CLOSED-LOOP SUPPLY CHAIN NETWORK DESIGN: CASE OF DURABLE PRODUCTS

Mohammad Jeihoonian

A thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy

Concordia University

Montréal, Québec, Canada

September 2016

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mr. Mohammad Jeihoonian**

Entitled: **Closed-loop supply chain network design: case of durable products**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Industrial Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

Dr. Wei-Ping Zhu Chair

Dr. Jean-Pierre Kenné External Examiner

Dr. Satyaveer S. Chauhan Examiner

Dr. Mingyuan Chen Examiner

Dr. Ivan Contreras Examiner

Dr. Masoumeh Kazemi Zanjani Supervisor

Dr. Michel Gendreau Co-supervisor

Approved _____

Chair of Department or Graduate Program Director

September 2016 _____

Dr. ,Dean

Faculty of Engineering and Computer Science

Abstract

Closed-loop supply chain network design: case of durable products

Mohammad Jeihoonian, Ph.D.

Concordia University, 2016

Closed loop supply chains comprise, in addition to the conventional forward flows from suppliers to end-users, a reverse flow of products, components, and materials from end-users to the manufacturers and secondary markets. Designing a closed-loop supply chain is a strategic level planning which considerably impacts on tactical and operational performance of the supply chain. It refers to the decisions taken on the location of facilities involved in the supply chain network along with the management of the physical flows associated with forward and product recovery channels. Our problem of interest is mainly motivated by the case of durable products including but not limited to large household appliances, computers, photocopying equipment, and aircraft engines. Such category of products has a modular structure, composed of independent components. As opposed to simple structured products, e.g., printer cartridges, that can only be recycled, each of the components in the reverse bill of materials of durable products can be recovered by a particular type of recovery process. Besides, durable products share a long life cycle characteristic which indeed makes designing their CLSC networks more complicated.

In this thesis, in keeping with the abovementioned motivation, we focus on designing closed-loop and reverse supply chains in the context of durable products that are of various quality conditions. The recovery decisions for product return include remanufacturing, part harvesting, bulk recycling, material recycling, and landfilling/incineration. Moreover, we take into account environmental concerns regarding

the harmful impacts of used products in the closed-loop supply chain planning. As the closed-loop supply chains typically encounter uncertainty in quality and quantity of the profitable return stream, we further aim to consider the impact of uncertainty in designing the recovery network. For such purposes, in the first phase, we address a closed-loop supply chain planning problem in the context of durable products with generic modular structures. The problem is formulated as a mixed-integer programming model which is then solved by an accelerated Benders decomposition-based algorithm. The performance of the proposed decomposition approach is enhanced through incorporating algorithmic features including valid inequalities, non-dominated optimality cuts, and local branching strategies.

Next, in the second phase, we propose a precise approach to model the uncertain quality status of returns, in which the availability of each component in the reverse bill of materials is modeled as discrete scenarios. We propose a two-stage stochastic programming model to address this problem setting. Then, since the cardinality of the scenario set grows exponentially with the number of involved components, we detail on a scenario reduction scheme to alleviate the computational burden of the proposed model. The stochastic problem is solved by a L-shaped algorithm enhanced through valid inequalities and Pareto-optimal cuts.

Finally, we investigate designing a dynamic reverse supply chain where the quantity of the return flows is uncertain. We introduce a multi-stage stochastic programming model and develop a heuristic inspired by scenario clustering decomposition scheme as the solution method. It revolves around decomposing the scenario tree into smaller sub-trees which consequently yields a number of sub-models in accordance with sub-trees. The resulting sub-models are then coordinated by Lagrangian penalty terms. On account of the fact that each sub-model per se is a hard to solve problem, a Benders decomposition-based algorithm is proposed to solve sub-models.

Preface

This thesis has been prepared in "Manuscript-based" format under the co-direction of Dr. Masoumeh Kazemi Zanjani from the department of Mechanical and Industrial Engineering, Concordia University and Dr. Michel Gendreau from department of Mathematics and Industrial Engineering, Polytechnique Montréal. This research was financially supported by Le Fonds de recherche du Québec-Nature et technologies (FRQNT) and the Natural Sciences and Engineering Research Council of Canada (NSERC). All the articles presented in this thesis were co-authored and reviewed prior to submission for publication by Dr. Masoumeh Kazemi Zanjani and Dr. Michel Gendreau. The author of this thesis acted as the principal researcher and performed the mathematical models development, programming of the solution algorithms, analysis and validation of the results, along with writing the first drafts of the articles.

The first article entitled "Accelerating Benders decomposition for closed-loop supply chain network design: case of used durable products with different quality levels", co-authored by Dr. Masoumeh Kazemi Zanjani and Dr. Michel Gendreau was published in *European Journal of Operational Research* in June 2016.

The second article entitled "Closed-loop supply chain network design under uncertain quality status: case of durable products", co-authored by Dr. Masoumeh Kazemi Zanjani and Dr. Michel Gendreau was accepted for publication in *International Journal of Production Economics* in July 2016.

The third article entitled "A decomposition algorithm for dynamic reverse supply

chain network design under uncertainty", co-authored by Dr. Masoumeh Kazemi Zanjani and Dr. Michel Gendreau was submitted to *Computers & Operations Research* in August 2016.

To my beloved parents and siblings

Alireza, Alieh, Mozhgan, Mehdi, and Pari

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Dr. Masoumeh Kazemi Zanjani who has continuously supported my PhD study with her immense knowledge and invaluable advice. I shall remain indebted for her boundless enthusiasm, encouragement, and patience. Furthermore, I would like to thank my co-advisor Prof. Michel Gendreau for his insightful guidance and incisive comments throughout my thesis. It would not have been possible to complete my PhD without their invaluable advice over the past four years.

In addition, I am grateful to Arlene Zimmerman, Leslie Hosein, Maureen Thuringer, and Sophie Merineau, as the administrative staff of the department of Mechanical and Industrial Engineering. I am also grateful to my fellow lab-mates and great friends at Concordia University particularly, Omid Sanei, Mostafa Pazoki, Bahman Fathi, Mohammad Tohidi, Armaghan Alibeyg, Dua Weraikat, Ehsan Rezabeigi, and Alireza Zandi Karimi.

Last but not least, my special thanks go to my adored family for their understanding, worthwhile support, and unconditional love during my studies.

Contents

List of Figures

List of Tables

Abbreviations

BOM	Bill of materials
CLSC	Closed-loop supply chain
DRSP	Dual recourse subproblem
DSP	Dual subproblem
EOL	End-of-life
MIP	Mixed-integer programming
MP	Master problem
MS-MIP	Multi-stage mixed-integer stochastic programming
OEM	Original equipment manufacturer
PCB	Printed Circuit Board
PSP	Primal subproblem
RSC	Reverse supply chain
RSP	Recourse subproblem
3PL	Third-party logistics

Chapter 1

Introduction

1.1 Overview

Sustainability of supply chains have gained a significant momentum during recent decades. In many industries, conventionally, EOL products especially solid items have been disposed of through burying under the ground as well as incineration, which are not sustainable practices. For instance, it is estimated that incinerators emit 446 kg of mercury annually in Canada [1]. In order to alleviate the environmental footprint of products that reach the end of their life cycle, legislative efforts have been made by many governments around the globe requiring OEMs to take the responsibility for the whole life cycle of their products. For example, in electrical and electronic sector, EU Directive (Directive 2003/108/EC) sets recycling targets ranging from 50% to 75% by weight. In addition to the environmental crisis, many firms have exploited new business opportunities that correspond to the recovery of the economic value residing in EOL products. As a well-known OEM that actively remanufactures its used products, IBM saves up to 80% per part by dismantling returned equipment compared to sourcing a new part from a supplier [2]. Needless to say, such business advantages call for quantitative decision making tools that address the challenges of

the underlying supply chain.

RSCs process products returns so as to recover value by re-processing them via a broad set of activities, such as remanufacturing, part harvesting, material recycling, etc. These practices are considered as key features of an economically and environmentally sustainable supply chain. Indeed, OEMs that have been most successful with their RSCs are those that closely coordinate it with the forward supply chain, initiating the CLSC [3]. Unlike forward supply chains, in CLSCs, end-users are the point of origin and hold final goods that generate value-added flows through various recovery options. Designing CLSCs revolves around the decisions to be made on the location of facilities in the forward and reverse chains along with the routing and the coordination of forward and reverse physical flows. The former is associated with the location and the capacity of collection, remanufacturing, recycling, and disposal facilities in the reverse channel to carry out recovery activities while the OEM operates a well-established forward chain. The latter addresses allocating the physical flows among the CLSC facilities. Addressing each of the above decisions locally might be quite straightforward. However, it considerably increases unnecessary infrastructure and processing costs, disregards the interdependence between forward and reverse chains, and hence results in a sub-optimal supply chain planning.

Motivated by the case of durable products, e.g., large household appliances, that are characterized by their modular structure and their long life cycle, the focus of this thesis is on designing CLSC and RSC networks applicable in the context of durable goods. Designing a recovery network for such category of products is a complex problem particularly attributed to the various types of strategies that can be adopted for the value recovery of several components in their reverse BOM. On the other hand, CLSCs deal with high degree of variability and uncertainty in quality and quantity of returns, which further complicate the planning of a recovery system for modular

structured products. Such variations are most often the consequence of different usage patterns and the enormous number of end-users dispersed over geographic regions. Even though a rigorous stream of academic research has been dedicated to the study of CLSC/RSC network design problems in both deterministic and non-deterministic settings, to the best of our knowledge, the impact of heterogeneity and uncertainty in quality and quantity of durable products have never been investigated. This is the critical point which distinguishes the current thesis from previous studies. More precisely, we contribute to the existing literature through addressing the problem of designing a comprehensive recovery network for the case of durable products while accounting the quality status of returns in a deterministic setting. We further focus on cases in which the quality state of returns are unknown a priori yet historical data on the condition of used items are available. Finally, we aim for investigating a RSC planning problem under stochastic returns quantity evolving over a planning horizon.

In what follows, we proceed with delineating the problem studied. Then, the research scope and objectives are presented. We present the outline of this thesis at the end of this chapter.

1.2 Problem description

In the CLSC network design problem of interest, the OEM operates a well-established forward supply chain that comprises module, part, and raw material suppliers, manufacturers, distribution facilities, and end-users. Each module supplier provides a particular type of new modules. Similarly, each type of new parts and raw materials are provided by the corresponding supplier. New durable goods are shipped form manufacturing facilities to end-users through distribution centers to meet their demands. The OEM seeks to adopt some plausible recovery options associated with the reverse BOM of returned products to abide by the legislative recovery target as

well as to capture the economic values by processing the returns. Thus, the existing forward network is extended to accommodate new facilities in the reverse network and accordingly to coordinate the physical forward and reverse flows in the extended supply chain network. The reverse channel composes collection, disassembly, remanufacturing, bulk recycling, material recycling, and disposal centers, referred to as the recovery facilities. The returns with heterogeneous quality status are acquired in disassembly facilities after being expected and sorted in collection centers. Given the reverse BOM, in disassembly centers, each returned item is disassembled into different components, i.e., modules and parts, along with raw materials. The most suitable recovery option for modules in a good condition is remanufacturing; the common practice for parts is part-harvesting; and the attractive disposition decision for raw materials is recycling. Depending on its quality state, each used product also contains a considerable amount of raw materials combined with other residues, e.g., electronic scrap. Such mix of residues cannot be easily processed through simple operations carried out in raw material recycling facilities. Bulk recycling that includes some unit operations, e.g., crushing and separation, is the viable recovery option for residues. Bulk recycling step is then followed by raw material recycling to recycle unprocessed raw materials. The recovered components and raw materials are then shipped to manufacturing facilities and/or are sold in secondary markets. The conceptual structure of the CLSC under investigation is schematically illustrated in Figure 1. The solid and dashed arcs indicate the forward and reverse flows, respectively.

In many cases, forecasting the quality status of the return stream is quite impossible due to limited accuracy of grading and classification errors. In such cases, that crude information of the quality status is available to the decision maker, considering the impact of random quality state on the grading decision is essential in order to mitigate uncertainty. This can be done through modeling the random quality status of

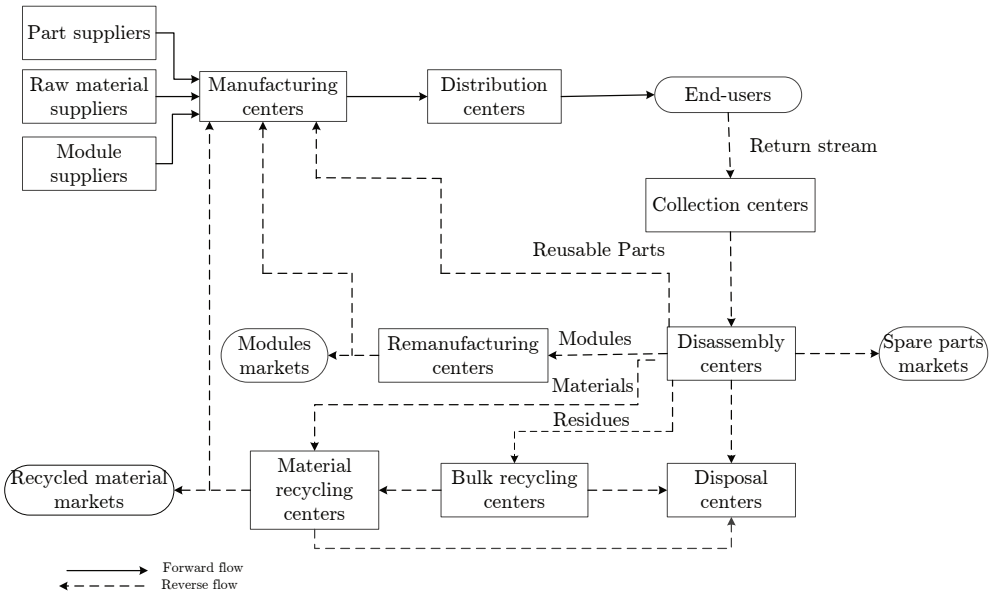

Figure 1: The CLSC network

used durable products by a probability distribution with finite support. In the supply chain described above, once the grading process is executed in disassembly centers, a complete information on the number of remanufacturable modules and reusable parts in addition to the amount of residues is available to the decision maker. Therefore, the CLSC network design problem can be seen as a two-stage decision making problem in which a set of decisions, e.g., the location of the recovery facilities, are taken before the realization of the quality status and the remaining decisions, e.g., reverse flows, are made after grading the return stream. The disassembled components, raw materials, and residues are shipped to appropriate recovery facilities. Similar to the first problem setting, the recovered components and raw materials are then shipped to manufacturing centers and/or are sold in secondary markets.

Given the uncertainty in quantity of returns over a planning horizon, the static setting described earlier, is extended into a multi-period setting where the stochastic return flows are of non-homogeneous quality status. In the dynamic setting, in each

period in the planning horizon, used products are acquired in disassembly centers such that the disassembled modules, parts, and residues are further processed through viable recovery practices. Moreover, studying this problem in a dynamic setting provides the possibility of closing some of the existing facilities or opening new ones over the planning horizon depending on the quantity of the return stream. This problem setting is restricted to RSC activities that entail taking back and recovery of used durable products. The processed items are sold in secondary markets.

1.3 Scope and objectives

To fill the void in the existing literature, the core objective of this study is to design a comprehensive CLSC/RSC that will aid in developing of efficient used product recovery systems in the context of durable products. Given the problem description, the specific objectives of this thesis are summarized as follows.

1. To take into account the reverse BOM of durable products and most of plausible recovery options that an OEM can adopt in practice,

2. To formulate the deterministic CLSC as a mathematical programming model to determine the location of the recovery facilities and to coordinate forward and reverse flows while accounting multiple quality levels and the legislative recovery target,

3. To develop a competitive decomposition algorithm to solve the deterministic mathematical model,

4. To explicitly incorporate the uncertain quality status of the return flows into the proposed mathematical model by means of two-stage stochastic programming approach,

5. To develop an efficient algorithm as the solution method for the stochastic CLSC planning problem,

6. To design a dynamic RSC network under uncertainty in quantity of returns through proposing a multi-stage stochastic programming model,

7. To develop a scenario clustering decomposition scheme that obtains high quality solutions for the resulting stochastic model in the previous objective,

8. To investigate the tractability of the proposed models and the performances of the solution methods based on a real-life industrial case.

1.4 Organization of the thesis

This manuscript has five chapters organized in the following sequence. Chapter 2 addresses the fundamental problem of this study that is designing a CLSC network, where the quality status of used products are categorized in multiple levels. The decisions to be made are the location of the recovery facilities in the reverse chain and the routing and the coordination of physical flows among the CLSC network entities. To this end, the problem is formulated as a MIP model in which the objective is to maximize the profit. In order to solve the proposed MIP model in a reasonable amount of time, an accelerated Benders decomposition-based algorithm is developed. Furthermore, we shed light on the impact of the legislative recovery target on the performance of the CLSC. In Chapter 3, through modeling uncertainty in the quality status of used durable products as binary scenarios, the deterministic problem is extended into a stochastic CLSC. This problem is reformulated as a two-stage mixed-integer stochastic program to maximize the expected profit. Given the large number of the quality state scenarios, a scenario reduction scheme adapted to the particular structure of the uncertainty set is applied to eliminate the most doubtful to occur

scenarios. The proposed model is then solved via an enhanced L-shaped decomposition algorithm. Moreover, some insight on the analysis of the scenario reduction scheme is provided. Chapter 4 presents a multi-stage mixed-integer stochastic programming model to address a dynamic RSC network design problem. The objective is to maximize the expected net profit over the entire planning horizon. The stochastic factor, i.e, quantity of returns, is modeled as a scenario tree allowing the adjustment of the decisions while more information on the uncertain parameter is available to the decision maker. The large-scale optimization problem is then solved by a heuristic scenario clustering decomposition approach. It employs a Benders decomposition-based algorithm as a suitable solution approach for each scenario cluster sub-model. Finally, Section 5 summarizes concluding remarks in addition to several avenues for future work.

Chapter 2

Accelerating Benders decomposition for closed-loop supply chain network design: case of used durable products with different quality levels

This was published as "Accelerating Benders decomposition for closed-loop supply chain network design: case of used durable products with different quality levels", *European Journal of Operational Research*, 251(3), 830-845, 2016.

Abstract

Durable products are characterized by their modular structured design as well as their long life cycle. Each class of components involved in the multi-indenture structure of such products requires a different recovery process. Moreover, due to their long life cycle, the return flows are of various quality levels. In this article, we study a closed-loop supply chain in the context of durable products with generic modular structures. To this end, we propose a mixed-integer programming model based on a generic disassembly tree where the number of each sub-assembly depends on the quality status of the return stream. The model determines the location of various types of facilities in the reverse network while coordinating forward and reverse flows. We also consider the legislative target for the recovery of used products as a constraint in the problem formulation. We present a Benders decomposition-based solution algorithm together with several algorithmic enhancements for this problem. Computational results illustrate the superior performance of the solution method.

2.1 Introduction

Landfilling of EOL durable products that contain large quantities of precious and depletable raw materials is a major concern in terms of sustainability and environmental footprints. In recent decades, OEMs in several countries, such as Germany and Japan, have been facing with legislations on the take-back of their EOL products. Meanwhile, they have started recognizing the product recovery as an opportunity for saving production costs through reusing the recovered parts in their forward flow in addition to having access to the secondary markets. Hence, the OEMs have been forced to extend the scope of traditional logistics to incorporate the return flows from customers to manufacturer. As pointed out by Guide and Van Wassenhove [3], OEMs

that have been most successful with their reverse supply chains are those that closely coordinate it with the forward supply chain, initiating the CLSC. In a CLSC, the role of the RSC is to collect used products from end-users, inspect and sort them as needed, ship them to various recovery facilities, and finally redistribute the recovered items into the forward supply chain or to the secondary markets.

This study is motivated by the recovery of durable products, such as aircraft, automobile, and large household appliances that are distinguished by their multi-indenture structure as well as their long life cycle. Such products can be disassembled into several components namely modules, parts, and precious raw materials. As opposed to simple waste, e.g., paper, carpet, and sand, that can only be recycled, each of the aforementioned components in the disassembly tree of durable products can be recovered by a particular recovery process. In the context of durable products with long life cycle, it can be expected that the majority of the return stream is composed of poor quality returns with a small number of recoverable modules and parts [4]. In other words, only a small portion of the return flows might belong to the warranty or damaged items involving a large number of high quality modules and parts. Since the remanufacturing cost increases as the quality of returns decrease, OEMs expect larger revenue through the recovery of high quality returns and thus might be less motivated for the acquisition and recovery of lower quality ones. However, the legislation, e.g., in Europe and Japan, sets targets for the recovery of used products. Failure to meet this target would incur penalties to the OEM and it has a negative impact on the image of the company from customers' point of view.

The existing CLSC network design models in the literature cover only a few recovery options, such as product remanufacturing and material recycling. In an attempt to fill the gap in the current literature on the design of CLSC networks, this paper proposes a MIP model which formulates CLSC network design for the case of used

durable products based on a generic disassembly tree. In addition to the location of collection, disassembly, and disposal facilities, the model will also decide on the location of a variety of recovery facilities such as remanufacturing, bulk recycling, and material recycling designated for each class of components of durable products.

Moreover, the variable quality status of the return stream has been incorporated in the structure of the aforementioned disassembly tree. More specifically, we assume that the return stream fits into various quality levels; warranty or damaged returns are usually categorized as the high quality stream, while EOL items are assigned to the poor quality stream. To the best of our knowledge, the issue of non-homogeneous quality of the components in the disassembly tree of complex structured products which affects the right choice of the recovery option has never been addressed in the CLSC design literature. As another contribution, we have also considered the legislative target imposed to OEMs for the recovery of returns in order to address environmental concerns regarding the harmful effects of leaving used products in the environment.

On account of the fact that the proposed MIP model is among the most large-scale CLSC network design models particularly due to various types of recovery facilities in the reverse network as well as the comprehensive generic disassembly tree, we have developed an efficient solution method based on Benders decomposition [5]. Regarding that the previous studies in the context of CLSC and RSC design have addressed less complicated networks, their Benders decomposition algorithmic schemes are limited to the generation of multiple optimality cuts, cut strengthening, and introducing the trust region restriction to the master problem [6, 7]. We, however, deploy a variety of enhancements including: 1) Adding valid inequalities to the master problem to reduce the number of feasibility cuts; 2) Generating enhanced Pareto-optimal cuts to exclude a larger space of the master problem; and, 3) Specializing the local branching

search to the proposed CLSC design model to concurrently improve both lower and upper bounds during the execution of Benders decomposition. Local branching as a neighborhood search algorithm offers a general search strategy that might be adapted to the type of problem being solved. It should be emphasized that we have adapted the local branching scheme described in the seminal work of Fischetti and Lodi [8] to the type of our problem of interest in the case of Benders decomposition approach to improve the convergence of the algorithm. To the best of our knowledge, this is the first attempt to combine the enhanced Pareto-optimality cuts of Papadakos [9] with the local branching within Benders decomposition algorithm.

In summary, the main contribution of our study is twofold. Firstly, we formulate a CLSC network design problem based on a generic disassembly tree corresponding to durable products with non-homogeneous quality status. To the best of our knowledge, the proposed model is the most generic CLSC strategic planning model in the sense that it involves all recovery options plausible in taking different sub-assemblies of a product while also considering legislative recovery target as an environmental goal. Secondly, we propose an exact solution algorithm based on Benders decomposition in conjunction with several computational enhancements such as improved Pareto-optimality cuts and local branching.

The remainder of this article is organized as follows. In the next section, we provide the overview of the most relevant literature. In Section 2.3, we provide more details on the problem investigated in this paper and then present its formulation. Section 2.4 describes the solution methodology. Computational experiments are presented in Section 2.5. Conclusion and future areas of research are provided in Section 2.6.

2.2 Literature review

In this section, we present a selective overview of the relevant literature on CLSC and RSC network design. For a detailed review, the interested reader is referred to [10–12]. We also provide a review of the relevant existing algorithmic refinements for Benders decomposition method.

2.2.1 CLSC and RSC network design models

As an early study, Fleischmann et al. [13] proposed a MIP model for designing a generic CLSC network including uncapacitated disassembly and remanufacturing facilities in the reverse channel. Krikke et al. [4] proposed a multi-objective CLSC network design in which the objective is to minimize cost and environmental impacts measured by energy and waste. Lu and Bostel [14] provided a MIP formulation to address a CLSC where the customers are directly served from hybrid manufacturing/remanufacturing facilities. For solving the problem, the authors developed a Lagrangian heuristic approach.

The multi-product CLSCs have been studied by [6, 15–18]. For instance, Min and Ko [17] focused on the design of a CLSC network in which a 3PL provider runs the reverse channel. The resulting model was solved by means of a genetic algorithm. More recently, Alumur et al. [19] presented a multi-product formulation for RSC network design while considering the reverse BOM. The proposed model was applied for a case study of washing machines and tumble dryers in Germany.

Listeş [20] proposed a two-stage stochastic programming model for designing a CLSC network under demand and return uncertainty. The proposed model was solved by the integer L-shaped algorithm. Salema et al. [21] addressed the problem of designing a CLSC network in which random demand and quantity of returns are

modeled as discrete scenarios with known probability distribution. For more recent papers on CLSC network design under uncertainty, the reader is referred to [22, 23].

Quality status of used products has been considered by Aras et al. [24]. The authors addressed a RSC network design problem in which used products are characterized with respect to different quality levels. A tabu search-simplex search method was developed as the solution approach. Likewise, a quality-dependent incentive policy for the collection of used items was presented in Aras and Aksen [25].

The classification of the contributions mentioned above is provided in Table 1. Columns 2 to 7 refer to location decisions in the CLSC and RSC network under investigation. Columns 4 to 7, in particular, indicate the type(s) of recovery options that have been considered in those articles. Column 8 verifies whether or not the recovery network design model is formulated based on a disassembly tree (reverse bill-of-material). Columns 9 and 10 indicate, respectively, if the variable quality status of the return stream and the environmental concern have been taken into consideration while designing the network. The last column represents the solution approach adopted. The overview of the existing literature reveals the extent to which the model we propose in this article goes beyond the literature. Only a few models address the multi-indenture structure of durable products and hence formulate the CLSC/RSC design problem based on a disassembly tree. Furthermore, none of those articles considered all disposition processes plausible for various types of dismantled components in such products as well as the recovery target as an environmental goal. Likewise, variable quality of the return stream and consequently its impact on the remanufacturing cost has been rarely investigated in the CLSC/RSC network design problem. On the methodological side, compared to Benders decomposition-based approaches applied in previous studies, the complex structure of the CLSC considered in this article requires developing a more sophisticated method to solve the resulting

Table 1: The most pertinent CLSC/RSC network design models in the literature

Article	Location Decisions						Disassembly tree	Quality status	Environmental criteria	Solution approach
	Collection	Disassembly	Recovery options							
			Rem.	BRec.	Rec.	Disp.				
[4]	✓	✓	✓		✓	✓	✓		✓	Commercial solver
[6]	✓		✓							EBD
[7]	✓								✓	EBD
[13]		✓	✓							Commercial Solver
[14]	✓		✓							Lagrangean heuristic
[15]	✓		✓							Commercial Solver
[16]	✓	✓								Commercial Solver
[17]	✓									Genetic algorithm
[18]	✓		✓			✓				Commercial Solver
[19]		✓	✓		✓		✓			Commercial solver
[20]	✓		✓							Integer L-shaped
[21]		✓	✓							Commercial solver
[22]	✓		✓							Commercial solver
[23]	✓	✓			✓	✓			✓	Commercial solver
[24]	✓							✓		Tabu serach
[25]	✓							✓		Tabu search
Our model	✓	✓	✓	✓	✓	✓	✓	✓	✓	BDLB

Rem.: Remanufacturing, BRec.: Bulk recycling, Rec.: Material recycling, Disp.: Disposal, EBD: Enhanced Benders decomposition, BDLB: BD with local branching and Pareto-optimality cut

MIP model.

2.2.2 Enhancing the performance of Benders decomposition

Benders decomposition is an exact solution method in which the variables of a MIP model are partitioned into two subsets such that when the integer variables are assigned numerical values, the problem reduces to a linear program. This procedure partitions the original mixed-integer programming model into a master problem and a linear subproblem referred to as the slave problem. These subproblems are then solved sequentially and iteratively until a termination criterion, such as a small gap, is satisfied.

Benders decomposition-based approaches have been widely used in solving supply chain network design problems: e.g., (i) Multi-commodity distribution system design [26], (ii) Stochastic supply chain design [27, 28], and (iii) Closed-loop supply chains [6, 7]. Over the years, since the pioneering paper of Geoffrion and Graves [26] in which the authors demonstrated that the master problem does not need to be solved to optimality, various techniques have been proposed to enhance the performance of Benders decomposition. The focus of the research has been mainly on circumventing the computational difficulty of the master problem, developing new cut selection schemes, and incorporating additional algorithmic features into the classical Benders algorithm.

McDaniel and Devine [29] suggested the relaxation of the integrality constraints in the master problem to obtain a set of initial optimality cuts and later on reintroduce them to the master problem and generate additional cuts until an optimal integer solution is found. As for cut selection schemes, Magnanti and Wong [30] discussed how to generate Pareto-optimal cuts considering the notion of core points when there are multiple optimal solutions to the dual subproblem. Despite the fact that more than

one dual subproblem is solved at each iteration, the proposed scheme proved to be quite efficient for solving network design type problems. Nonetheless, this approach suffers from two major drawbacks: 1) The normalization constraint added to the auxiliary dual subproblem is numerically unstable [31]; and, 2) The dual subproblem is required to be bounded. In other words, this approach cannot be employed to enrich feasibility cuts. In order to resolve the first issue, Papadakos [9] demonstrated that, through a different core point choice at each iteration, the normalization constraint can be disregarded in the auxiliary dual subproblem. Moreover, the author showed that the convex combination of the current master problem solution and the previous core point suffices to obtain a new core point. Considering the fact that the dual subproblem can be transformed into a pure feasibility problem, Fischetti et al. [32] addressed both feasibility and optimality cuts in an unified framework. The aim is to identify the "minimal source of infeasibility" and consequently to find the small set of constraints that would be sufficient to exclude the master problem solution. More recently, Sherali and Lunday [33] proposed the idea of generating maximal non-dominated cuts to speed up the Benders algorithm. As an algorithmic add-on, Rei et al. [34] recently proposed to apply local branching [8] throughout the solution process to improve both lower and upper bounds simultaneously and accordingly accelerate Benders decomposition algorithm.

2.3 Problem statement

2.3.1 Durable product structure

As pointed out in the preceding sections, durable products have a multi-indenture structure. They consist of multiple and various types of components as shown in the disassembly tree in Figure 11. Once the durable product is dismantled, it yields

various modules, parts, residues, solid raw materials, in addition to non-recoverable components. Modules, e.g., washing machine motor and clutch, are units of products that undergo the remanufacturing processes. In this study we assume that modules are brought up to the brand-new status through the remanufacturing processes. We also assume that poor quality modules can be recovered through bulk recycling. Parts, e.g., washing tube or PCB in a washing machine, are another category of components in the disassembly process. It is also assumed that each product yields different numbers of a specific part depending on its quality level. If parts are not qualified for harvesting, they would undergo the bulk recycling processes. Solid raw materials in the product, such as plastic, iron, and copper are separated after the product is dismantled. Such materials can directly undergo the appropriate recycling processes. Nevertheless, a big fraction of materials might be combined with other residues and thus it is not easy to extract them through simple activities in material recycling units. Bulk recycling is the appropriate recovery option for such residues. It encompasses shredding and different separation methods that first transform the residues into flakes and then separate different categories of materials based on their physical properties. Components with no value are salvaged (e.g., landfilling and incineration).

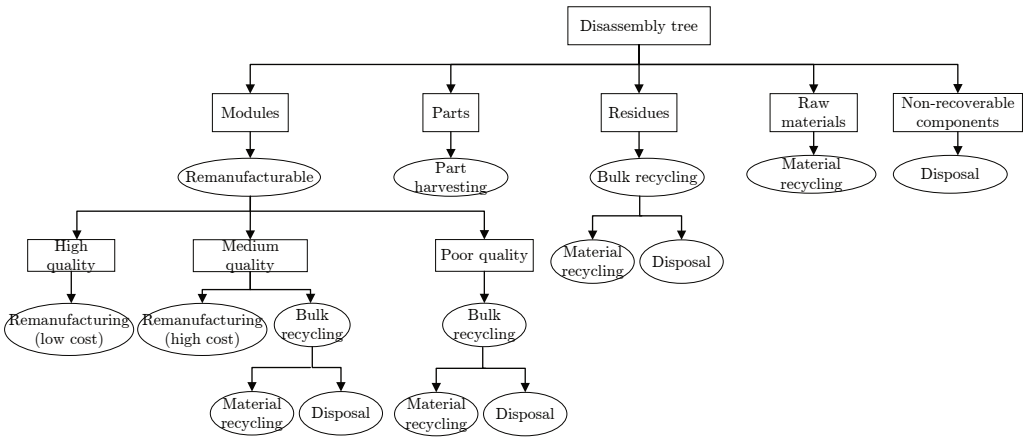

Figure 2: Disassembly tree of a generic durable product

2.3.2 CLSC network configuration

In the CLSC network under consideration, the OEM manages an established forward supply chain including suppliers, manufacturing facilities, distribution centers, and end-users. The brand-new durable products are shipped from manufacturing facilities to end-users through distribution centers to meet the demand. In the reverse chain, used products with different quality levels are acquired by collection centers. In disassembly centers, used products are dismantled considering the disassembly tree and consequently each component will be processed in the appropriate recovery facility. The recovered components can then be: (i) delivered to manufacturing facilities to deploy in manufacturing processes, and (ii) offered at secondary markets. Given the above description, the conceptual structure of the CLSC is schematically illustrated in Figure 12. The solid arcs indicate the forward flows while the dashed ones denote the reverse flows in the CLSC under consideration.

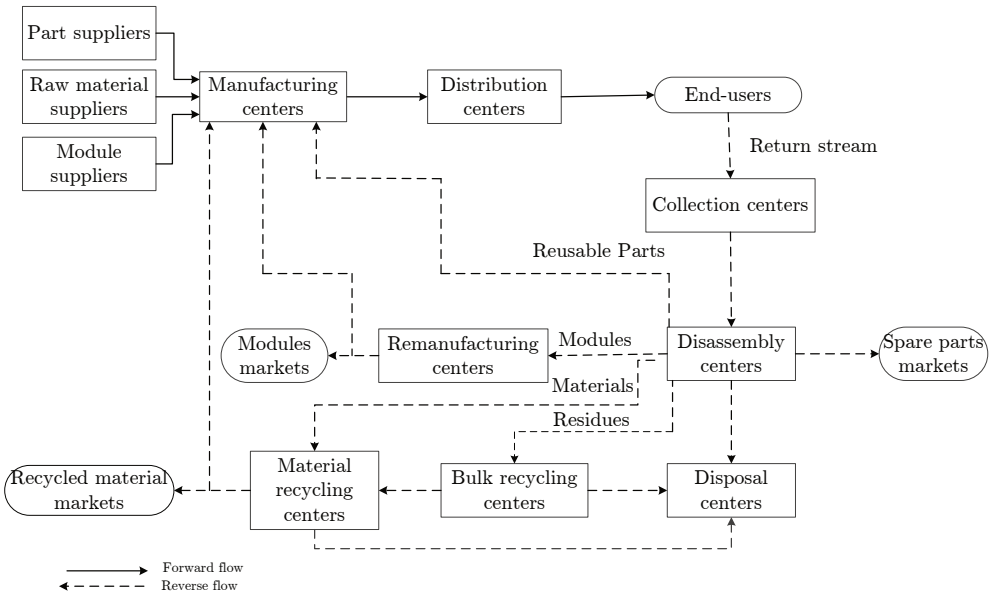

Figure 3: Conceptual framework for the CLSC network

2.3.3 Problem formulation

Given the network configuration provided in Section 3.2, the CLSC design model is looking for (i) the location of collection, disassembly, remanufacturing, bulk recycling, material recycling, and disposal facilities in the reverse network, and (ii) routing/coordinating the forward and reverse flows. The objective function of the proposed MIP model is to maximize the net profit which incorporates the revenue from selling new products and recovered components in addition to the fixed opening costs of facilities as well as processing procurement, and transportation costs in the network. The proposed mathematical model is subjected to the following major restrictions.

- Demand satisfaction constraints for brand-new products and recovered items,
- Flow balance constraints at different facilities,
- Capacity limitations of facilities,
- Non-negativity and binary restrictions on decision variables.

Furthermore, according to legislative regulations, the company is obliged to acquire a substantial portion of the return stream for recovery purposes. This condition reflects the environmental concerns regarding the harmful effects of leaving used durable products in the environment. It should be noted manufacturers might not be willing to invest on the recovery of low profitable poor quality returns due to their small salvage value. However, through imposing this condition as a constraint to the CLSC design model, we can ensure that poor quality returns are also treated in the reverse network. This constraint has been called "the environmentally friendly constraint" provided as constraint (13) in the following model.

The following assumptions support the MIP model provided in this section:

- The OEM operates a well-established forward network,

- Demands at end-users and secondary markets are known and must be fully fulfilled,
- The return fraction of each end-user zone is known,
- Returns are categorized with respect to various quality levels,
- The unit cost of collection, disassembly, and remanufacturing costs as well as the unit acquisition price of the return stream are quality-dependent,
- The model is considered in a single-product and single-period setting.

The problem notations including sets, parameters, and decision variables are listed in the *Appendix*.

Mathematical model

The objective function of the MIP model is to maximize the net profit defined as the difference between the total income and the total cost.

Total revenue

$$\sum_{j\in J}\sum_{k\in K} Pk_k QK_{jk} + \sum_{m\in M}\sum_{w\in W}\sum_{l\in L} Pw_l QW_{mwl} + \sum_{a\in A}\sum_{s\in S}\sum_{p\in P} Ps_p QS_{asp} + \sum_{g\in G}\sum_{e\in E}\sum_{r\in R} Pe_r QE_{ger} \quad (1)$$

Total cost

Fixed cost

$$\sum_{c\in C} fc_c YC_c + \sum_{a\in A} fa_a YA_a + \sum_{m\in M} fm_m YM_m + \sum_{g\in G} fg_g YG_g + \sum_{b\in B} fb_b YB_b + \sum_{d\in D} fd_d YD_d \quad (2)$$

Procurement cost

$$\sum_{z\in Z}\sum_{i\in I}\sum_{p\in P} cz_{zp}QI_{zip} + \sum_{u\in U}\sum_{i\in I}\sum_{r\in R} cu_{ur}NI_{uir}$$
$$+ \sum_{x\in X}\sum_{i\in I}\sum_{l\in L} cx_{xl}XI_{xil} + \sum_{c\in C}\sum_{a\in A}\sum_{q\in Q} Pr_{q}QA_{caq} \tag{3}$$

Processing cost

$$\sum_{i\in I}\sum_{j\in J} ci_{i}QJ_{ij} + \sum_{j\in J}\sum_{k\in K} cj_{j}QK_{jk} + \sum_{k\in K}\sum_{c\in C}\sum_{q\in Q} cc_{cq}QC_{kcq}$$
$$+ \sum_{c\in C}\sum_{a\in A}\sum_{q\in Q} ca_{aq}QA_{caq} + \sum_{a\in A}\sum_{g\in G}\sum_{r\in R} cg_{gr}QG_{agr} + \sum_{b\in B}\sum_{g\in G}\sum_{r\in R} cg_{gr}NG_{bgr}$$
$$+ \sum_{a\in A}\sum_{m\in M}\sum_{l\in L}\sum_{q\in Q} cm_{mlq}QM_{amlq} + \sum_{a\in A}\sum_{b\in B} cb_{b}QB_{ab} + \sum_{a\in A}\sum_{d\in D} cd_{d}QD_{ad}$$
$$+ \sum_{b\in B}\sum_{d\in D} cd_{d}ND_{bd} + \sum_{g\in G}\sum_{d\in D}\sum_{r\in R} cd_{d}XD_{gdr} \tag{4}$$

Transportation cost

$$\sum_{z\in Z}\sum_{i\in I}\sum_{p\in P} ti_{zip}QI_{zip} + \sum_{u\in U}\sum_{i\in I}\sum_{r\in R} ri_{uir}NI_{uir} + \sum_{x\in X}\sum_{i\in I}\sum_{l\in L} si_{xil}XI_{xil}$$
$$+ \sum_{i\in I}\sum_{j\in J} tj_{ij}QJ_{ij} + \sum_{j\in J}\sum_{k\in K} tk_{jk}QK_{jk} + \sum_{k\in K}\sum_{c\in C}\sum_{q\in Q} tc_{kcq}QC_{kcq}$$
$$+ \sum_{c\in C}\sum_{a\in A}\sum_{q\in Q} ta_{caq}QA_{caq} + \sum_{a\in A}\sum_{m\in M}\sum_{l\in L}\sum_{q\in Q} tm_{aml}QM_{amlq} + \sum_{a\in A}\sum_{b\in B} tb_{ab}QB_{ab}$$
$$+ \sum_{a\in A}\sum_{g\in G}\sum_{r\in R} tg_{agr}QG_{agr} + \sum_{a\in A}\sum_{d\in D} td_{ad}QD_{ad} + \sum_{b\in B}\sum_{g\in G}\sum_{r\in R} rg_{bgr}NG_{bgr}$$
$$+ \sum_{g\in G}\sum_{d\in D}\sum_{r\in R} sd_{gd}XD_{gdr} + \sum_{b\in B}\sum_{d\in D} rd_{bd}ND_{bd} + \sum_{a\in A}\sum_{s\in S}\sum_{p\in P} ts_{asp}QS_{asp}$$
$$+ \sum_{m\in M}\sum_{w\in W}\sum_{l\in L} tw_{mwl}QW_{mwl} + \sum_{g\in G}\sum_{e\in E}\sum_{r\in R} te_{ger}QE_{ger}$$

$$+\sum_{a\in A}\sum_{i\in I}\sum_{p\in P} tz_{aip}QZ_{aip}+\sum_{m\in M}\sum_{i\in I}\sum_{l\in L} tx_{mil}QX_{mil}+\sum_{g\in G}\sum_{i\in I}\sum_{r\in R} tu_{gir}QU_{gir} \quad (5)$$

Subject to:

Manufacturing centers

$$\sum_{z\in Z} QI_{zip}+\sum_{a\in A} QZ_{aip}=\phi_p\sum_{j\in J} QJ_{ij} \quad \forall\, i\in I, \forall\, p\in P \quad (6)$$

$$\sum_{u\in U} NI_{uir}+\sum_{g\in G} QU_{gir}=\mu_r\sum_{j\in J} QJ_{ij} \quad \forall\, i\in I, \forall\, r\in R \quad (7)$$

$$\sum_{x\in X} XI_{xil}+\sum_{m\in M} QX_{mil}=\omega_l\sum_{j\in J} QJ_{ij} \quad \forall\, i\in I, \forall\, l\in L \quad (8)$$

Distribution centers

$$\sum_{i\in I} QJ_{ij}=\sum_{k\in K} QK_{jk} \quad \forall\, j\in J \quad (9)$$

$$\sum_{j\in J} QK_{jk}=dk_k \quad \forall\, k\in K \quad (10)$$

Collection centers

$$\sum_{c\in C} QC_{kcq}=\psi_q dk_k \quad \forall\, k\in K, \forall\, q\in Q \quad (11)$$

$$\sum_{k\in K} QC_{kcq}\geq\sum_{a\in A} QA_{caq} \quad \forall\, c\in C, \forall\, q\in Q \quad (12)$$

$$\sum_{c\in C}\sum_{a\in A}\sum_{q\in Q} QA_{caq}\geq\theta\sum_{k\in K}\sum_{c\in C}\sum_{q\in Q} QC_{kcq} \quad (13)$$

Disassembly centers

$$\sum_{c\in C}\sum_{q\in Q}\gamma_{pq}QA_{caq}=\sum_{i\in I} QZ_{aip}+\sum_{s\in S} QS_{asp} \quad \forall\, a\in A, \forall\, p\in P \quad (14)$$

$$\sum_{a\in A} QS_{asp} = ds_{sp} \quad \forall\, s \in S, \forall\, p \in P \tag{15}$$

$$\sum_{c\in C}\sum_{q\in Q} \sigma_q QA_{caq} = \sum_{d\in D} QD_{ad} \quad \forall\, a \in A \tag{16}$$

$$\sum_{c\in C}\sum_{q\in Q} \beta_q QA_{caq} = \sum_{b\in B} QB_{ab} \quad \forall\, a \in A \tag{17}$$

$$\sum_{c\in C}\sum_{q\in Q} \alpha_{rq} QA_{caq} = \sum_{g\in G} QG_{agr} \quad \forall\, a \in A, \forall\, r \in R \tag{18}$$

$$\sum_{c\in C} \delta_{lq} QA_{caq} = \sum_{m\in M} QM_{amlq} \quad \forall\, a \in A, \forall\, l \in L, \forall\, q \in Q \tag{19}$$

Remanufacturing centers

$$\sum_{a\in A}\sum_{q\in Q} QM_{amlq} = \sum_{w\in W} QW_{mwl} + \sum_{i\in I} QX_{mil} \quad \forall\, m \in M, \forall\, l \in L \tag{20}$$

$$\sum_{m\in M} QW_{mwl} = dw_{wl} \quad \forall\, w \in W, \forall\, l \in L \tag{21}$$

Bulk recycling centers

$$\sum_{a\in A} \eta_r QB_{ab} = \sum_{g\in G} NG_{bgr} \quad \forall\, b \in B, \forall\, r \in R \tag{22}$$

$$\sum_{a\in A} QB_{ab} = \sum_{g\in G}\sum_{r\in R} NG_{bgr} + \sum_{d\in D} ND_{bd} \quad \forall\, b \in B \tag{23}$$

Material recycling centers

$$\sum_{a\in A} \tau_r QG_{agr} + \sum_{b\in B} \tau_r NG_{bgr} = \sum_{d\in D} XD_{gdr} \quad \forall\, g \in G, \forall\, r \in R \tag{24}$$

$$\sum_{g\in G} QE_{ger} = de_{er} \quad \forall\, e \in E, \forall\, r \in R \tag{25}$$

$$\sum_{a\in A} QG_{agr} + \sum_{b\in B} NG_{bgr} = \sum_{i\in I} QU_{gir} + \sum_{e\in E} QE_{ger}$$
$$+ \sum_{d\in D} XD_{gdr} \quad \forall\, g \in G, \forall\, r \in R \tag{26}$$

Capacity constraints

$$\sum_{i\in I} QI_{zip} \leq caz_{zp} \quad \forall\, z \in Z, \forall\, p \in P \tag{27}$$

$$\sum_{i\in I} NI_{uir} \leq cau_{ur} \quad \forall\, u \in U, \forall\, r \in R \tag{28}$$

$$\sum_{i\in I} XI_{xil} \leq cax_{xl} \quad \forall\, x \in X, \forall\, l \in L \tag{29}$$

$$\sum_{j\in J} QJ_{ij} \leq cai_{i} \quad \forall\, i \in I \tag{30}$$

$$\sum_{i\in I} QJ_{ij} \leq caj_{j} \quad \forall\, j \in J \tag{31}$$

$$\sum_{k\in K}\sum_{q\in Q} QC_{kcq} \leq cac_{c} YC_{c} \quad \forall\, c \in C \tag{32}$$

$$\sum_{c\in C}\sum_{q\in Q} QA_{caq} \leq caa_{a} YA_{a} \quad \forall\, a \in A \tag{33}$$

$$\sum_{a\in A} QD_{ad} + \sum_{b\in B} ND_{bd} + \sum_{g\in G}\sum_{r\in R} XD_{gdr} \leq cad_{d} YD_{d} \quad \forall\, d \in D \tag{34}$$

$$\sum_{a\in A} QB_{ab} \leq cab_{b} YB_{b} \quad \forall\, b \in B \tag{35}$$

$$\sum_{a\in A} QG_{agr} + \sum_{b\in B} NG_{bgr} \leq cag_{gr} YG_{g} \quad \forall\, g \in G, \forall\, r \in R \tag{36}$$

$$\sum_{a\in A}\sum_{q\in Q} QM_{amlq} \leq cam_{ml} YM_{m} \quad \forall\, m \in M, \forall\, l \in L \tag{37}$$

Constraints (6)-(8) ensure that the total outgoing flow from each manufacturing center is equal to the total incoming flow into this facility from suppliers and reverse network. Constraint (9)-(10) ensure flow balance at each distribution center as well as demand satisfaction at each end-user zone. Constraint (11) ensures that all the

returned products are collected at the collection centers. Constraint (12) ensures that the sum of the flow to the disassembly facilities, i.e., acquired returns, cannot exceed the total amount of returned products available in collection centers. Constraint (13) is the environmentally friendly restriction imposing that the total amount of acquired returns must be at least equal to a certain percentage of the total amount of the return stream in collection centers. Constraints (14)-(19) ensure flow conservation at each disassembly center and demand satisfaction at spare parts markets. Constrain (20) ensures that the total incoming flow to each remanufacturing center is equal to the total outgoing flow to modules secondary markets and manufacturing facilities. Constraint (21) ensures that the demands of all secondary markets for remanufactured modules are satisfied. Constraints (22)-(23) ensure flow conservation at each bulk recycling center. Constraints (24)-(26) are flow conservation restrictions at each material recycling center. Constraints (27)-(31) impose capacity restrictions on forward chain facilities. Constraints (32)-(37) ensure that the total incoming flow to an open facility in the reverse network cannot exceed its capacity.

2.4 Solution methodology

The proposed model (1)-(37) has a conspicuous special property that facilitates the application of Benders decomposition as the solution method [26]. For a given vector of locations of reverse chain facilities, the remaining problem is a network type problem which can be solved much easier than the MIP model. In what follows, we present the details of the Benders reformulation of the MIP model along with the proposed algorithmic enhancements.

2.4.1 Benders reformulation

For the sake of simplicity, let P be the vector of unit prices of selling brand-new and recovered components at the marketplaces. Let F be the vector of fixed costs of opening facilities in the reverse network. Furthermore, let C be the vector of other types of costs and let $\boldsymbol{QX}$ be the set of forward and reverse flows variables. Let $\boldsymbol{Y}$ be the set of binary decision variables representing, respectively, the locations of collection, disassembly, remanufacturing, bulk recycling, material recycling, and disposal centers. Furthermore, let $\overline{\boldsymbol{Y}}$ denote the vector of fixed $\boldsymbol{Y}$. The resulting PSP that determines the routing of the forward and reverse flows can be stated as follows.

$$\text{Max} \quad (1), (3)-(5)$$

$$\text{s.t.} \quad (6)-(31)$$

$$\sum_{k\in K}\sum_{q\in Q} QC_{kcq} \leq cac_c\overline{YC}_c \quad \forall\, c \in C \tag{38}$$

$$\sum_{c\in C}\sum_{q\in Q} QA_{caq} \leq caa_a\overline{YA}_a \quad \forall\, a \in A \tag{39}$$

$$\sum_{a\in A} QD_{ad} + \sum_{b\in B} ND_{bd} + \sum_{g\in G}\sum_{r\in R} XD_{gdr} \leq cad_d\overline{YD}_d \quad \forall\, d \in D \tag{40}$$

$$\sum_{a\in A} QB_{ab} \leq cab_b\overline{YB}_b \quad \forall\, b \in B \tag{41}$$

$$\sum_{a\in A} QG_{agr} + \sum_{b\in B} NG_{bgr} \leq cag_{gr}\overline{YG}_g \quad \forall\, g \in G, \forall\, r \in R \tag{42}$$

$$\sum_{a\in A}\sum_{q\in Q} QM_{amlq} \leq cam_{ml}\overline{YM}_m \quad \forall\, m \in M, \forall\, l \in L \tag{43}$$

Let $\boldsymbol{v^1}$,..., $\boldsymbol{v^{26}}$ and $\boldsymbol{v^{27}}$,..., $\boldsymbol{v^{32}}$ be the set of dual decision variables associated with constraint (6)-(31) and (38)-(43), respectively. The DSP can be formulated as follows.

$$\text{Min} \quad Z_{\boldsymbol{v}}(\overline{\boldsymbol{Y}}) = \sum_{k\in K} dk_k v_k^5 + \sum_{k\in K}\sum_{q\in Q} \psi_q dk_k v_{kq}^6 + \sum_{s\in S}\sum_{p\in P} ds_{sp} v_{sp}^{10}$$
$$+ \sum_{w\in W}\sum_{l\in L} dw_{wl} v_{wl}^{16} + \sum_{e\in E}\sum_{r\in R} de_{er} v_{er}^{20} + \sum_{z\in Z}\sum_{p\in P} caz_{zp} v_{zp}^{22}$$
$$+ \sum_{u\in U}\sum_{r\in R} cau_{ur} v_{ur}^{23} + \sum_{x\in X}\sum_{l\in L} cax_{xl} v_{xl}^{24} + \sum_{i\in I} cai_i v_i^{25} + \sum_{j\in J} caj_j v_j^{26}$$
$$+ \sum_{c\in C} cac_c \overline{YC}_c v_c^{27} + \sum_{a\in A} caa_a \overline{YA}_a v_a^{28} + \sum_{d\in D} cad_d \overline{YD}_d v_d^{29}$$
$$+ \sum_{b\in B} cab_b \overline{YB}_b v_b^{30} + \sum_{g\in G}\sum_{r\in R} cag_{gr} \overline{YG}_g v_{gr}^{31}$$
$$+ \sum_{m\in M}\sum_{l\in L} cam_{ml} \overline{YM}_m v_{ml}^{32} \tag{44}$$

$$\text{s.t.} \quad (\boldsymbol{v^1}, \boldsymbol{v^2}, ..., \boldsymbol{v^{32}}) \in \Delta \tag{45}$$

where the unrestricted dual variable vectors $\boldsymbol{v^5}$, $\boldsymbol{v^6}$, $\boldsymbol{v^{10}}$, $\boldsymbol{v^{16}}$, and $\boldsymbol{v^{20}}$ are, respectively, associated with constraints (10), (11), (15), (21), and (25). The non-negative dual variable vectors $\boldsymbol{v^{22}}$,..., $\boldsymbol{v^{26}}$ and $\boldsymbol{v^{27}}$,..., $\boldsymbol{v^{32}}$ are, respectively, associated with constraints (27)-(31) and (38)-(43). Besides, Δ indicates the polyhedron defined by the constraints of the DSP. If Δ is empty, the DSP is infeasible and according to duality theory in linear programming, the PSP is either infeasible or unbounded. However, the proposed MIP model is not unbounded. Let $\pi(.)$ represent the part of the dual subproblem objective function (44) which is independent of the location variables. Introducing an extra variable Γ, we can formulate the MP that determines the CLSC network configuration as follows.

$$\text{Max} \quad \Gamma - \sum_{c\in C} fc_c YC_c - \sum_{a\in A} fa_a YA_a - \sum_{m\in M} fm_m YM_m$$

$$- \sum_{g \in G} fg_g YG_g - \sum_{b \in B} fb_b YB_b - \sum_{d \in D} fd_d YD_d \tag{46}$$

$$\text{s.t.} \quad \Gamma \leq \pi(\hat{\boldsymbol{\upsilon}}^{\boldsymbol{n}^T}) + \sum_{c \in C} cac_c YC_c \hat{\upsilon}_c^{27} + \sum_{a \in A} caa_a YA_a \hat{\upsilon}_a^{28} + \sum_{d \in D} cad_d YD_d \hat{\upsilon}_d^{29}$$
$$+ \sum_{b \in B} cab_b YB_b \hat{\upsilon}_b^{30} + \sum_{g \in G} \sum_{r \in R} cag_{gr} YG_g \hat{\upsilon}_{gr}^{31}$$
$$+ \sum_{m \in M} \sum_{l \in L} cam_{ml} YM_m \hat{\upsilon}_{ml}^{32} \tag{47}$$

$$0 \leq \pi(\hat{\boldsymbol{\kappa}}^{\boldsymbol{n}^T}) + \sum_{c \in C} cac_c YC_c \hat{\kappa}_c^{27} + \sum_{a \in A} caa_a YA_a \hat{\kappa}_a^{28} + \sum_{d \in D} cad_d YD_d \hat{\kappa}_d^{29}$$
$$+ \sum_{b \in B} cab_b YB_b \hat{\kappa}_b^{30} + \sum_{g \in G} \sum_{r \in R} cag_{gr} YG_g \hat{\kappa}_{gr}^{31}$$
$$+ \sum_{m \in M} \sum_{l \in L} cam_{ml} YM_m \hat{\kappa}_{ml}^{32} \tag{48}$$

$$\boldsymbol{Y} \in \{0, 1\} \tag{49}$$

where $\boldsymbol{\kappa}$ indicates extreme rays of Δ when the DSP is unbounded. Now, let Δ_p and Δ_r represent the sets of extreme points and extreme rays of Δ, respectively. Moreover, let V denote the capacities of several types of facilities in constraints (32)-(37) including collection, disassembly, remanufacturing, bulk recycling, material recycling, and disposal centers. The compact representation of the MP can be stated as follows.

$$\text{Max} \quad \Gamma - F^T \boldsymbol{Y} \tag{50}$$

$$\text{s.t.} \quad \Gamma \leq \pi(\hat{\boldsymbol{\upsilon}}^{\boldsymbol{n}^T}) + \hat{\boldsymbol{\upsilon}}^{\boldsymbol{m}^T} V \boldsymbol{Y} \quad (\boldsymbol{\upsilon}^{\boldsymbol{n}}, \boldsymbol{\upsilon}^{\boldsymbol{m}} | \boldsymbol{n} \neq \boldsymbol{m}) \in \Delta_p \tag{51}$$

$$0 \leq \pi(\hat{\boldsymbol{\kappa}}^{\boldsymbol{n}^T}) + \hat{\boldsymbol{\kappa}}^{\boldsymbol{m}^T} V \boldsymbol{Y} \quad (\boldsymbol{\kappa}^{\boldsymbol{n}}, \boldsymbol{\kappa}^{\boldsymbol{m}} | \boldsymbol{n} \neq \boldsymbol{m}) \in \Delta_r \tag{52}$$

$$\boldsymbol{Y} \in \{0, 1\} \tag{53}$$

Observe that the polyhedron Δ might have a vast number of extreme points and rays. An efficient iterative algorithm is to dynamically generate only a subset of optimality

and feasibility cuts. This approach is very effective since only a subset of these cuts will be active for the MP and most of them are redundant. Starting from empty subsets of extreme points and rays, each iteration of the algorithm first solves the MP. It provides an updated upper bound on the optimal solution of MIP. Then, the DSP is solved using the solution of the MP. If it is bounded, an optimal solution corresponds to an extreme point of Δ_p is identified and the optimality cut (51) is introduced to the MP. Otherwise, the feasibility cut (52) associated with an extreme ray of Δ_r is added to the MP.

2.4.2 Algorithmic enhancement

As Benders decomposition is known to be a method which converges quite slowly [26], we provide various algorithmic enhancements in order to accelerate the solution algorithm.

2.4.2.1 Valid inequalities

Considering the structure of the MIP model, we can introduce the following valid inequalities to the MP to restrict its feasible region. These cuts give useful information concerning the subproblem to MP and thus help it to find a better network configuration particularly at early iterations of the Benders algorithm. Therefore, we can expect that the presence of the valid inequalities reduces the number of feasibility cuts (52) during the execution of the solution process.

$$\sum_{c \in C} cac_c YC_c \geq \sum_{k \in K} \sum_{q \in Q} \psi_q dk_k \tag{54}$$

$$\sum_{a \in A} caa_a YA_a \geq \theta \sum_{k \in K} \sum_{q \in Q} \psi_q dk_k \tag{55}$$

$$\sum_{m\in M}\sum_{l\in L} cam_{ml} YM_m \geq \sum_{w\in W}\sum_{l\in L} dw_{wl} \tag{56}$$

$$\sum_{g\in G}\sum_{r\in R} cag_{gr} YG_g \geq \sum_{e\in E}\sum_{r\in R} de_{er} \tag{57}$$

$$\sum_{b\in B} YB_b \geq 1 \tag{58}$$

$$\sum_{d\in D} YD_d \geq 1 \tag{59}$$

Constraints (54) and (55) ensure that the selected collection and disassembly centers provide enough capacity to acquire returns. Constraints (56) and (57) ensure enough capacity for satisfying the demands of recovered components at their corresponding secondary markets through opening adequate recovery facilities. According to constraints (58) and (59), at least one bulk recycling and one disposal center must be opened in the CLSC network.

2.4.2.2 Pareto-optimal cuts

As the PSP is usually degenerate due to its typical network structure, the DSP might have multiple optimal solutions [30]. Hence, several valid optimality cuts of different strength associated to the set of alternative optimal solutions can be generated. Given the dominating cut definition [30], the optimal cut corresponds to the dual solution vectors $(\hat{\boldsymbol{v}}_1^{\boldsymbol{n}^T}, \hat{\boldsymbol{v}}_1^{\boldsymbol{m}^T})$ dominates the cut generated from the dual solution $(\hat{\boldsymbol{v}}_2^{\boldsymbol{n}^T}, \hat{\boldsymbol{v}}_2^{\boldsymbol{m}^T})$ if and only if

$$\pi(\hat{\boldsymbol{v}}_1^{\boldsymbol{n}^T}) + \hat{\boldsymbol{v}}_1^{\boldsymbol{m}^T} V\boldsymbol{Y} \leq \pi(\hat{\boldsymbol{v}}_2^{\boldsymbol{n}^T}) + \hat{\boldsymbol{v}}_2^{\boldsymbol{m}^T} V\boldsymbol{Y}; \quad \boldsymbol{n} \neq \boldsymbol{m}$$

for all $\boldsymbol{Y}$ with strict inequality for at least one point . A Pareto-optimal cut is an optimality cut which cannot be dominated by any other cut. It is usually expected

that appending Pareto-optimal cuts expedite the convergence of the Benders algorithm. Let $\mathbf{Y^{LP}}$ be the polyhedron defined by $0 \leq YC_c \leq 1, \forall\ c \in C; 0 \leq YA_a \leq 1, \forall\, a \in A; 0 \leq YM_m \leq 1, \forall\, m \in M; 0 \leq YB_b \leq 1, \forall\ b \in B; 0 \leq YG_g \leq 1, \forall\ g \in G;$, and $0 \leq YD_d \leq 1, \forall\ d \in D$ as well as constraints (6)-(37). Let $ri(\mathbf{Y^{LP}})$ indicate the relative interior of $\mathbf{Y^{LP}}$. A Pareto-optimal cut can be obtained by solving the following auxiliary dual problem.

$$\text{Min} \quad \pi(\boldsymbol{v}^{\boldsymbol{n}^T}) + \boldsymbol{v}^{\boldsymbol{m}^T} V \boldsymbol{Y^0} \tag{60}$$

$$\text{s.t.} \quad \pi(\boldsymbol{v}^{\boldsymbol{n}^T}) + \boldsymbol{v}^{\boldsymbol{m}^T} V \overline{\boldsymbol{Y}} = Z_{\boldsymbol{v}}(\overline{\boldsymbol{Y}}) \tag{61}$$

$$(\boldsymbol{v^1}, \boldsymbol{v^2}, ..., \boldsymbol{v^{32}}) \in \Delta \tag{62}$$

where $\boldsymbol{Y^0} \in ri(\mathbf{Y^{LP}})$, $\boldsymbol{n} \neq \boldsymbol{m}$, and Δ indicates the polyhedron defined by the constraints of the DSP. As mentioned earlier, the normalization constraint (61) might be quite dense and numerically unstable. Nonetheless, Papadakos [9] demonstrated that this constraint can be omitted through choosing a different core point on the objective function (60) every time the Pareto-optimal cut generation step is executed. It was also showed that any convex combination of the current master problem solution and an initial core point suffices to obtain a valid core point ([9], Theorem 8). The modified auxiliary dual subproblem can then be restated as follows.

$$\begin{aligned} \text{Min} \quad & \pi(\boldsymbol{v}^{\boldsymbol{n}^T}) + \boldsymbol{v}^{\boldsymbol{m}^T} V \boldsymbol{Y^0} \\ \text{s.t.} \quad & (\boldsymbol{v^1}, \boldsymbol{v^2}, ..., \boldsymbol{v^{32}}) \in \Delta \end{aligned} \tag{63}$$

where $\boldsymbol{n} \neq \boldsymbol{m}$ and Δ indicates the polyhedron defined by the constraints of the DSP. The optimal solution to (63) is used to generate an optimality cut, which is a Pareto-optimal one in the $ri(\mathbf{Y^{LP}})$. It should be noted that since the description of the convex hull of $\mathbf{Y}$ is not available a priori and also finding a core point in the convex

hull is difficult, Pareto-optimal cuts would be generated using $\boldsymbol{Y^0} \in ri(\mathbf{Y^{LP}})$. Such optimality cuts are non-dominated ones in $\mathbf{Y^{LP}}$. We refer to this accelerated Benders decomposition-based algorithm as "BD1" described in Algorithm 7. It should be emphasized that λ^c denotes the non-negative parameter to be used in the convex combination that updates the core point throughout the solution process for generating the Pareto-optimality cuts. It has been shown in the literature (see, e.g. [9, 35]) that 0.5 is a suitable choice for this parameter that gives the best results in most of cases. Hence, we use the same value in our algorithms.

Algorithm 1 -BD1

$UB \leftarrow \infty$, $LB \leftarrow -\infty$, $\lambda^c \leftarrow 0.5$
Find an initial core point $\boldsymbol{Y^0}$
Add valid inequalities (54)-(59) to the MP
while $(UB - LB)/UB \leq \epsilon$ **do**
Solve the auxiliary DSP (63)
Add Pareto-optimal cut (51) to the MP
Solve the MP
$UB \leftarrow \overline{\Gamma} - F^T\overline{\boldsymbol{Y}}$
Solve the DSP
if the DSP is unbounded **then**
    Add the feasibility cut (52) to the MP
    $\boldsymbol{Y^0} \leftarrow \lambda^c \boldsymbol{Y^0} + \xi$
else
    Add the optimality cut (51) to the MP
    $LB \leftarrow$ max (LB, $Z_{\boldsymbol{v}}(\overline{\boldsymbol{Y}}) - F^T\overline{\boldsymbol{Y}}$)
    $\boldsymbol{Y^0} \leftarrow \lambda^c \boldsymbol{Y^0} + (1 - \lambda^c)\overline{\boldsymbol{Y}}$
end if
end while
Solve the corresponding PSP

2.4.2.3 Local branching

Pilot computational tests have shown that although the Pareto-optimal cuts enhance the performance of the classical Benders decomposition algorithm, the upper bound still slowly decreases throughout the solution process. Therefore, we consider the

incorporation of local branching cuts [8, 34] throughout "BD1". Given a feasible reference point of an integer programming model, the main idea of local branching is to divide the feasible space of the problem into a series of smaller subproblems which can be solved by any appropriate generic solver. Therefore, one might be able to identify a better feasible solution in the neighborhood of the reference point within an acceptable computational time. In the context of Benders decomposition, each time the local branching search is executed, we may find better lower bounds as well as multiple optimality cuts that naturally lead to improved lower and upper bounds. We proceed with a detailed discussion of the local branching procedure.

The local branching is performed once the solution to the MP yields a feasible PSP, i.e., the DSP is bounded and optimal. We use this feasible solution as a reference point to create local branching subproblems. Let $\overline{\boldsymbol{Y}}^1$ be an optimal solution to the MP. Introducing the following disjunction, we can divide the feasible region of the MIP model (1)-(37) into two reduced subproblems.

$$\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^1) \leq \kappa_v \vee \Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^1) \geq \kappa_v + 1$$

The reduced subproblem created after adding the left branching cut to the MIP, namely the left branch subproblem, can be solved efficiently by CPLEX. The extended representation of this compact cut is stated as follows.

$$\begin{aligned}\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^1) = & \sum_{c \in C \setminus \overline{C}} YC_c + \sum_{c \in \overline{C}} (1 - YC_c) + \sum_{a \in A \setminus \overline{A}} YA_a + \sum_{a \in \overline{A}} (1 - YA_a) \\ & + \sum_{m \in M \setminus \overline{M}} YM_m + \sum_{m \in \overline{M}} ((1 - YM_m) + \sum_{b \in B \setminus \overline{B}} YB_b + \sum_{b \in \overline{B}} (1 - YB_b) \\ & + \sum_{g \in G \setminus \overline{G}} YG_g + \sum_{g \in \overline{G}} (1 - YG_g) + \sum_{d \in D \setminus \overline{D}} YD_d + \sum_{d \in \overline{D}} (1 - YD_d) \leq \kappa_v\end{aligned}$$

where $\overline{C}$, $\overline{A}$, $\overline{M}$, $\overline{B}$, $\overline{G}$, and $\overline{D}$ represent the set of open facilities in the reverse chain obtained after solving the current MP. Assigning a relatively low value to κ_v, imposing a time limit on the left branch subproblem, and a small optimality gap ϵ_k, each time a subproblem is solved, we ensure that the local branching procedure quickly explores different parts of the feasible region of the MIP. Let $\overline{\boldsymbol{Y}}^2$ be the solution to the local branching subproblem. After solving the local branching subproblem, one of the following cases might arise.

Case 1: If the optimal solution of the current subproblem has been identified within the time limit and the optimality gap, the left branching constraint will be replaced by the right branching one, i.e., $\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^1) \geq \kappa_v + 1$. The solution $\overline{\boldsymbol{Y}}^2$ is considered as the new reference point and the branching scheme will be applied to this solution, i.e., $\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^2) \leq \kappa_v$. We proceed the local branching search through solving the new local branching subproblem.

Case 2: If the current subproblem is proven infeasible, the left branching constraint will be replaced by the right one, i.e., $\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^1) \geq \kappa_v+1$. Moreover, the diversification procedure (*div.*) will be performed through increasing the size of the feasible region of the current subproblem by $\lceil \kappa_v/2 \rceil$. We proceed the local branching search through solving the new local branching subproblem.

Case 3: If the time limit is reached and the feasible solution to the current subproblem has been improved although it is not an optimal one, the left branching constraint will be eliminated without imposing the right branching one. Moreover, the "tabu" cut $\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^2) \geq 1$ will be introduced into the current subproblem to remove $\overline{\boldsymbol{Y}}^2$. Then, the new subproblem will be created by defining a left branching constraint associated with the new reference point, i.e., $\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^2) \leq \kappa_v$. We proceed the local branching search through solving the new local branching subproblem.

Case 4: If the time limit exceeds without improvement in the value of the objective

function of the current local branching subproblem, the right-hand side of the left branching constraint will be decreased by "1" and the tabu cut will also be added to the current subproblem to eliminate $\overline{\boldsymbol{Y}}^2$ from further consideration. The current subproblem will then be resolved in an attempt to find a better solution. In case no improved solution is found even in this new reduced neighborhood, the diversification procedure (*div.*) will be applied by enlarging the size of the feasible region by "1".

In addition, the tabu constraint is imposed at the beginning of the execution of the local branching procedure in order to exclude the solution to the current MP from further exploration.

The branching scheme is repeated through iterations of "BD1" until a specified number of local branching subproblems or diversifications will be satisfied. We remark that since local branching subproblems created by adding neighborhood constraints to MIP are quite hard to solve, the MIP model is only used in two iterations of the Benders algorithm to define the local branching subproblems, namely "MIP phase". As for the rest of iterations of the Benders algorithm, local branching subproblems are created by adding the neighborhood constraints to the MP, namely "MP phase". After each call to local branching procedure, several new feasible solutions, if any, are identified. They can be used to create a pool of optimality cuts (51), which will then be added to the MP to improve the quality of upper bound. This local branching-based Benders decomposition algorithm as described in Algorithm 8 is referred to as "BD2". The local branching procedure is also outlined in Algorithm 3. In this algorithm, the feasible solution to the local branching subproblems, when the first and the third cases arise, would be stored in $\overline{\boldsymbol{QX}}_h$ and $\overline{\boldsymbol{Y}}_h$ indicating, respectively, the CLSC network flows and locations. At the end of each local branching procedure, we obtain a lower (upper) bound through evaluating the objective function of MIP (MP) regarding these feasible solutions.

Algorithm 2 -BD2

$UB \leftarrow \infty$, $LB \leftarrow -\infty$, $t \leftarrow 1$, $MaxIter \leftarrow 2$
Find an initial core point $\boldsymbol{Y^0}$
Add valid inequalities (54)-(59) to the MP
while $(UB - LB)/UB \leq \epsilon$ **do**
Solve the auxiliary DSP (63)
Add Pareto-optimal cut (51) to the MP
Solve the MP
$UB \leftarrow \overline{\Gamma} - F^T\overline{\boldsymbol{Y}}$
Solve the DSP
if the DSP is unbounded **then**
    Add the feasibility cut (52) to the MP
    $\boldsymbol{Y^0} \leftarrow \lambda^c \boldsymbol{Y^0} + \xi$
else
    Add the optimality cut (51) to the MP
    $LB \leftarrow$ max (LB, $Z_{\boldsymbol{v}}(\overline{\boldsymbol{Y}}) - F^T\overline{\boldsymbol{Y}}$)
    $\boldsymbol{Y^0} \leftarrow \lambda^c \boldsymbol{Y^0} + (1 - \lambda^c)\overline{\boldsymbol{Y}}$
    $\overline{\boldsymbol{Y}}^1 \leftarrow \overline{\boldsymbol{Y}}$
    if $t \leq MaxIter$ **then**
        $t \leftarrow t + 1$
        MIP phase $\leftarrow$ true
        Perform the *LocBran.* procedure
    else
        MP phase $\leftarrow$ true
        Perform the *LocBran.* procedure
    end if
    Add the pool of optimality cuts (51) to the MP
end if
end while
Solve the corresponding PSP

2.5 Case example

We evaluate the tractability of the proposed model and the performance of the accelerated Benders decomposition-based algorithm for a case of large household appliances, i.e., used washing machines, inspired by [19] and [36]. The concerned washing machines consist of two modules, ten parts, and three types of solid materials. Table 2 presents the disassembly tree of a returned washing machine. The next section de-

Algorithm 3 -LocBran.

$rhs \leftarrow \kappa_v$, $Itr \leftarrow 1$, $dv \leftarrow 1$, $div. \leftarrow$ false, $h \leftarrow 1$
Add $\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^1) \geq 1$
while $(Itr \leq Sub.) \vee (dv \leq Ndiv.)$ **do**
Add $\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^1) \leq rhs$
Solve the resulting subproblem under a time limit as well as ϵ_k and label its solution $\overline{\boldsymbol{Y}}^2$, if any
if Case 1 **then**
Reverse the last local branching constraint into $\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^1) \geq \kappa_v + 1$
$\overline{\boldsymbol{Y}}^1 \leftarrow \overline{\boldsymbol{Y}}^2$, $div. \leftarrow$ false, $rhs \leftarrow \kappa_v$, $\overline{\boldsymbol{Y}}_h \leftarrow \overline{\boldsymbol{Y}}^2$, $h \leftarrow h + 1$, $Itr \leftarrow Itr + 1$
end if
if Case 2 **then**
Reverse the last local branching constraint into $\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^1) \geq \kappa_v + 1$
$rhs \leftarrow \kappa_v + \lceil \kappa_v / 2 \rceil$, $dv \leftarrow dv + 1$
end if
if Case 3 **then**
Eliminate the last local branching constraint $\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^1) \leq \kappa_v$
Add $\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^2) \geq 1$ to the current subproblem
$\overline{\boldsymbol{Y}}^1 \leftarrow \overline{\boldsymbol{Y}}^2$, $div. \leftarrow$ false, $rhs \leftarrow \kappa_v$, $\overline{\boldsymbol{Y}}_h \leftarrow \overline{\boldsymbol{Y}}^2$, $h \leftarrow h + 1$, $Itr \leftarrow Itr + 1$
end if
if Case 4 **then**
Eliminate the last local branching constraint $\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^1) \leq \kappa_v$
Add $\Delta(\boldsymbol{Y}, \overline{\boldsymbol{Y}}^2) \geq 1$ to the current subproblem
if $div.$ **then**
$dv \leftarrow dv + 1$, $rhs \leftarrow \kappa_v + 1$
else
$rhs \leftarrow \kappa_v - 1$
end if
$div. \leftarrow$ true
end if
end while
if MIP phase **then**
$LB \leftarrow \max\limits_{1 \leq h \leq Itr} P^T \overline{\boldsymbol{QX}}_h - C^T \overline{\boldsymbol{QX}}_h - F^T \overline{\boldsymbol{Y}}_h$
else
$UB \leftarrow \min\limits_{1 \leq h \leq Itr} \overline{\Gamma_h} - F^T \overline{\boldsymbol{Y}}_h$
end if
Generate pool of optimality cuts using $\overline{\boldsymbol{Y}}_h$

Table 2: Separable components of a used washing machine

Description	Value
ϕ_p	washing tube:1, cover:1, balance:1, frame:1, hose:1, condenser:1, small electric parts:1, electric wire:1, transformer:1, PCB:1
μ_r	plastics:6 kg, steel:3 kg, copper:1 kg
ω_l	motor:1, clutch:1

scribes test instances settings and then it is followed by a summary of computational results.

2.5.1 Experimental design

We assume three quality levels, namely, poor, medium, and high for the return stream. Demands of the brand-new washing machines and remanufactured modules are selected at random from {600, 601, 602, ..., 1000} and {50, 51, 52, ..., 150}, respectively. Demands of spare parts and recycled raw materials are also randomly determined from {30, 31, 32, ..., 100}. Capacities of facilities in the forward network are randomly generated following a reasonable relationship with the disassembly tree and demands of end-users. Capacities of reverse network facilitates are also randomly generated considering end-users and secondary markets demands as well as return ratios and recovery coefficients. Denote by "U" the uniform distribution, shipping costs are considered to be $U(5,\ 10)$ for each washing machine and $U(1,\ 4)$ for each unit of components. Other parameters are generated as summarized in Tables 3 and 4. Note that fixed costs of opening facilities in the reverse network are generated considering the capacity of facilities. More specifically, the higher the capacity of a facility, the larger infrastructural cost it will require. The legislative recovery target, i.e., θ, is 0.70.

We also consider seven major classes within each three different test instances as shown in Table 5. These test instances vary according to the number of CLSC facilities

Table 3: Quality level-dependent parameters

Description	Quality levels		
	High	Medium	Poor
ψ_q	$U(0.1, 0.2)$	$U(0.2, 0.3)$	$U(0.3, 0.4)$
δ_{lq}	1, 1	1, 0	0, 0
γ_{pq}	1, 1, 1, 1, 1, 1, 1, 1, 1, 1	1, 1, 0, 0, 0, 1, 1, 0, 0, 1	0, 0, 1, 0, 0, 0, 0, 0, 0, 1
α_{rq}	5, 2, 1	4, 1, 1	3, 1, 0
β_q	2	4	6
σ_q	0.1	0.2	0.4
cc_{cq}	1	1.5	2
ca_{aq}	1	1.5	2
cm_{mlq}	3	4	5
Pr_q	175	125	75

Table 4: Other case example parameters

Description	Value	Description	Value
fc_c	$U(400000, 600000)$	fa_a	$U(400000, 600000)$
fm_m	$U(700000, 900000)$	fb_b	$U(400000, 600000)$
fg_g	$U(400000, 600000)$	fd_d	$U(200000, 400000)$
Pk_k	$U(600, 1300)$	Ps_p	$U(50, 70)$
Pw_l	$U(100, 120)$	Pe_r	$U(20, 30)$
cx_{xl}	$U(70, 90)$	cz_{zp}	$U(30, 50)$
cu_{ur}	$U(10, 20)$	ci_i	$U(6, 7)$
cj_j	$U(1, 2)$	cb_b	$U(1.5, 2.5)$
cg_{gr}	$U(1.5, 2.5)$	cd_d	$U(1.5, 2.5)$
η_r	$U(0.2, 0.3)$	τ_r	$U(0.05, 0.15)$

as well as the number of the first and secondary markets. All twenty-one test instances have been randomly generated from the uniformly distributed demand, return ratio, and various parameters provided in Tables 3 and 4 through Monte Carlo sampling method. It should also be emphasized that we have chosen the bounds of most of the MIP model parameters based on the current literature of CLSC/RSC network design ([6], [19], [20]) to ensure the stability of the generated test instances. Table 6 presents the number of constraints and variables, including binary and continuous ones, in each problem set.

Table 5: Test problem classes

Set	Z	U	X	I	J	K	C	A	M	B	G	D	S	W	E
1	10	3	2	5	10	60	10	10	10	10	10	5	30	30	30
2	10	3	2	5	10	80	10	10	10	10	10	5	40	40	40
3	10	3	2	5	15	100	15	15	15	15	15	7	50	50	50
4	10	3	2	5	15	120	15	15	15	15	15	7	60	60	60
5	10	3	2	5	20	130	20	20	20	20	20	10	65	65	65
6	10	3	2	5	20	140	20	20	20	20	20	10	70	70	70
7	10	3	2	5	25	150	25	25	25	25	25	12	75	75	75

Table 6: Size of test problems

Set	# of constraints	# of continuous variables	# of binary variables
1	1349	10010	55
2	1579	12310	55
3	2041	22960	82
4	2271	26410	82
5	2619	39210	110
6	2734	41510	110
7	3081	56860	137

2.5.2 Computational results

The proposed algorithmic scheme is implemented in C++ using Concert Technology with IBM-ILOG CPLEX 12.51. All the experiments are conducted on an Intel Pentium 1.90 GHz machine with 4 GB RAM. The relative optimality gap, i.e., $\epsilon = 1\%$, as well as a maximum time of 3600 seconds were imposed as the stopping criteria for both "BD1" and "BD2". Furthermore, all 21 test instances were solved by CPLEX 12.51 in a maximum time limit of 7200 seconds and within the stopping gap tolerance of 1%. Table 7 includes the value of the local branching parameters, such as the number of local branching subproblems (Sub.) to be solved at each iteration of "BD2", the maximum number of diversifications (Ndiv.), time limit for solving each local branching subproblem (Time), the value of κ_v, and the optimality gap $\epsilon_{\kappa_v}\%$ each time this procedure is called in both phases of "MIP" and "MP".

Table 8 summarizes the computational statistics obtained after solving each test

Table 7: The value of parameters of local branching procedure

Set	MIP phase					MP phase				
	Sub.	Ndiv.	Time (sec)	κ_v	ϵ_{κ_v}	Sub.	Ndiv.	Time (sec)	κ_v	ϵ_{κ_v}
1	3	3	20	3	1%	1	3	-	10	0%
2	3	3	20	3	1%	1	3	-	10	0%
3	3	3	60	5	1%	1	3	-	25	0%
4	4	3	60	5	1%	1	3	-	25	0%
5	4	3	60	7	1%	1	3	-	30	0%
6	5	3	80	7	1%	1	3	-	30	0%
7	5	3	100	8	1%	1	3	-	40	0%

instance with "BD1", "BD2", and CPLEX. In this table, the resolution time in seconds (Time), the number of iterations (Iter.), and the value of the profit objective function are reported for both algorithms. Column "GAP" under "BD1" represents the relative difference between lower and upper bounds within the dedicated time limit, i.e., one hour. We also present the CPU time in seconds in addition to the profit reported by CPLEX after 7200 seconds. It should be stated that we have also examined the performance of the classical Benders decomposition method as well as the accelerated Benders algorithm described in Sherali and Lunday [33] on our test problems. However, neither the computational statistics of the former nor the latter have not been presented in Table 8 due to the poor performance of these two algorithms. The most important findings concerning Table 8 are summarized as follows.

- CPLEX is only able to find the optimal solution of the test instances of the first and the second problem sets indicated by asterisk symbol in the last column. However, it fails to solve the test instances of other sets to optimality within the dedicated time limit and gap tolerance. The best value of the objective function, i.e., profit, obtained by CPLEX after 2 hours when applied to solve other sets is also reported in the last column.

Table 8: Comparison of both algorithms and CPLEX

Set	"BD1"				"BD2"			CPLEX	
	Time (sec)	Iter.	Profit	GAP	Time (sec)	Iter.	Profit	Time (sec)	Profit
1	28	40	11,282,100	$\leq 1\%$	101	16	11,282,100	621	11,282,100*
	30	54	11,600,600	$\leq 1\%$	140	19	11,600,600	485	11,600,600*
	22	35	12,524,000	$\leq 1\%$	98	15	12,524,000	583	12,524,000*
2	289	128	17,071,000	$\leq 1\%$	231	35	17,071,000	605	17,071,000*
	1359	209	15,568,800	$\leq 1\%$	814	77	15,568,800	953	15,568,800*
	70	82	15,072,600	$\leq 1\%$	185	29	15,072,600	709	15,072,600*
3	≥ 3600	90	22,416,000	1.61%	924	20	22,494,300	≥ 7200	22,321,100
	≥ 3600	77	22,258,400	2.65%	670	11	22,271,200	≥ 7200	22,196,200
	≥ 3600	75	22,868,800	3.15%	587	14	22,951,700	≥ 7200	22,801,400
4	≥ 3600	110	28,496,100	2.32%	772	14	28,531,900	≥ 7200	28,475,000
	≥ 3600	91	20,482,300	2.42%	751	13	20,532,400	≥ 7200	20,260,000
	≥ 3600	139	22,473,400	2.34%	832	12	22,560,200	≥ 7200	22,415,500
5	≥ 3600	65	22,697,800	3.58%	1087	16	22,862,600	≥ 7200	22,601,400
	≥ 3600	97	27,909,000	2.00%	967	17	27,909,900	≥ 7200	27,702,800
	≥ 3600	107	20,563,300	2.84%	1023	16	20,610,300	≥ 7200	20,530,700
6	≥ 3600	74	31,624,100	1.62%	1835	19	31,682,100	≥ 7200	31,476,600
	≥ 3600	96	28,978,500	2.34%	1624	17	28,982,300	≥ 7200	28,556,300
	≥ 3600	59	31,723,300	2.69%	1498	15	31,829,700	≥ 7200	31,711,900
7	≥ 3600	19	31,700,500	2.88%	2510	14	31,741,300	≥ 7200	30,799,800
	≥ 3600	16	30,323,400	4.10%	2938	15	30,799,100	≥ 7200	29,708,100
	≥ 3600	57	30,501,300	2.45%	1963	14	30,573,000	≥ 7200	27,530,000
Average	-	82	22,768,348	-	1026	20	22,831,005	-	22,485,995

- "BD1" demonstrates better performance in terms of CPU time, i.e., on average, four times faster than "BD2" only in solving the test instances of the first problem set. However, "BD2" implementation reduces the number of iterations, on average, 2.5 times. We can also observe the rapid convergence of this algorithm when applied to solve the test instances of the second set. In particular, "BD2" is on average 1.4 times faster than "BD1" in terms of running time. Moreover, the average number of iterations during "BD2" execution is 3 times smaller than that of "BD1".

- As for the other five sets, "BD1" is unable to solve the test instances to optimality even in 1 hour. Columns GAP and profit indicate, respectively, the relative difference between the lower and upper bounds and the the value of the objective function obtained by "BD1" after one hour. However, "BD2" solves all test instances within the alloted optimality gap of 1% in considerably smaller number of iterations. Note that the average relative gap between profit values reported by "BD2" and CPLEX for the test instances of the last five sets is 1.82%. It indicates the extent to which the best feasible solution obtained by CPLEX within 2 hours is far from the optimal solution identified by "BD2". The average solution time of "BD2" is 18 minutes when applied to solve the test instances of problem sets 3 to 6. In addition, the average solution time of "BD2" is 42 minutes in solving the test instances of the last set, i.e., the largest set (see Figure 13).

The average number of iterations when solving each problem set with both decomposition algorithms is illustrated in Figure 5. In summary, it can be observed that local branching search considerably improves the performance of Benders decomposition algorithm.

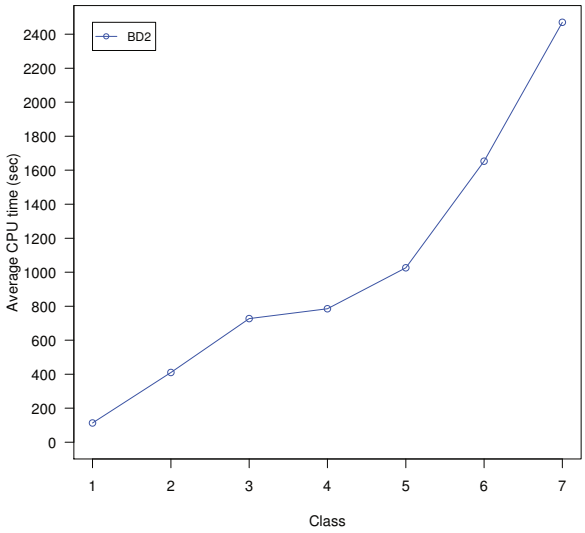

Figure 4: CPU time vs. Problem sets

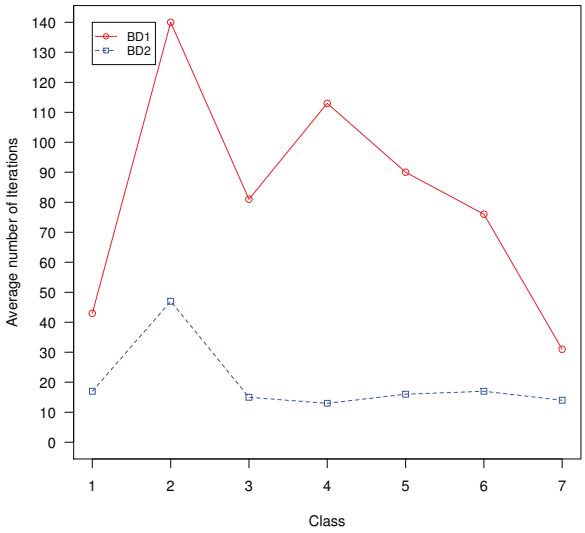

Figure 5: Iterations vs. Problem sets

2.5.3 The impact of recovery target on the CLSC performance

Recall that the costly regulatory restrictions are forcing OEMs on taking back of a certain percentage of the return stream. It, therefore, imposes the environmentally friendly constraint (13) in the CLSC network design model. We investigate the impact of the recovery target θ on the amount of the acquisition in disassembly centers. To this end, we use three values, 0.70 (the base case model), 0.77, and 0.80, as the recovery target levels. We also consider problem sets 1, 4, and 7, respectively, as small-, medium-, and large- sized networks and then choose one test instance from each problem set. These test instances vary in the total amount of end-user demands and the total quantity of returns in addition to the network size.

As expected, the OEM favors the recovery of high quality returns due to their

profitability, i.e., the total amount of the high quality returns is acquired in disassembly centers by the OEM in all three levels of the recovery target. Likewise, 100% of medium quality return stream is treated in the reverse network. However, the company is not willing to use all of potential recovery capacities to take back the poor quality return flows due to their low profitability. As shown in Figure 6, in the base case model, 41%, 44%, and 40% of poor quality returns are desirable for recovery processes, respectively, for the small-, medium-, and large-sized network. As the legislation requires more restrictive recovery targets, the OEM is forced to take a larger amount of poor quality returns. We can see from Figure 6 that the acquisition amount of poor quality returns significantly increases for each instance in accordance with the change in the recovery target. For example, with a target level of 0.80, the return acquisition reaches the value of 80% for the medium-sized network.

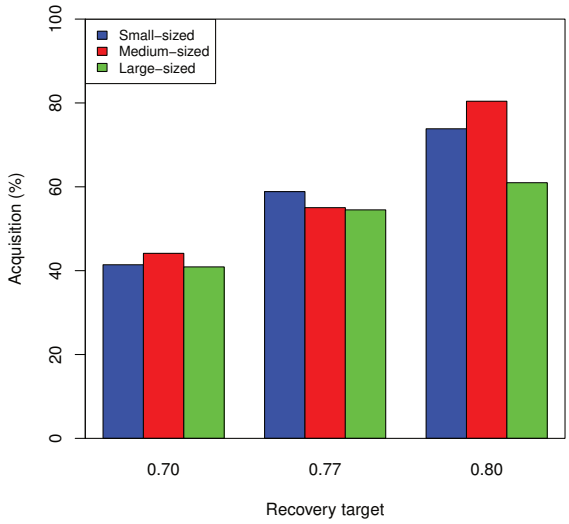

Figure 6: The percentage of returns acquisition

We also analyze the sensitivity of the objective function CLSC profit to changes in the recovery target through considering the same test instances. For this analysis, we

compare the profit of the base case model ($\theta = 0.70$) with those of 0.77 and 0.80 as the recovery target. Figure 7 illustrates profit decrease for each test instance comparing to the base case model. We observe that the profit of designing the network decreases for all concerned network sizes. For instance, the profit of designing the large-sized network decreases by 1.30% and 1.61%, respectively, in the target levels of 0.77 and 0.80. The implication is that increasing the recovery target requires installing more facilities or high capacity ones in the reverse network to process a larger amount of return flows, which incurs additional infrastructural costs. It also incurs extra processing costs at various types of facilities in the reverse channel. Observe from Figure 7 that the profit variation in the medium-sized network is considerably greater than the other test instances. This can be explained by the fact that in the large-sized network the revenue from the forward channel is remarkably higher than the infrastructural cost in the reverse channel. Therefore, the profit slightly decreases with higher levels of the recovery target. In a similar manner, since the total quantity of returns in the small-sized network is quite low, the costs of installing new facilities in the reverse channel has less impact on the profit compared to the medium-sized network.

Another observation to be made concerns the impact of the recovery target on the design of the CLSC. As shown above, with higher levels of the recovery target, more poor quality flow is sent to disassembly centers, which affects the configuration of the CLSC network and thus the total profit. Considering the disassembly tree, the viable recovery process of the additional poor quality components is to install high capacity and costly bulk recycling centers and consequently material recycling, and disposal facilities in the reverse network. One practical alternative for the OEM is outsourcing the bulk recycling processes to a third-party logistics provider. In this way, the OEM also avoids installing extra material recycling and disposal centers for

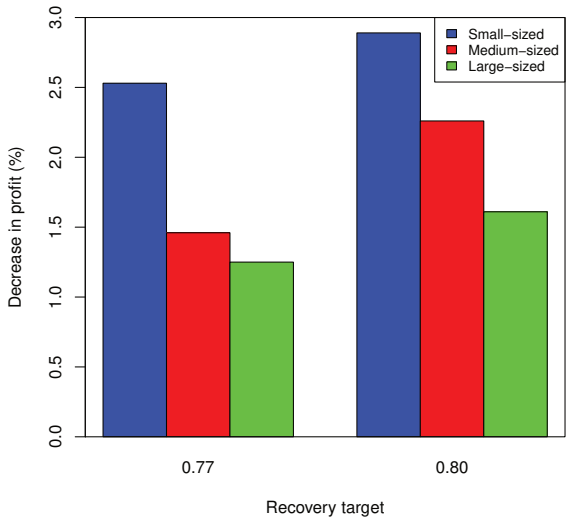

Figure 7: Profit variation

processing the additional amount of poor quality returns under new regulations.

2.6 Conclusion

In this article, we addressed a CLSC network design problem which is applicable in the context of durable products. We considered several features of practical relevance namely, a generic disassembly tree, all types of recovery processes plausible for each product component, the legislative recovery target, in addition to the non-homogeneity in the quality state of the return stream. Due to the generic features of the disassembly tree under discussion, the proposed model is not limited to applications for specific industries. It can also give some insight to decision makers how to design a CLSC with comprehensive recovery options.

In order to solve such a large-scale optimization problem, we developed two different variants of Benders decomposition algorithm, namely "BD1", and "BD2". The

former only takes advantage of generating Pareto-optimal cuts while the latter incorporates local branching search into the solution process. The performance of both methods were compared with CPLEX on a set of twenty-one test instances. As shown by the computational results, both solution algorithms outperformed CPLEX. When a comparison was made between two variants, "BD1" could only solve six test instances. However, "BD2" was able to solve all test instances to optimality in a reasonable amount of time. Particularly, the average number of iterations during the execution of "BD2" is 4 times smaller than that of "BD1". The improved convergence behavior of "BD2" is mainly due to efforts devoted to the local branching phases.

Many OEMs most often design and manufacture a family of products (e.g., LG and Samsung). This common practice in industry leads to introduce a comprehensive recovery system that can be readily generalizable to a family of products or equivalently a CLSC in a multi-product setting. Moreover, the dynamic nature of the CLSC network design problem which is mainly due to fluctuations/uncertainty in demand and quantity of returns over a planning horizon might not be often neglected. In this regard, multi-stage stochastic programming approach and modular capacity facilities could be the right approaches to be adopted. Hence, a natural extension of the current study would be to extend the proposed model into a multi-period setting. Furthermore, offering incentives to customers in order to encourage them to return used durable products prior to reaching their end-of-life would be another interesting area to investigate. Such incentives that can be offered in form of discounts on future purchases to customers would increase the percentage of high quality level returns. It should be mentioned that we are currently extending this work to incorporate the random quality state of used durable products into the proposed model and to solve the resulting large-scale stochastic problem through developing efficient algorithms. Application of the proposed model and solution methodology to a real industrial case

would also be an interesting extension of the current study.

Acknowledgment

This research was supported by Le Fonds de recherche du Québec-Nature et technologies (FRQNT) and the Natural Sciences and Engineering Research Council of Canada (NSERC). This support is gratefully acknowledged.

2.7 Appendix

Indices

Z: Set of part suppliers

U: Set of raw material suppliers

X: Set of module suppliers

I: Set of manufacturing centers

J: Set of distribution centers

K: Set of end-user zones

C: Set of collection centers

A: Set of disassembly centers

M: Set of remanufacturing centers

B: Set of bulk recycling centers

G: Set of material recycling centers

D: Set of disposal centers

W: Set of secondary markets for modules

S: Set of secondary markets for spare parts

E: Set of secondary markets for recycled materials

L: Set of modules

P: Set of parts

R: Set of raw materials

Q: Set of quality levels of returns

Parameters

ϕ_p: The number of part p in each unit of product

μ_r: The volume of material r in each unit of product

ω_l: The number of module l in each unit of product

ψ_q: The rate of return of each quality level q

β_q: The mass of residues in the returned product with quality level q shipped to bulk recycling centers from disassembly centers

α_{rq}: The mass of recyclable material r in the returned product with quality level q shipped to material recycling centers from disassembly centers

σ_q: The ratio of non-recoverable returns with quality level q shipped to disposal centers from disassembly centers

γ_{pq}: The number of part p in the returned product with quality level q shipped to secondary markets and manufacturing centers from disassembly centers

δ_{lq}: The number of remanufacturable module l in the returned product with quality level q shipped to remanufacturing centers from disassembly centers

η_r: The ratio of recyclable material r shipped to material recycling centers from bulk recycling centers

τ_r: The ratio of non-recyclable material r shipped to disposal centers from bulk and material recycling centers

θ: The legislative target for recovery of the return stream

fc_c: Fixed cost of opening collection center c

fa_a: Fixed cost of opening disassembly center a

fm_m: Fixed cost of opening remanufacturing center m

fg_g: Fixed cost of opening material recycling center g

fb_b: Fixed cost of opening bulk recycling center b

fd_d: Fixed cost of opening disposal center d

tc_{kc}: Shipping cost per unit of the returned product from end-user k to collection center c

ta_{ca}: Shipping cost per unit of the returned product from collection center c to disassembly center a

ts_{asp}: Shipping cost per unit of part p from disassembly center a to spare market s

tz_{aip}: Shipping cost per unit of part p from disassembly center a to manufacturing center i

td_{ad}: Shipping cost per kg of materials from disassembly center a to disposal center d

tg_{agr}: Shipping cost per kg of recyclable material r from disassembly center a to material recycling center g

te_{ger}: Shipping cost per kg of recycled material r from recycling center g to recycled material market e

tu_{gir}: Shipping cost per kg of recycled material r from material recycling center g to manufacturing centers i

sd_{gd}: Shipping cost per kg of waste from material recycling center g to disposal center d

tb_{ab}: Shipping cost per kg of residues from disassembly center a to bulk recycling center b

rg_{bgr}: Shipping cost per kg of recyclable material r from bulk recycling center b to material recycling center g

rd_{bd}: Shipping cost per kg of waste from bulk recycling center b to disposal center d

tm_{aml}: Shipping cost per unit of module l from disassembly center a to remanufacturing center m

tw_{mwl}: Shipping cost per unit of module l from remanufacturing center m to secondary market w

tx_{mil}: Shipping cost per unit of module l from remanufacturing center m to manufacturing center i

ti_{zip}: Shipping cost per unit of part p from part supplier z to manufacturing center i

ri_{uir}: Shipping cost per kg of material r from material supplier u to manufacturing center i

si_{xil}: Shipping cost per unit of module l from module supplier x to manufacturing center i

tj_{ij}: Shipping cost per unit of the new product from manufacturing center i to distribution center j

tk_{jk}: Shipping cost per unit of the new product from distribution center j to end-user k

caz_{zp}: Capacity of part supplier z for part p

cau_{ur}: Capacity of raw material supplier u for raw material r

cax_{xl}: Capacity of module supplier x for module l

cai_i: Capacity of manufacturing center i

caj_j: Capacity of distribution center j

cac_c: Capacity of collection center c

caa_a: Capacity of disassembly center a

cad_d: Capacity of disposal center d

cab_b: Capacity of bulk recycling center b

cag_{gr}: Capacity of material recycling center g for raw material r

cam_{ml}: Capacity of remanufacturing center m for module l

dk_k: Demand the new product at end-user zone k

ds_{sp}: Demand for part p at spare market s

de_{er}: Demand for material r at recycled material market e

dw_{wl}: Demand for module l at secondary market w

cz_{zp}: Procurement cost per unit of part p supplied by part supplier z

cu_{ur}: Procurement cost per kg of material r supplied by raw material supplier u

cx_{xl}: Procurement cost per unit of module l supplied by module supplier x

ci_i: Production cost per unit of product at manufacturing center i

cj_j: Distribution cost per unit of product at distribution center j

cc_{cq}: Processing cost per unit of the returned product with quality level q at collection center c

ca_{aq}: Processing cost per unit of the returned product with quality level q at disassembly center a

cd_d: Disposal cost at disposal center d

cg_{gr}: Recycling cost per kg of material r at material recycling center g

cb_b: Processing cost per kg of residues at recycling center b

cm_{mlq}: Remanufacturing cost per unit of module l with quality level q at remanufacturing center m

Pk_k: Unit price of the new product at end-user zone k

Ps_p: Unit price of part p at spare markets

Pe_r: Unit price of material r at recycled material markets

Pw_l: Unit price of module l at secondary markets

Pr_q: Unit acquisition price of the returned product with quality q

Decision variables

QI_{zip}: The number of part p shipped from part supplier z to manufacturing center i

NI_{uir}: The quantity of raw material r shipped from raw material supplier u to manufacturing center i

XI_{xil}: he number of module l shipped from module supplier x to manufacturing center i

QJ_{ij}: The quantity of products shipped from manufacturing center i to distribution center j

QK_{jk}: The quantity of products shipped from distribution center j to end-user zone k

QC_{kcq}: The quantity of returns with quality level q shipped from end-user zone k to collection center c

QA_{caq}: The quantity of returns with quality level q shipped from collection center c to disassembly center a

QS_{asp}: The number of part p shipped from disassembly center a to spare parts market s

QZ_{aip}: The number of part p shipped from disassembly center a to manufacturing center i

QM_{amlq}: The number of module l with quality level q shipped from disassembly center a to remanufacturing center m

QW_{mwl}: The number of module l shipped from remanufacturing center m to secondary market w

QX_{mil}: The number of module l shipped from remanufacturing center m to manufacturing center i

QB_{ab}: The quantity of residues shipped from disassembly center a to bulk recycling center b

QG_{agr}: The quantity of recyclable material r shipped from disassembly center a to material recycling center g

NG_{bgr}: The quantity of recyclable material r shipped from bulk recycling center b to material recycling center g

QE_{ger}: The quantity of recycled material r shipped from material recycling center g to recycled material market e

QU_{gir}: The quantity of recycled material r shipped from material recycling center g to manufacturing center i

QD_{ad}: The quantity of non-recoverable components shipped from disassembly center a to disposal center d

ND_{bd}: The quantity of residues shipped from bulk recycling center b to disposal center d

XD_{gdr}: The quantity of raw material r shipped from material recycling center g to disposal center d

YC_c: A binary variable which is equal to one if collection center c is opened and zero otherwise

YA_a: A binary variable which is equal to one if disassembly center a is opened and zero otherwise

YM_m: A binary variable which is equal to one if remanufacturing center m is opened and zero otherwise

YB_b: A binary variable which is equal to one if bulk recycling center b is opened and zero otherwise

YG_g: A binary variable which is equal to one if material recycling center g is opened and zero otherwise

YD_d: A binary variable which is equal to one if disposal center d is opened and zero otherwise

Chapter 3

Closed-loop supply chain network design under uncertain quality status: case of durable products

This chapter is dedicated to the article entitled *"Closed-loop supply chain network design under uncertain quality status: case of durable products"*. This article has been accepted for publication in the *International Journal of Production Economics*. The titles, figures, and mathematical formulations have been revised to keep the coherence through the manuscript.

Abstract

This paper proposes a two-stage stochastic mixed-integer programming model for a closed-loop supply chain network design problem in the context of modular structured products in which the reverse network entails several types of recovery options. It accounts for uncertainty in the quality status of the return stream, modeled as binary scenarios for each component in the reverse bill of material corresponding to such products. To deal with the intractable number of scenarios in the proposed model, a scenario reduction scheme is adapted to the problem of interest to preserve the most pertinent scenarios based on a modified Euclidean distance measure. The reduced stochastic large-scale optimization problem is then solved via a L-shaped algorithm enhanced with surrogate constraints and Pareto-optimal cuts. Numerical results indicate that the scenario reduction algorithm provides good quality solutions to the stochastic problem in a reasonable amount of time through applying the enhanced L-shaped method.

3.1 Introduction

In response to sustainability of supply chains, design and management of CLSC have attracted a growing interest over the recent decade. It has been recognized that CLSCs comprise forward channel along with the so-called "RSC". The focus of the RSC is on taking back of end-of-life and end-of-use products (cores) from consumers, and recovering added value by reusing the entire product, and/or some of its components, such as modules and parts [37]. The prime importance of CLSCs is attributed to the environmental footprint of cores as well as the profitability of recovery practices. Needless to say, the prosperity of such business practices requires placing appropriate

reverse logistics infrastructures and managing arising return flows efficiently. Therefore, the design of CLSCs is becoming increasingly important.

CLSC network design refers to decisions for locating various types of facilities in both forward and reverse chains in addition to efficiently routing and coordinating physical forward and reverse flows. Designing a CLSC network for durable products, that are characterized by their modular structured design and their long life cycle (e.g., computers and large household appliances), is a complex problem. Such category of products can be disassembled into several components (i.e., modules and parts) in addition to raw materials concerning the reverse BOM. Consequently, the reverse supply chain of such products includes various types of recovery facilities associated with different components in the reverse BOM of durable products. We note that in the context of CLSC/RSC network design, most studies are limited to involving a few recovery activities, e.g., remanufacturing and material recycling, in designing their networks. To fill the void in such a line of research, we, however, incorporate various recovery options, which an OEM can adopt in tackling the return stream. These recovery processes are plausible in taking different sub-assemblies of a typical durable product.

Indeed, a high level of uncertainty is a characteristic for various product recovery systems [38]. A clear distinction that is made between CLSCs and the traditional forward supply chains lies in uncertain condition (quality) of cores. This issue also adds to the complexity of the CLSC network design for case of used durable products. In contrast, the existing stream of literature explicitly lacks accounting such unavoidable aspect in CLSC/RSC network design. That is, in the modeling framework, some simplifying assumptions have been made with respect to the quality variation of the return flows, thereby alleviating the complexity of the proposed model. More precisely, the concerned literature has focused on classifying cores with respect to

deterministic disjoint quality levels [24, 25, 39], which ignores the uncertainty. In a similar fashion, the uncertain quality of returns has been addressed through a finite number of grading process outcomes, each one occurs with a positive probability, such that every grading outcome is a combination of disjoint quality levels in diverse percentages [40]. Another attempt to model uncertain quality has been considering the rate of recovery as a random variable [41]. Nonetheless, the above-mentioned approaches are rough approximations of the uncertain quality status of cores.

Observing this major drawback, in this study, as our prime contribution, we propose a more precise approach to model the uncertain quality status, where the availability of each component in the reverse BOM is modeled as a discrete scenario following a Bernoulli distribution. This novel approach leads to an explicit incorporation of the uncertain quality status of returns. In this regard, we formulate this large-scale optimization problem as a two-stage stochastic mixed-integer program with recourse [42]. As far as the authors are aware, the design of a CLSC concerning the explicit modeling of the uncertain quality of each component in the reverse BOM has never been investigated in the context of durable products. Even though it is not the prime concern of this paper, it should be stated that the uncertain demand of brand-new products, quantity of returns, and economic parameters, e.g., unit prices can also be taken into account as other sources of uncertainty in designing the CLSC network.

On account of the fact that proposing the aforementioned approach for modeling the uncertain quality exponentially increases the number of scenarios, as the second contribution, we implement fast forward selection algorithm [43, 44] adapted to our problem setting as a scenario reduction scheme to preserve the most pertinent scenarios. Further, on the methodological side, our third contribution is to develop an enhanced solution approach based on L-shaped method [45], which shows a consistent

performance efficiency in our experimentations. The classical L-shaped algorithm results from applying Benders decomposition [5] to two-stage stochastic programming models with continuous variables. This decomposition approach typically requires algorithmic refinements to accelerate its convergence. We, therefore, provide enhancement strategies that include adding induced constraints to the master problem to restrict its feasible region and also generating Pareto-optimal cuts to strengthen the deepness of optimality cuts throughout the execution of the solution algorithm.

The remainder of this paper is organized as follows. In the next section, we provide the review of the relevant literature. In Section 3.3, we introduce the problem setting and its two-stage stochastic programming formulation. Section 3.4 elaborates on the scenario reduction algorithm. In Section 3.5, a detailed description of the enhanced L-shaped method is given. Section 3.6 presents computational experiments on the performance of the proposed solution method for a large household appliance example. In the last Section, we provide concluding remarks and future research directions.

3.2 Literature review

Many efforts have been made to model and optimize deterministic CLSC/ RSC network design problems. The current literature in the context of product recovery offers a variety of problems spanning from recycling of simple waste, such as carpet and sand to different recovery options of more complex products [4, 6, 13, 15, 17, 24, 25, 39, 46]. For an extensive review of proposed models and cases, the reader is referred to [10, 11].

The overview of the existing literature implies that in most of prevailing studies uncertainty consideration is limited to demand and quantity of returns. Observing a few exceptions [40, 41], the impact of uncertainty in quality of returns on designing CLSCs, regardless of its substantial impact, has received scant attention in the literature.

The most common approach that makes an explicit attempt to incorporate uncertainty in design parameters is stochastic optimization approaches. Since the seminal work of Salema et al. [21], in which the authors addressed a CLSC network design problem under uncertain demand and quantity of returns, some researchers have developed two-stage stochastic programming formulations to model such stochastic parameters. Details of these studies can be found in [20, 22, 23, 47, 48]. The analysis of the current literature suggests the following observations: 1) most of them are case oriented logistical networks; 2) the number of scenarios is quite small (e.g., twelve scenarios in [20]); and, 3) concerning the size of test problems, the optimal solution is found by virtue of commercial softwares. It should be stated that [20] is an exception to the last observation such that an integer L-shaped algorithm was devised as the solution approach. Nonetheless, its algorithmic framework is only suited for addressing situations with a very small number of scenarios.

As we noted earlier, only a few recent studies have captured uncertainty in the quality status of returns. While accounting a multi-period setting, Zeballos et al. [40] addressed a stochastic CLSC network design problem for which the solutions are obtained relying on a commercial software. Chen et al. [41] considered the rate of recovery as a measure to reflect the quality status of cores. Accordingly, the authors modeled the random recovery rate as a set of scenarios for a CLSC network design, including collection and remanufacturing facilities in the reverse channel. The resulting two-stage stochastic quadratic formulation was solved via the integer L-shaped method integrated with the sample average approximation scheme. As highlighted in the preceding section, we, however, capture the uncertain quality status of the return stream in a more precise setting with regard to the reverse BOM in the context of modular structured products. Our approach differs in the scope from [40] and [41] in the sense that: 1) it discriminates the quality state of cores in terms of the availability

of components in the reverse BOM of a durable product; 2) it involves various types of recovery options plausible for the sub-assemblies of a durable product; 3) the choice of the recovery process for each component depends on its quality status; and 4) the enhanced L-shaped method together with the proposed scenario reduction strategy is capable of serving as a viable tool for designing a realistic-scale CLSC network with quite crude information of quality status.

3.3 Problem description and formulation

In what follows, first, the description of the CLSC network applicable for durable products is presented. Then, we articulate how to model the random quality status of cores. Finally, we formulate the CLSC network design problem as a two-stage stochastic mixed-integer program with recourse.

3.3.1 CLSC network design for durable products

In the CLSC network design problem of interest, an organization operates a well-established forward channel in which the forward network comprises components and raw materials suppliers, manufacturing facilities, distribution centers, and end-user locations. The organization tends to adopt some suitable recovery practices to satisfy the directive recycling target stipulated by the legislator as well as to reclaim the economic value from used components. Hence, the aim is to extend the existing forward network to accommodate the recovery facilities and consequently to coordinate the physical forward and reverse flows in the extended supply chain network. The reverse network consists of collection, disassembly, remanufacturing, bulk recycling, material recycling, and disposal centers, referred to as the recovery facilities. The returned durable products received at end-user zones are shipped to disassembly centers

through collection centers. In our problem setting, the quality status of a returned durable product is defined as the availability of modules and parts for remanufacturing and part harvesting/reusing practices along with the mass of residues for bulk recycling processes. In disassembly centers, the inspected return stream is disassembled into different components based on the reverse BOM. The recoverable modules and recyclable materials are then sent to remanufacturing and recycling centers for further processes. Besides, the bulk of mixed residues would typically be processed in bulk recycling centers to separate the precious raw materials from mixed scrap, e.g., electronic scrap. The bulk recycling step is followed by material recycling and landfilling/incineration at disposal points. The material recycling step is designed to recover the raw materials. The non-valuable remains, i.e., process waste, should be safely disposed of at disposal centers. The ultimate purpose of remanufacturing modules, part harvesting, and recycling raw materials is twofold: 1) shipping to manufacturing facilities to deploy in manufacturing of brand-new products; and, 2) selling on potential secondary markets. Given the above description, the conceptual structure of the CLSC under consideration is schematically illustrated in Figure 8. The solid arcs indicate the forward flows while the dashed ones denote the reverse flows in the CLSC.

3.3.2 Modeling random quality states of the return stream

The quality status of returns is indeed affected by the changes having been made during the lifetime of durable products. More specifically, in many cases, due to the usage and the deterioration rates during the long life cycle of such products, it is quite impossible to foresee the exact number of recoverable components in a returned durable product. Moreover, the quality status can only be revealed after grading the returned items in disassembly centers. In this study, the random quality state

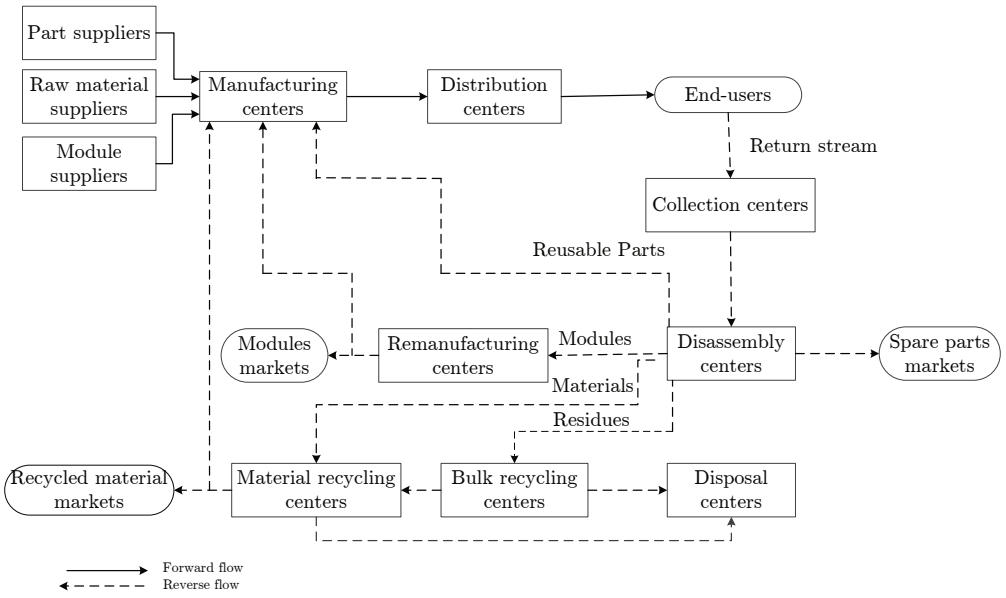

Figure 8: General structure of the CLSC network

is defined as the availability of each component in the reverse BOM and modeled as discrete scenarios with Bernoulli probability distribution. Let P and L denote, respectively, the set of parts and modules in the brand-new durable product. Further, let γ_p, δ_l, and β denote, respectively, the number of reusable part p, the number of remanufacturable module l, and the mass of residues in the returned durable product. Now that we represent the random quality vector by $\boldsymbol{\xi}$ where $\boldsymbol{\xi} = \{\gamma_p | \forall\ p \in P; \delta_l | \forall\ l \in L; \beta\}$. We also represent each particular realization (scenario) of the random quality status by $\gamma_p(\boldsymbol{\xi}_s)$, $\delta_l(\boldsymbol{\xi}_s)$, and $\beta(\boldsymbol{\xi}_s)$. Each particular scenario s is associated with a non-negative probability π_s such that $\sum_{s \in S} \pi_s = 1$. Once the grading process is executed in disassembly centers, it is realized whether or not a particular part/module is suitable for the effective recovery process. For the sake of clarification, the following assumptions are made.

- The grading process yields a good condition component with the probability of functionality $\hat{p}$ otherwise a poor state one with the failure probability $\hat{q} = 1 - \hat{p}$,

- The probability of functionality of each component (module/part) is the same in all returned items,
- The condition of different returned items are independent and identically distributed,
- Durable goods are homogeneous.

As an example, we point out to a used twin tub of a washing machine such that each washing tube unit is 3.5 kg in weight. Every unit can independently be either functional with probability $\hat{p}$ or defective with probability $\hat{q}$. Hence, the outcome of the grading process for the washing tube follows a Bernoulli distribution. Likewise, the quality status of other components in the reverse BOM is an independent random variable following a Bernoulli distribution. For every realization of the random quality vector, i.e., $\boldsymbol{\xi_s}$, we define an indicator function for each unit j of part p and another indicator function for each unit k of module l as follows.

$$I(p^j) = \begin{cases} 1 & \text{if unit } j \text{ is in a functional state;} \quad j = 1, ..., n^p \\ 0 & \text{otherwise} \end{cases}$$

$$I(l^k) = \begin{cases} 1 & \text{if unit } k \text{ is in a functional state;} \quad k = 1, ..., n^l \\ 0 & \text{otherwise} \end{cases}$$

It allows to consider, respectively, the number of reusable part p as $\gamma_{ps} = \sum_{j=1}^{n^p} I(p^j)$ and the number of remanufacturable module l as $\delta_{ls} = \sum_{k=1}^{n^l} I(l^k)$. For example, in the aforementioned washing machine case, a possible outcome of the grading process might be one fully functional and one defective washing tube. Thus, for this specific part, γ is equal to 1. On the other hand, the indicator function of the defective unit takes 0 with probability $\hat{q}$. The defective unit will be considered as residues and the viable recovery option for this unit will be bulk recycling process. In this regard,

β will be increased with the corresponding weight of the unit, i.e., 3.5 kg and thus β is equal to 3.5 kg. In other words, all defective units observed after the grading operation are considered as residues and β is equal to the total summation of their corresponding weights.

The scenario generation approach described above yields 2^n scenarios for a typical durable product that entails n different types of components. The probability of each scenario can be calculated as follows.

$$\pi_s = \hat{p}^{\gamma_{ps}+\delta_{ls}}.\hat{q}^{n^p+n^l-\gamma_{ps}-\delta_{ls}}$$

For example, suppose another washing machine that comprises ten different types of parts, namely P1, P2, ..., P10 and two different types of modules, namely L1 and L2 such that it involves only one unit per each type of parts and modules. Further, each component can be functional with probability 0.3, i.e., $\hat{p} = 0.3$. Consequently, the grading process results in 2^{12} scenarios. Table 9 presents five scenarios and their corresponding probability of occurrence among all possible realization of quality state scenarios for the grading process of this washing machine.

Table 9: Example of quality state scenarios

Scenario	Component type												Probability
	L1	L2	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	
1	1	0	1	0	0	1	1	1	0	1	0	1	0.000037
2	1	1	1	0	1	0	0	0	1	1	0	1	0.000037
3	0	1	1	1	1	1	1	0	1	1	0	0	0.000016
4	0	1	1	0	1	0	0	0	0	0	1	0	0.000467
5	1	1	1	0	1	1	1	0	1	1	1	1	0.000002

As shown in Table 9, in the first scenario (the first row of the table), there are six parts in functional condition and the others are considered as residues and should be treated in bulk recycling to recover their precious materials. Therefore, the value of the parameter γ is equal to 6. Likewise, only one module, i.e., L1, is remanufacturable

and thus δ is equal to 1. The parameter β is then equal to the total weights of the components taking value of 0 in the first scenario, i.e., P2, P3, P7, P9, and L2. The probability of the first scenario is calculated as $0.3^7 \times 0.7^5 = 0.000037$.

3.3.3 Two-stage stochastic programming formulation

Given the described scenario generation approach for random quality status, we can formulate our problem setting as a two-stage stochastic program with recourse. In a general two-stage stochastic programming model, the first stage decisions are taken when the decision maker does not have enough information about the outcome of uncertain parameters, while, the second stage decisions are implemented after the uncertainty is realized. In other words, the "second stage" decision is made when the complete information with respect to uncertain parameter(s) is available.

Referring to Figure 8, in our problem setting, in the first stage, the location of collection, disassembly, remanufacturing, bulk recycling, material recycling, and disposal facilities in the reverse network should be determined before the complete information on the quality status of returns is available. Thus, the binary location decisions are first stage decisions. Moreover, since the demand is deterministic, the uncertainty in quality status of returns does not affect the decisions on the quantity of brand-new durable products shipped from manufacturing facilities to end-users via distribution centers. Therefore, such forward flow decisions are also among the first stage decisions. Likewise, as the quantity of returns is deterministic, the reverse flow from end-users to collection sites and the flow from collection centers to disassembly facilities are also first stage decisions. Lastly, regardless of the quality state of returns, they contain precious raw materials that can also be recycled in material recycling centers. Accordingly, the flow of recyclable materials from disassembly centers to material recycling sites is considered as a first stage decision variable. Once the

returns are graded in disassembly centers, a complete information on the number of recoverable modules and parts in addition to the mass of residues is available to the decision maker. Consequently, the remaining flows in the reverse channel are referred to as the second stage decisions (recourse actions). The physical flows from various suppliers to manufacturing centers are also considered as second stage decisions due to the impact of uncertain quality states on them. Further assumptions about our problem of interest described earlier in this section are as follows.

- The planning horizon is considered as the life cycle of durable goods (e.g., 5 years for large household appliances),
- The demand is deterministic and must be fully satisfied at end-users zones while the deterministic demands of used parts, remanufactured modules, and recycled raw materials at secondary markets can be partially fulfilled,
- The rate of return is known a priori,
- The capacities of all facilities in the underlying CLSC are limited and fixed (i.e., there is no planned capacity expansion). Further, opening any type of facilities in the reverse network incurs a fixed location cost,
- Mandated by the governmental restrictions, the OEM is obliged to acquire a substantial portion of the return stream for recovery purposes,
- All prices, procurement, processing, and shipping costs are deterministic.

The notations used in the mathematical model is listed in the *Appendices*, in the section entitled *Problem notations*. The two-stage stochastic mixed-integer programming model with recourse can be stated as follows.

$$
\begin{aligned}
\text{Max} \quad & \sum_{j\in J}\sum_{k\in K} Pk_k XK_{jk} - \sum_{i\in I}\sum_{j\in J}(ci_i + tj_{ij})XJ_{ij} \\
& - \sum_{j\in J}\sum_{k\in K}(cj_j + tk_{jk})XK_{jk} - \sum_{k\in K}\sum_{c\in C}(cc_c + tc_{kc})XC_{kc} \\
& - \sum_{c\in C}\sum_{a\in A}(Pr + ca_a + ta_{ca})XA_{ca} - \sum_{a\in A}\sum_{g\in G}\sum_{r\in R}(cg_{gr} + tg_{agr})XG_{agr} \\
& - \sum_{c\in C} fc_c YC_c - \sum_{a\in A} fa_a YA_a - \sum_{m\in M} fm_m YM_m - \sum_{b\in B} fb_b YB_b \\
& - \sum_{g\in G} fg_g YG_g - \sum_{d\in D} fd_d YD_d \\
& + \sum_{s\in S} \pi_s Q(XJ_{ij}, XK_{jk}, XC_{kc}, XA_{ca}, XG_{agr}, YC_c, YA_a, YB_b, YM_m, \\
& YG_g, YD_d, \boldsymbol{\xi}_s) \qquad (64)
\end{aligned}
$$

$$
\text{s.t.} \quad \sum_{i\in I} XJ_{ij} = \sum_{k\in K} XK_{jk} \quad \forall\, j \in J \qquad (65)
$$

$$
\sum_{j\in J} XK_{jk} = dk_k \quad \forall\, k \in K \qquad (66)
$$

$$
\sum_{c\in C} XC_{kc} = \psi dk_k \quad \forall\, k \in K \qquad (67)
$$

$$
\sum_{k\in K} XC_{kc} \geq \sum_{a\in A} XA_{ca} \quad \forall\, c \in C \qquad (68)
$$

$$
\sum_{c\in C}\sum_{a\in A} XA_{ca} \geq \lambda \sum_{k\in K}\sum_{c\in C} XC_{kc} \qquad (69)
$$

$$
\sum_{c\in C} \alpha_r XA_{ca} = \sum_{g\in G} XG_{agr} \quad \forall\, a \in A, \forall\, r \in R \qquad (70)
$$

$$
\sum_{j\in J} XJ_{ij} \leq cai_i \quad \forall\, i \in I \qquad (71)
$$

$$
\sum_{i\in I} XJ_{ij} \leq caj_j \quad \forall\, j \in J \qquad (72)
$$

$$
\sum_{k\in K} XC_{kc} \leq cac_c YC_c \quad \forall\, c \in C \qquad (73)
$$

$$\sum_{c\in C} XA_{ca} \le caa_a YA_a \quad \forall\, a \in A \tag{74}$$

where $Q(XJ_{ij}, XK_{jk}, XC_{kc}, XA_{ca}, XG_{agr}, YC_c, YA_a, YM_m, YB_b, YG_g, YD_d, \boldsymbol{\xi}_s)$ is the optimal value of the following problem:

$$\begin{aligned}
\text{Max} \quad & \sum_{m\in M}\sum_{w\in W}\sum_{l\in L} Pw_l QW_{mwls} + \sum_{a\in A}\sum_{o\in O}\sum_{p\in P} Ps_p QS_{aops} \\
& + \sum_{g\in G}\sum_{e\in E}\sum_{r\in R} Pe_r QE_{gers} - \sum_{z\in Z}\sum_{i\in I}\sum_{p\in P}(cz_{zp} + ti_{zip})QI_{zips} \\
& - \sum_{u\in U}\sum_{i\in I}\sum_{r\in R}(cu_{ur} + ri_{uir})NI_{uirs} - \sum_{h\in H}\sum_{i\in I}\sum_{l\in L}(cx_{hl} + si_{hil})HI_{hils} \\
& - \sum_{a\in A}\sum_{m\in M}\sum_{l\in L}(cm_{ml} + tm_{aml})QM_{amls} - \sum_{a\in A}\sum_{b\in B}(cb_b + tb_{ab})QB_{abs} \\
& - \sum_{b\in B}\sum_{g\in G}\sum_{r\in R}(cg_{gr} + rg_{bgr})NG_{bgrs} - \sum_{b\in B}\sum_{d\in D}(cd_d + rd_{bd})ND_{bds} \\
& - \sum_{g\in G}\sum_{d\in D}\sum_{r\in R}(cd_d + sd_{gd})GD_{gdrs} \qquad (75) \\
& - \sum_{a\in A}\sum_{o\in O}\sum_{p\in P} ts_{aop} QS_{aops} - \sum_{m\in M}\sum_{w\in W}\sum_{l\in L} tw_{mwl} QW_{mwls} \\
& - \sum_{g\in G}\sum_{e\in E}\sum_{r\in R} te_{ger} QE_{gers} - \sum_{a\in A}\sum_{i\in I}\sum_{p\in P} tz_{aip} QZ_{aips} \\
& - \sum_{m\in M}\sum_{i\in I}\sum_{l\in L} tx_{mil} QX_{mils} - \sum_{g\in G}\sum_{i\in I}\sum_{r\in R} tu_{gir} QU_{girs} \qquad (76)
\end{aligned}$$

$$\text{s.t.} \quad \sum_{z\in Z} QI_{zips} + \sum_{a\in A} QZ_{aips} = \phi_p \sum_{j\in J} XJ_{ij} \quad \forall\, i \in I, \forall\, p \in P, \forall\, s \in S \tag{77}$$

$$\sum_{u\in U} NI_{uirs} + \sum_{g\in G} QU_{girs} = \mu_r \sum_{j\in J} XJ_{ij} \quad \forall\, i \in I, \forall\, r \in R, \forall\, s \in S \tag{78}$$

$$\sum_{h\in H} HI_{hils} + \sum_{m\in M} QX_{mils} = \omega_l \sum_{j\in J} XJ_{ij} \quad \forall\, i \in I, \forall\, l \in L, \forall\, s \in S \tag{79}$$

$$\sum_{c\in C} \gamma_{ps} XA_{ca} = \sum_{i\in I} QZ_{aips} + \sum_{o\in O} QS_{aops} \quad \forall\, a \in A, \forall\, p \in P, \forall\, s \in S \tag{80}$$

$$\sum_{a\in A} QS_{aops} \leq ds_{op} \quad \forall\, o \in O, \forall\, p \in P, \forall\, s \in S \tag{81}$$

$$\sum_{c\in C} \beta_s XA_{ca} = \sum_{b\in B} QB_{abs} \quad \forall\, a \in A, \forall\, s \in S \tag{82}$$

$$\sum_{c\in C} \delta_{ls} XA_{ca} = \sum_{m\in M} QM_{amls} \quad \forall\, a \in A, \forall\, l \in L, \forall\, s \in S \tag{83}$$

$$\sum_{a\in A} QM_{amls} = \sum_{w\in W} QW_{mwls} + \sum_{i\in I} QX_{mils} \;\forall\, m \in M, \forall\, l \in L, \forall\, s \in S \tag{84}$$

$$\sum_{m\in M} QW_{mwls} \leq dw_{wl} \quad \forall\, w \in W, \forall\, l \in L, \forall\, s \in S \tag{85}$$

$$\sum_{a\in A} \eta_r QB_{abs} = \sum_{g\in G} NG_{bgrs} \quad \forall\, b \in B, \forall\, r \in R, \forall\, s \in S \tag{86}$$

$$\sum_{a\in A} QB_{abs} = \sum_{g\in G}\sum_{r\in R} NG_{bgrs} + \sum_{d\in D} ND_{bds} \quad \forall\, b \in B, \forall\, s \in S \tag{87}$$

$$\sum_{a\in A} \tau_r XG_{agr} + \sum_{b\in B} \tau_r NG_{bgrs} = \sum_{d\in D} GD_{gdrs} \;\forall\, g \in G, \forall\, r \in R, \forall\, s \in S \tag{88}$$

$$\sum_{g\in G} QE_{gers} \leq de_{er} \quad \forall\, e \in E, \forall\, r \in R, \forall\, s \in S \tag{89}$$

$$\sum_{a\in A} XG_{agr} + \sum_{b\in B} NG_{bgrs} = \sum_{i\in I} QU_{girs} + \sum_{e\in E} QE_{gers}$$
$$+ \sum_{d\in D} GD_{gdrs} \quad \forall\, g \in G, \forall\, r \in R, \forall\, s \in S \tag{90}$$

$$\sum_{i\in I} QI_{zips} \leq caz_{zp} \quad \forall\, z \in Z, \forall\, p \in P, \forall\, s \in S \tag{91}$$

$$\sum_{i\in I} NI_{uirs} \leq cau_{ur} \quad \forall\, u \in U, \forall\, r \in R, \forall\, s \in S \tag{92}$$

$$\sum_{i\in I} HI_{hils} \leq cax_{hl} \quad \forall\, h \in H, \forall\, l \in L, \forall\, s \in S \tag{93}$$

$$\sum_{a\in A} QM_{amls} \leq cam_{ml} YM_m \quad \forall\, m \in M, \forall\, l \in L, \forall\, s \in S \tag{94}$$

$$\sum_{a\in A} QB_{abs} \leq cab_b YB_b \quad \forall\, b \in B, \forall\, s \in S \tag{95}$$

$$\sum_{a\in A} XG_{agr} + \sum_{b\in B} NG_{bgrs} \leq cag_{gr} YG_g \quad \forall\, g \in G, \forall\, r \in R, \forall\, s \in S \tag{96}$$

$$\sum_{b\in B} ND_{bds} + \sum_{g\in G}\sum_{r\in R} GD_{gdrs} \leq cad_d YD_d \quad \forall\, d \in D, \forall\, s \in S \tag{97}$$

In the two-stage stochastic programming model (64)-(97), the objective function is to maximize the expected profit for all realized quality state scenarios. The objective function is composed of the revenue from selling brand-new products and recovered components and recycled materials in addition to the fixed costs of opening facilities as well as processing, procurement, and shipping costs. Constraint (65)-(66) ensure flow balance at each distribution center and demand satisfaction at each end-user zone. Constraint (67) ensures that all the returned products are collected at the collection centers. Constraint (68) ensures that the total flow to the disassembly facilities, i.e., acquired returns, cannot exceed the total amount of returned products available in collection centers. Constraint (69) ensures that the OEM acquires a substantial portion of the return stream for recovery purposes. This constraint reflects the environmental concerns regarding the harmful effects of leaving used durable products in the environment. Constraint (70) ensures that the total flow outgoing from disassembly centers to all recycling centers is equal to the incoming flow to each disassembly center from all collection centers, multiplied by recyclable mass coefficient α_r. Constraints (71)-(74) are capacity restrictions. Constraints (77)-(79) ensure that the total outgoing flow from each manufacturing center is equal to the total incoming flow into this facility from suppliers and reverse channel. Constraints (80)-(83) ensure flow conservation at each disassembly center. Constraints (84)-(85) ensure the flow conservation at each remanufacturing facility. Constraints (86)-(87) ensure flow conservation at each bulk recycling center. Constraints (88)-(90) are flow conservation restrictions at each material recycling center. Constraints (91)-(97) impose capacity restrictions on supply chain facilities. Constraints (81), (85), and (89) represent partial demand satisfaction of recovered components and recycled raw

materials at secondary markets.

The two-stage stochastic mixed-integer programming model (64)-(97) involves an extremely large number of recourse problems which yield a computationally intractable model. Therefore, a scenario-based reduction scheme is required to reduce the difficulty of solving the underlying model.

3.4 Scenario reduction algorithm

In this section, we provide a detailed exposition of a scenario reduction scheme based on fast forward selection algorithm adapted to the particular structure of the uncertainty set for quality status of returns. The main idea behind the scenario reduction scheme is to preserve the most pertinent scenarios through eliminating the doubtful scenarios to occur. Consequently, it determines the best approximation of the initial set of scenarios with respect to a probability distance measure, i.e., most often Monge-Kantorovich distance.

We let Ω be the probability distribution carried by n-dimensional scenarios $s_i = (\xi_{s_i}^{l_1}, \xi_{s_i}^{l_2}, ..., \xi_{s_i}^{l_L}, \xi_{s_i}^{p_1}, \xi_{s_i}^{p_2}, ..., \xi_{s_i}^{p_P})$ in which l and p indicate, respectively, index of modules and parts. For instance, l_1 denotes the first type of modules while p_3 represents the third type of parts in the reverse BOM. Each scenario s_i is associated with probability π_i for $i = \{1, ..., |S|\}$ such that $\sum_{i=1}^{|S|} \pi_i = 1$. Further, we let $\overline{\Omega}$ be the set of reduced probability distribution, compared to Ω, carried by finitely scenarios $s_j = (\xi_{s_j}^{l_1}, \xi_{s_j}^{l_2}, ..., \xi_{s_j}^{l_L}, \xi_{s_j}^{p_1}, \xi_{s_j}^{p_2}, ..., \xi_{s_j}^{p_P})$ with probability $\overline{\pi_j}$ for $j \in \{1, ..., |S|\} \setminus J$, where J denotes the index set of eliminated scenarios. The minimal Monge-Kantorovich distance between Ω and $\overline{\Omega}$ is then attained as follows (Theorem 2.1. in [43]).

$$D(\Omega, \overline{\Omega}) = \sum_{i \in J} \pi_i . \min_{j \notin J} c(s_i, s_j) \tag{98}$$

and the probability of the preserved scenario s_j of $\overline{\Omega}$, $j \notin J$, is given by the so-called *redistribution rule*:

$$\overline{\pi_j} = \pi_j + \sum_{i \in J_j} \pi_i \tag{99}$$

where $c(s_i, s_j)$ is a distance metric between s_i and s_j and $J_j = \{i \in J : j = j(i)\}$ and $j(i) \in \arg\min_{j \notin J} c(s_i, s_j); \forall i \in J$. The interpretation of the redistribution rule, equation (99), is that the modified value of the probability of a preserved scenario is equal to sum of its initial probability and all probabilities of eliminated scenarios that are closest to it concerning the distance metric c.

The reduction problem (98) states that the initial scenario set involving 2^n number of scenarios is covered by two sets $J \subset \{1, ..., |S|\}$ and $\{1, ..., |S|\} \backslash J$ such that the cover has the minimum value, i.e., $D(\Omega, \overline{\Omega})$. This problem is therefore a set covering problem, which is NP-hard. In [43], an efficient heuristic algorithm based on Monge-Kantorovich distance has been developed to determine the optimal scenario set reduction. The concept of fast forward selection algorithm is the recursive selection of scenarios that will not be eliminated. The first scenario to be preserved is the one that has the minimum sum of the distances to the unselected scenarios. In the subsequent steps, the distance of the unselected scenarios is updated by comparing these scenarios to the selected set. Thus, the sum of the distances of the unselected scenarios is calculated and the next scenario to be preserved is selected akin to the first scenario in the selected set. This algorithm terminates once a specified number (e.g., m) of scenarios has been selected to be preserved. In the last step, the probabilities of each non-selected scenario is added to its closest selected scenario concerning the redistribution rule (99).

The first step of forward selection algorithm involves constructing the distance matrix. Euclidean distance has been a common metric used in stochastic optimization

literature. In this study, the distance norm is adapted to the particular structure of the uncertainty set. More specifically, we differentiate between modules and parts in the reverse BOM of a durable product. The rationale is the different characteristics and economic values residing in these components. Hence, from the practical standpoint, the modified distance matrix better reflects the heterogeneity residing in each quality state scenario compared to a classical distance matrix. To this end, the following modified Euclidean norm is considered as a distance measure between every pair of scenarios s_i and s_j.

$$
\begin{aligned}
c(s_i, s_j) &= ||\xi^{\boldsymbol{l}}_{s_i} - \xi^{\boldsymbol{l}}_{s_j}||_2 + ||\xi^{\boldsymbol{p}}_{s_i} - \xi^{\boldsymbol{p}}_{s_j}||_2 \\
&= \sqrt{(\xi^{l_1}_{s_i} - \xi^{l_1}_{s_j})^2 + ... + (\xi^{l_L}_{s_i} - \xi^{l_L}_{s_j})^2} + \sqrt{(\xi^{p_1}_{s_i} - \xi^{p_1}_{s_j})^2 + ... + (\xi^{p_P}_{s_i} - \xi^{p_P}_{s_j})^2}
\end{aligned}
$$

The fast forward selection algorithm is outlined in Algorithm 4. This algorithm starts with an empty selected set of scenarios and iteratively updates it by adding the scenario minimizing Monge-Kantorovich distance between original and selected sets. In the first step, the distance matrix corresponding to the original set of scenarios is constructed using the modified Euclidean distance metric described above. Then, the minimum distance is computed (line 2 in the algorithm description) and immediately the first scenario to be preserved is identified. Consequently, the set of selected scenarios is updated. In the following steps, the distance matrix is updated (line 4) and distances between scenarios in the original and selected sets are calculated (line 5). The scenario with the minimum distance value is selected as the next scenario not to be eliminated and the selected set is updated. The probability of the scenarios in the selected set is computed using the redistribution rule.

Algorithm 4 - Fast forward selection

1: **Step 1:** $c_{ij}^{[1]} := c(s_i, s_j) \quad \forall i, j \in \{1, ..., |S|\}; i \neq j$

2: $z_j^{[1]} := \sum_{\substack{i=1 \\ i \neq j}}^{|S|} \pi_i c_{ij}^{[1]} \quad \forall j \in \{1, ..., |S|\}$

3: $j_1 \in \underset{j \in \{1,...,|S|\}}{\arg\min}\, z_j^{[1]}, J^{[1]} := \{1, ..., |S|\} \backslash \{j_1\}$

4: **Step** k**:** $c_{ij}^{[k]} := \min\{c_{ij}^{[k-1]}, c_{ij_{k-1}}^{[k-1]}\} \quad \forall i, j \in J^{[k-1]}; i \neq j$

5: $z_j^{[k]} := \sum_{i \in J^{[k-1]} \backslash \{j\}} \pi_i c_{ij}^{[k]} \quad \forall j \in J^{[k-1]}$

6: $j_k \in \underset{j \in J^{[k-1]}}{\arg\min}\, z_j^{[k]}, J^{[k]} := J^{[k-1]} \backslash \{j_k\}$

7: **Step** $m + 1$**:** Applying the redistribution rule (99)

3.5 Solution methodology

The mixed-integer programming model (64)-(97) is a large-scale optimization problem. It is particularly due to several binary decision variables corresponding to location of facilities in the reverse network, the generic reverse BOM of the returned durable product, and the large yet tractable number of reduced recourse problems. This model can be tackled by an efficient solution approach which we devise based on L-shaped algorithm. In classical L-shaped method, the deterministic equivalent (original) problem is decomposed into a MP and a set of RSPs associated with each random scenario defined in the original model. The MP comprises the first stage variables, an artificial variable, and a set of first stage constraints. This problem is the reformulation of the original model which is solved by a cutting plane algorithm such that, at each iteration, whenever a feasible solution to the original problem is found, an optimality cut associated with the set of scenarios are added to the MP. Otherwise, a number of feasibility cuts corresponding to infeasible scenarios are added. The solution to the MP and the expected value of the solutions to the recourse problems gives, respectively, upper and lower bounds to the original problem. The solution process terminates once an optimal solution is found or a prescribed optimality gap

is satisfied.

In what follows, we first provide the classical L-shaped reformulation, then we present algorithmic enhancements in order to speed-up the classical L-shaped algorithm.

3.5.1 L-shaped reformulation

In L-shaped scheme, all the second stage flow variables are projected out and the master problem includes the first stage facility locations and flow variables along with a surrogate variable. We let $\overline{\boldsymbol{X}}$ and $\overline{\boldsymbol{Y}}$ denote a tentative first stage solution. The corresponding RSP for each scenario s can be stated as follows.

$$\text{Max} \quad Q(\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}}, \boldsymbol{\xi}_s) \tag{100}$$

$$\text{s.t.} \quad (81), (84)-(87), (89), (91)-(93)$$

$$\sum_{z\in Z} QI_{zips} + \sum_{a\in A} QZ_{aips} = \phi_p \sum_{j\in J} \overline{XJ_{ij}} \quad \forall\, i \in I, \forall\, p \in P \tag{101}$$

$$\sum_{u\in U} NI_{uirs} + \sum_{g\in G} QU_{girs} = \mu_r \sum_{j\in J} \overline{XJ_{ij}} \quad \forall\, i \in I, \forall\, r \in R \tag{102}$$

$$\sum_{h\in H} HI_{hils} + \sum_{m\in M} QX_{mils} = \omega_l \sum_{j\in J} \overline{XJ_{ij}} \quad \forall\, i \in I, \forall\, l \in L \tag{103}$$

$$\sum_{c\in C} \gamma_{ps} \overline{XA_{ca}} = \sum_{i\in I} QZ_{aips} + \sum_{o\in O} QS_{aops} \quad \forall\, a \in A, \forall\, p \in P \tag{104}$$

$$\sum_{c\in C} \beta_s \overline{XA_{ca}} = \sum_{b\in B} QB_{abs} \quad \forall\, a \in A \tag{105}$$

$$\sum_{c\in C} \delta_{ls} \overline{XA_{ca}} = \sum_{m\in M} QM_{amls} \quad \forall\, a \in A, \forall\, l \in L \tag{106}$$

$$\sum_{b\in B} \tau_r NG_{bgrs} - \sum_{d\in D} GD_{gdrs} = -\sum_{a\in A} \tau_r \overline{XG_{agr}} \quad \forall\, g \in G, \forall\, r \in R \tag{107}$$

$$\sum_{a\in A}\overline{XG_{agr}} = \sum_{i\in I} QU_{girs} + \sum_{e\in E} QE_{gers} + \sum_{d\in D} GD_{gdrs}$$
$$- \sum_{b\in B} NG_{bgrs} \quad \forall\, g \in G, \forall\, r \in R \tag{108}$$

$$\sum_{a\in A} QM_{amls} \le cam_{ml}\overline{YM_m} \quad \forall\, m \in M, \forall\, l \in L \tag{109}$$

$$\sum_{a\in A} QB_{abs} \le cab_b\overline{YB_b} \quad \forall\, b \in B \tag{110}$$

$$\sum_{b\in B} NG_{bgrs} \le cag_{gr}\overline{YG_g} - \sum_{a\in A}\overline{XG_{agr}} \quad \forall\, g \in G, \forall\, r \in R \tag{111}$$

$$\sum_{b\in B} ND_{bds} + \sum_{g\in G}\sum_{r\in R} GD_{gdrs} \le cad_d\overline{YD_d} \quad \forall\, d \in D \tag{112}$$

To formulate the dual of the recourse subproblem, we define $\boldsymbol{\upsilon^1}$,..., $\boldsymbol{\upsilon^{21}}$, particularly in which $\upsilon^5_{ops}, \upsilon^9_{wls}, \upsilon^{13}_{ers}, \upsilon^{15}_{zps}, \upsilon^{16}_{urs}, \upsilon^{17}_{hls}, \upsilon^{18}_{mls}, \upsilon^{19}_{bs}, \upsilon^{20}_{grs}, \upsilon^{21}_{ds} \in \mathbb{R}^+$, as the set of dual variable vectors corresponding to the constraints of RSP. The dual problem for each scenario s, i.e., DRSP, can then be formulated as follows.

$$\begin{aligned}
\text{Min} \quad Z_{\boldsymbol{\upsilon}}(\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}}, \boldsymbol{\xi}_s) = {} & \sum_{i\in I}\sum_{j\in J}\overline{XJ_{ij}}\left(\sum_{p\in P}\phi_p\upsilon^1_{ips} + \sum_{r\in R}\mu_r\upsilon^2_{irs} + \sum_{l\in L}\omega_l\upsilon^3_{ils}\right) \\
& + \sum_{a\in A}\sum_{c\in C}\overline{XA_{ca}}\left(\sum_{p\in P}\gamma_{ps}\upsilon^4_{aps} + \beta_s\upsilon^6_{as} + \sum_{l\in L}\delta_l\upsilon^7_{als}\right) + \sum_{o\in O}\sum_{p\in P} ds_{op}\upsilon^5_{ops} \\
& + \sum_{w\in W}\sum_{l\in L} dw_{wl}\upsilon^9_{wls} + \sum_{g\in G}\sum_{a\in A}\overline{XG_{agr}}\left(\sum_{r\in R}\upsilon^{12}_{grs} + \sum_{r\in R}\upsilon^{14}_{grs}\right) \\
& + \sum_{e\in E}\sum_{r\in R} de_{er}\upsilon^{13}_{ers} + \sum_{z\in Z}\sum_{p\in P} caz_{zp}\upsilon^{15}_{zps} + \sum_{u\in U}\sum_{r\in R} cau_{ur}\upsilon^{16}_{urs} \\
& + \sum_{h\in H}\sum_{l\in L} cax_{hl}\upsilon^{17}_{hls} + \sum_{m\in M}\sum_{l\in L} cam_{ml}\overline{YM_m}\upsilon^{18}_{mls} + \sum_{b\in B} cab_b\overline{YB_b}\upsilon^{19}_{bs} \\
& + \sum_{g\in G}\sum_{r\in R}\left(cag_{gr}\overline{YG_g} - \sum_{a\in A}\overline{XG_{agr}}\right)\upsilon^{20}_{grs} + \sum_{d\in D} cad_d\overline{YD_d}\upsilon^{21}_{ds}
\end{aligned} \tag{113}$$

$$\text{s.t.} \quad (\boldsymbol{\upsilon^1}, \boldsymbol{\upsilon^2}, ..., \boldsymbol{\upsilon^{21}}) \in \Delta_s \tag{114}$$

where Δ_s is the polyhedron defined by the constraints of the DRSP. We let $\rho_s(.)$ denote the first stage variables-free terms in the objective function of the DRSP and we introduce a surrogate variable θ representing an upper bound on the expected recourse function $\mathbb{E}_{\boldsymbol{\xi}}[Q(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\xi})]$. Thus, we can reformulate the master problem of the two-stage stochastic programming model (64)-(97) as follows.

$$
\begin{aligned}
\text{Max} \quad & \theta + \sum_{j\in J}\sum_{k\in K} Pk_k XK_{jk} - \sum_{i\in I}\sum_{j\in J} ci_i XJ_{ij} - \sum_{i\in I}\sum_{j\in J} tj_{ij} XJ_{ij} \\
& - \sum_{j\in J}\sum_{k\in K} cj_j XK_{jk} - \sum_{j\in J}\sum_{k\in K} tk_{jk} XK_{jk} - \sum_{k\in K}\sum_{c\in C} cc_c XC_{kc} \\
& - \sum_{k\in K}\sum_{c\in C} tc_{kc} XC_{kc} - \sum_{c\in C}\sum_{a\in A} ca_a XA_{ca} - \sum_{c\in C}\sum_{a\in A} ta_{ca} XA_{ca} \\
& - \sum_{c\in C}\sum_{a\in A} PrXA_{ca} - \sum_{a\in A}\sum_{g\in G}\sum_{r\in R} cg_{gr} XG_{agr} - \sum_{a\in A}\sum_{g\in G}\sum_{r\in R} tg_{agr} XG_{agr} \\
& - \sum_{c\in C} fc_c YC_c - \sum_{a\in A} fa_a YA_a - \sum_{m\in M} fm_m YM_m - \sum_{b\in B} fb_b YB_b \\
& - \sum_{g\in G} fg_g YG_g - \sum_{d\in D} fd_d YD_d \qquad (115)
\end{aligned}
$$

$$
\text{s.t.} \quad (65) - (74)
$$

$$
\begin{aligned}
& \theta \leq \sum_{s\in S} \pi_s \Bigg(\rho_s(\hat{\boldsymbol{\upsilon}}^{\boldsymbol{n}^T}) + \sum_{i\in I}\sum_{j\in J} XJ_{ij} \Bigg(\sum_{p\in P} \phi_p \hat{\upsilon}^1_{ips} + \sum_{r\in R} \mu_r \hat{\upsilon}^2_{irs} + \sum_{l\in L} \omega_l \hat{\upsilon}^3_{ils} \Bigg) \\
& + \sum_{a\in A}\sum_{c\in C} XA_{ca} \Bigg(\sum_{p\in P} \gamma_{ps} \hat{\upsilon}^4_{aps} + \beta_s \hat{\upsilon}^6_{as} + \sum_{l\in L} \delta_l \hat{\upsilon}^7_{als} \Bigg) \\
& + \sum_{g\in G}\sum_{a\in A} XG_{agr} \Bigg(\sum_{r\in R} \hat{\upsilon}^{12}_{grs} + \sum_{r\in R} \hat{\upsilon}^{14}_{grs} \Bigg) + \sum_{m\in M}\sum_{l\in L} cam_{ml} YM_m \hat{\upsilon}^{18}_{mls} \\
& + \sum_{b\in B} cab_b YB_b \hat{\upsilon}^{19}_{bs} + \sum_{g\in G}\sum_{r\in R} \Bigg(cag_{gr} YG_g - \sum_{a\in A} XG_{agr} \Bigg) \hat{\upsilon}^{20}_{grs} + \sum_{d\in D} cad_d YD_d \hat{\upsilon}^{21}_{ds} \Bigg)
\end{aligned} \qquad (116)
$$

$$
0 \leq \rho_{s'}(\hat{\boldsymbol{\kappa}}^{\boldsymbol{n}^T}) + \sum_{i\in I}\sum_{j\in J} XJ_{ij} \Bigg(\sum_{p\in P} \phi_p \hat{\kappa}^1_{ips'} + \sum_{r\in R} \mu_r \hat{\kappa}^2_{irs'} + \sum_{l\in L} \omega_l \hat{\kappa}^3_{ils'} \Bigg)
$$

$$+\sum_{a\in A}\sum_{c\in C} XA_{ca}\left(\sum_{p\in P}\gamma_{ps}\hat{\kappa}^{4}_{aps'}+\beta_s\hat{\kappa}^{6}_{as'}+\sum_{l\in L}\delta_l\hat{\kappa}^{7}_{als'}\right)$$
$$+\sum_{g\in G}\sum_{a\in A} XG_{agr}\left(\sum_{r\in R}\hat{\kappa}^{12}_{grs'}+\sum_{r\in R}\hat{\kappa}^{14}_{grs'}\right)+\sum_{m\in M}\sum_{l\in L} cam_{ml}YM_m\hat{\kappa}^{18}_{mls'}$$
$$+\sum_{b\in B} cab_bYB_b\hat{\kappa}^{19}_{bs'}+\sum_{g\in G}\sum_{r\in R}\left(cag_{gr}YG_g-\sum_{a\in A}XG_{agr}\right)\hat{\kappa}^{20}_{grs'}$$
$$+\sum_{d\in D} cad_dYD_d\hat{\kappa}^{21}_{ds'}\quad \forall s'\in S \tag{117}$$
$$\boldsymbol{X}\in\mathbb{R}^{+},\boldsymbol{Y}\in\{0,1\} \tag{118}$$

where $\boldsymbol{\kappa}$ indicates extreme rays of Δ whenever the DRSP is unbounded for a given first stage solution in scenario s'. As shown above, the set of optimality cuts for each scenario s has been aggregated to produce the optimality cut (115), while (116) represents the feasibility cut for each infeasible scenario s'.

We also let F and C be the cost vectors in the objective function of the first stage problem. The classical L-shaped method is therefore outlined in Algorithm 5. Pilot computational tests have shown a slow convergence of L-shaped method. To circumvent this issue, we proceed with proposing two different enhancement strategies in the next section.

3.5.2 Algorithmic refinements

3.5.2.1 Induced constraints

We note that various types of valid inequalities (induced constraints) can be added to the MP. In the early iterations of L-shaped algorithm, because the iterative algorithm is initialized from empty subsets of extreme rays and extreme points, the solution to the MP might be infeasible for the original model, which leads to the generation of the feasibility cuts. Induced constraints restrict the feasible region of the MP and transfer

Algorithm 5 - Classical L-shaped algorithm

$UB \leftarrow \infty$, $LB \leftarrow -\infty$
while $(UB - LB)/UB \leq \epsilon$ **do**
Solve the MP
$UB \leftarrow \bar{\theta} - F^T\overline{\boldsymbol{Y}} + C^T\overline{\boldsymbol{X}}$
Solve the DRSP for each scenario s
if the DRSP is unbounded **then**
Add the corresponding feasibility cut (116) to the MP
else
Generate the corresponding optimality cut
end if
if The DRSP was optimal for all realiztion of scenarios **then**
Add the aggregated optimality cut (115) to the MP
$LB \leftarrow \max\ (LB, \sum_s \pi_s Z_{\boldsymbol{v}}(\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}}, \boldsymbol{\xi}_s) + F^T\overline{\boldsymbol{Y}} + C^T\overline{\boldsymbol{X}})$
end if
end while

more information on RSPs to the MP. Hence, they dramatically diminish the number of feasibility cuts throughout the solution process and enhance the convergence of L-shaped method by helping the MP to identify close to optimal solutions. Given the model (64)-(97) described in the preceding section, the following constraints can be added to the MP as induced constraints.

$$\sum_{c\in C} cac_c YC_c \geq \sum_{k\in K} \psi dk_k \tag{119}$$

$$\sum_{a\in A} caa_a YA_a \geq \lambda\psi \sum_{k\in K} dk_k \tag{120}$$

$$\sum_{m\in M} cam_{ml} YM_m \geq \delta_{ls}\lambda\psi \sum_{k\in K} dk_k \quad \forall\, l \in L, \forall\, s \in S \tag{121}$$

$$\sum_{b\in B} cab_b YB_b \geq \beta_s\lambda\psi \sum_{k\in K} dk_k \quad \forall\, s \in S \tag{122}$$

$$\sum_{g\in G} cag_{gr} YG_g \geq \alpha_r\lambda\psi \sum_{k\in K} dk_k \quad \forall\, r \in R \tag{123}$$

$$\sum_{d\in D} YD_d \geq 1 \tag{124}$$

The set of inequalities (119)-(123) ensure installing enough capacity in the reverse channel. Constraint (124) ensures that at least one disposal center is installed in the reverse network.

3.5.2.2 Pareto-optimal cuts

One of the well-known strategies to enhance the convergence of L-shaped algorithm is to strengthen the deepness of the optimality cuts. In some applications, multiple optimal solutions might exist to the DRSP, each providing a potentially different cut. To choose the deepest cut among various optimality cuts corresponding to the multiple optimal solutions, Magnanti and Wong [30] proposed a cut selection scheme, which improves the convergence of the Benders decomposition algorithm. In the context of our problem of interest, the definition of a stronger cut can be expressed as follows.

Definition 3.1. Given that $\boldsymbol{X}$ and $\boldsymbol{Y}$ represent, respectively, the set of first stage flows and locations variables in model (64)-(97), the optimality cut generated from the dual solution vectors $(\boldsymbol{v_1^1}, ..., \boldsymbol{v_1^7}, \boldsymbol{v_1^9}, \boldsymbol{v_1^{12}}, ..., \boldsymbol{v_1^{21}}) \in \Delta_s$ dominates the cut generated from $(\boldsymbol{v_2^1}, ..., \boldsymbol{v_2^7}, \boldsymbol{v_2^9}, \boldsymbol{v_2^{12}}, ..., \boldsymbol{v_2^{21}}) \in \Delta_s$ if and only if

$$
\begin{aligned}
&\sum_{s\in S}\pi_s\Bigg(\rho_s(\hat{\boldsymbol{v}}_{\mathbf{1}}^{\boldsymbol{n}^T}) + \sum_{i\in I}\sum_{j\in J} XJ_{ij}\Bigg(\sum_{p\in P}\phi_p\hat{v}^1_{1ips} + \sum_{r\in R}\mu_r\hat{v}^2_{1irs} + \sum_{l\in L}\omega_l\hat{v}^3_{1ils}\Bigg)\\
&+ \sum_{a\in A}\sum_{c\in C} XA_{ca}\Bigg(\sum_{p\in P}\gamma_{ps}\hat{v}^4_{1aps} + \beta_s\hat{v}^6_{1as} + \sum_{l\in L}\delta_l\hat{v}^7_{1als}\Bigg)\\
&+ \sum_{g\in G}\sum_{a\in A} XG_{agr}\Bigg(\sum_{r\in R}\hat{v}^{12}_{1grs} + \sum_{r\in R}\hat{v}^{14}_{1grs}\Bigg) + \sum_{m\in M}\sum_{l\in L} cam_{ml}YM_m\hat{v}^{18}_{1mls}\\
&+ \sum_{b\in B} cab_bYB_b\hat{v}^{19}_{1bs} + \sum_{g\in G}\sum_{r\in R}\Bigg(cag_{gr}YG_g - \sum_{a\in A}XG_{agr}\Bigg)\hat{v}^{20}_{1grs} + \sum_{d\in D} cad_dYD_d\hat{v}^{21}_{1ds}\Bigg)\\
&\leq \sum_{s\in S}\pi_s\Bigg(\rho_s(\hat{\boldsymbol{v}}_{\mathbf{2}}^{\boldsymbol{n}^T}) + \sum_{i\in I}\sum_{j\in J} XJ_{ij}\Bigg(\sum_{p\in P}\phi_p\hat{v}^1_{2ips} + \sum_{r\in R}\mu_r\hat{v}^2_{2irs} + \sum_{l\in L}\omega_l\hat{v}^3_{2ils}\Bigg)
\end{aligned}
$$

$$+\sum_{a\in A}\sum_{c\in C} XA_{ca}\left(\sum_{p\in P}\gamma_{ps}\hat{\upsilon}^{4}_{2aps}+\beta_{s}\hat{\upsilon}^{6}_{2as}+\sum_{l\in L}\delta_{l}\hat{\upsilon}^{7}_{2als}\right)$$
$$+\sum_{g\in G}\sum_{a\in A} XG_{agr}\left(\sum_{r\in R}\hat{\upsilon}^{12}_{2grs}+\sum_{r\in R}\hat{\upsilon}^{14}_{2grs}\right)+\sum_{m\in M}\sum_{l\in L} cam_{ml}YM_{m}\hat{\upsilon}^{18}_{2mls}$$
$$+\sum_{b\in B} cab_{b}YB_{b}\hat{\upsilon}^{19}_{2bs}+\sum_{g\in G}\sum_{r\in R}\left(cag_{gr}YG_{g}-\sum_{a\in A}XG_{agr}\right)\hat{\upsilon}^{20}_{2grs}+\sum_{d\in D} cad_{d}YD_{d}\hat{\upsilon}^{21}_{2ds}\Bigg)$$

for all $\boldsymbol{X}$ and $\boldsymbol{Y}$ with strict inequality for at least one extreme point. A Pareto-optimal cut is not dominated by any other cut. Now, let Λ be a polyhedron stated as $\Lambda = \{(\boldsymbol{X},\boldsymbol{Y}) : (65)-(74) \quad \text{are satisifed}\}$.

Definition 3.2. Core point: any point $(\boldsymbol{X^0},\boldsymbol{Y^0})$ contained in the relative interior of the convex hull of Λ is said to be a core point, i.e., $(\boldsymbol{X^0},\boldsymbol{Y^0}) \in ri(\Lambda^c)$, in which $ri(.)$ and Λ^c, respectively, denote the relative interior and the convex hull of Λ.

Given the above definitions, the Pareto-cut selection scheme based on [30] for the CLSC network design problem under investigation is presented in the *Appendices*, in the section entitled *Magnanti and Wong problem*. In this study, we use the Pareto-optimal cut generation approach presented in Papadakos [9]. In [9], it has been shown that the normalization constraint (see constraint (127)) can be disregarded through varying the value of the core point at each iteration of the solution process. In this approach, once the solution to the MP yields feasible RSPs, the value of the core point can be updated through the convex combination of the MP solution and the previous value of the core point. In this regard the auxiliary dual recourse subproblem (auxiliary-DRSP) can be stated as follows.

$$\text{Min} \quad Z_{\boldsymbol{\upsilon}}(\boldsymbol{X^0},\boldsymbol{Y^0},\boldsymbol{\xi}_s)=\sum_{i\in I}\sum_{j\in J} XJ^{0}_{ij}\left(\sum_{p\in P}\phi_{p}\upsilon^{1}_{ips}+\sum_{r\in R}\mu_{r}\upsilon^{2}_{irs}+\sum_{l\in L}\omega_{l}\upsilon^{3}_{ils}\right)$$
$$+\sum_{a\in A}\sum_{c\in C} XA^{0}_{ca}\left(\sum_{p\in P}\gamma_{ps}\upsilon^{4}_{aps}+\beta_{s}\upsilon^{6}_{as}+\sum_{l\in L}\delta_{l}\upsilon^{7}_{als}\right)+\sum_{o\in O}\sum_{p\in P} ds_{op}\upsilon^{5}_{ops}$$

$$
\begin{aligned}
&+ \sum_{w\in W}\sum_{l\in L} dw_{wl} v^{9}_{wls} + \sum_{g\in G}\sum_{a\in A} XG^{0}_{agr}\left(\sum_{r\in R} v^{12}_{grs} + \sum_{r\in R} v^{14}_{grs}\right) \\
&+ \sum_{e\in E}\sum_{r\in R} de_{er} v^{13}_{ers} + \sum_{z\in Z}\sum_{p\in P} caz_{zp} v^{15}_{zps} + \sum_{u\in U}\sum_{r\in R} cau_{ur} v^{16}_{urs} \\
&+ \sum_{h\in H}\sum_{l\in L} cax_{hl} v^{17}_{hls} + \sum_{m\in M}\sum_{l\in L} cam_{ml} YM^{0}_{m} v^{18}_{mls} + \sum_{b\in B} cab_{b} YB^{0}_{b} v^{19}_{bs} \\
&+ \sum_{g\in G}\sum_{r\in R}\left(cag_{gr} YG^{0}_{g} - \sum_{a\in A} XG^{0}_{agr}\right) v^{20}_{grs} + \sum_{d\in D} cad_{d} YD^{0}_{d} v^{21}_{ds} \\
\text{s.t.}\quad & (\boldsymbol{v^{1}}, \boldsymbol{v^{2}}, ..., \boldsymbol{v^{21}}) \in \Delta_{s} \qquad (125)
\end{aligned}
$$

The optimal solution to auxiliary-DRSP (124) is used to generate the Pareto-optimal cut. We let non-negative parameter λ^{c} indicate the weight of the core point ($\boldsymbol{X^{0}}$, $\boldsymbol{Y^{0}}$) in the convex combination that updates the value of the core point throughout the solution process. Empirically, it has been shown that 0.5 most often yields the best results ([9]). An outline of the proposed enhanced L-shaped method is presented in Algorithm 6.

3.6 Numerical results

In this section, we illustrate some numerical experiments to provide an analysis of the CLSC network design problem under investigation. To this end, first, we address a typical large household appliance example, i.e., washing machine, as a suitable case of durable products and provide a description of randomly generated data sets based on the CLSC/RSC design literature ([19], [20]), which ensures varying values of input parameters. Then, we proceed with a detailed representation of the performance of the proposed solution algorithm on two reduced sets of scenarios with different sizes, i.e., 500 and 1000, respectively. Finally, using the enhanced L-shaped algorithm for each scenario set, we evaluate the performance of the scenario reduction scheme.

Algorithm 6 - Enhanced L-shaped algorithm

$UB \leftarrow \infty$, $LB \leftarrow -\infty$, $\lambda^c \leftarrow 0.5$
Add induced constraints (118)-(123) to the MP
Start with an initial core point $(\boldsymbol{X^0}, \boldsymbol{Y^0})$
while $(UB - LB)/UB \leq \epsilon$ **do**
Solve auxiliary-DRSP (124) for each scenario s
Add the aggregated Pareto-optimal cut (115) to the MP
Solve the MP
$UB \leftarrow \bar{\theta} - F^T\overline{\boldsymbol{Y}} + C^T\overline{\boldsymbol{X}}$
Solve the DRSP for each scenario s
if the DRSP is unbounded **then**
    Add the corresponding feasibility cut (116) to the MP
else
    Generate the corresponding optimality cut
end if
if The DRSP was optimal for all realiztion of scenarios **then**
    Add the aggregated optimality cut (115) to the MP
    $LB \leftarrow \max\,(LB, \sum_s \pi_s Z_{\boldsymbol{v}}(\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}}, \boldsymbol{\xi}_s) + F^T\overline{\boldsymbol{Y}} + C^T\overline{\boldsymbol{X}})$
    $(\boldsymbol{X^0}, \boldsymbol{Y^0}) \leftarrow \lambda^c(\boldsymbol{X^0}, \boldsymbol{Y^0}) + (1 - \lambda^c)(\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}})$
else
    $(\boldsymbol{X^0}, \boldsymbol{Y^0}) \leftarrow \lambda^c(\boldsymbol{X^0}, \boldsymbol{Y^0}) + \zeta$
end if
end while

The fast forward selection algorithm and the accelerated L-shaped method are implemented in C++ programming language. More particularly, the proposed decomposition algorithm is implemented in C++ using Concert Technology with IBM-ILOG CPLEX 12.60. We also employ the default settings of CPLEX and conduct all the experiments on an Intel Quad Core 3.40 GHz with 8 GB RAM.

3.6.1 Computational experiments

We consider the recovery network of a typical washing machine that has been adapted from Park et al. [36]. This washing machine entails ten parts (e.g., washing tube) and two modules (e.g., motor) as components and three types of raw materials (e.g., steel). The number of each component in addition to the volume of each raw material are

shown in Table 10. For instance, this washing machine contains a single washing tube which is 3.5 kg in weight. Recalling the scenario generation approach described in Section 3.2 and the number of components of the washing machine in our experiments, i.e., twelve components, the grading process leads to 2^{12}, equally 4096, quality state scenarios which is relatively large. Moreover, the return ratio parameter, i.e., ψ, is 60%.

Moreover, the capacities of facilities in the forward network are randomly generated considering the first market demand and the value of parameters given in Table 10. The same rule applies for the capacities of facilities in the reverse network with a few exceptions such that their values are generated according to the reverse BOM, demands, and the return ratio. For example, the capacity of collection centers are chosen between $Uniform(3 \times MeanCac, 5 \times MeanCac)$ where $MeanCac = \sum_k \psi dk_k/|C|$. Demands of recycled raw materials at the corresponding secondary marketplaces are also randomly generated considering the reverse BOM and the rate of return. Finally, shipping costs are selected from $Uniform(4, 7)$ for the washing machine, $Uniform(1, 4)$ for each type of components, and $Uniform(0.1, 0.5)$ for raw materials, bulk of residues, and waste.

In the *Appendices*, in the section entitled *Parameter settings*, the following parameters of the proposed mathematical formulation are presented in separated tables. The revenues from selling the new washing machine, recovered components, and recycled materials at the corresponding markets are carefully estimated vis-à-vis recent market data. The unit procurement costs for each new component and raw material are assumed to be halves of the unit prices. Further, using the values presented in [20] and [19], the processing costs and other parameters, such as demands, the rate of return, and fixed costs of opening facilities are also given in the *Appendices*. We note that fixed costs are generated with regard to the capacity of facilities, i.e., opening a

facility with high capacity level requires to invest a greater amount of infrastructural costs.

Table 10: Components and raw materials of the case example

Description	Value
ϕ_p	washing tube:1 (3.5 kg), cover:1 (2.5 kg), balance:1 (2.5 kg), frame:1 (11.5 kg), condenser:1 (0.5 kg), hose:1 (1 kg), small electric parts:1 (1 kg), electric wire:1 (1 kg), transformer:1 (1 kg), PCB:1 (0.5 kg)
ω_l	motor:1 (5 kg), clutch:1 (4 kg)
μ_r	plastic:3 kg, steel:2 kg, copper:1 kg

We also apply fast forward selection algorithm described in an earlier section where two reduced scenario sets of sizes 500 and 1000 are selected out of the set of 4096 scenarios. Then, we consider five classes of problems, each with 5 randomly generated test instances, for both sets of scenarios as shown in Table 11. We also show detailed information on the size of classes in Table 12.

3.6.2 Analysis of the enhanced L-shaped algorithm

To assess the computational efficiency of the proposed enhanced L-shaped method, we define an optimality gap $\epsilon\%$ in addition to a time limit as the stopping criteria

Table 11: Problem classes

Class	$\|Z\|$	$\|U\|$	$\|H\|$	$\|I\|$	$\|J\|$	$\|K\|$	$\|C\|$	$\|A\|$	$\|M\|$	$\|B\|$	$\|G\|$	$\|D\|$	$\|O\|$	$\|W\|$	$\|E\|$	$\|S\|$
C1	10	3	2	5	10	60	10	10	10	10	10	5	30	30	30	500
																1000
C2	10	3	2	5	10	80	10	10	10	10	10	5	40	40	40	500
																1000
C3	10	3	2	5	15	100	15	15	15	15	15	7	50	50	50	500
																1000
C4	10	3	2	5	15	120	15	15	15	15	15	7	60	60	60	500
																1000
C5	10	3	2	5	20	140	20	20	20	20	20	10	70	70	70	500
																1000

Table 12: Size of the deterministic equivalent problems

Class	$\|S\|$	Constraints	Continuous Vars.	Binary Vars.
C1	500	476706	3261650	55
	1000	953206	6521650	55
C2	500	551746	4012050	55
	1000	1103246	8022050	55
C3	500	705326	7276475	82
	1000	1410326	14548975	82
C4	500	780366	8402075	82
	1000	1560366	16799575	82
C5	500	934446	13022300	110
	1000	1868446	26037300	110

for this algorithm. More precisely, the solution process terminates once either the optimality gap falls below 0.5% or the solution time exceeds 3600 seconds. As for the core point $(\boldsymbol{X^0}, \boldsymbol{Y^0})$, point $\boldsymbol{Y^0}$ is fixed to 0.5 for all the first stage binary variables at the beginning of the solution approach. Moreover, to pick a suitable value for point $\boldsymbol{X^0}$, after fixing the value of $\boldsymbol{Y^0}$ to 0.5, the resulting problem (64)-(74) (excluding the recourse function) is solved imposing a small positive value as the lower bound for $\boldsymbol{X^0}$ to ensure generating an interior point. We also solve all 50 test instances with CPLEX in a maximum time limit of 18000 seconds and within the stopping gap tolerance of 0.5% to avoid tail-off effect. Tables 13 and 14 present, respectively, computational results for the reduced sets of 500 and 1000 scenarios. These tables also show computational statistics of CPLEX including CPU time in seconds followed by the value of the objective function reported by CPLEX within the dedicated time limit. The last column entitled by "Gap" represents the relative difference between the values of the objective function reported by the enhanced L-shaped algorithm and CPLEX for each test instance within their dedicated time limits.

$$Gap = 100 \times |(V_{2SP} - CPX_{2SP})|/V_{2SP}$$

Table 13: Computational results on problem classes for $|S| = 500$

Class	Enhanced L-shaped			CPLEX		
	Runtime (sec)	Iterations	Profit	Runtime (sec)	Profit	Gap (%)
C1	320	22	23,583,400	$\geq$ 18000	23,524,200	0.25
	170	12	26,655,700	$\geq$ 18000	26,460,800	0.73
	280	19	25,156,500	$\geq$ 18000	24,860,600	1.18
	269	19	24,681,000	$\geq$ 18000	24,634,700	0.19
	244	17	26,100,800	$\geq$ 18000	26,097,100	0.01
C2	443	27	35,837,400	$\geq$ 18000	35,460,700	1.05
	527	32	34,890,800	$\geq$ 18000	34,703,900	0.54
	224	14	36,404,200	$\geq$ 18000	35,984,800	1.15
	278	17	34,448,900	$\geq$ 18000	34,297,300	0.44
	523	31	37,958,400	$\geq$ 18000	37,771,900	0.49
C3	969	32	44,375,400	$\geq$ 18000	4,181,020	90.58
	521	17	40,886,900	$\geq$ 18000	1,050,180	97.43
	1119	36	45,205,700	$\geq$ 18000	6,253,750	86.17
	704	24	42,834,000	$\geq$ 18000	3,127,920	92.70
	531	18	43,602,900	$\geq$ 18000	5,294,590	87.86
C4	1382	36	51,663,200	$\geq$ 18000	10,866,000	78.97
	1154	31	56,215,500	$\geq$ 18000	15,416,000	72.58
	763	21	58,737,500	$\geq$ 18000	19,074,500	67.52
	764	21	54,991,600	$\geq$ 18000	17,596,300	68.00
	823	23	55,213,400	$\geq$ 18000	17,773,400	67.81
C5	2835	38	65,911,400	$\geq$ 18000	12,020,400	81.76
	2900	41	60,880,800	$\geq$ 18000	9,697,270	84.07
	2348	31	63,561,500	$\geq$ 18000	12,181,600	80.83
	2680	33	58,883,900	$\geq$ 18000	4,692,950	92.03
	2530	36	62,967,200	$\geq$ 18000	11,292,800	82.06

where V_{2SP} and CPX_{2SP} indicate, respectively, the optimal solution of model (64)-(97), which is obtained by applying the accelerated L-shaped method and CPLEX to the reduced sets of scenarios.

As far as the results reported in Table 13 are concerned, the proposed L-shaped method solves all 25 test instances of different sizes to optimality in a reasonable amount of time while CPLEX is only able to find a feasible solution for them. In particular, the feasible solution identified by CPLEX for test problems of C3 to C5

Table 14: Computational results on problem classes for $|S| = 1000$

Class	Enhanced L-shaped			CPLEX		
	Runtime (sec)	Iterations	Profit	Runtime (sec)	Profit	Gap (%)
C1	598	21	23,722,300	$\geq$ 18000	23,687,100	0.15
	331	12	26,802,200	$\geq$ 18000	26,573,900	0.85
	580	19	25,299,200	$\geq$ 18000	24,629,600	2.65
	556	18	24,823,600	$\geq$ 18000	24,660,100	0.66
	455	16	26,242,400	$\geq$ 18000	25,686,000	2.12
C2	857	26	36,006,600	$\geq$ 18000	35,746,500	0.72
	900	27	35,106,500	$\geq$ 18000	34,721,000	1.10
	460	14	36,585,700	$\geq$ 18000	36,173,300	1.13
	580	17	34,634,300	$\geq$ 18000	34,266,800	1.06
	1324	39	38,117,700	$\geq$ 18000	37,964,300	0.40
C3	1848	30	44,619,300	$\geq$ 18000	4,142,150	90.71
	1154	19	41,121,600	$\geq$ 18000	No solution	-
	1720	28	45,440,300	$\geq$ 18000	No solution	-
	1438	24	43,077,100	$\geq$ 18000	3,070,370	92.87
	970	16	43,835,900	$\geq$ 18000	5,170,890	88.20
C4	1944	26	51,663,200	$\geq$ 18000	15,081,600	70.81
	2472	33	56,215,500	$\geq$ 18000	18,885,000	66.41
	1663	22	58,737,500	$\geq$ 18000	21,676,300	63.10
	1768	23	54,991,600	$\geq$ 18000	17,898,900	67.45
	1588	22	55,213,400	$\geq$ 18000	17,942,400	67.50
C5	$\geq$ 3600	26	66,258,600	$\geq$ 18000	M	-
	$\geq$ 3600	27	65,256,200	$\geq$ 18000	M	-
	$\geq$ 3600	25	63,761,100	$\geq$ 18000	M	-
	$\geq$ 3600	25	59,212,300	$\geq$ 18000	M	-
	$\geq$ 3600	27	63,307,000	$\geq$ 18000	M	-

M: out-of-memory

are considerably far from the optimal solution given by the L-shaped method observing the huge average gap between the values of the objective function reported by two methods as indicated in the last column. This can be explained by the fact that the deterministic equivalent problem which CPLEX attempts to solve involves a large number of recourse problems associated with the representative quality state scenarios. Hence, model (64)-(97), even with reduced number of scenarios, is itself a very difficult to solve problem for the commercial software. This observation supports the call for an efficient solution approach. As opposed to CPLEX that is unable to find high quality solutions within 5h CPU time, the proposed enhanced L-shaped algorithm can easily handle realistic size problems such that the average runtime in classes C1 and C2 is less than 600 seconds, which verifies the advantage of the algorithmic refinement strategies. Note that, in the case of the largest test problems (class C5), we observe, on average, a 77.42% solution time increase compared to the other four classes. Nonetheless, all test instances are solvable in the allotted time by the proposed decomposition scheme.

Likewise, Table 14 reports the same statistics in the case of 1000 scenarios. Analysis of this table leads to similar implications as in the case of 500 scenarios. However, concerning the size of the set of scenarios, i.e., 1000, few exceptions arise that are required to be clarified. Firstly, the runtime for the accelerated L-shaped method increases 12.77% on average over all test instances excluding the fifth class. Secondly, the test instances of the last class cannot be solved within one hour until the dedicated optimality gap, i.e., $\epsilon = 0.5\%$, is reached by the solution algorithm. However, for this class of problems, the relative difference between the lower and upper bounds of the solution process after an hour is quite tight (0.54% on average over test instances of C5), which is not far from the stopping optimality gap of 0.5%. Moreover, when the computational results of CPLEX are considered, we observe an out-of-memory state

in the last class. Note that, in two test instances of the third class, CPLEX cannot generate any feasible solution within five hours, denoted by "No solution". The above discussion demonstrates the effectiveness of the proposed accelerated L-shaped algorithm for solving our problem of interest.

Moreover, we note that due to the undesirable performance of the classical L-shaped algorithm on pilot tests in terms of solution time, its numerical results have been excluded from this section. For example, the CPU time of the classical L-shaped method when it applies to solve the first test instance of C1 in Table 13 is 1950 seconds. This is further illustrated in Figure 9 through observing the plotted convergence of the classical (C-Lshaped) and enhanced (E-Lshaped) algorithms on solving this test instance. As shown in Figure 9, the number of iterations to achieve optimality has been reduced by 77% in the enhanced L-shaped algorithm. In this typical example, the enhanced variant converges to the optimal solution in 17% of the total solution time required for the classical method.

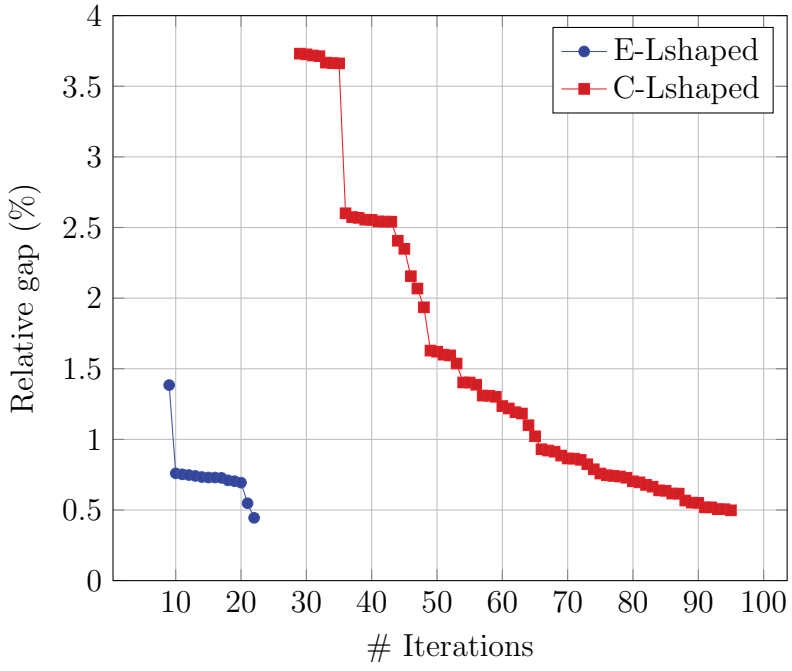

Figure 9: Convergence of the gap for the 1^{st} instance of C1

Finally, from the network configuration point of view, investigating the computational results in both reduced scenario sets indicates that the facilities installed in the reverse channel can be classified in two categories in terms of the number of active facilities. As the first group of facilities, collection, remanufacturing, and bulk recycling centers account for, on average, 60% of the facilities installed in the CLSC. While, as the second group, disassembly, material recycling, and disposal centers are, on average, 40% of the active facilities in the reverse network of the CLSC. It should be noted that, for each type of the recovery facilities, the number of facilities that are opened in the reverse channel increases in accordance with the size of the test instance. Furthermore, the total number of components, i.e., parts and modules, expected to be processed in the reverse network increases averagely 54% and 56%, respectively, in the set of 1000 scenarios compared to 500 ones for each class of problems. In contrary, there exists a smooth decline, i.e., 5%, in the total volume of residues needed to be treated in bulk recycling centers.

3.6.3 Sensitivity analysis

Observe that the quantity of returned products can be also subject to change. Hence, we analyze the sensitivity of solutions to the variations in the value of the return ratio, i.e., ψ, for the first test instance of C2 in Tables 13 and 14. To this end, we assume that the rate of return is normally distributed with the mean value of 0.60 and the variance of 1% mean over all end-user zones. This normal distribution is then approximated by a 3-point discrete distribution (high, average, and low ratio) by using the Gaussian quadrature method [49]. The average point is the return ratio of 0.60 which is considered as the base case. The lower and the upper cases are also solved to optimality by means of the proposed solution algorithm.

As shown in Figure 10, when 500 scenarios are concerned, the higher level of the

return ratio leads to an increase in the number of various types of facilities opened in the reverse network, which in turn incurs additional set-up costs. It also increases the total processing and shipping costs of the reverse chain. On the other hand, the lower level of the return ratio has an opposite impact on the configuration the CLSC network as it results in smaller number of active facilities in the reverse channel. Investigating the results of 1000 scenarios instance indicates a similar observation. The implication is that the level of the return stream plays a key role on the profitability of the CLSC in terms of changing the value of the total cost of designing the network. More precisely, the profit that the OEM gains in the lower return ratio case is of 6.75% and 13.53% greater than the base and the upper cases when $|S| = 500$. We note that in the case of $|S| = 1000$, the increase in the value of the profit function is slightly greater than the smaller-sized scenario set.

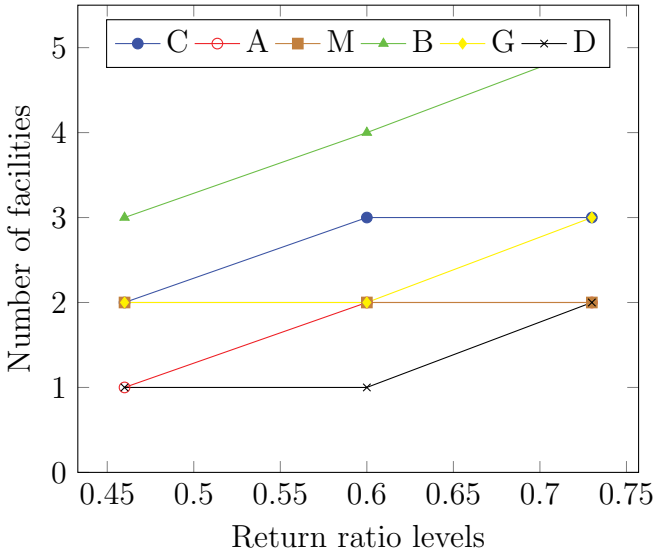

Figure 10: Impact of different return ratio levels on the CLSC configuration

3.6.4 Analysis of fast forward selection algorithm

Herein, we evaluate the performance of fast forward selection algorithm through comparing the numerical results of the reduced scenario sets presented in the previous

section. To this end, we compare the values of the expected profit obtained by considering reduced sets of scenarios with the one that we estimate through considering all possible scenarios for the random quality status of the return stream. Recall that V_{2SP} indicates the optimal solution of model (64)-(97), which is obtained by using the enhanced L-shaped algorithm. Now, the first stage variables in (64)-(97) are substituted with the corresponding optimal values obtained by applying the proposed L-shaped algorithm for the reduced two-stage stochastic program. Then, the resulting recourse subproblems are solved for all 4096 quality state scenarios. The expected value of the profit function is therefore calculated over all scenarios and denoted by EV_{2SP}. The relative difference (RD) measure, representing the gap between EV_{2SP} and V_{2SP}, can be stated as follows.

$$RD = 100 \times |(V_{2SP} - EV_{2SP})|/V_{2SP}$$

It should be noted that the RD measure indicates how good the reduced scenario set, obtained by the proposed fast forward selection algorithm, represents the whole set of scenarios. The steps described above are repeated for each test instance of our experiments. Table 15 summarizes the average value of RD over all five test instances for each class of problems, in cases of 500 and 1000 scenarios.

Table 15: The average value of RD

# Scenarios	C1	C2	C3	C4	C5
500	1.25%	1.17%	1.21%	1.14%	1.19%
1000	0.69%	0.65%	0.67%	0.71%	0.70%

As can be seen Table 15, the RD values are quite insignificant, i.e., less than 2%. It means that both reduced sets of 500 and 1000 scenarios provide good quality solutions to our stochastic problem. This observation verifies the capability of fast forward selection algorithm in finding reduced sets of scenarios, that are reliable

representations of 4096 quality state scenarios.

Moreover, analyzing these results, it can be inferred that the larger reduced scenario set, i.e., 1000 scenarios, provides a better approximation to the true stochastic CLSC network design problem involving a large number of scenarios. However, it should be remembered that, in the case of 1000 scenarios, the average CPU time increases by 12.77% over solvable test instances against that of 500 scenarios. As evident from the above discussion, if the decision maker prefers to obtain more accurate solutions to model (64)-(97), he/she should pay the cost of computational time. The proposed solution framework along with the scenario reduction method allows the decision maker to have a perfect insight on possible trade-offs between computational time and the accuracy of solutions, and therefore to select a solution that is consistent with his/her willingness to invest computational efforts to design the CLSC network.

3.7 Conclusion

In this paper, we introduced a CLSC network design problem under uncertain quality status of the return stream, which is applicable to the case of durable products. The underlying uncertainty is considered as the availability of each component in the reverse BOM and modeled as discrete scenarios with Bernoulli probability distribution. Accordingly, we proposed a two-stage mixed-integer stochastic program to explicitly address uncertainty in this problem. In order to tackle the intractable number of scenarios resulting from several components that exist in the reverse BOM of a typical durable product, we adapted fast forward selection algorithm to our problem of interest to preserve the most pertinent binary scenarios in the deterministic equivalent problem. Moreover, we developed a solution method based on L-shaped algorithm, further enhanced with additional acceleration strategies, including valid inequalities and non-dominated optimality cuts. Our computational experiments demonstrated

an outstanding capability of the proposed algorithm for the CLSC network design problem.

We believe that our numerical results are general in nature and remain valid in the context of any durable product. The proposed solution framework together with the employed scenario reduction technique can be used to solve realistic-sized problems. More specifically, the computational experiments performed show that the average solution time are 1012 and 1962 seconds, receptively, for cases of 500 and 1000 scenarios. Furthermore, our findings indicate that the adapted fast forward selection algorithm is potent enough to construct a subset of the complete scenario set that leads to good quality solutions.

This research can be extended in several directions. Given a planning horizon discretized by a set of finite time periods, the proposed model can be extended to a dynamic CLSC network design problem in which the uncertainty in quantity of returns and demands can be addressed along with uncertain quality status of cores. While accounting a multi-period setting to extend the current study into a dynamic CLSC, longer time horizons, e.g., fifteen years, should be considered to enhance the accuracy of the design decisions. As investments related with facilities have, usually, higher time horizons than five years; thus, in such cases, annualized costs have to be used. More precisely, it is of a great importance to consider the depreciation of the capital invested while accounting a planning horizon [22]. Another avenue of research is to deal with the uncertain quality state through the robust optimization approach. This might alleviate the computational complexity of the stochastic programming method.

Acknowledgment

The authors are grateful for the financial supports provided by Le Fonds de recherche du Québec-Nature et technologies (FRQNT) and the Natural Sciences and Engineering Research Council of Canada (NSERC).

3.8 Appendices

3.8.1 Problem notations

Indices

Z: Set of part suppliers

U: Set of raw material suppliers

H: Set of module suppliers

I: Set of manufacturing centers

J: Set of distribution centers

K: Set of end-user zones

C: Set of collection centers

A: Set of disassembly centers

M: Set of remanufacturing centers

B: Set of bulk recycling centers

G: Set of material recycling centers

D: Set of disposal centers

W: Set of secondary markets for modules

O: Set of secondary markets for spare parts

E: Set of secondary markets for recycled materials

L: Set of modules

P: Set of parts

R: Set of raw materials

S: Set of quality state scenarios

Parameters

ϕ_p: The number of part p in each unit of product

μ_r: The volume of material r in each unit of product

ω_l: The number of module l in each unit of product

ψ: The rate of return

β_s: The mass of residues in the returned product in scenario s shipped to bulk recycling centers from disassembly centers

α_r: The mass of recyclable material r in the returned product shipped to material recycling centers from disassembly centers

γ_{ps}: The number of reusable part p in the returned product in scenario s shipped to secondary markets and manufacturing centers from disassembly centers

δ_{ls}: The number of remanufacturable module l in the returned product in scenario s shipped to remanufacturing centers from disassembly centers

η_r: The ratio of recyclable material r shipped to material recycling centers from bulk recycling centers

τ_r: The ratio of non-recyclable material r shipped to disposal centers from bulk and material recycling centers

λ: The legislative target for recovery of the return stream

fc_c: Fixed cost of opening collection center c

fa_a: Fixed cost of opening disassembly center a

fm_m: Fixed cost of opening remanufacturing center m

fg_g: Fixed cost of opening material recycling center g

fb_b: Fixed cost of opening bulk recycling center b

fd_d: Fixed cost of opening disposal center d

tc_{kc}: Shipping cost per unit of the returned product from end-user k to collection center c

ta_{ca}: Shipping cost per unit of the returned product from collection center c to disassembly center a

ts_{aop}: Shipping cost per unit of part p from disassembly center a to spare market o

tz_{aip}: Shipping cost per unit of part p from disassembly center a to manufacturing center i

tg_{agr}: Shipping cost per kg of recyclable material r from disassembly center a to material recycling center g

te_{ger}: Shipping cost per kg of recycled material r from recycling center g to recycled material market e

tu_{gir}: Shipping cost per kg of recycled material r from material recycling center g to manufacturing centers i

sd_{gd}: Shipping cost per kg of wastes from material recycling center g to disposal center d

tb_{ab}: Shipping cost per kg of residues from disassembly center a to bulk recycling center b

rg_{bgr}: Shipping cost per kg of recyclable material r from bulk recycling center b to material recycling center g

rd_{bd}: Shipping cost per kg of wastes from bulk recycling center b to disposal center d

tm_{aml}: Shipping cost per unit of module l from disassembly center a to remanufacturing center m

tw_{mwl}: Shipping cost per unit of module l from remanufacturing center m to secondary market w

tx_{mil}: Shipping cost per unit of module l from remanufacturing center m to manufacturing center i

ti_{zip}: Shipping cost per unit of part p from part supplier z to manufacturing center i

ri_{uir}: Shipping cost per kg of material r from material supplier u to manufacturing center i

si_{hil}: Shipping cost per unit of module l from module supplier h to manufacturing center i

tj_{ij}: Shipping cost per unit of the new product from manufacturing center i to distribution center j

tk_{jk}: Shipping cost per unit of the new product from distribution center j to end-user k

caz_{zp}: Capacity of part supplier z for part p

cau_{ur}: Capacity of raw material supplier u for raw material r

cax_{hl}: Capacity of module supplier h for module l

cai_{i}: Capacity of manufacturing center i

caj_{j}: Capacity of distribution center j

cac_{c}: Capacity of collection center c

caa_{a}: Capacity of disassembly center a

cad_{d}: Capacity of disposal center d

cab_{b}: Capacity of bulk recycling center b

cag_{gr}: Capacity of material recycling center g for raw material r

cam_{ml}: Capacity of remanufacturing center m for module l

dk_{k}: Demand for the new product at end-user zone k

ds_{op}: Demand for part p at spare market o

de_{er}: Demand for material r at recycled material market e

dw_{wl}: Demand for module l at secondary market w

cz_{zp}: Procurement cost per unit of part p supplied by part supplier z

cu_{ur}: Procurement cost per kg of material r supplied by raw material supplier u

cx_{hl}: Procurement cost per unit of module l supplied by module supplier h

ci_i: Production cost per unit of product at manufacturing center i

cj_j: Distribution cost per unit of product at distribution center j

cc_c: Processing cost per unit of the returned product at collection center c

ca_a: Processing cost per unit of the returned product at disassembly center a

cd_d: Disposal cost at disposal center d

cg_{gr}: Recycling cost per kg of material r at material recycling center g

cb_b: Processing cost per kg of residues at recycling center b

cm_{ml}: Remanufacturing cost per unit of module l at remanufacturing center m

Pk_k: Unit price of the new product at end-user zone k

Ps_p: Unit price of part p at spare parts markets

Pe_r: Unit price of material r at recycled material markets

Pw_l: Unit price of module l at secondary markets

Pr: Unit acquisition price of the returned product

The first stage decision variables

YC_c: A binary variable which is equal to one if collection center c is opened and zero otherwise

YA_a: A binary variable which is equal to one if disassembly center a is opened and zero otherwise

YM_m: A binary variable which is equal to one if remanufacturing center m is opened and zero otherwise

YB_b: A binary variable which is equal to one if bulk recycling center b is opened and zero otherwise

YG_g: A binary variable which is equal to one if material recycling center g is opened

and zero otherwise

YD_d: A binary variable which is equal to one if disposal center d is opened and zero otherwise

XJ_{ij}: The quantity of products shipped from manufacturing center i to distribution center j

XK_{jk}: The quantity of products shipped from distribution center j to end-user zone k

XC_{kc}: The quantity of returns shipped from end-user zone k to collection center c

XA_{ca}: The quantity of returns shipped from collection center c to disassembly center a

XG_{agr}: The quantity of recyclable material r shipped from disassembly center a to material recycling center g

The second stage decision variables

QI_{zips}: The number of part p shipped from part supplier z to manufacturing center i in scenario s

NI_{uirs}: The quantity of raw material r shipped from raw material supplier u to manufacturing center i in scenario s

HI_{hils}: The number of module l shipped from module supplier h to manufacturing center i in scenario s

QS_{aops}: The number of part p shipped from disassembly center a to spare parts market o in scenario s

QZ_{aips}: The number of part p shipped from disassembly center a to manufacturing center i in scenario s

QM_{amls}: The number of module l shipped from disassembly center a to remanufacturing center m in scenario s

QW_{mwls}: The number of module l shipped from remanufacturing center m to secondary market w in scenario s

QX_{mils}: The number of module l shipped from remanufacturing center m to manufacturing center i in scenario s

QB_{abs}: The quantity of residues shipped from disassembly center a to bulk recycling center b in scenario s

NG_{bgrs}: The quantity of recyclable material r shipped from bulk recycling center b to material recycling center g in scenario s

QE_{gers}: The quantity of recycled material r shipped from material recycling center g to recycled material market e in scenario s

QU_{girs}: The quantity of recycled material r shipped from material recycling center g to manufacturing center i in scenario s

ND_{bds}: The quantity of residues shipped from bulk recycling center b to disposal center d in scenario s

GD_{gdrs}: The quantity of raw material r shipped from material recycling center g to disposal center d in scenario s

3.8.2 Magnanti and Wong problem

Considering Magnanti and Wong's approach, throughout the L-shaped algorithm, when the solution to the MP yields feasible RSPs for all scenarios, the following auxiliary dual subproblem has to be solved for each representative scenario.

$$\text{Min} \quad Z_{\boldsymbol{v}}(\boldsymbol{X^0}, \boldsymbol{Y^0}, \boldsymbol{\xi}_s) = \sum_{i\in I}\sum_{j\in J} X J_{ij}^0 \left(\sum_{p\in P} \phi_p v_{ips}^1 + \sum_{r\in R} \mu_r v_{irs}^2 + \sum_{l\in L} \omega_l v_{ils}^3 \right)$$

$$
\begin{aligned}
&+\sum_{a\in A}\sum_{c\in C} XA^0_{ca}\left(\sum_{p\in P}\gamma_{ps}v^4_{aps}+\beta_s v^6_{as}+\sum_{l\in L}\delta_l v^7_{als}\right)+\sum_{o\in O}\sum_{p\in P} ds_{op}v^5_{ops}\\
&+\sum_{w\in W}\sum_{l\in L} dw_{wl}v^9_{wls}+\sum_{g\in G}\sum_{a\in A} XG^0_{agr}\left(\sum_{r\in R} v^{12}_{grs}+\sum_{r\in R} v^{14}_{grs}\right)\\
&+\sum_{e\in E}\sum_{r\in R} de_{er}v^{13}_{ers}+\sum_{z\in Z}\sum_{p\in P} caz_{zp}v^{15}_{zps}+\sum_{u\in U}\sum_{r\in R} cau_{ur}v^{16}_{urs}\\
&+\sum_{h\in H}\sum_{l\in L} cax_{hl}v^{17}_{hls}+\sum_{m\in M}\sum_{l\in L} cam_{ml}YM^0_m v^{18}_{mls}+\sum_{b\in B} cab_b YB^0_b v^{19}_{bs}\\
&+\sum_{g\in G}\sum_{r\in R}\left(cag_{gr}YG^0_g-\sum_{a\in A} XG^0_{agr}\right)v^{20}_{grs}+\sum_{d\in D} cad_d YD^0_d v^{21}_{ds} \qquad (126)\\
\text{s.t.}\quad &\sum_{i\in I}\sum_{j\in J}\overline{XJ_{ij}}\left(\sum_{p\in P}\phi_p v^1_{ips}+\sum_{r\in R}\mu_r v^2_{irs}+\sum_{l\in L}\omega_l v^3_{ils}\right)\\
&+\sum_{a\in A}\sum_{c\in C}\overline{XA_{ca}}\left(\sum_{p\in P}\gamma_{ps}v^4_{aps}+\beta_s v^6_{as}+\sum_{l\in L}\delta_l v^7_{als}\right)+\sum_{o\in O}\sum_{p\in P} ds_{op}v^5_{ops}\\
&+\sum_{w\in W}\sum_{l\in L} dw_{wl}v^9_{wls}+\sum_{g\in G}\sum_{a\in A}\overline{XG_{agr}}\left(\sum_{r\in R} v^{12}_{grs}+\sum_{r\in R} v^{14}_{grs}\right)\\
&+\sum_{e\in E}\sum_{r\in R} de_{er}v^{13}_{ers}+\sum_{z\in Z}\sum_{p\in P} caz_{zp}v^{15}_{zps}+\sum_{u\in U}\sum_{r\in R} cau_{ur}v^{16}_{urs}\\
&+\sum_{h\in H}\sum_{l\in L} cax_{hl}v^{17}_{hls}+\sum_{m\in M}\sum_{l\in L} cam_{ml}\overline{YM_m}v^{18}_{mls}+\sum_{b\in B} cab_b\overline{YB^b}v^{19}_{bs}\\
&+\sum_{g\in G}\sum_{r\in R}\left(cag_{gr}\overline{YG_g}-\sum_{a\in A}\overline{XG_{agr}}\right)v^{20}_{grs}\\
&+\sum_{d\in D} cad_d\overline{YD_d}v^{21}_{ds}=Z^*_{\boldsymbol{v}}(\overline{\boldsymbol{X}},\overline{\boldsymbol{Y}},\boldsymbol{\xi}_s) \qquad (127)\\
&(\boldsymbol{v^1},\boldsymbol{v^2},....,\boldsymbol{v^{21}})\in\Delta_s \qquad (128)
\end{aligned}
$$

where $Z^*_{\boldsymbol{v}}(\overline{\boldsymbol{X}},\overline{\boldsymbol{Y}},\boldsymbol{\xi}_s)$ denotes the optimal value of the DRSP for the concerned scenario. The normalization constraint (126) ensures that the optimal solution of (125)-(127) is selected from the set of alternative optimal solutions to the DRSP. The aggregated optimality cut (115) generated using the solution of (125)-(127) is a Pareto-optimal cut.

3.8.3 Parameter settings

Tables 16 to 19 summarize parameter settings of model (64)-(74).

Table 16: Parameter settings for modules

Description	Value	
	Motor	Clutch
cx_{hl}	75	35
Pw_l	150	75

Table 17: Parameter settings for raw materials

Description	Value		
	Plastic	Steel	Copper
α_r	1.5 kg	1 kg	0.5 kg
cu_{ur}	0.75	0.5	3
pe_r	1.5	1	6

Table 18: Parameter settings for parts

Type of part	Value	
	cz_{zp}	Ps_p
Washing tube	20	40
Cover	5	10
Balance	25	50
Frame	5	10
Condenser	15	30
Transformer	15	30
Small electric	5	10
Hose	20	40
Electric wire	20	40
PCB board	35	70

Table 19: Values of other parameters

Description	Value	Description	Value
ci_i	4	cj_j	1
cc_c	1	ca_a	2
cm_m	3	cb_b	2
cg_{gr}	2	cd_d	2
η_r	0.3	τ_r	0.2
Pr	200	λ	0.7
dk_k	$\{600, 601, ..., 1200\}$	ds_{op}	$\{30, 31, ..., 100\}$
dw_{wl}	$\{30, 31, ..., 100\}$	Pk_k	Uniform(700, 1300)
fc_c	Uniform(400000, 600000)	fa_a	Uniform(400000, 600000)
fm_m	Uniform(700000, 900000)	fb_b	Uniform(400000, 600000)
fg_g	Uniform(400000, 600000)	fd_d	Uniform(200000, 400000)

Chapter 4

A decomposition algorithm for dynamic reverse supply chain network design under uncertainty

This chapter is dedicated to the article entitled *"A decomposition algorithm for dynamic reverse supply chain network design under uncertainty"*. This article was submitted to the *Computers & Operations Research* in July 2016. The titles, figures, and mathematical formulations have been revised to keep the coherence through the manuscript.

Abstract

Motivated by the recovery practices of modular-structured products, this study addresses designing a reverse supply chain network while incorporating uncertainty in quantity of the return stream over a planning horizon. The stochastic parameter is modeled as a scenario tree in which each stage of decision making corresponds to a unique time period. The concerned problem is formulated as a multi-stage mixed-integer stochastic programming model. Considering a scenario clustering decomposition scheme, the proposed model is decomposed into smaller scenario cluster sub-models such that the sub-models are associated with a number of sub-trees that share a certain number of predecessor nodes in the scenario tree. The sub-models are coordinated into an implementable solution via a Lagrangean progressive hedging-based method which employs a Benders decomposition-based algorithm as a viable solution approach for each scenario cluster sub-model. Based on a realistic scale case, computational results indicate a consistent performance efficiency of the proposed scenario clustering decomposition approach.

4.1 Introduction

RSC network design refers to the decisions in terms of locations of facilities associated with the collection and recovery of EOL products in addition to the allocation of physical flows among these facilities and secondary markets. Designing RSC networks for durable products (e.g., large household appliances) that are distinguished by their modular structure and their long life cycle is a complex problem. It is explained by the fact that such category of products can be dissembled into several components, namely modules and parts along with a bulk of damaged yet recyclable components referred to as residues. Depending on the category and quality status of each component

in the reverse BOM, a particular recovery process would be desired to reclaim the economic value residing in a specific component. For example, remanufacturing is a typical option for a used module in a good condition. Nonetheless, a poor quality (damaged) module is considered as residues that can be recycled to separate the precious raw materials from mixed scrap. Observing the variation in market demands of brand-new durable products, a similar tendency can be expected in generating end-of-life durable goods. In this regard, a dynamic perspective should be considered to accommodate such fluctuations in the RSC planning over a planning horizon. To date, a few contributions have addressed this concern [17, 19, 50]. Considering an application-oriented approach, Salema et al. [50] proposed a graph-based scheme to design a dynamic recovery network to capture the fluctuations in the rate of returns in a deterministic setting. In another attempt, Alumur et al. [19] developed a MIP formulation to model a RSC network design in a multi-period setting while considering the reverse BOM. The proposed model was also analyzed for a real-life industrial case.

An inherent characteristic of the recovery systems is the uncertainty in quality/quantity of returns. Needless to say, a successful designing of RSC networks requires the inclusion of such critical factors into the decision making problem. Most studies in the literature have utilized two-stage stochastic programming approaches to explicitly deal with uncertainty in static (single-period) settings [20, 21, 41, 48, 51]. The overview of the current literature indicates that most of the previous research in the context of designing RSCs under uncertainty is limited to single-period settings. In such studies, the common sources of uncertainty entail quality/quantity of returns, demands, and economical parameters such as shipping costs. For instance, Fonseca et al. [48] provided a two-stage stochastic programming model for designing a RSC network under uncertainty in transportation costs. The dynamic nature of the product returns combined with the uncertainty in the quantity of returns lead to a natural

extension of the static RSC network design into a dynamic setting under uncertainty which consequently calls for multi-stage stochastic programming [42] as a suitable approach to be adopted. In a multi-stage stochastic program, modeling the uncertain parameter, e.g., quantity of returns, as a scenario tree allows the adjustment of the decisions while more information on the uncertain parameter is available to the decision maker. In this line of research, Cardoso et al. [22] proposed a MS-MIP model to maximize the expected net present value of designing a close-loop supply chain network over a planning horizon under uncertainty in demand. In a similar vein, more recently, Zeballos et al. [23] used a scenario tree approach for discretization of stochastic demand and quantity of returns over the planning horizon. The resulting MS-MIP model was solved by a commercial software. As it can be observed, the number of studies in the context of designing a dynamic recovery network while accounting uncertainty is limited. To fill the existing void of research, on the modeling side, the first contribution of this study is to address the problem of designing a RSC in a multi-period setting considering the reverse BOM of durable products. In the underlying problem of interest, the quantity of returns is stochastic and non-stationary during the planning horizon. It is worth noting that a push market is assumed for the recovered modules, parts, and materials, which is a realistic assumption in many industries. Hence, the demand for the recovered items is considered as a deterministic yet dynamic parameter. Through modeling the uncertain factor as a scenario tree, the problem is modeled as a MS-MIP in which one seeks to maximize the expected profit. The non-homogeneity characteristic of the components of a durable good is also incorporated in the design decisions through defining a finite number of quality levels. To the best of our knowledge, none of the aforementioned studies in dynamic RSC planning have addressed the impact of the quality status of components on the choice of the recovery option while accounting uncertainty in the quantity of returns.

One complicating aspect of MS-MIP models is their computational intractability, particularly due to the exponential growth in the number of decision variables over the stages of the scenario tree of the stochastic parameter(s). Scenario clustering decomposition schemes have been shown to successfully solve large-scale multi-stage stochastic programming problems [52–55]. The prime idea of the scenario clustering decomposition is to divide the scenario tree into a set of scenario clusters such that they share some ancestor nodes. In most studies in the context of dynamic recovery network design, the size of test instances is quite small allowing the plain use of MIP solvers. Therefore, as the second contribution, on the methodological side, a heuristic inspired by a scenario clustering decomposition scheme [54, 55] is provided to solve the resulting large-scale MS-MIP problem. This algorithm revolves around decomposing the scenario tree into smaller sub-trees. The MS-MIP model would consequently be broken down into smaller sub-models corresponding to each sub-tree. Afterwards, the scenario cluster sub-models are coordinated by Lagrangean penalty terms in the objective function and a progressive hedging-based scheme [56] is applied for updating Lagrangean multipliers [57]. It is noteworthy to state that each scenario cluster sub-model per se is a hard to solve problem. Hence, as the third contribution, a Benders decomposition-based (BD) solution algorithm [5] is developed for tackling each scenario cluster sub-model which is enhanced with a Pareto-optimal cuts selection strategy [9].

The remainder of this article is organized as follows. In the next section, the description of the problem investigated in this article is provided and its formulation is introduced. Section 4.3 elaborates the details of the solution methodology including the scenario clustering decomposition and Benders decomposition schemes. Computational experiments on a case of large household appliances, i.e., washing machines, is presented in Section 4.4. Finally, Section 4.5 concludes this paper.

4.2 Problem statement

4.2.1 Problem description

Considering a dynamic RSC network design context, as shown in Figure 11, in each period in the planning horizon, used products that are of non-homogeneous quality status are acquired in collection zones and then shipped to disassembly centers. The returns are then graded into multiple quality levels in disassembly centers. As noted in the preceding section, depending on the quality level of the component in the reverse BOM of a durable product, it can be sent to a particular facility for the recovery process. Hence, high quality modules are sent to remanufacturing centers and high quality parts are used for part harvesting to make them "like-new" components. These components are then offered at a lower price compared to the brand-new components at their corresponding marketplaces. For instance, in a washing machine, its motor and washing tube are categorized, respectively, as modules and parts. The high quality level motor is profitable for remanufacturing while a poor quality washing tube is sent to the bulk recycling center. Damaged components are also shipped to bulk recycling facilities to recycle precious raw materials. The unprocessed raw materials are then purchased by a 3PL provider. More precisely, it is assumed that there exists an infinite demand for recyclable raw materials in markets. It is also assumed that the waste of residues is safely disposed of in bulk recycling facilities at zero cost, as it is a sunk cost.

In a deterministic setting, the design decisions in each period revolve around the location of each facility including disassembly, remanufacturing, and bulk recycling centers to be installed in the RSC network. It should be noted that a dynamic RSC design provides the flexibility to adjust the number and the location of facilities as the quantity of returns evolves over time. More precisely, depending on the quantity of

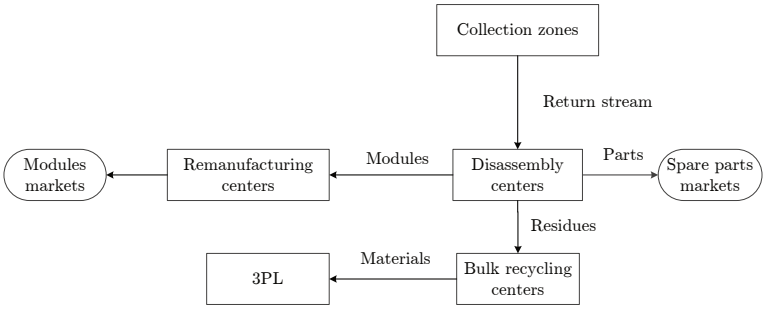

Figure 11: The RSC network

returns over the planning horizon, either some of the existing facilities are closed (in case of a reduced return stream) or new facilities are opened (in case of an increasing return stream). Furthermore, the planning decisions include the physical flows and inventory levels at each facility. The objective function is to maximize the profit over the planning horizon. The organization gains revenues from remanufacturing when the remanufactured modules are sold in the secondary markets; from reusable parts when they are sold to spare parts markets; from bulk recycling when the unprocessed raw materials are purchased by the third-party logistics provider. The total cost comprises the fixed costs of the installation of facilities along with inventory holding, processing, and transportation costs in the RSC network. Furthermore, the following assumptions are made regarding the problem setting.

- Demands of remanufactured modules and reusable parts are deterministic yet dynamic over the planning horizon;
- The return stream is categorized with respect to a finite set of quality levels;
- The unit collection, disassembly, and remanufacturing costs are quality status-dependent;
- Capacities of facilities are not subject to change within time periods.

4.2.2 Modeling uncertain returns

In the problem of interest, it is assumed that the quantity of returns is uncertain and dynamic; hence, it evolves as a discrete time stochastic process over the planning horizon. As noted earlier, the dynamic and uncertain nature of returns quantities require the adjustment of the design decisions during the planning horizon. To this end, the planning horizon is discretized into a finite set of time periods such that the decisions are implemented at the end of each time period. Considering the multi-period setting together with uncertainty, the stochastic quantity of returns parameter can be interpreted as a scenario tree in which each stage indicates the realization of the uncertain parameter. It is assumed that each stage corresponds to a single time period. In a given stage, each node represents a distinguishable state of random return concerning the available information up to this stage. In the underlying problem, each node is directly connected to two other nodes while moving away from the root node. In other words, each node in the scenario tree has only one sibling except the root node. Besides, a return quantity scenario is defined as the full path from the root node, i.e., the current state of world, to a leaf node at the last stage of the scenario tree. Figure 12 illustrates a scenario tree with four stages.

4.2.3 Problem formulation

Given the RSC network design problem described in Section 4.2.1 and the scenario tree representing the uncertain quantity of returns, in this section, the latter problem is formulated as a MS-MIP model. In this model, the location of facilities depends on the quantity of returns, hence this decision is defined for each possible realization of the stochastic parameter in each period, represented by a node in each stage of the scenario tree. In a similar fashion, the quantity of acquisition, disassembly quantity,

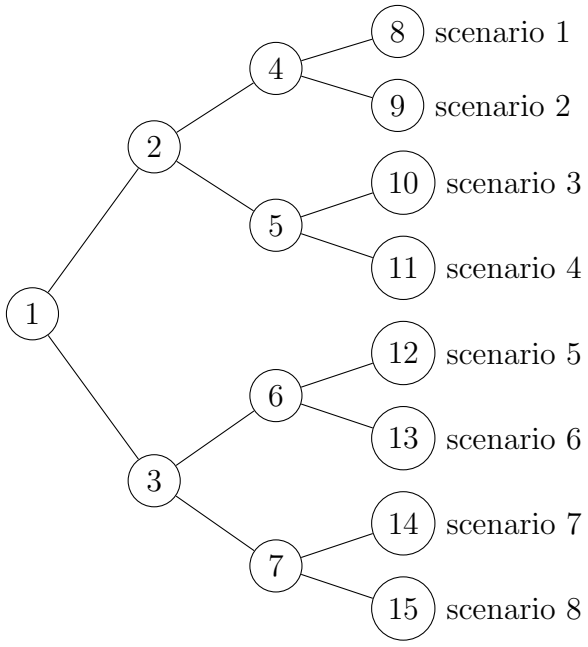

Figure 12: Scenario tree for the random quantity of returns

the flow between the disassembly and recovery facilities, shipped quantities of recovered items, as well as the inventory levels of recovered items at different facilities also depend on the return quantity, thus are defined for each node. Furthermore, in any given period in the planning horizon (represented as a stage in the scenario tree), the decision maker cannot foresee future outcomes of the return quantity; therefore, location and flow decisions must satisfy the non-anticipativity condition (NAC). The latter indicates that such decisions in a given period (stage) are identical for scenarios with a common ancestor node in that period. For instance, in Figure 12, scenarios 1 and 2 share node 2 in stage 2; therefore, the location and flow decisions must be identical for both scenarios at this stage and consequently are defined exclusively for the ancestor node 2. On the other hand, since inventory decisions are state variables that depend on the main design/flow decisions, the NAC would automatically apply. The problem notation is provided in the *Appendices*, in the section entitled *Problem*

notations. The compact formulation of the MS-MIP model corresponding to the dynamic RSC network design problem under investigation can be stated as follows. It should be noted that the NAC is implicitly taken into consideration while using a compact formulation of a multi-stage stochastic problem. On the contrary, the latter condition must be explicitly stated in a split-variable or clustered formulation.

Total revenue

$$\sum_{n\in Tree} pr(n)\Bigg\{\sum_{t\in T}\sum_{a\in A}\sum_{o\in O}\sum_{p\in P} Ps_p QS_{aopt}(n) + \sum_{t\in T}\sum_{d\in D}\sum_{w\in W}\sum_{l\in L} Pw_l QW_{dwlt}(n) + \sum_{t\in T}\sum_{b\in B}\sum_{r\in R} Pe_r BR_{brt}(n)\Bigg\} \tag{129}$$

Total cost

Fixed cost

$$\sum_{n\in Tree} pr(n)\Bigg\{\sum_{t\in T}\sum_{a\in A} fa_a YA_{at}(n) + \sum_{t\in T}\sum_{d\in D} fd_d YD_{dt}(n) + \sum_{t\in T}\sum_{b\in B} fb_b YB_{bt}(n)\Bigg\} \tag{130}$$

Processing cost

$$\sum_{n\in Tree} pr(n)\Bigg\{\sum_{t\in T}\sum_{c\in C}\sum_{a\in A}\sum_{q\in Q} ca_{aq} QA_{caqt}(n) + \sum_{t\in T}\sum_{a\in A}\sum_{d\in D}\sum_{l\in L} cd_{dl} QD_{adlt}(n) \sum_{t\in T}\sum_{a\in A}\sum_{b\in B} cb_b QB_{abt}(n)\Bigg\} \tag{131}$$

Inventory holding cost

$$\sum_{n\in Tree} pr(n)\Bigg\{\sum_{t\in T}\sum_{a\in A}\sum_{p\in P} hp_p IP_{apt}(n) + \sum_{t\in T}\sum_{a\in A}\sum_{l\in L} hl_l IL_{alt}(n) + \sum_{t\in T}\sum_{a\in A} hbIB_{at}(n) + \sum_{t\in T}\sum_{d\in D}\sum_{l\in L} hl_l ID_{dlt}(n)\Bigg\} \tag{132}$$

Transportation cost

$$\sum_{n\in Tree} pr(n)\Bigg\{\sum_{t\in T}\sum_{c\in C}\sum_{a\in A}\sum_{q\in Q} ta_{ca} QA_{caqt}(n) \sum_{t\in T}\sum_{a\in A}\sum_{o\in O}\sum_{p\in P} ts_{aop} QS_{aopt}(n) + \sum_{t\in T}\sum_{a\in A}\sum_{d\in D}\sum_{l\in L} td_{adl} QD_{adlt}(n) + \sum_{t\in T}\sum_{a\in A}\sum_{b\in B} tbQB_{abt}(n) + \sum_{t\in T}\sum_{d\in D}\sum_{w\in W}\sum_{l\in L} tw_{dwl} QW_{dwlt}(n)\Bigg\} \tag{133}$$

Supply constraints

$$\sum_{a\in A} QA_{caqt}(n) = \psi_{cqt}(n) \quad c\in C, q\in Q, t\in T, n\in Tree \tag{134}$$

Flow balance constraints

$$IP_{apt}(n) = IP_{ap(t-1)}(m) + \sum_{c\in C}\sum_{q\in Q} \gamma_{pq} QA_{caqt}(n) - \sum_{o\in O} QS_{aopt}(n) \quad a\in A,$$

$$p\in P, t\in T, n\in Tree, m = a(n) \tag{135}$$

$$IL_{alt}(n) = IL_{al(t-1)}(m) + \sum_{c\in C}\sum_{q\in Q} \delta_{lq} QA_{cat}(n) - \sum_{d\in D} QD_{adlt}(n) \quad a\in A,$$

$$l\in L, t\in T, n\in Tree, m = a(n) \tag{136}$$

$$IB_{at}(n) = IB_{a(t-1)}(m) + \sum_{c\in C}\sum_{q\in Q} \beta_q QA_{caqt}(n) - \sum_{b\in B} QB_{abt}(n) \quad a\in A, t\in T,$$

$$n\in Tree, m = a(n) \tag{137}$$

$$ID_{dlt}(n) = ID_{dl(t-1)}(m) + \sum_{a \in A} QD_{adlt}(n) - \sum_{w \in W} QW_{dwlt}(n) \quad d \in D,$$

$$l \in L, t \in T, n \in Tree, m = a(n) \tag{138}$$

$$\sum_{a \in A} \eta_r QB_{abt}(n) = BR_{brt}(n) \quad b \in B, r \in R, t \in T, n \in Tree \tag{139}$$

Demand constraints

$$\sum_{a \in A} QS_{aopt}(n) = ds_{opt} \quad o \in O, p \in P, t \in T, n \in Tree \tag{140}$$

$$\sum_{d \in D} QW_{dwlt}(n) = dw_{wlt} \quad w \in W, l \in L, t \in T, n \in Tree \tag{141}$$

Capacity constraints of facilities

$$\sum_{c \in C} \sum_{q \in Q} QA_{caqt}(n) \leq caa_a YA_{at}(n) \quad a \in A, t \in T, n \in Tree \tag{142}$$

$$\sum_{a \in A} QD_{azlt}(n) \leq cad_{dl} YD_{dt}(n) \quad d \in D, l \in L, t \in T, n \in Tree \tag{143}$$

$$\sum_{a \in A} QB_{abt}(n) \leq cab_b YB_{bt}(n) \quad b \in B, t \in T, n \in Tree \tag{144}$$

In model (129)-(144), the objective function accounts for maximizing the expected profit. Constraint (134) ensures the acquisition of the return stream for each node and each time period. Constraints (135)-(138) are inventory balance restrictions, respectively, for parts, modules, and residues at disassembly centers in addition to remanufacturing facilities. Flow balance restriction in each bulk recycling center is imposed by Constraints (139). Constraints (140)-(141) ensure the demand satisfaction of parts and remanufactured modules at their corresponding marketplaces in each time period. Constraints (142)-(144) impose capacity restriction on disassembly, remanufacturing, and bulk recycling centers.

4.3 Solution methodology

Solving the MS-MIP model (129)-(144) by a commercial solver for real-size instances is a challenge. This is due to the existence of the three sets of binary variables that increase exponentially in number as the number of stages in the planning horizon is increased. As noted earlier, the computational complexity has motivated the authors to propose a heuristic scenario clustering decomposition (HSCD) algorithm. This algorithm comprises two major steps: (1) Scenario cluster decomposition (SCD) and (2) Scenario cluster coordination (SCC). In the SCD step, first, the scenario tree is partitioned into a set of scenario cluster sub-trees. Then, for each sub-tree, the corresponding MS-MIP model is represented in a compact formulation. Furthermore, the NACs corresponding to common nodes in the original scenario tree are introduced in the objective function of each scenario cluster sub-model. Finally, in the SCC step, the aforementioned sub-models are coordinated into an implementable solution by means of a Lagrangian Progressive Hedging-based algorithm. The details of the aforementioned steps are provided as follows.

4.3.1 Step 1: SCD

4.3.1.1 Decomposing the scenario tree

Definition 4.1. Given that n^{δ} and S represent, respectively, the set of nodes that belong to stage δ and the set of scenario clusters, according to [54, 55], a break stage δ^* is defined as a stage in the scenario tree such that the following equation holds: $|S| = |n^{\delta^*+1}|$.

In Figure 13, if the second stage is chosen as the break stage, i.e., $\delta^* = 2$, four scenario clusters ($|S| = 4$) are obtained such that each shares node 1. Furthermore, the first and the second scenario cluster sub-trees share node 2 and the other two

share node 3 in the original scenario tree (see Figure 13).

Let introduce N^s as the set of nodes that belong to scenario cluster s, $\Delta = \{1, 2, .., \delta^*\}$, N_1 as the set of nodes corresponding to the stages in Δ, $N_2 = N \backslash N_1$, $N_1^s = N_1 \cap N^s$, $N_2^s = N_2 \cap N^s$. Moreover, let ζ^ω be the likelihood of scenario ω, Ω_s be the set of scenarios in scenario cluster sub-tree s, and $\zeta^s(n) = \sum_{\omega \in \Omega_s} \zeta^\omega$.

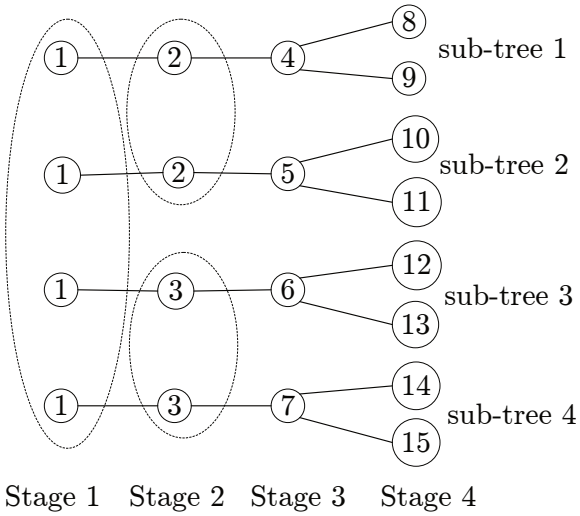

Figure 13: The scenario cluster sub-trees

4.3.1.2 Formulating the scenario cluster sub-model

Following the scenario tree decomposition, the MS-MIP model is formulated for each sub-tree in a compact representation. The NACs for any node in N_2 are implicitly considered by formulating each scenario cluster sub-tree in a compact representation. However, these constraints are required to explicitly be taken into account for every node in N_1. The purpose of introducing the NACs is to coordinate and link $|S|$ scenario cluster sub-models into an implementable solution. Let $X_t^s(n)$ be the vector of flow and location variables in the MS-MIP (129)-(144). Let P be the vector of unit prices of selling brand-new and recovered components at the marketplaces. Let F and C be, respectively, the vector of fixed costs of opening facilities in the

reverse network and the vector of procurement, processing, inventory carrying, and transportation costs. Moreover, let η^n be scenario cluster sub-trees that share node n, $\underline{s}_{\eta^n} = min\{s|\forall s \in \eta^n\}$, $\overline{s}_{\eta^n} = max\{s|\forall s \in \eta^n\}$. The NACs can therefore be stated as follows.

$$X_t^s(n) - X_t^{s+1}(n) \leq 0 \quad \forall s = \underline{s}_{\eta^n}, ..., (\underline{s}_{\eta^n}) - 1, t \in \Delta, n \in N_1 \tag{145}$$

$$X_t^{\overline{s}_{\eta^n}}(n) - X_t^{\underline{s}_{\eta^n}}(n) \leq 0 \quad t \in \Delta, n \in N_1 \tag{146}$$

For instance, in the sub-trees shown in Figure 13, the NACs for the location of disassembly centers, i.e., $YA_{at}(n)$, is expressed as follows.

$$YA_{a2}^1(2) - YA_{a2}^2(2) \leq 0 \quad \forall a \in A$$

$$YA_{a2}^2(2) - YA_{a2}^1(2) \leq 0 \quad \forall a \in A$$

$$YA_{a2}^3(3) - YA_{a2}^4(3) \leq 0 \quad \forall a \in A$$

$$YA_{a2}^4(3) - YA_{a2}^3(3) \leq 0 \quad \forall a \in A$$

By dualizing the NACs and using a Lagrangean multiplier vector, i.e., $\mu_t^s(n)$, the MS-MIP model (129)-(144) can be reformulated as the following multi-stage scenario cluster Lagrangean decomposition (MSCLD) problem. [54, 55].

$$Z_{MSCLD}(\mu, s) = Max \sum_{s=1}^{|S|} \sum_{n \in N_1^s} \sum_{t \in \Delta} \zeta^s(n)\{PX_t^s(n) - FX_t^s(n) - CX_t^s(n)\}$$
$$+ \sum_{s=1}^{|S|} \sum_{n \in N_2^s} \sum_{t \notin \Delta} pr(n)\{PX_t^s(n) - FX_t^s(n) - CX_t^s(n)\}$$

$$\sum_{s=\underline{s}_{\eta^n}}^{(\bar{s}_{\eta^n})-1} \sum_{n\in N_1} \sum_{t\in\Delta} \mu_t^s(n)\{X_t^{s+1}(n) - X_t^s(n)\}$$

$$+ \sum_{n\in N_1} \sum_{t\in\Delta} \mu_t^{\bar{s}_{\eta^n}}(n)\{X_t^{\underline{s}_{\eta^n}}(n) - X_t^{\bar{s}_{\eta^n}}(n)\}$$

$$s.t. \quad (134) - (144) \quad \forall t \in \Delta, n \in \{N_1, N_2\} \tag{147}$$

As it can be seen, (147) is a relaxation of MS-MIP (129)-(144) for all $\mu_t^s(n) \geq 0; \quad \forall s \in S, n \in N_1^s$, and $t \in \Delta$. Thus, the value of its objective function, $Z_{MSCLD}(\mu, s)$, is an upper bound on the optimal solution of the original MS-MIP model.

Definition 4.2. The dual problem (Lagrangean dual) of the original MS-MIP model with respect to NACs (145)-(146), for a given break stage δ^*, can be represented as

$$Z_{MSCLD} = Min_{\mu\geq 0} Z_{MSCLD}(\mu, s) \tag{148}$$

The Lagrangean dual problem (148) is solved by an iterative sub-gradient-based scheme to identify an upper bound on the original MS-MIP model (129)-(144). It should be stated that model (147) can further be decomposed into $|S|$ subproblems in accordance with each scenario cluster sub-tree. Its objective function can also be attained through summing up each individual sub-model objective function as follows.

$$Z_{MSCLD}(\mu, s) = \sum_{s=1}^{|S|} Z_{MSCLD}^s(\mu) \tag{149}$$

4.3.2 Step 2: SCC

4.3.2.1 Lagrangean progressive hedging-based algorithm (LPHA)

In order to update Lagrangean multipliers, a Lagrangean progressive hedging-based scheme is considered as presented in Escudero et al. [54, 57]. The progressive hedging algorithm was firstly introduced in the seminal work of Rockafellar and Wets [56] for solving multi-stage stochastic linear programming models.

Definition 4.3. The classical sub-gradient vector $g_t(n)$ in which $n \in N_1^s, t \in \Delta$ can be defined as follows [54].

$$g_t(n) = \begin{pmatrix} X_t^{\underline{s}_{\eta^n}}(n) - X_t^{\bar{s}_{\eta^n}}(n) \\ X_t^{(\underline{s}_{\eta^n})+1}(n) - X_t^{\underline{s}_{\eta^n}}(n) \\ . \\ . \\ . \\ X_t^{\bar{s}_{\eta^n}}(n) - X_t^{((\bar{s}_{\eta^n})-1)}(n) \end{pmatrix}$$

In LPHA scheme, apart from the classical sub-gradient vector, a new modified vector is defined.

Definition 4.4. Denoted by $\overline{g}_t(n)$, a non-necessary feasible pseudo sub-gradient vector can be represented as follows.

$$\overline{g}_t(n) = \begin{pmatrix} \overline{X}_t(n) - X_t^{\bar{s}_{\eta^n}}(n) \\ \overline{X}_t(n) - X_t^{\underline{s}_{\eta^n}}(n) \\ . \\ . \\ . \\ \overline{X}_t(n) - X_t^{((\bar{s}_{\eta^n})-1)}(n) \end{pmatrix}$$

where $\overline{X}_t(n) = \sum_{s \in \eta^n} \zeta^s(n) X^s(n)$ such that, $n \in N_1^s$ and $t \in \Delta$. In fact, $\overline{X}_t(n)$ indicates an approximate expected value over the set of scenario cluster sub-trees that

share node n. Denoting by $\underline{Z}_{MSCLD}$ a lower bound on (147), the details of LPHA method are summarized in Algorithm 7.

Algorithm 7 - LPHA

Step 0: **Initialization** $\mu_t^s(n) \leftarrow 0 \quad \forall s \in S, n \in N_1^s, t \in \Delta$, $\bar{\alpha} \leftarrow 1$, Iteration counter $k \leftarrow 1$
Solve $|S|$ scenario cluster sub-models (147) independently
Compute the initial value of $Z_{MSCLD}(\mu, s)$
while termination conditions are not satisfied **do**
Step 1: Calculate $g_t^k(n)$ and $\bar{g}_t^k(n) \quad \forall n \in N_1^s, t \in \Delta$
Step 2: Update Lagrangean multipliers
$\mu_t^{k+1}(n) \leftarrow max\{0, \mu_t^k(n) + \bar{\alpha}.\frac{(Z_{MSCLD}(\mu^k,s)-\underline{Z}_{MSCLD})}{\left\|\bar{g}_t^k(n)\right\|^2}.\bar{g}k_t(n)\}$
Step 3: Solve $|S|$ scenario cluster sub-models with μ^{k+1} and update $Z_{MSCLD}(\mu^{k+1}, s)$
Step 4: $k \leftarrow k + 1$
end while

The termination conditions require that either the sub-gradient vector $g_t^k(n)$ is less than a threshold, i.e., 0.01, or the value of Z_{MSCLD} is not improved in a sequence of consecutive iterations, i.e., 5 iterations.

Each iteration of LPHA calls for the solutions of $|S|$ independent sub-models. In the context of the dynamic RSC network design investigated in this study, each sub-model is itself a large-scale optimization problem which cannot be solved by the plain use of MIP engines, e.g., CPLEX. Therefore, a Benders decomposition-based algorithm tailored to the particular structure of each scenario cluster sub-model is proposed in the next section. This algorithm is nested within the LPHA algorithm to solve the $|S|$ scenario cluster sub-models.

4.3.2.2 Benders decomposition-based algorithm

The hallmark of Benders decomposition is to exploit the decomposable structure present in the formulation of the MIP model. In other words, in a MIP model, integer/binary variables are seen as complicating variables such that when fixing them,

the MIP model reduces to smaller subproblems (PSP), which can be solved individually to generate cutting planes for the master problem (MP). The PSP and MP are then solved sequentially and iteratively until a termination criterion is satisfied. As the classical Benders decomposition turns out to converge slowly in the underlying problem due to the degeneracy of PSP, a cut selection strategy based on the work of Papadakos [9] is proposed to expedite the convergence of the solution process.

4.3.2.2.1 Benders reformulation

Given a particular scenario cluster sub-tree $s = (\underline{s}_{\eta^n}) + 1, ..., \overline{s}_{\eta^n}$ in (149) and a vector of feasible location decisions, i.e., $\overline{\boldsymbol{Y^s}} = \{YA^s_{at}(n), YD^s_{dt}(n), YB^s_{bt}(n)\}$, the PSP can be formulated as follows.

$$
\begin{aligned}
Z^s_{MSCLD}(\mu) = Max \sum_{n\in N^s_1}\sum_{t\in\Delta}\zeta^s(n)\Bigg\{ & \sum_{a\in A}\sum_{o\in O}\sum_{p\in P} Ps_p QS^s_{aopt}(n) \\
& + \sum_{d\in D}\sum_{w\in W}\sum_{l\in L} Pw_l QW^s_{dwlt}(n) + \sum_{b\in B}\sum_{r\in R} Pe_r BR^s_{brt}(n) - \sum_{c\in C}\sum_{a\in A}\sum_{q\in Q} ca_{aq} QA^s_{caqt}(n) \\
& + \sum_{a\in A}\sum_{d\in D}\sum_{l\in L} cd_{dl} QD^s_{adlt}(n) - \sum_{a\in A}\sum_{b\in B} cb_b QB^s_{abt}(n) - \sum_{a\in A}\sum_{p\in P} hp_p IP^s_{apt}(n) \\
& - \sum_{a\in A}\sum_{l\in L} hl_l IL^s_{alt}(n) - \sum_{a\in A} hbIB^s_{at}(n) - \sum_{d\in D}\sum_{l\in L} hl_l ID^s_{dlt}(n) \\
& - \sum_{c\in C}\sum_{a\in A}\sum_{q\in Q} ta_{ca} QA^s_{caqt}(n) - \sum_{a\in A}\sum_{o\in O}\sum_{p\in P} ts_{aop} QS^s_{aopt}(n) \\
& - \sum_{a\in A}\sum_{d\in D}\sum_{l\in L} td_{adl} QD^s_{adlt}(n) - \sum_{a\in A}\sum_{b\in B} tbQB^s_{abt}(n) - \sum_{d\in D}\sum_{w\in W}\sum_{l\in L} tw_{dwl} QW^s_{dwlt}(n) \\
& + \sum_{n\in N^s_2}\sum_{t\notin\Delta} pr(n)\Bigg\{ \sum_{a\in A}\sum_{o\in O}\sum_{p\in P} Ps_p QS^s_{aopt}(n) - \sum_{d\in D}\sum_{w\in W}\sum_{l\in L} Pw_l QW^s_{dwlt}(n) \\
& - \sum_{b\in B}\sum_{r\in R} Pe_r BR^s_{brt}(n) - \sum_{c\in C}\sum_{a\in A}\sum_{q\in Q} ca_{aq} QA^s_{caqt}(n) + \sum_{a\in A}\sum_{d\in D}\sum_{l\in L} cd_{dl} QD^s_{adlt}(n)
\end{aligned}
$$

$$
\begin{aligned}
&- \sum_{a\in A}\sum_{b\in B} cb_b QB^s_{abt}(n) - \sum_{a\in A}\sum_{p\in P} hp_p IP^s_{apt}(n) - \sum_{a\in A}\sum_{l\in L} hl_l IL^s_{alt}(n) \\
&- \sum_{a\in A} hbIB^s_{at}(n) - \sum_{z\in Z}\sum_{l\in L} hl_l ID^s_{dlt}(n) - \sum_{c\in C}\sum_{a\in A}\sum_{q\in Q} ta_{ca} QA^s_{caqt}(n) \\
&- \sum_{a\in A}\sum_{o\in O}\sum_{p\in P} ts_{aop} QS^s_{aopt}(n) - \sum_{a\in A}\sum_{d\in D}\sum_{l\in L} td_{adl} QD^s_{adlt}(n) - \sum_{a\in A}\sum_{b\in B} tbQB^s_{abt}(n) \\
&- \sum_{d\in D}\sum_{w\in W}\sum_{l\in L} tw_{dwl} QW^s_{dwlt}(n) \Bigg\} \\
&+ \sum_{n\in N_1}\sum_{t\in\Delta} \Bigg\{ \sum_{a\in A}\sum_{o\in O}\sum_{p\in P} (\mu^{1,s-1}_{aopt}(n) - \mu^{1,s}_{aopt}(n)) QS^s_{aopt}(n) \\
&+ \sum_{d\in D}\sum_{w\in W}\sum_{l\in L} (\mu^{2,s-1}_{dwlt}(n) - \mu^{2,s}_{dwlt}(n)) QW^s_{dwlt}(n) \\
&+ \sum_{b\in B}\sum_{r\in R} (\mu^{3,s-1}_{brt}(n) - \mu^{3,s}_{brt}(n)) BR^s_{brt}(n) \\
&+ \sum_{c\in C}\sum_{a\in A}\sum_{q\in Q} (\mu^{4,s-1}_{caqt}(n) - \mu^{4,s}_{caqt}(n)) QA^s_{caqt}(n) \\
&+ \sum_{a\in A}\sum_{d\in D}\sum_{l\in L} (\mu^{5,s-1}_{adlt}(n) - \mu^{5,s}_{adlt}(n)) QD^s_{adlt}(n) \\
&+ \sum_{a\in A}\sum_{b\in B} (\mu^{6,s-1}_{abt}(n) - \mu^{6,s}_{abt}(n)) QB^s_{abt}(n) \Bigg\} \qquad (150)
\end{aligned}
$$

$$
s.t. \quad (134) - (141) \quad \forall t \in \Delta\backslash\{1\}, n \in N^s
$$

$$
\sum_{c\in C}\sum_{q\in Q} QA^s_{caqt}(n) \le caa_a \overline{YA}^s_{at}(n) \quad a \in A, t \in \Delta, n \in N^s \qquad (151)
$$

$$
\sum_{a\in A} QD^s_{azlt}(n) \le cad_{dl} \overline{YD}^s_{dt}(n) \quad d \in D, l \in L, t \in \Delta, n \in N^s \qquad (152)
$$

$$
\sum_{a\in A} QB^s_{abt}(n) \le cab_b \overline{YB}^s_{bt}(n) \quad b \in B, t \in \Delta, n \in N^s \qquad (153)
$$

where $\boldsymbol{\mu^{1,s}}$,..., $\boldsymbol{\mu^{6,s}}$ denote the set of Lagrangean multipliers for scenario cluster sub-tree s. Note that for sub-tree $s = \underline{s}_{\eta^n}$, the terms of the objective function (150) that correspond to Lagrangean multipliers are written as the following compact representation: $\sum_{n\in N_1}\sum_{t\in\Delta}\{\mu^{\overline{s}_{\eta^n}}_t(n) - \mu^{\underline{s}_{\eta^n}}_t(n)\} X^{\underline{s}_{\eta^n}}_t(n)$.

Let $\boldsymbol{\upsilon^{1,s}}$,..., $\boldsymbol{\upsilon^{11,s}}$ be the set of dual variables corresponding to constraints (134)-(141) and (151)-(153) in which $\boldsymbol{\upsilon^{9,s}}$, $\boldsymbol{\upsilon^{10,s}}$, and $\boldsymbol{\upsilon^{11,s}}$ are non-negative. The dual subproblem (DSP) can be formulated as follows .

$$
\begin{aligned}
Z^s_{\boldsymbol{\upsilon}}(\overline{\boldsymbol{Y^s}}) = \text{Min} \quad & \sum_{c\in C}\sum_{q\in Q}\sum_{t\in\Delta}\sum_{n\in N^s}\psi_{cqt}(n)\upsilon^{1,s}_{cqt}(n) \\
& + \sum_{o\in O}\sum_{p\in P}\sum_{t\in\Delta}\sum_{n\in N^s} ds_{opt}\upsilon^{7,s}_{opt}(n) + \sum_{w\in W}\sum_{l\in L}\sum_{t\in\Delta}\sum_{n\in N^s} dw_{wlt}\upsilon^{8,s}_{wlt}(n) \\
& + \sum_{a\in A}\sum_{t\in\Delta}\sum_{n\in N^s} caa_a\overline{YA}^s_{at}(n)\upsilon^{9,s}_{at}(n) + \sum_{d\in D}\sum_{l\in L}\sum_{t\in\Delta}\sum_{n\in N^s} cad_{dl}\overline{YD}^s_{dt}(n)\upsilon^{10,s}_{dl}(n) \\
& + \sum_{b\in B}\sum_{t\in\Delta}\sum_{n\in N^s} cab_b\overline{YB}^s_{bt}(n)\upsilon^{11,s}_{bt}(n) \qquad (154)
\end{aligned}
$$

$$
s.t. \quad (\boldsymbol{\upsilon^{1,s}}, \boldsymbol{\upsilon^{2,s}}, ..., \boldsymbol{\upsilon^{11,s}}) \in \Lambda^s \qquad (155)
$$

where Λ^s denotes the polyhedron defined by the constraints of the DSP for a particular scenario cluster sub-tree s. Let θ^s be a surrogate variable that is an upper bound on (150). Furthermore, let $\rho(.)$ entails all terms in the objective function of DSP (154) independent of the location variables. The master problem (MP) can be stated as follows.

$$
\begin{aligned}
\max \quad & \theta^s - \sum_{n\in N^s_1}\sum_{t\in\Delta}\zeta^s(n)\left\{\sum_{a\in A} fa_a YA^s_{at}(n) + \sum_{d\in D} fd_d YD^s_{dt}(n) + \sum_{b\in B} fb_b YB^s_{bt}(n)\right\} \\
& - \sum_{n\in N^s_2}\sum_{t\notin\Delta} pr(n)\left\{\sum_{a\in A} fa_a YA^s_{at}(n) + \sum_{d\in D} fd_d YD^s_{dt}(n) + \sum_{b\in B} fb_b YB^s_{bt}(n)\right\} \\
& + \sum_{n\in N_1}\sum_{t\in\Delta}\left\{\sum_{a\in A}(\mu^{7,s-1}_{at}(n) - \mu^{7,s}_{at}(n))YA^s_{at}(n)\right. \\
& + \sum_{d\in D}(\mu^{8,s-1}_{dt}(n) - \mu^{8,s}_{dt}(n))YD^s_{dt}(n) \\
& \left. + \sum_{b\in B}(\mu^{9,s-1}_{bt}(n) - \mu^{9,s}_{bt}(n))YB^s_{bt}(n)\right\} \qquad (156)
\end{aligned}
$$

$$
\begin{aligned}
s.t. \quad & \theta^s \leq \rho(\hat{\boldsymbol{v}}^{\boldsymbol{s}^T}) + \sum_{a\in A}\sum_{t\in\Delta}\sum_{n\in N^s} caa_a Y A^s_{at}(n)\hat{v}^{9,s}_{at}(n) \\
& + \sum_{d\in D}\sum_{l\in L}\sum_{t\in\Delta}\sum_{n\in N^s} cad_{dl} Y D^s_{dt}(n)\hat{v}^{10,s}_{dl}(n) \\
& + \sum_{b\in B}\sum_{t\in\Delta}\sum_{n\in N^s} cab_b Y B^s_{bt}(n)\hat{v}^{11,s}_{bt}(n) \qquad (157) \\
& 0 \leq \rho(\hat{\boldsymbol{\kappa}}^{\boldsymbol{s}^T}) + \sum_{a\in A}\sum_{t\in\Delta}\sum_{n\in N^s} caa_a Y A^s_{at}(n)\hat{\kappa}^{9,s}_{at}(n) \\
& + \sum_{d\in D}\sum_{l\in L}\sum_{t\in\Delta}\sum_{n\in N^s} cad_{dl} Y D^s_{dt}(n)\hat{\kappa}^{10,s}_{dl}(n) \\
& + \sum_{b\in B}\sum_{t\in\Delta}\sum_{n\in N^s} cab_b Y B^s_{bt}(n)\hat{\kappa}^{11,s}_{bt}(n) \qquad (158) \\
& \boldsymbol{Y} \in \{0,1\} \qquad (159)
\end{aligned}
$$

where $\boldsymbol{\kappa^s}$ denotes extreme rays of Λ^s when the DSP is unbounded for a given location solution and scenario cluster sub-tree s. As for sub-tree $s = \underline{s}_{\eta^n}$, the terms of (156) that entail Lagrangean multipliers are written as follows.

$$
\begin{aligned}
& \sum_{n\in N_1}\sum_{t\in\Delta}\Bigg\{\sum_{a\in A}(\mu^{7,\bar{s}_{\eta^n}}_{at}(n) - \mu^{7,\underline{s}_{\eta^n}}_{at}(n)) Y A^{\underline{s}_{\eta^n}}_{at}(n) + \sum_{d\in D}(\mu^{8,\bar{s}_{\eta^n}}_{dt}(n) - \mu^{8,\underline{s}_{\eta^n}}_{dt}(n)) Y D^{\underline{s}_{\eta^n}}_{dt}(n) \\
& + \sum_{b\in B}(\mu^{9,\bar{s}_{\eta^n}}_{bt}(n) - \mu^{9,\underline{s}_{\eta^n}}_{bt}(n)) Y B^{\underline{s}_{\eta^n}}_{bt}(n)\Bigg\}
\end{aligned}
$$

At each iteration of the Benders decomposition algorithm, if the DSP is bounded, an optimality cut (157) is generated given a vector of optimal dual solutions. Otherwise, a feasibility cut (158) is introduced to the MP to eliminate values of location decisions for which the PSP is infeasible.

4.3.2.2.2 Pareto-optimal cuts

The degeneracy of the PSP implies that there exist multiple optimal solutions for the DSP such that each of these leads to a distinct optimality cut. An efficient implementation of Benders decomposition algorithm requires a cut selection scheme to choose the deepest cut among the various optimality cuts which can be generated by arbitrarily taking optimal dual solutions. Papadakos [9] proposed a dual selection strategy to expedite the Benders algorithm. In the context of the underlying problem, let Γ indicates the polyhedron defined as $\Gamma = \{\boldsymbol{Y} : (159) \text{ holds}\}$.

Definition 4.5 A core point is defined as any point $\boldsymbol{Y^0}$ in the relative interior of the convex hull of Γ, i.e., $\boldsymbol{Y^0} \in ri(\Gamma^c)$. Γ^c and $ri(.)$ indicate the convex hull and the relative interior of Γ, respectively.

Definition 4.6 An optimality cut (157) associated with $(\boldsymbol{v_1^{1,s}}, \boldsymbol{v_1^{7,s}}, \boldsymbol{v_1^{8,s}}, \boldsymbol{v_1^{9,s}}, \boldsymbol{v_1^{10,s}}, \boldsymbol{v_1^{11,s}}) \in \Lambda^s$ dominates the one that corresponds to $(\boldsymbol{v_2^{1,s}}, \boldsymbol{v_2^{7,s}}, \boldsymbol{v_2^{8,s}}, \boldsymbol{v_2^{9,s}}, \boldsymbol{v_2^{10,s}}, \boldsymbol{v_2^{11,s}}) \in \Lambda^s$ if and only if

$$
\begin{aligned}
&\rho(\hat{\boldsymbol{v}}_{\boldsymbol{1}}^{(\boldsymbol{m},\boldsymbol{s})^T}) + \sum_{a\in A}\sum_{t\in\Delta}\sum_{n\in N^s} caa_a Y A_{at}^s(n)\hat{v}_{1at}^{9,s}(n) \\
&+ \sum_{d\in D}\sum_{l\in L}\sum_{t\in\Delta}\sum_{n\in N^s} cad_{dl} Y D_{dt}^s(n)\hat{v}_{1dl}^{10,s}(n) + \sum_{b\in B}\sum_{t\in\Delta}\sum_{n\in N^s} cab_b Y B_{bt}^s(n)\hat{v}_{1bt}^{11,s}(n) \\
&\leq \rho(\hat{\boldsymbol{v}}_{\boldsymbol{2}}^{(\boldsymbol{m},\boldsymbol{s})^T}) + \sum_{a\in A}\sum_{t\in\Delta}\sum_{n\in N^s} caa_a Y A_{at}^s(n)\hat{v}_{2at}^{9,s}(n) \\
&+ \sum_{d\in D}\sum_{l\in L}\sum_{t\in\Delta}\sum_{n\in N^s} cad_{dl} Y D_{dt}^s(n)\hat{v}_{2dl}^{10,s}(n) + \sum_{b\in B}\sum_{t\in\Delta}\sum_{n\in N^s} cab_b Y B_{bt}^s(n)\hat{v}_{2bt}^{11,s}(n)
\end{aligned}
$$

for all $\boldsymbol{Y}$ with a strict inequality for at least one extreme point. A Pareto-optimal cut by definition is an optimality cut that is not dominated by any other cut. It can

be obtained through using the optimal solution of the following auxiliary DSP.

$$
\begin{aligned}
Z^s_{\boldsymbol{\upsilon}}(\boldsymbol{Y}^0) = \text{Min} \quad & \sum_{c\in C}\sum_{q\in Q}\sum_{t\in\Delta}\sum_{n\in N^s} \psi_{cqt}(n)\upsilon^{1,s}_{cqt}(n) \\
& + \sum_{o\in O}\sum_{p\in P}\sum_{t\in\Delta}\sum_{n\in N^s} ds_{opt}\upsilon^{7,s}_{opt}(n) + \sum_{w\in W}\sum_{l\in L}\sum_{t\in\Delta}\sum_{n\in N^s} dw_{wlt}\upsilon^{8,s}_{wlt}(n) \\
& + \sum_{a\in A}\sum_{t\in\Delta}\sum_{n\in N^s} ca_{at}YA^0_{at}(n)\upsilon^{9,s}_{at}(n) + \sum_{d\in D}\sum_{l\in L}\sum_{t\in\Delta}\sum_{n\in N^s} cad_{dl}YD^0_{dt}(n)\upsilon^{10,s}_{dl}(n) \\
& + \sum_{b\in B}\sum_{t\in\Delta}\sum_{n\in N^s} cab_{b}YB^0_{bt}(n)\upsilon^{11,s}_{bt}(n) \\
s.t. \quad & (\boldsymbol{\upsilon}^{\boldsymbol{1,s}}, \boldsymbol{\upsilon}^{\boldsymbol{2,s}}, ..., \boldsymbol{\upsilon}^{\boldsymbol{11,s}}) \in \Lambda^s
\end{aligned} \tag{160}
$$

In this modified Benders decomposition algorithm, one starts with an initial core point, i.e., $\boldsymbol{Y}^0 = \{0.5\}$, to build a Part-optimal cut to be added to the MP. In the subsequent iterations, when the solution to the MP yields a feasible PSP, the auxiliary DSP (160) is solved, using a new core point that is the convex combination of the MP solution and the previous value of the core point, to generate a non-dominated cut. To this end, a non-negative parameter, i.e., λ, is considered as the weight of the core point $\boldsymbol{Y}^0$ in the convex combination to update the value of the core point throughout the solution process. The value of this parameter is assigned to be 0.5 [46, 51]. The description of the proposed Benders decomposition-based method is outlined in Algorithm 8.

4.4 Numerical example

In this section, the performance of applying the solution scheme on the proposed model is investigated with respect to a set of test problems. To this end, first, the specific settings of the concerned case example is provided. The example is a typical large household appliance, i.e., a washing machine, that follows the settings of a

Algorithm 8 - Benders decomposition-based algorithm

$UB \leftarrow \infty$, $LB \leftarrow -\infty$, $\boldsymbol{Y^0} = \{0.5\}$, $\lambda \leftarrow 0.5$
while $(UB - LB)/UB \leq \epsilon$ **do**
Solve auxiliary-DRSP (160)
Add Pareto-optimal cut (157) to the MP
Solve the MP
Update UB
Solve the DSP
if the DSP is unbounded **then**
    Add the feasibility cut (158) to the MP
    $\boldsymbol{Y^0} \leftarrow \lambda \boldsymbol{Y^0} + \xi$
else
    Add the optimality cut (157) to the MP
    Update LB, if necessary
    $\boldsymbol{Y^0} \leftarrow \lambda \boldsymbol{Y^0} + (1 - \lambda)\overline{\boldsymbol{Y}}$
end if
end while
Solve the PSP

case study presented in [36]. It should be noted that the parameter settings of the proposed MS-MIP model are carefully estimated vis-à-vis recent market data and current CLSC/RSC network design literature ([19, 20]). Then, it is followed by the computational results section. In this study, all algorithms are implemented in C++ programming language using Concert Technology with IBM-ILOG CPLEX 12.60 on an Intel Quad Core 3.40 GHz with 8 GB RAM. Moreover, the default settings of CPLEX are employed to solve the DSP and the MP in the Benders decomposition algorithm.

4.4.1 Experimental design

The BOM of the washing machines is described in Table 20. More specifically, each washing machine consists of ten parts (e.g., balance) and two modules (e.g., motor). All used machines acquired in collection points are of two quality levels, i.e., high and poor.

Table 20: Components of the case example

Description	Value
Parts	washing tube:1 (3.5 kg), cover:1 (2.5 kg), balance:1 (2.5 kg), frame:1 (11.5 kg), condenser:1 (0.5 kg), hose:1 (1 kg), small electric parts:1 (1 kg), electric wire:1 (1 kg), transformer:1 (1 kg), PCB board:1 (0.5 kg)
Modules	motor:1 (5 kg), clutch:1 (4 kg)

The uncertain parameter, i.e., the quantity of returns, is normally distributed with a mean of 400 and a variance equal to 20% of the mean for high quality returns. As for the poor quality ones, the mean is considered to be 600 while the variance is equal to 20% of the mean. These normal distributions are then approximated by a 2-point discrete distribution (high and low ratio) through using the Gaussian quadrature method [49]. The time horizon is divided into three equal time periods such that each of them spans five years. Consequently, the time horizon is clustered into four stages (stage zero is the present time). Moreover, the scenario tree of the stochastic quantity of returns entails fifteen nodes and eight scenarios.

In the *Appendices*, in the section entitled *Parameter settings*, as depicted in Tables 24 - 28, a summary of other parameters used in the case example is provided. Besides, shipping costs are selected from $Uniform(4, 7)$ for the used washing machines, $Uniform(1, 4)$ for each type of components, and $Uniform(0.1, 0.5)$ for bulk of residues. Capacities of facilities are randomly generated aligned with the stochastic quantity of returns and the BOM. For example, the capacity of disassembly centers are chosen between $Uniform(2 \times MeanCaa, 3 \times MeanCaa)$ where $MeanCaa = |C| \times (400 + 600)/|A|$. Moreover, the fixed cost of installing a facility is proportional to its capacity, so that a facility with high capacity level requires a greater investment.

In order to carry out the experiments, four main classes within each five test instances are considered as shown in Table 21. The detailed information on the size

of the classes including the number of constraints, continuous variables, and binary variables are shown in Table 22. It is worth noting that the largest class of test instances (C4) reasonably reflects real-size RSCs in the context of durable products.

Table 21: Description of classes

Class	C	A	D	B	O	W
1	40	5	5	2	20	20
2	50	10	10	5	25	25
3	60	10	10	5	30	30
4	70	15	15	7	35	35

Table 22: Size of test instances

Class	# Constraint	# Continuous variable	# Binary variable
1	5992	28014	168
2	8680	68390	350
3	9800	80290	350
4	12432	140434	518

4.4.2 Computational results

On each of the twenty test instances, the proposed decomposition scheme, i.e., HSCD, is applied to find an upper bound within the stopping criteria, i.e., either the sub-gradient vector is less than 1% or the current value of the upper bound is not improved in 5 iterations. The second stage is chosen as the break stage leading to four scenario cluster sub-models. As for the resolution of each scenario cluster sub-model, the Benders decomposition-based algorithm described in the preceding section is employed with the stopping criteria of either 1% optimality gap or 3600 seconds time limitation. Alternatively, for the sake of comparison, considering the first stage as the break stage, each test instance is also solved using the Benders decomposition-based algorithm where the termination condition is either a time limit of 24 hours or an optimality gap of 1% for each of the resulting two scenario clusters. It should be

noted that by decomposing the scenario tree in the first stage, two sub-models are obtained such that each of them can be independently solved. The optimal solution of each sub-model individually yields a sub-optimal solution to the optimal solution of the MS-MIP model. Note that it is not required to impose any NAC in the first stage of the tree as it corresponds to time period zero where the initial inventory levels are zero.

Table 23 presents the results reported by HSCD and Benders decomposition algorithm algorithms for all twenty test instances. For the former approach, columns "Time" and "#Iteration" indicate the total CPU time in seconds and the number of iterations, respectively. The fourth column shows the best upper bound on the MS-MIP model identified through applying HSCD. As for the latter approach, column "Time" indicates the amount of time required to solve the MS-MIP model (129)-(144) within 1% of optimality gap while column "Profit" gives the value of the objective function within the dedicated time limit and the optimality gap. Moreover, column "gp(%)" denotes the relative difference between upper and lower bounds reported by Benders decomposition algorithm within 24 hours running. It should be stated that the runtime of BD approach is considered as the maximum of the solution times of the two scenario cluster sub-models in each test instance. The last column, "Gap(%), expresses the relative difference in percentage between the solutions obtained by the two approaches.

The results show that the performance of the HSCD scheme is quite promising in the sense that it provides an upper bound on the objective function of the MS-MIP model in significantly less amount of time compared to the BD approach, i.e, on average 2 hours over all test instances. Given the low gap values in the last column (0.58% on average), even though the solutions are not necessarily feasible, they are quite close to those provided by the BD approach. More precisely, the

Table 23: Comparison of HSCD and BD algorithms

Class	HSCD			BD			Gap(%)
	Time (sec)	# Iteration	$Z_{MSCLD}(\mu, s)$	Time (sec)	gp(%)	Profit	
	132	10	7,662,040	110	≤ 1	7,650,870	0.14
	183	9	6,412,000	120	≤ 1	6,351,610	0.94
C1	319	10	7,600,490	1371	≤ 1	7,596,140	0.06
	160	8	7,564,320	182	≤ 1	7,443,070	1.60
	422	10	6,881,080	2019	≤ 1	6,864,140	0.25
	5695	12	7,201,340	$\geq 24hr$	1.15	7,106,850	1.31
	2231	10	6,622,640	54701	≤ 1	6,611,560	0.17
C2	8391	15	6,568,970	$\geq 24hr$	1.10	6,558,930	0.15
	7729	13	6,296,800	$\geq 24hr$	1.40	6,280,200	0.26
	7100	15	7,095,570	$\geq 24hr$	1.12	6,978,190	1.65
	4190	13	9,854,950	$\geq 24hr$	1.50	9,684,640	1.73
	9113	16	9,617,400	$\geq 24hr$	1.14	9,562470	0.57
C3	5445	13	9,577,230	$\geq 24hr$	1.10	9,568740	0.09
	7184	14	9,643,130	61971	≤ 1	9,626,300	0.17
	7081	14	9,481,810	$\geq 24hr$	1.20	9,378,090	1.10
	13593	19	8,676,980	$\geq 24hr$	1.98	8,631,030	0.53
	16050	21	9,434,940	$\geq 24hr$	1.63	9,431,570	0.04
C4	17171	23	8,612,290	$\geq 24hr$	1.20	8,611,640	0.01
	19265	25	9,874,380	$\geq 24hr$	1.53	9,839,310	0.35
	14622	19	8,617,780	$\geq 24hr$	1.11	8,578,770	0.46
Average	7304	15	-	-	-	-	0.58

average solution time of HSCD in solving the test instances of C1, C2, and C3 is, respectively, 4 minutes, 1.7 hours, and 1.9 hours. It increases to 4.5 hours for the last class of test problems. Furthermore, the infeasibility rate of the dualized NACs in the HSCD algorithm is on average 1% or less for each class of test problems.

On other hand, except the test instances of the first class, the second instance of C2, and the fourth instance of C3, the BD algorithm is unable to obtain the optimal solution of the concerned test instances within the dedicated time limit and the optimality gap. Particularly, once the results of C1 for which the optimal solutions are given by the decomposition method are concerned, HSCD provides the upper bounds close to the optimal solution, i.e., 0.58% on average. As for other instances, high quality feasible solutions are reported by BD after the 24 hours time limit. More

specifically, the optimality gap of the algorithm reported is less on 1.8% on average for such instances. In the targeted instances, HSCD also provides an upper bound close to the feasible solution identified by Benders decomposition as shown in the last column of Table 23.

4.5 Concluding remarks

In this study, a reverse supply chain network problem in a multi-period setting was addressed for taking back and recovery of used products that are of heterogeneous quality states. The concerned problem arises in the context of durable products which typically are composed of many components. As the inherent uncertainty in quantity of returns is assumed to evolve as a discrete time stochastic process during the planning horizon, a scenario tree was generated to model the uncertain parameter. The resulting multi-stage decision making problem was modeled as a MS-MIP model to address the decisions on the location of facilities and the quantity of flows in the reverse supply chain network.

In order to solve the proposed model for realistic sizes, a heuristic scenario clustering decomposition was proposed which mainly decomposes the scenario tree into a set of cluster of scenarios. The scenario clusters were independently solved by the Benders decomposition-based algorithm and coordinated in an implementable solution thorough a Lagrangean Progressive hedging-based scheme. The proposed solution scheme provided good upper bounds on the objective function of the original stochastic model in a reasonable amount of running time. It can be noticed not only by the closeness of the upper bounds to the solutions reported by the Benders decomposition-based algorithm, but also by the fact that the infeasibility rate of the dualized NACs is small for each class of test problems.

Given the multi-period setting, the underlying problem can be extended through

accounting uncertainty in quality status of the return stream and demands. Another promising venue of research is to address the willingness of durable goods holders to return their used units by means of financial incentives.

Acknowledgment

The authors gratefully acknowledge the financial support from Le Fonds de recherche du Québec-Nature et technologies (FRQNT) and the Natural Sciences and Engineering Research Council of Canada (NSERC).

4.6 Appendices

4.6.1 Problem notations

Indices

$Tree$: Scenario tree

C: Set of collection zones

A: Set of disassembly centers

D: Set of remanufacturing centers

B: Set of bulk recycling centers

W: Set of secondary markets for modules

O: Set of secondary markets for spare parts

L: Set of modules

P: Set of parts

R: Set of raw materials

n, m: Nodes of the scenario tree

$a(n)$: Immediate predecessor of node n in the scenario tree

T: Set of time periods

Parameters

$\psi_{cqt}(n)$: Quantity of returns with quality level q in collection zone c in period t at node n of the scenario tree

β_q: The mass of residues in the returned product with quality level q shipped to bulk recycling centers from disassembly centers

γ_{pq}: The number of reusable part p in the returned product with quality level q shipped to secondary markets from disassembly centers

δ_{lq}: The number of remanufacturable module l in the returned product with quality level q shipped to remanufacturing centers from disassembly centers

η_r: The ratio of recyclable material r

fa_a: Fixed cost of opening disassembly center a

fd_d: Fixed cost of opening remanufacturing center d

fb_b: Fixed cost of opening bulk recycling center b

ta_{ca}: Shipping cost per unit of the returned product from collection zone c to disassembly center a

tb_{ab}: Shipping cost per kg of residues from disassembly center a to bulk recycling center b

ts_{aop}: Shipping cost per unit of part p from disassembly center a to spare market o

td_{adl}: Shipping cost per unit of module l from disassembly center a to remanufacturing center d

tw_{dwl}: Shipping cost per unit of module l from remanufacturing center d to secondary market w

caa_a: Capacity of disassembly center a

cad_{dl}: Capacity of remanufacturing center d for module l

ds_{opt}: Demand for part p at spare market o in period t

dw_{wlt}: Demand for module l at secondary market w in period t

ca_{aq}: Processing cost per unit of the returned product with quality level q at disassembly center a

cd_{dl}: Remanufacturing cost per unit of module l at remanufacturing center d

hp_p: Unit holding cost of part p in disassembly centers

hl_l: Unit holding cost of module l in disassembly centers/remanufacturing centers

hb: Unit holding cost of residues in disassembly centers

Ps_p: Unit price of part p at spare parts markets

Pw_l: Unit price of module l at module markets

Pe_r: Unit price of selling recyclable raw materials to the third-party provider

$pr(n)$: Probability of node n of the scenario tree

Decision variables

$YA_{at}(n)$: A binary variable which is equal to one if disassembly center a is opened in period t at node n of the scenario tree and zero otherwise

$YD_{dht}(n)$: A binary variable which is equal to one if remanufacturing center z is opened in period t at node n of the scenario tree and zero otherwise

$YB_{bt}(n)$: A binary variable which is equal to one if bulk recycling center b is opened in period t at node n of the scenario tree and zero otherwise

$QA_{caqt}(n)$: The quantity of returns with quality level q shipped from collection zone c to disassembly center a in period t at node n of the scenario tree

$QS_{aopt}(n)$: The number of part p shipped from disassembly center a to spare parts market o in period t at node n of the scenario tree

$QD_{adlt}(n)$: The number of module l shipped from disassembly center a to remanufacturing center d in period t at node n of the scenario tree

$QB_{abt}(n)$: The quantity of residues shipped from disassembly center a to bulk recycling center b in period t at node n of the scenario tree

$BR_{brt}(n)$: The quantity of recyclable material r purchased by the third-party provider from bulk recycling center b in period t at node n of the scenario tree

$QW_{dwlt}(n)$: The number of module l shipped from remanufacturing center d to secondary market w in period t at node n of the scenario tree

$IP_{apt}(n)$: Inventory level of part p in disassembly center a by the end of period t at node n of the scenario tree

$IL_{alt}(n)$: Inventory level of module l in disassembly center a by the end of period t at node n of the scenario tree

$IB_{at}(n)$: Inventory level of residues in disassembly center a by the end of period t at node n of the scenario tree

$ID_{dlt}(n)$: Inventory level of module l in remanufacturing center d by the end of period t at node n of the scenario tree

4.6.2 Parameter settings

Tables 24 to 28 present a summary of parameter settings of the proposed MS-MIP model

Table 24: Settings for modules

Description	Value	
	Motor	Clutch
Pw_l	150	75
hl_l	3	3

Table 25: Settings for raw materials

Description	Value		
	Plastic	Steel	Copper
pe_r	0.75	0.5	3
η_r	0.3	0.3	0.3

Table 26: Settings for parts

Type of part	Value	
	Ps_p	hp_p
Washing tube	40	1.5
Cover	10	1.5
Balance	50	1.5
Frame	10	1.5
Condenser	30	1.5
Transformer	30	1.5
Small electric	10	1.5
Hose	40	1.5
Electric wire	40	1.5
PCB board	70	1.5

Table 27: Settings for quality level-dependent parameters

Parameter	Quality levels	
	High	Poor
δ_{lq}	1, 1	0, 1
γ_{pq}	1, 1, 1, 1, 1, 1, 1, 1, 1, 1	0, 0, 1, 0, 0, 0, 0, 0, 0, 1
β_q	3	30
ca_{aq}	1	2

Table 28: Other parameter settings

Description	Value	Description	Value
cb_b	2	hb	1
ds_{opt}	$\{200, 201, ..., 400\}$	dw_{wlt}	$\{200, 201, ..., 400\}$
fa_a	$Uniform$(400000, 600000)	fd_d	$Uniform$(700000, 900000)
fb_b	$Uniform$(200000, 400000)	cd_{dl}	3

Chapter 5

Conclusion and Future Work

5.1 Concluding Remarks

This thesis addressed closed-loop and reverse supply chain planning problems applicable in the context of durable products distinguished by their modular structure and their long life cycle. It accounted for several recovery activities plausible in practice associated with the generic reverse bill of materials of such category of products. Inspired by a real-life case example, i.e., washing machines, we investigated the performance of the solution approaches on instances with realistic sizes. It was shown that the problem instances can be solved within reasonable amount of times utilizing the proposed decomposition-based solution schemes.

In the second chapter, given a generic reverse bill of materials, we presented a closed-loop supply chain network design problem in which the returns stream is of non-homogeneous quality status. Then, on the methodological side, we developed a variant of Benders decomposition algorithm, which was able to solve all test instances to optimality less than an hour of running time. Our further analysis indicated that a viable strategy to recycle the bulk of residues under strict regulations is to outsource the administration of the recycling precess to a third-party provider.

In the third chapter, focusing on a profit-oriented approach, we proposed a two-stage mixed-integer stochastic programming model to explicitly incorporate random quality status of returns into the closed-loop supply chain planning problem. More precisely, the random quality state was modeled as a set of discrete scenarios with Bernoulli probability distribution. Observing a large number of scenarios that can be realized depending on the number of components in the reverse bill of material of a typical durable product, we adapted fast forward selection algorithm to the problem of interest to preserve the most pertinent scenarios. Then, as the stochastic model with smaller number of recourse problems was not amenable to solve with commercial optimization packages, we developed an enhanced decomposition solution approach based on L-shaped algorithm. Computational results denoted the effectiveness of the proposed solution algorithm. Moreover, the solution method along with the scenario reduction scheme provided insight on possible trade-offs between computational time and the accuracy of solutions.

Finally, in the fourth chapter, we considered a multi-period reverse supply chain network design problem to determine the location of facilities and the quantity of physical flows in the reverse channel in each period. The returned products were characterized by high and poor quality levels. However, the quantity of returns was unknown a priori and evolved over the planning horizon calling for adopting a multi-stage stochastic programming approach. The resulting large-scale optimization problem was solved via a heuristic scenario clustering decomposition approach such that scenario cluster sub-models were coordinated by a Lagrangean Progressive hedging-based method. Furthermore, each scenario cluster sub-model was solved by a Benders decomposition-based algorithm. Results demonstrated that the solution scheme provided a strong upper bound while the infeasibility rate of the dualized non-anticipativity constraints was quite small for each test instance.

5.2 Future research directions

Immediate extensions of this thesis can revolve around the following directions.

- Investigating financial incentive mechanisms that can be differentiated based on the quality status of used products, e.g., subsidy per return, is an interesting area for further research. Such incentives can be integrated with a hybrid drop-off/pickup collection strategy, which would have significant cost and profit implications,

- Considering the variants of the proposed models accounting for uncertainty in demands at the secondary markets and economic parameters such as transportation cost,

- Developing appropriate and quantifiable metrics to further assess the environmental impact of the closed-loop/reverse supply chain networks, e.g., CO_2 emission and obnoxious effects of the recovery facilities,

- Another interesting research direction is to focus on a multi-product setting where the closed-loop/reverse supply chain networks are generalized to a family of durable products,

- Developing efficient solution algorithms for the variants mentioned above would be another promising area of research.

Bibliography

[1] A. Knox, An overview of incineration and EFW technology as applied to the management of municipal solid waste (MSW). ONEIA Energy Subcommitte .

[2] M. Fleischmann, J. A. Van Nunen, B. Gräve, Integrating closed-loop supply chains and spare-parts management at IBM, Interfaces 33 (6) (2003) 44–56.

[3] V. D. R. Guide Jr, L. N. Van Wassenhove, The reverse supply chain, Harvard Business Review 80 (2) (2002) 25–26.

[4] H. Krikke, J. Bloemhof-Ruwaard, L. Van Wassenhove, Concurrent product and closed-loop supply chain design with an application to refrigerators, International journal of production research 41 (16) (2003) 3689–3719.

[5] J. F. Benders, Partitioning procedures for solving mixed-variables programming problems, Numerische Mathematik 4 (1) (1962) 238–252.

[6] H. Üster, G. Easwaran, E. Akçali, S. Cetinkaya, Benders decomposition with alternative multiple cuts for a multi-product closed-loop supply chain network design model, Naval Research Logistics (NRL) 54 (8) (2007) 890–907.

[7] M. Pishvaee, J. Razmi, S. A. Torabi, An accelerated Benders decomposition algorithm for sustainable supply chain network design under uncertainty: A case study of medical needle and syringe supply chain, Transportation Research Part E: Logistics and Transportation Review 67 (2014) 14–38.

[8] M. Fischetti, A. Lodi, Local branching, Mathematical programming 98 (1-3) (2003) 23–47.

[9] N. Papadakos, Practical enhancements to the Magnanti–Wong method, Operations Research Letters 36 (4) (2008) 444–449.

[10] E. Akçalı, S. Çetinkaya, H. Üster, Network design for reverse and closed-loop supply chains: An annotated bibliography of models and solution approaches, Networks 53 (3) (2009) 231–248.

[11] N. Aras, T. Boyaci, V. Verter, Designing the reverse logistics network. In: Ferguson, M., Souza, G. (Eds.), Closed-Loop Supply Chains: New Developments to Improve the Sustainability of Business Practices. CRC Press (2010) 67–97.

[12] K. Govindan, H. Soleimani, D. Kannan, Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future, European Journal of Operational Research 240 (3) (2014) 603–626.

[13] M. Fleischmann, P. Beullens, J. M. Bloemhof-Ruwaard, L. N. Wassenhove, The impact of product recovery on logistics network design, Production and Operations Management 10 (2) (2001) 156–173.

[14] Z. Lu, N. Bostel, A facility location model for logistics systems including reverse flows: The case of remanufacturing activities, Computers & Operations Research 34 (2) (2007) 299–323.

[15] V. Jayaraman, V. Guide Jr, R. Srivastava, A closed-loop logistics model for remanufacturing, Journal of the Operational Research Society 50 (5) (1999) 497–508.

[16] N. Ö. Demirel, H. Gökçen, A mixed integer programming model for remanufacturing in reverse logistics environment, The International Journal of Advanced Manufacturing Technology 39 (11-12) (2008) 1197–1206.

[17] H. Min, H.-J. Ko, The dynamic design of a reverse logistics network from the perspective of third-party logistics service providers, International Journal of Production Economics 113 (1) (2008) 176–192.

[18] H. Soleimani, M. Seyyed-Esfahani, M. A. Shirazi, A new multi-criteria scenario-based solution approach for stochastic forward/reverse supply chain network design, Annals of Operations Research (2013) 1–23.

[19] S. A. Alumur, S. Nickel, F. Saldanha-da Gama, V. Verter, Multi-period reverse logistics network design, European Journal of Operational Research 220 (1) (2012) 67–78.

[20] O. Listeş, A generic stochastic model for supply-and-return network design, Computers & Operations Research 34 (2) (2007) 417–442.

[21] M. I. G. Salema, A. P. Barbosa-Povoa, A. Q. Novais, An optimization model for the design of a capacitated multi-product reverse logistics network with uncertainty, European Journal of Operational Research 179 (3) (2007) 1063–1077.

[22] S. R. Cardoso, A. P. F. Barbosa-Póvoa, S. Relvas, Design and planning of supply chains with integration of reverse logistics activities under demand uncertainty, European Journal of Operational Research 226 (3) (2013) 436–451.

[23] L. J. Zeballos, C. A. Méndez, A. P. Barbosa-Povoa, A. Q. Novais, Multi-period design and planning of closed-loop supply chains with uncertain supply and demand, Computers & Chemical Engineering 66 (2014) 151–164.

[24] N. Aras, D. Aksen, A. Gönül Tanuğur, Locating collection centers for incentive-dependent returns under a pick-up policy with capacitated vehicles, European Journal of Operational Research 191 (3) (2008) 1223–1240.

[25] N. Aras, D. Aksen, Locating collection centers for distance-and incentive-dependent returns, International Journal of Production Economics 111 (2) (2008) 316–333.

[26] A. M. Geoffrion, G. W. Graves, Multicommodity distribution system design by Benders decomposition, Management science 20 (5) (1974) 822–844.

[27] F. Oliveira, I. E. Grossmann, S. Hamacher, Accelerating Benders stochastic decomposition for the optimization under uncertainty of the petroleum product supply chain, Computers & Operations Research 49 (2014) 47–58.

[28] T. Santoso, S. Ahmed, M. Goetschalckx, A. Shapiro, A stochastic programming approach for supply chain network design under uncertainty, European Journal of Operational Research 167 (1) (2005) 96–115.

[29] D. McDaniel, M. Devine, A modified Benders' partitioning algorithm for mixed integer programming, Management Science 24 (3) (1977) 312–319.

[30] T. L. Magnanti, R. T. Wong, Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria, Operations Research 29 (3) (1981) 464–484.

[31] M. Fischetti, D. Salvagnin, A. Zanette, Minimal infeasible subsystems and Benders cuts, Technical Report, University of Padova .

[32] M. Fischetti, D. Salvagnin, A. Zanette, A note on the selection of Benders' cuts, Mathematical Programming 124 (1-2) (2010) 175–182.

[33] H. D. Sherali, B. J. Lunday, On generating maximal nondominated Benders cuts, Annals of Operations Research 210 (1) (2013) 57–72.

[34] W. Rei, J.-F. Cordeau, M. Gendreau, P. Soriano, Accelerating Benders decomposition by local branching, INFORMS Journal on Computing 21 (2) (2009) 333–345.

[35] N. Papadakos, Integrated airline scheduling, Computers & Operations Research 36 (1) (2009) 176–195.

[36] P.-J. Park, K. Tahara, I.-T. Jeong, K.-M. Lee, Comparison of four methods for integrating environmental and economic aspects in the end-of-life stage of a washing machine, Resources, conservation and recycling 48 (1) (2006) 71–85.

[37] V. D. R. Guide Jr, L. N. Van Wassenhove, OR FORUM-the evolution of closed-loop supply chain research, Operations research 57 (1) (2009) 10–18.

[38] M. Fleischmann, H. R. Krikke, R. Dekker, S. D. P. Flapper, A characterisation of logistics networks for product recovery, Omega 28 (6) (2000) 653–666.

[39] K. Das, A. H. Chowdhury, Designing a reverse logistics network for optimal collection, recovery and quality-based product-mix planning, International Journal of Production Economics 135 (1) (2012) 209–221.

[40] L. J. Zeballos, M. I. Gomes, A. P. Barbosa-Povoa, A. Q. Novais, Addressing the uncertain quality and quantity of returns in closed-loop supply chains, Computers & Chemical Engineering 47 (2012) 237–247.

[41] W. Chen, B. Kucukyazici, V. Verter, M. J. Sáenz, Supply chain design for unlocking the value of remanufacturing under uncertainty, European Journal of Operational Research 247 (3) (2015) 804–819.

[42] J. R. Birge, F. Louveaux, Introduction to stochastic programming, Springer Science & Business Media, 2011.

[43] H. Heitsch, W. Römisch, Scenario reduction algorithms in stochastic programming, Computational optimization and applications 24 (2-3) (2003) 187–206.

[44] J. Dupačová, N. Gröwe-Kuska, W. Römisch, Scenario reduction in stochastic programming, Mathematical programming 95 (3) (2003) 493–511.

[45] R. M. Van Slyke, R. Wets, L-shaped linear programs with applications to optimal control and stochastic programming, SIAM Journal on Applied Mathematics 17 (4) (1969) 638–663.

[46] M. Jeihoonian, M. Kazemi Zanjani, M. Gendreau, Accelerating Benders Decomposition for Closed-Loop Supply Chain Network Design: Case of Used Durable Products with Different Quality Levels, European Journal of Operational Research 251 (3) (2016) 830–845.

[47] M. S. Pishvaee, F. Jolai, J. Razmi, A stochastic optimization model for integrated forward/reverse logistics network design, Journal of Manufacturing Systems 28 (4) (2009) 107–114.

[48] M. C. Fonseca, Á. García-Sánchez, M. Ortega-Mier, F. Saldanha-da Gama, A stochastic bi-objective location model for strategic reverse logistics, Top 18 (1) (2010) 158–184.

[49] A. C. Miller III, T. R. Rice, Discrete approximations of probability distributions, Management science 29 (3) (1983) 352–362.

[50] M. I. G. Salema, A. P. Barbosa-Povoa, A. Q. Novais, Simultaneous design and planning of supply chains with reverse flows: a generic modelling framework, European Journal of Operational Research 203 (2) (2010) 336–349.

[51] M. Jeihoonian, M. Kazemi Zanjani, M. Gendreau, Closed-Loop Supply Chain Network Design under Uncertain Quality Status: Case of Durable Products, International Journal of Production Economics (In press).

[52] L. F. Escudero, M. A. Garín, M. Merino, G. Pérez, On BFC-MSMIP strategies for scenario cluster partitioning, and twin node family branching selection and bounding for multistage stochastic mixed integer programming, Computers & Operations Research 37 (4) (2010) 738–753.

[53] P.-L. Carpentier, M. Gendreau, F. Bastin, Optimal scenario set partitioning for multistage stochastic programming with the progressive hedging algorithm, Technical report CIRRELT-2013-55, CIRRELT .

[54] L. F. Escudero, M. A. Garín, A. Unzueta, Cluster Lagrangean decomposition in multistage stochastic optimization, Computers & Operations Research 67 (2016) 48–62.

[55] M. Kazemi Zanjani, O. S. Bajgiran, M. Nourelfath, A hybrid scenario cluster decomposition algorithm for supply chain tactical planning under uncertainty, European Journal of Operational Research 252 (2) (2016) 466–476.

[56] R. T. Rockafellar, R. J.-B. Wets, Scenarios and policy aggregation in optimization under uncertainty, Mathematics of operations research 16 (1) (1991) 119–147.

[57] L. F. Escudero, M. A. Garín, G. Pérez, A. Unzueta, Scenario Cluster Decomposition of the Lagrangian dual in two-stage stochastic mixed 0–1 optimization, Computers & Operations Research 40 (1) (2013) 362–377.