',count6)
writeln('')
fc3_time:=gettime-starttime

minimize(XPRS_LIN,cost)

writeln(!=====================================================================)
writeln('LP RELAXATION WITH ADDED (L,S) INEQUALITIES AND ALL FLOW COVER INEQUALITIES')
writeln(!=====================================================================)
writeln('Total cost for LP relaxation with added (l,s)
and all flow cover inequalities: ',getobjval)
writeln('')
writeln('Integrality gap after adding all cuts: ',((optval-getobjval)/optval)*100)
writeln('Closed gap by all cuts: ',(getobjval-linrelaxval2)/(optval-linrelaxval2)*100)
a_time:=c_time + fc1_time + fc2_time + fc3_time
fc_time:=fc1_time + fc2_time + fc3_time
writeln('Time spent by (l,S):', c_time)
writeln('Time spent by FC:',    fc_time)
writeln('Time spent by All:',   a_time)
writeln('') fclose(F_OUTPUT)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!PROCEDURES!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!=====================================================================
!PROCEDURE: GET SOLUTION
!=====================================================================

procedure getsolution(t: integer)
 linearx_r(t):=getsol(x_r(t))
 linearx_m(t):=getsol(x_m(t))
 lineary_r(t):=getsol(y_r(t))
 lineary_m(t):=getsol(y_m(t))
end-procedure

!=====================================================================
!PROCEDURE: FLOW COVER INEQUALITIES
!=====================================================================

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!PROCEDURE 1!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

procedure flowcover_1(lambda:integer)
```

```
declarations
 w_r:array(period) of mpvar
 s_r:set of integer
 l_r:set of integer
 maxi_r:real
end-declarations

s_r:={}
l_r:={}

forall(t in period)w_r(t) is_binary

!Objective function
obj_r:=sum(t in period)(linearx_r(t)+(maxlist(bigm_r(t)-lambda,0)
*(1-lineary_r(t))))*w_r(t)

!Constraint
c_r:=sum(t in period)bigm_r(t)*w_r(t)=sum(t in period)return(t)+lambda

maximize(obj_r)
c_r:=0

if (status(getprobstat)='Optimum found')then
 forall(t in period)do
  if(round(getsol(w_r(t)))=1)then
   s_r+={t}
  end-if
 end-do

 if(s_r<>{}) and (getobjval>sum(t in period)return(t))then
  sum(t in s_r)x_r(t)-sum(t in s_r)maxlist(bigm_r(t)-lambda,0)*y_r(t)<=
  sum(t in period)return(t)-sum(t in s_r)maxlist(bigm_r(t)-lambda,0)
  writeln('The flow cover inequality 1 is added')
  count1:=count1+1
  writeln('Lambda=',lambda,', S_r=',s_r)
  if (sum(t in s_r)solopt_x_r(t)-sum(t in s_r)maxlist(bigm_r(t)-lambda,0)
  *solopt_y_r(t)> sum(t in period)return(t)-
  sum(t in s_r)maxlist(bigm_r(t)-lambda,0)) then
   writeln('---->Cuts off the optimal solution')
  end-if
 end-if

 if (s_r<>{}) then
 maxi_r:=0
  forall (t in s_r) do
   if (bigm_r(t)>=maxi_r) then
    maxi_r:=bigm_r(t)
   end-if
  end-do

 forall(t in period) do
  if (not(t in s_r)) and (linearx_r(t)
  -(maxlist(bigm_r(t),maxi_r)-lambda)*lineary_r(t)>0) then
   l_r+={t}
  end-if
 end-do

   if (maxi_r>=lambda) and (l_r<>{}) and
   (getobjval+sum(t in l_r)(linearx_r(t)-(maxlist(bigm_r(t),maxi_r)-lambda)*lineary_r(t))>
   sum(t in period)return(t)) then
    sum(t in s_r)x_r(t)+sum(t in l_r)x_r(t)
    -sum(t in s_r)(maxlist(bigm_r(t)-lambda,0)*y_r(t))
    -sum(t in l_r)((maxlist(bigm_r(t),maxi_r)-lambda)*y_r(t))
    <=sum(t in period)return(t)-sum(t in s_r)maxlist(bigm_r(t)-lambda,0)
```

```
     writeln('The extended flow cover inequality 1 is added')
     count2:=count2+1
     writeln('Lambda=',lambda,', S_r=',s_r,',  L_r=',l_r)
    if (sum(t in s_r)solopt_x_r(t)+sum(t in l_r)solopt_x_r(t)
    -sum(t in s_r)(maxlist(bigm_r(t)-lambda,0)*solopt_y_r(t))
    -sum(t in l_r)((maxlist(bigm_r(t),maxi_r)-lambda)*solopt_y_r(t))
    >sum(t in period)return(t)-sum(t in s_r)maxlist(bigm_r(t)-lambda,0)) then
     writeln('---->Cuts off the optimal solution')
    end-if
  end-if
 end-if
end-if

end-procedure

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!PROCEDURE 2!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

procedure flowcover_2(lambda:integer)

declarations
 u_m:array(period) of mpvar
 s_m:set of integer
 l_m:set of integer
 maxi_2:real
end-declarations

s_m:={}
l_m:={}

forall(t in period) u_m(t) is_binary

!Objective function
obj_m:=sum(t in period)(linearx_m(t)+(maxlist(bigm_m(t)-lambda,0)
*(1-lineary_m(t))))*u_m(t)

!Constraint
c_m:=sum(t in period)bigm_m(t)*u_m(t)=sum(t in period)(demand(t)-return(t))+lambda

maximize(obj_m)
c_m:=0

if (status(getprobstat)='Optimum found')then
 forall(t in period)do
  if(round(getsol(u_m(t)))=1)then
   s_m+={t}
  end-if
 end-do

 if(s_m<>{}) and (getobjval>sum(t in period)linearx_m(t))then
  sum(t in s_m)x_m(t)-sum(t in s_m)maxlist(bigm_m(t)-lambda,0)*y_m(t)
  +sum(t in s_m)maxlist(bigm_m(t)-lambda,0)<=sum(t in period)x_m(t)
  writeln('The flow cover inequality 2 is added')
  count3:=count3+1
  writeln('Lambda=',lambda,', S_m=',s_m)
  if (sum(t in s_m)solopt_x_m(t)-sum(t in s_m)maxlist(bigm_m(t)-lambda,0)*solopt_y_m(t)
  +sum(t in s_m)maxlist(bigm_m(t)-lambda,0)>sum(t in period)solopt_x_m(t)) then
   writeln('---->Cuts off the optimal solution')
  end-if
 end-if

 if (s_m<>{}) then
 maxi_2:=0
 forall (t in s_m) do
  if (bigm_m(t)>=maxi_2) then
```

```
   maxi_2:=bigm_m(t)
  end-if
 end-do

 forall(t in period) do
  if (not(t in s_m)) and
  (linearx_m(t)-(maxlist(maxi_2,bigm_m(t))-lambda)*lineary_m(t)>0) then
   l_m+={t}
  end-if
 end-do

 if(maxi_2>=lambda)and (l_m<>{}) and (getobjval+sum(t in l_m)(linearx_m(t)
 -(maxlist(maxi_2,bigm_m(t))-lambda)*lineary_m(t))>sum(t in period)linearx_m(t)) then
  sum(t in s_m)x_m(t)-sum(t in s_m)maxlist(bigm_m(t)-lambda,0)*y_m(t)
  +sum(t in s_m)maxlist(bigm_m(t)-lambda,0)
  +sum(t in l_m)((x_m(t))-(maxlist(maxi_2,bigm_m(t))-lambda)*y_m(t))
  <=sum(t in period)x_m(t)
   writeln('The extended flow cover inequality 2 is added')
   count4:=count4+1
   writeln('Lambda=',lambda,', S_m=',s_m,',  L_m=',l_m)
   if (sum(t in s_m)solopt_x_m(t)-sum(t in s_m)maxlist(bigm_m(t)-lambda,0)*solopt_y_m(t)
   +sum(t in s_m)maxlist(bigm_m(t)-lambda,0)+sum(t in l_m)((solopt_x_m(t))
   -(maxlist(maxi_2,bigm_m(t))-lambda)*solopt_y_m(t))
   >sum(t in period)solopt_x_m(t)) then
    writeln('---->Cuts off the optimal solution')
   end-if
  end-if
 end-if
end-if

end-procedure

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!PROCEDURE 3!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

procedure flowcover_3(lambda:integer)

declarations
 u_r:array(period) of mpvar
 u_m:array(period) of mpvar
 s_r:set of integer
 s_m:set of integer
 l_r:set of integer
 l_m:set of integer
 maxi_3:real
 maxi_4:real
end-declarations

s_r:={}
s_m:={}
l_r:={}
l_m:={}

forall(t in period)do
 u_r(t) is_binary
 u_m(t) is_binary
end-do

!Objective function
obj_rm1:=sum(t in period)(linearx_r(t)+(maxlist(bigm_r(t)-lambda,0)
*(1-lineary_r(t))))*u_r(t)
+sum(t in period)(linearx_m(t)+(maxlist(bigm_m(t)-lambda,0)*(1-lineary_m(t))))*u_m(t)

!Constraint
c_rm1:=sum(t in period)bigm_r(t)*u_r(t)+sum(t in period)bigm_m(t)*u_m(t)
```

```
=sum(t in period)demand(t)+lambda
forall(t in period)do
u_r(t)+u_m(t)=1
end-do

maximize(obj_rm1)
c_rm1:=0

if (status(getprobstat)='Optimum found')then
 forall(t in period)do
  if(round(getsol(u_r(t)))=1)then
   s_r+={t}
  end-if
 end-do

 forall(t in period)do
  if(round(getsol(u_m(t)))=1)then
   s_m+={t}
  end-if
 end-do

 if(s_r<>{}) and (s_m<>{}) and
 (getobjval>sum(t in period)(linearx_r(t)+linearx_m(t)))then
  sum(t in s_r)x_r(t)-sum(t in s_r)maxlist(bigm_r(t)-lambda,0)*y_r(t)
  +sum(t in s_r)maxlist(bigm_r(t)-lambda,0)
  +sum(t in s_m)x_m(t)-sum(t in s_m)maxlist(bigm_m(t)-lambda,0)*y_m(t)
  +sum(t in s_m)maxlist(bigm_m(t)-lambda,0)<=sum(t in period)(x_r(t)+x_m(t))
   writeln('The flow cover inequality 3 is added')
   count5:=count5+1
   writeln('Lambda=',lambda,', S_r=',s_r,', S_m=',s_m)
   if (sum(t in s_r)solopt_x_r(t)-sum(t in s_r)maxlist(bigm_r(t)-lambda,0)*solopt_y_r(t)
   +sum(t in s_r)maxlist(bigm_r(t)-lambda,0)
   +sum(t in s_m)solopt_x_m(t)-sum(t in s_m)maxlist(bigm_m(t)-lambda,0)*solopt_y_m(t)
   +sum(t in s_m)maxlist(bigm_m(t)-lambda,0)
   >sum(t in period)(solopt_x_r(t)+solopt_x_m(t))) then
    writeln('---->Cuts off the optimal solution')
   end-if
 end-if

 if (s_r<>{}) and (s_m<>{}) then
 maxi_3:=0
 maxi_4:=0

  forall (t in s_r) do
   if (bigm_r(t)>=maxi_3) then
    maxi_3:=bigm_r(t)
   end-if
  end-do

  forall (t in s_m) do
   if (bigm_m(t)>=maxi_4) then
    maxi_4:=bigm_m(t)
   end-if
  end-do

 forall(t in period) do
  if (not(t in s_r)) and
  (linearx_r(t)-(maxlist(maxi_3,maxi_4,bigm_r(t))-lambda)*lineary_r(t)>0) then
   l_r+={t}
  end-if
 end-do

 forall(t in period) do
  if (not(t in s_m)) and
```

```
 (linearx_m(t)-(maxlist(maxi_3,maxi_4,bigm_m(t))-lambda)*lineary_m(t)>0) then
   l_m+={t}
  end-if
 end-do

 if(maxi_3>=lambda) and (maxi_4>=lambda) and (l_r<>{}) and (l_m<>{}) and
 (getobjval+sum(t in l_r)(linearx_r(t)-(maxlist(maxi_3,maxi_4,bigm_r(t))-lambda)
 *lineary_r(t))+sum(t in l_m)(linearx_m(t)-(maxlist(maxi_3,maxi_4,bigm_m(t))-lambda)
 *lineary_m(t))>sum(t in period)(linearx_r(t)+linearx_m(t)))then
  sum(t in s_r)x_r(t)-sum(t in s_r)maxlist(bigm_r(t)-lambda,0)*y_r(t)
  +sum(t in s_r)maxlist(bigm_r(t)-lambda,0)
  +sum(t in l_r)(x_r(t)-(maxlist(maxi_3,maxi_4,bigm_r(t))-lambda)*y_r(t))
  +sum(t in s_m)x_m(t)-sum(t in s_m)maxlist(bigm_m(t)-lambda,0)*y_m(t)
  +sum(t in s_m)maxlist(bigm_m(t)-lambda,0)
  +sum(t in l_m)(x_m(t)-(maxlist(maxi_3,maxi_4,bigm_m(t))-lambda)*y_m(t))
  <=sum(t in period)(x_r(t)+x_m(t))
   writeln('The extended flow cover inequality 3 is added')
   count6:=count6+1
   writeln('Lambda=',lambda,', S_r=',s_r,', L_r=',l_r,', S_m=',s_m,', L_m=',l_m)
   if (sum(t in s_r)solopt_x_r(t)-sum(t in s_r)maxlist(bigm_r(t)-lambda,0)
    *solopt_y_r(t)+sum(t in s_r)maxlist(bigm_r(t)-lambda,0)
    +sum(t in l_r)(solopt_x_r(t)-(maxlist(maxi_3,maxi_4,bigm_r(t))-lambda)*solopt_y_r(t))
    +sum(t in s_m)solopt_x_m(t)-sum(t in s_m)maxlist(bigm_m(t)-lambda,0)*solopt_y_m(t)
    +sum(t in s_m)maxlist(bigm_m(t)-lambda,0)
    +sum(t in l_m)(solopt_x_m(t)-(maxlist(maxi_3,maxi_4,bigm_m(t))-lambda)*solopt_y_m(t))
    >sum(t in period)(solopt_x_r(t)+solopt_x_m(t))) then
    writeln('---->Cuts off the optimal solution')
   end-if
  end-if
 end-if
end-if
end-procedure
end-model

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!END MODEL!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```