CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

# Accelerating Benders Decomposition for Closed-Loop Supply Chain Network Design: Case of Used Durable Products with Different Quality Levels

**Mohammad Jeihoonian**
**Masoumeh Kazemi Zanjani**
**Michel Gendreau**

**November 2014**

**CIRRELT-2014-58**

**Bureaux de Montréal :**
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

**Bureaux de Québec :**
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

# Accelerating Benders Decomposition for Closed-Loop Supply Chain Network Design: Case of Used Durable Products with Different Quality Levels

**Mohammad Jeihoonian[1,2,*], Masoumeh Kazemi Zanjani[1,2] , Michel Gendreau[1,3]**

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Mechanical and Industrial Engineering, Concordia University, 1515, St-Catherine, St. West, EV4.243, Montréal, Canada H3G 1M8

[3] Department of Mathematics and Industrial Engineering, Polytechnique Montréal, P.O. Box 6079, Station Centre-Ville, Montréal, Canada H3C 3A7

**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 generic closed-loop supply chain which incorporates the variable quality of used durable products and their disassembly tree. To this end, we propose a mixed-integer programming model to determine the location of different 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.

**Keywords**. Closed-loop supply chain design, durable products, disassembly tree, Benders decomposition, local branching.

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



---

* Corresponding author: Mohammad.Jeihoonian@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2014

## 1. Introduction

Landfilling of End-Of-Life (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, original Equipment Manufacturers (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 include not only the conventional forward flow, but also the reverse flow from the customers to the manufacturer. As pointed out by Guide and Van Wassenhove [1], OEMs that have been most successful with their reverse supply chains are those that closely coordinate it with the forward supply chain, initiating the closed-loop supply chain (CLSC). In a CLSC, the role of the reverse supply chain (RSC) is to collect used products from end-users, inspect and sort them as needed, ship them to various recovery options, 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 different recovery process. In the context of durable products with long life cycle, it can be expected that the majority of the returned items is composed of poor quality returns with a small number of recoverable modules and parts [2]. In other words, only a small portion of the return stream 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 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 mixed-integer programming (MIP) model which formulates CLSC network design for the case of 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 of returns has been rarely investigated in the design of CLSC and RSC networks. Hence, as another contribution, we assume that the return stream fits into various quality levels. More precisely, warranty or damaged returns are usually categorized as the high quality stream, while EOL items are assigned to the poor quality stream. We also take into account the legislative target for the recovery returns as a constraint in the MIP model 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 several types of recovery facilities in the reverse network as well as the comprehensive generic disassembly tree, we develop an accelerated solution method based on Benders decomposition [3]. Regarding that the previous studies in the context of CLSC and RSC design addressed less complicated networks, their Benders decomposition algorithmic scheme were limited to the generation of multiple optimality cuts, cut strengthening, and introducing valid inequalities and the trust region constraint to the master problem [4, 5]. 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 Pareto-optimal cuts to exclude a larger space of the master problem; and, 3) Local branching search to concurrently improve both lower and upper bounds during the execution of the solution algorithm to solve such a large-scale optimization problem.

In summary, the main contribution of our study is twofold. First of all, we formulate a CLSC network design problem based on a generic disassembly tree corresponding to durable products incorporating remanufacturing, part harvesting, bulk and material recycling, and landfill as recovery options.

Moreover, we address the non-homogeneous quality state of returns. 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 subassemblies of a product. Secondly, we propose an exact solution algorithm in conjunction with several computational enhancements.

The remainder of this article is organized as follows. In the next section, we provide the overview of the most relevant literature. In Section 3, we provide more details on the problem investigated in this paper and then present its formulation. Section 4 describes the solution methodology. Computational experiments are presented in Section 5. Conclusion and future areas of research are provided in Section 6.

## 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 [6–8]. We also provide a review of relevant existing algorithmic improvements for Benders decomposition method.

### *2.1. CLSC and RSC network design models*

As an early study in CLSC network design, Fleischmann et al. [9] proposed a MIP model for designing a generic CLSC network including uncapacitated disassembly and remanufacturing facilities in the reverse channel. Krikke et al. [2] proposed a multi-objective CLSC network design in which the objective is to minimize cost and environmental impacts measured by energy and waste. Pishvaee et al. [5] developed a Benders decomposition-based algorithm to determine the location and capacity decisions in a medical RSC.

The multi-product CLSCs were studied by [4, 10, 11]. More recently, Alumur et al. [12] presented a multi-product formulation for RSC network design while considering the reverse bill-of-material. The proposed model was applied for a case study of washing machines and tumble dryers in Germany.

Listeş [13] 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. For recent papers on CLSC network design under uncertainty, the reader is referred to [14, 15].

Quality status of used products has been considered by Aras et al. [16]. 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 [17].

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 take into account 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 consider 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 the Benders decomposition-based approaches applied in previous CLSC/RSC network design studies, the complex structure of the CLSC considered in this article requires developing a more sophisticated method to solve the resulting MIP model.

### *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 model into a pure integer and linear subproblems, consisting of the integer and the continuous variables of the original problem referred to as master problem and dual subproblem, respectively. These subproblems are then solved sequentially and iteratively until a termination criterion, such as a small gap, is satisfied.

Several techniques have been proposed to enhance the performance of Benders decomposition. McDaniel and Devine [18] suggested the relaxation of the integrality constraints in the master problem to obtain a set of initial optimality cuts. Magnanti and Wong [19] 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. This approach suffers from two major drawbacks: 1) The normalization constraint added to the auxiliary dual subproblem is numerically unstable; 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 [20] 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. Concerning the difficulty of solving the master problem and the values of lower and upper bounds, Rei et al. [21] recently applied a local branching strategy, introduced for the first time by Fischetti and Lodi [22], to improve both lower and upper bounds simultaneously and accordingly accelerate the Benders decomposition algorithm. Finally, Sherali and Lunday [23] proposed the idea of generating maximal non-dominated cuts to speed up the Benders algorithm.

## 3. Problem statement

### *3.1. Durable product structure*

As pointed out earlier, durable products have a multi-indenture structure. They consist of multiple and various types of components as shown in the disassembly tree in Figure 1. 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, such as a washing tube or PBC boards 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 are 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).

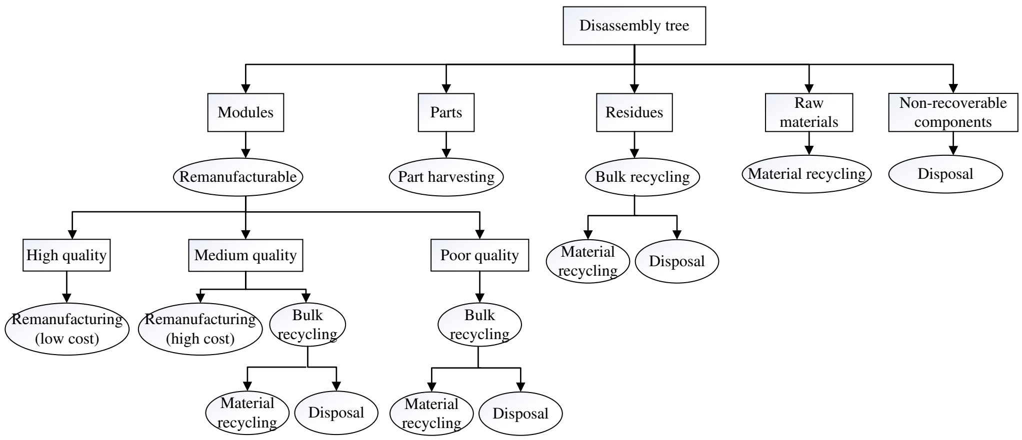


Figure 1: Disassembly tree of a generic durable product

### *3.2. CLSC network configuration*

In the CLSC network under consideration, the OEM manages an established forward supply chain that consists of three categories of suppliers, manufacturing facilities, distribution centers, and end-users. The brand-new durable products are shipped form 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. According to governmental regulations concerning environmental issues, it is supposed that a substantial portion of the total quantity of returns acquired in collection points must be treated in the reverse network. It leads to introducing the environmentally friendly constraint into the CLSC network design model while taking the variable quality of the return stream into consideration. 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 used for two purposes: 1) Conveying to manufacturing facilities to deploy in manufacturing of the brand-new products; and, 2) Selling at secondary markets. Given the above description, the conceptual structure of the CLSC is schematically illustrated in Figure 2. The solid arcs indicate the forward flows while the dashed ones denote the reverse flows in the CLSC under consideration.

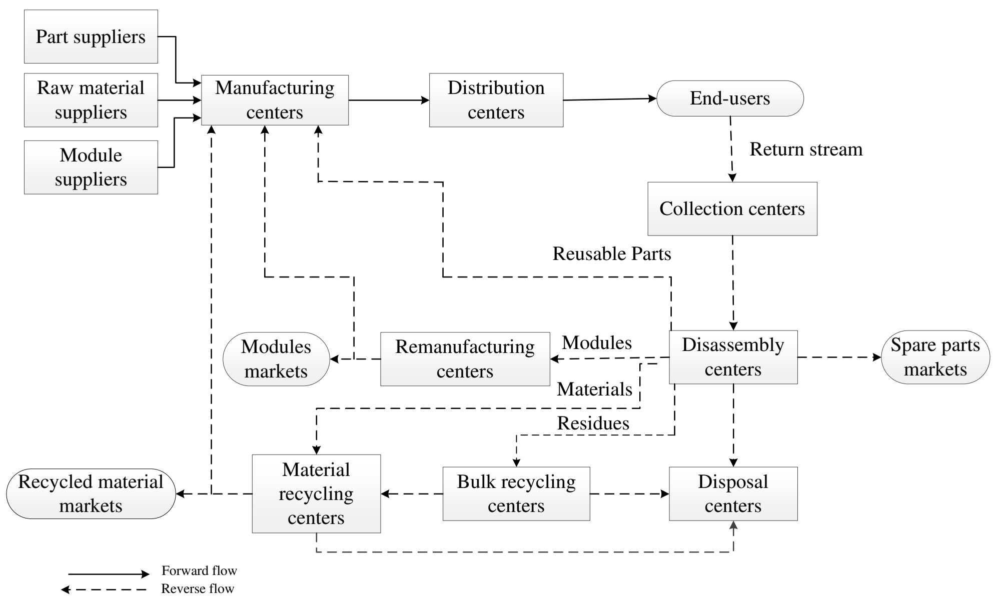


Figure 2: Conceptual framework for the CLSC network

### *3.3. Problem formulation*

Prior to presenting the mathematical model, the notations are summarized as follows. Let $Z$ be the set of part suppliers, $U$ be the set of raw material suppliers, $X$ be the set of module suppliers, $I$ be the set of manufacturing centers, $J$ be the set of distribution centers, $K$ be the set of end-user zones, $C$ be the set of collection centers, $A$ be the set of disassembly centers, $S$ be the set of spare parts markets, $M$ be the set of remanufacturing centers, $W$ be the set of remanufactured modules markets, $B$ be the set of bulk recycling centers, $G$ be the set of material recycling centers, $E$ be the set of recycled materials markets, $D$ be the set of disposal centers, $P$ be the set of parts in the product, $R$ be the set of raw materials in the product, $L$ be the set of modules in the product, and finally, $Q$ be the set of quality levels of returns. Let $[Pk_k, Pw_l, Ps_p, Pe_r]$ be unit prices. Let $[fc_c, fa_a, fm_m, fb_b, fg_g, fd_d]$ be fixed costs of opening collection, disassembly, remanufacturing, bulk recycling, material recycling, and disposal centers. Let $[cz_{zp}, cu_{ur}, cx_{xl}, Pr_q]$ be unit procurement costs of new parts, raw materials, modules from suppliers, and acquisition price of returns. Let $[ci_i, cj_j, cc_{cq}, ca_{aq}, cm_{mlq}, cb_b, cg_{gr}, cd_d]$ be unit processing costs at different CLSC facilities. Let $[ti_{zip}, ri_{uir}, si_{xil}, tj_{ij}, tk_{jk}, tc_{kc}, ta_{ca}, ts_{asp}, tz_{aip}, tb_{ab}, tm_{aml}, tg_{agr}, td_{ad}, rg_{bgr}, sd_{gd}, rd_{bd}, tx_{mil}, te_{ger}, tw_{mwl}, tu_{gir}]$ be unit transportation costs between network entities. Let $[dk_k, dw_{wl}, ds_{sp}, de_{er}]$ be demands of the brand-new product and recovered components at their corresponding marketplaces. Let $[caz_{zp}, cau_{ur}, cax_{xl}, cai_i, caj_j, cac_c, caa_a, cab_b, cad_d, cam_{ml}, cag_{gr}]$ be capacities of CLSC facilities. Let $\phi_p$ be the number of part $p$ in each unit of product, $\mu_r$ be the mass of material $r$ in each unit of product, $\omega_l$ be the number of module $l$ in each unit of product, $\psi_q$ be the rate of return of each quality level $q$, $\beta_q$ be the mass of residues in the returned product with quality level $q$ shipped to bulk recycling centers from disassembly centers, $\alpha_{rq}$ be the mass of recyclable material $r$ in the returned product with quality level $q$ shipped to material recycling centers from disassembly centers, $\sigma_q$ be the non-recoverable mass of the returned product with quality level $q$ shipped to disposal centers from disassembly centers, $\gamma_{pq}$ be the number of part $p$ in the returned product with quality level $q$ shipped to spare markets and manufacturing centers from disassembly centers, $\delta_{lq}$ be the umber of remanufacturable module $l$ in the returned product with quality level $q$ shipped to remanufacturing centers from disassembly centers, $\eta_r$ be the ratio of recyclable material $r$ shipped to material recycling centers from bulk recycling centers, $\tau_r$ be the ratio of non-recyclable material $r$ shipped to disposal centers from bulk and material recycling centers, and finally, $\theta$ be the recovery target set by the government.

Decisions to be made are the quantity of flows between network entities and the location of different types of facilities in the reverse network summarized as follows. Let $QI_{zip}$ be the quantity of part $p$ shipped from part supplier $z$ to manufacturing center $i$, $NI_{uir}$ be the quantity of material $r$ shipped from material supplier $u$ to manufacturing center $i$, $XI_{xil}$ be the quantity of module $l$ shipped from module supplier $x$ to manufacturing center $i$, $QJ_{ij}$ be the quantity of products shipped from manufacturing center $i$ to distribution center $j$, $QK_{jk}$ be the quantity of products shipped from distribution center $j$ to end-user zone $k$, $QC_{kcq}$ be the quantity of returns with quality level $q$ shipped from end-user zone $k$ to collection center $c$, $QA_{caq}$ be the quantity of returns with quality level $q$ shipped from collection center $c$ to disassembly center $a$, $QS_{asp}$ be the number of part $p$ shipped from disassembly center $a$ to spare parts market $s$, $QZ_{aip}$ be the number of part $p$

shipped from disassembly center $a$ to manufacturing center $i$, $QM_{amlq}$ be the quantity of module $l$ with quality level $q$ shipped from disassembly center $a$ to remanufacturing center $m$, $QW_{mwl}$ be the number of module $l$ shipped from remanufacturing center $m$ to secondary market $w$, $QX_{mil}$ be the number of module $l$ shipped from remanufacturing center $m$ to manufacturing center $i$, $QB_{ab}$ be the quantity of residues shipped from disassembly center $a$ to bulk recycling center $b$, $QG_{agr}$ be the quantity of recyclable material $r$ shipped from disassembly center $a$ to material recycling center $g$, $NG_{bgr}$ be the quantity of recyclable material $r$ shipped from bulk recycling center $b$ to material recycling center $g$, $QE_{ger}$ be the quantity of recycled material $r$ shipped from material recycling center $g$ to recycled material market $e$, $QU_{gir}$ be the quantity of recycled material $r$ shipped from material recycling center $g$ to manufacturing center $i$, $QD_{ad}$ be the quantity of non-recoverable components shipped from disassembly center $a$ to disposal center $d$, $ND_{bd}$ be the quantity of residues shipped from bulk recycling center $b$ to disposal center $d$, and finally, $XD_{gdr}$ be the quantity of material $r$ shipped from material recycling center $g$ to disposal center $d$. Now, let $YC_c$ take value one if collection center $c$ is opened and zero otherwise, $YA_a$ take value one if disassembly center $a$ is opened and zero otherwise, $YM_m$ take value one if remanufacturing center $m$ is opened and zero otherwise, $YB_b$ take value one if bulk recycling center $b$ is opened and zero otherwise, $YG_g$ take value one if material recycling center $g$ is opened and zero otherwise, and finally $YD_d$ take value one if disposal center $d$ is opened and zero otherwise.

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

**Total revenue**

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

**Total cost**

*Fixed cost*

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

*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*

$$\sum_{z\in Z}\sum_{i\in I}\sum_{p\in P} ti_{zip}QI_{zip} + \sum_{u\in U}\sum_{i\in I}\sum_{r\in R} ri_{uir}NI_{uir} + \sum_{x\in X}\sum_{i\in I}\sum_{l\in L} si_{xil}XI_{xil} + \sum_{i\in I}\sum_{j\in J} tj_{ij}QJ_{ij} + \sum_{j\in J}\sum_{k\in K} tk_{jk}QK_{jk} + \sum_{k\in K}\sum_{c\in C}\sum_{q\in Q} tc_{kcq}QC_{kcq} + \sum_{c\in C}\sum_{a\in A}\sum_{q\in Q} ta_{caq}QA_{caq} + \sum_{a\in A}\sum_{m\in M}\sum_{l\in L}\sum_{q\in Q} tm_{aml}QM_{amlq} + \sum_{a\in A}\sum_{b\in B} tb_{ab}QB_{ab} + \sum_{a\in A}\sum_{g\in G}\sum_{r\in R} tg_{agr}QG_{agr} + \sum_{a\in A}\sum_{d\in D} td_{ad}QD_{ad} + \sum_{b\in B}\sum_{g\in G}\sum_{r\in R} rg_{bgr}NG_{bgr} + \sum_{g\in G}\sum_{d\in D}\sum_{r\in R} sd_{gd}XD_{gdr} + \sum_{b\in B}\sum_{d\in D} rd_{bd}ND_{bd} + \sum_{a\in A}\sum_{s\in S}\sum_{p\in P} ts_{asp}QS_{asp} + \sum_{m\in M}\sum_{w\in W}\sum_{l\in L} tw_{mwl}QW_{mwl} + \sum_{g\in G}\sum_{e\in E}\sum_{r\in R} te_{ger}QE_{ger} + \sum_{a\in A}\sum_{i\in I}\sum_{p\in P} tz_{aip}QZ_{aip} + \sum_{m\in M}\sum_{i\in I}\sum_{l\in L} tx_{mil}QX_{mil} + \sum_{g\in G}\sum_{i\in I}\sum_{r\in R} tu_{gir}QU_{gir} \tag{5}$$

Constraints are grouped into two major categories: flow balance and capacity restrictions.

**Flow balance constraints**

*Manufacturing centers*

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

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

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

*Distribution centers*

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

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

*Collection centers*

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

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

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

*Disassembly centers*

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

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

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

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

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

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

*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**

*Forward chain facilities*

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

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

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

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

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

*Reverse chain facilities*

$$\sum_{k\in K}\sum_{q\in Q} QC_{kcq} \le cac_c YC_c \quad \forall\, c\in C \tag{32}$$

$$\sum_{c\in C}\sum_{q\in Q} QA_{caq} \le caa_a YA_a \quad \forall\, a\in A \tag{33}$$

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

$$\sum_{a\in A} QB_{ab} \le cab_b YB_b \quad \forall\, b\in B \tag{35}$$

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

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

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 return stream in collection centers as set by the government. Constraints (14)-(19) ensure flow conservation at each disassembly center. 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. Constraint (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.

## 4. Solution methodology

The proposed model (1)-(37) has a conspicuous special property that facilitates the application of Benders decomposition as the solution method. 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.

### *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 $\bar{\boldsymbol{Y}}$ denote the vector of fixed $\boldsymbol{Y}$. The resulting primal subproblem (PSP) that determines the routing of the forward and reverse flows can be stated as follows.

$$\begin{aligned}
\max \quad & (1), (3) - (5) \\
\text{s.t.} \quad & (6) - (31) \\
& \sum_{k \in K} \sum_{q \in Q} QC_{kcq} \leq cac_c \overline{YC}_c \quad \forall\, c \in C && (38) \\
& \sum_{c \in C} \sum_{q \in Q} QA_{caq} \leq caa_a \overline{YA}_a \quad \forall\, a \in A && (39) \\
& \sum_{a \in A} QD_{ad} + \sum_{b \in B} ND_{bd} + \sum_{g \in G} \sum_{r \in R} XD_{gdr} \leq cad_d \overline{YD}_d \quad \forall\, d \in D && (40) \\
& \sum_{a \in A} QB_{ab} \leq cab_b \overline{YB}_b \quad \forall\, b \in B && (41) \\
& \sum_{a \in A} QG_{agr} + \sum_{b \in B} NG_{bgr} \leq cag_{gr} \overline{YG}_g \quad \forall\, g \in G, \forall\, r \in R && (42) \\
& \sum_{a \in A} \sum_{q \in Q} QM_{amlq} \leq cam_{ml} \overline{YM}_m \quad \forall\, m \in M, \forall\, l \in L && (43)
\end{aligned}$$

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

$$\begin{aligned}
\min \quad & Z_{\boldsymbol{\upsilon}}(\bar{\boldsymbol{Y}}) = \sum_{k \in K} dk_k \upsilon_k^5 + \sum_{k \in K} \sum_{q \in Q} \psi_q dk_k \upsilon_{kq}^6 + \sum_{s \in S} \sum_{p \in P} ds_{sp} \upsilon_{sp}^{10} + \sum_{w \in W} \sum_{l \in L} dw_{wl} \upsilon_{wl}^{16} \\
& + \sum_{e \in E} \sum_{r \in R} de_{er} \upsilon_{er}^{20} + \sum_{z \in Z} \sum_{p \in P} caz_{zp} \upsilon_{zp}^{22} + \sum_{u \in U} \sum_{r \in R} cau_{ur} \upsilon_{ur}^{23} \\
& + \sum_{x \in X} \sum_{l \in L} cax_{xl} \upsilon_{xl}^{24} + \sum_{i \in I} cai_i \upsilon_i^{25} + \sum_{j \in J} caj_j \upsilon_j^{26} + \sum_{c \in C} cac_c \overline{YC}_c \upsilon_c^{27} \\
& + \sum_{a \in A} caa_a \overline{YA}_a \upsilon_a^{28} + \sum_{d \in D} cad_d \overline{YD}_d \upsilon_d^{29} + \sum_{b \in B} cab_b \overline{YB}_b \upsilon_b^{30} \\
& + \sum_{g \in G} \sum_{r \in R} cag_{gr} \overline{YG}_g \upsilon_{gr}^{31} + \sum_{m \in M} \sum_{l \in L} cam_{ml} \overline{YM}_m \upsilon_{ml}^{32} && (44) \\
\text{s.t.} \quad & (\boldsymbol{\upsilon^1}, \boldsymbol{\upsilon^2}, ...., \boldsymbol{\upsilon^{32}}) \in \Delta && (45)
\end{aligned}$$

where the unrestricted dual variable vectors $\boldsymbol{\upsilon^5}$, $\boldsymbol{\upsilon^6}$, $\boldsymbol{\upsilon^{10}}$, $\boldsymbol{\upsilon^{16}}$, and $\boldsymbol{\upsilon^{20}}$ are, respectively, associated with constraints (10), (11), (15), (21), and (25). The non-negative dual variable vectors $\boldsymbol{\upsilon^{22}}$,..., $\boldsymbol{\upsilon^{26}}$ and $\boldsymbol{\upsilon^{27}}$,..., $\boldsymbol{\upsilon^{32}}$ are, respectively, associated with constraints (27)-(31) and (38)-(43). Besides, $\Delta$ indicates the polyhedron defined by the constraints of the DSP. If $\Delta$ 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 $\Gamma$, we can formulate the master problem (MP) that determines the CLSC network configuration as follows.

$$\max \quad \Gamma - \sum_{c\in C} fc_c YC_c - \sum_{a\in A} fa_a YA_a - \sum_{m\in M} fm_m YM_m$$
$$- \sum_{g\in G} fg_g YG_g - \sum_{b\in B} fb_b YB_b - \sum_{d\in D} fd_d YD_d \tag{46}$$
$$\text{s.t.} \quad \Gamma \le \pi(\hat{\boldsymbol{\upsilon}}^{\boldsymbol{n}^T}) + \sum_{c\in C} cac_c YC_c \hat{\upsilon}_c^{27} + \sum_{a\in A} caa_a YA_a \hat{\upsilon}_a^{28} + \sum_{d\in D} cad_d YD_d \hat{\upsilon}_d^{29}$$
$$+ \sum_{b\in B} cab_b YB_b \hat{\upsilon}_b^{30} + \sum_{g\in G}\sum_{r\in R} cag_{gr} YG_g \hat{\upsilon}_{gr}^{31} + \sum_{m\in M}\sum_{l\in L} cam_{ml} YM_m \hat{\upsilon}_{ml}^{32} \tag{47}$$
$$0 \le \pi(\hat{\boldsymbol{\kappa}}^{\boldsymbol{n}^T}) + \sum_{c\in C} cac_c YC_c \hat{\kappa}_c^{27} + \sum_{a\in A} caa_a YA_a \hat{\kappa}_a^{28} + \sum_{d\in D} cad_d YD_d \hat{\kappa}_d^{29}$$
$$+ \sum_{b\in B} cab_b YB_b \hat{\kappa}_b^{30} + \sum_{g\in G}\sum_{r\in R} cag_{gr} YG_g \hat{\kappa}_{gr}^{31} + \sum_{m\in M}\sum_{l\in L} cam_{ml} YM_m \hat{\kappa}_{ml}^{32} \tag{48}$$
$$\boldsymbol{Y} \in \{0,1\} \tag{49}$$

where $\boldsymbol{\kappa}$ indicate extreme rays of $\Delta$ when the DSP is unbounded. Now, let $\Delta_p$ and $\Delta_r$ represent the sets of extreme points and extreme rays of $\Delta$, 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.

$$\max \quad \Gamma - F^T\boldsymbol{Y} \tag{50}$$
$$\text{s.t.} \quad \Gamma \le \pi(\hat{\boldsymbol{\upsilon}}^{\boldsymbol{n}^T}) + \hat{\boldsymbol{\upsilon}}^{\boldsymbol{m}^T} V\boldsymbol{Y} \quad (\boldsymbol{\upsilon}^{\boldsymbol{n}}, \boldsymbol{\upsilon}^{\boldsymbol{m}} | \boldsymbol{n} \ne \boldsymbol{m}) \in \Delta_p \tag{51}$$
$$0 \le \pi(\hat{\boldsymbol{\kappa}}^{\boldsymbol{n}^T}) + \hat{\boldsymbol{\kappa}}^{\boldsymbol{m}^T} V\boldsymbol{Y} \quad (\boldsymbol{\kappa}^{\boldsymbol{n}}, \boldsymbol{\kappa}^{\boldsymbol{m}} | \boldsymbol{n} \ne \boldsymbol{m}) \in \Delta_r \tag{52}$$
$$\boldsymbol{Y} \in \{0,1\} \tag{53}$$

Observe that the polyhedron $\Delta$ 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 generally 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 $\Delta_p$ is identified and leads to the optimality cut (51). Otherwise, the feasibility cut (52) associated with an extreme ray of $\Delta_r$ would be added to the MP.

### 4.2. Algorithmic enhancement

As Benders decomposition is known to be a method which converges quite slowly, we provide different algorithmic enhancement in order to accelerate the solution algorithm.

#### 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. We expect that the presence of the valid inequalities reduce the number of feasibility cuts (52) during the execution of the Benders algorithm.

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

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

$$\sum_{a \in A} caa_a YA_a \geq \sum_{s \in S} \sum_{p \in P} ds_{sp} \tag{56}$$

$$\sum_{m \in M} \sum_{l \in L} cam_{ml} YM_m \geq \sum_{w \in W} \sum_{l \in L} dw_{wl} \tag{57}$$

$$\sum_{g \in G} \sum_{r \in R} cag_{gr} YG_g \geq \sum_{e \in E} \sum_{r \in R} de_{er} \tag{58}$$

$$\sum_{b \in B} YB_b \geq 1 \tag{59}$$

$$\sum_{d \in D} YD_d \geq 1 \tag{60}$$

Constraints (54) and (55) ensure that the selected collection and disassembly centers provide enough capacity to acquire returns. Constraint (56)-(58) ensure enough capacity for satisfying the demands of recovered components at their corresponding secondary markets through opening adequate recovery facilities. According to constraints (59) and (60), at least one bulk recycling and one disposal center must be opened in the CLSC network.

#### 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. As a result, several valid optimality cuts of different strength associated to the set of alternative optimal solutions can be generated. According to Magnanti and Wong [19], the optimal cut corresponds to the dual solution vectors $(\hat{\boldsymbol{\upsilon}}_1^{\boldsymbol{n}^T}, \hat{\boldsymbol{\upsilon}}_1^{\boldsymbol{m}^T})$ dominate the cut generated from the dual solution $(\hat{\boldsymbol{\upsilon}}_2^{\boldsymbol{n}^T}, \hat{\boldsymbol{\upsilon}}_2^{\boldsymbol{m}^T})$ 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 $0 \leq YC_c \leq 1, \forall\ c \in C; 0 \leq YA_a \leq 1, \forall a \in A; 0 \leq YM_m \leq 1, \forall m \in M; 0 \leq YB_b \leq 1, \forall\ b \in B; 0 \leq YG_g \leq 1, \forall\ g \in G; 0 \leq YD_d \leq 1, \forall\ d \in D$. 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.

$$\min \quad \pi(\boldsymbol{v}^{\boldsymbol{n}^T}) + \boldsymbol{v}^{\boldsymbol{m}^T} V\boldsymbol{Y^0} \tag{61}$$

$$\text{s.t.} \quad \pi(\boldsymbol{v}^{\boldsymbol{n}^T}) + \boldsymbol{v}^{\boldsymbol{m}^T} V\bar{\boldsymbol{Y}} = Z_{\boldsymbol{v}}(\bar{\boldsymbol{Y}}) \tag{62}$$

$$(\boldsymbol{v^1}, \boldsymbol{v^2}, ...., \boldsymbol{v^{32}}) \in \Delta \tag{63}$$

where $\boldsymbol{Y^0} \in ri(\mathbf{Y^{LP}})$, $\boldsymbol{n} \neq \boldsymbol{m}$, and $\Delta$ indicates the polyhedron defined by the constraints of the DSP. As mentioned earlier, the normalization constraint (62) might be quite dense and numerically unstable. Nonetheless, Papadakos [20] demonstrated that this constraint can be omitted through choosing a different core point on the objective function (61) 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 ([20], Theorem 8). The modified auxiliary dual subproblem can be restated as follows.

$$\begin{aligned} \min \quad & \pi(\boldsymbol{v}^{\boldsymbol{n}^T}) + \boldsymbol{v}^{\boldsymbol{m}^T} V\boldsymbol{Y^0} \\ \text{s.t.} \quad & (\boldsymbol{v^1}, \boldsymbol{v^2}, ...., \boldsymbol{v^{32}}) \in \Delta \end{aligned} \tag{64}$$

where $\boldsymbol{n} \neq \boldsymbol{m}$ and $\Delta$ indicates the polyhedron defined by the constraints of the DSP. The optimal solution to (64) 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 the identification of 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 1.

**Algorithm 1** -BD1

$UB \leftarrow \infty$, $LB \leftarrow -\infty$, $\lambda^c \leftarrow 0.5$
Find an initial core point $\boldsymbol{Y^0}$
**while** $(UB - LB)/UB \leq \epsilon$ **do**
Solve the auxiliary DSP (64)
Add Pareto-optimal cut (51) to the MP
Solve the MP
$UB \leftarrow \bar{\Gamma} - F^T \bar{\boldsymbol{Y}}$
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, $Z_{\boldsymbol{v}}(\bar{\boldsymbol{Y}}) - F^T \bar{\boldsymbol{Y}}$)
  $\boldsymbol{Y^0} \leftarrow \lambda^c \boldsymbol{Y^0} + (1 - \lambda^c) \bar{\boldsymbol{Y}}$
**end if**
**end while**
Solve the corresponding PSP

*4.2.3. Local branching*

The preliminary computational results revealed that even though the Pareto-optimal cuts enhance the performance of the classic Benders decomposition algorithm, the upper bound still slowly decreases throughout the solution process. Therefore, we consider the incorporation of local branching cuts [21, 22] 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 lead naturally 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 $\bar{\boldsymbol{Y}}^1$ be an optimal solution to the MP. Introducing the following disjunction, we can devide the feasible region of the MIP model (1)-(37) into two reduced subproblems.

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

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.

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

where $\bar{C}$, $\bar{A}$, $\bar{M}$, $\bar{B}$, $\bar{G}$, and $\bar{D}$ represent the set of open facilities in the reverse chain obtained after solving the current MP. Assigning a relatively low value to $\kappa_v$, imposing a time limit on the left branch subproblem, and a small optimality gap $\epsilon_k$, 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 $\bar{\boldsymbol{Y}}^2$ 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., $\Delta(\boldsymbol{Y},\bar{\boldsymbol{Y}}^1) \ge \kappa_v + 1$. The solution $\bar{\boldsymbol{Y}}^2$ is considered as the new reference point and the branching scheme will be applied to this solution, i.e., $\Delta(\boldsymbol{Y},\bar{\boldsymbol{Y}}^2) \le \kappa_v$. 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., $\Delta(\boldsymbol{Y},\bar{\boldsymbol{Y}}^1) \ge \kappa_v + 1$. 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 $\Delta(\boldsymbol{Y},\bar{\boldsymbol{Y}}^2) \ge 1$ will be introduced into the current subproblem to remove $\bar{\boldsymbol{Y}}^2$. Then, the new subproblem will be created by defining a left branching constraint associated with the new reference point, i.e., $\Delta(\boldsymbol{Y},\bar{\boldsymbol{Y}}^2) \le \kappa_v$. 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 $\bar{\boldsymbol{Y}}^2$ 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 2 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 $\bar{\boldsymbol{QX}}_h$ and $\bar{\boldsymbol{Y}}_h$ 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.

## 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 [12] and [24]. The washing machines under consideration consist of two modules, ten parts, and three types of solid materials. Table 1 displays the disassembly tree of a

**Algorithm 2** -BD2

$UB \leftarrow \infty$, $LB \leftarrow -\infty$, $t \leftarrow 1$, $MaxIter \leftarrow 2$
Find an initial core point $\boldsymbol{Y^0}$
**while** $(UB - LB)/UB \leq \epsilon$ **do**
  Solve the auxiliary DSP (64)
  Add Pareto-optimal cut (51) to the MP
  Solve the MP
  $UB \leftarrow \bar{\Gamma} - F^T\bar{\boldsymbol{Y}}$
  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, $Z_{\boldsymbol{v}}(\bar{\boldsymbol{Y}}) - F^T\bar{\boldsymbol{Y}}$)
    $\boldsymbol{Y^0} \leftarrow \lambda^c \boldsymbol{Y^0} + (1-\lambda^c)\bar{\boldsymbol{Y}}$
    $\bar{\boldsymbol{Y}}^1 \leftarrow \bar{\boldsymbol{Y}}$
    **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

Table 1: Separable components of a used washing machine

| Parameter | Value |
|---|---|
| $\phi_p$ | washing tube:1, cover:1, balance:1, frame:1, hose:1, condenser:1, small electric parts:1, electric wire:1, transformer:1, PCB board:1 |
| $\mu_r$ | plastics:6 kg, steel:3 kg, copper:1 kg |
| $\omega_l$ | motor:1, clutch:1 |

returned washing machine. The next section describes test instances settings and then it is followed by a summary of computational results.

### *5.1. Experimental design*

We assume three quality levels, namely, low, 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 determined through {30, 31, 32, ..., 100}. Capacities of facilities in the forward network are randomly generated following a reasonable

**Algorithm 3** -LocBran.

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

relationship with demands of end-users and the disassembly tree. Capacities of reverse network facilitates are also randomly generated based upon 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 2 and 3. Note that fixed costs of opening facilities in the reverse network are generated considering the capacity of facilities. In other words, the higher the capacity of a facility, the larger infrastructural cost it will require.

We also consider seven major classes within each three different test in-

Table 2: Quality level dependent parameters

| Parameter | Quality levels | | |
|---|---|---|---|
| | High | Medium | Poor |
| $\psi_q$ | $U(0.1, 0.2)$ | $U(0.2, 0.3)$ | $U(0.3, 0.4)$ |
| $\delta_{lq}$ | 1, 1 | 1, 0 | 0, 0 |
| $\gamma_{pq}$ | 1, 1, 1, 1, 1,<br>1, 1, 1, 1, 1 | 1, 1, 0, 0, 0,<br>1, 1, 0, 0, 1 | 0, 0, 1, 0, 0,<br>0, 0, 0, 0, 1 |
| $\alpha_{rq}$ | 5, 2, 1 | 4, 1, 1 | 3, 1, 0 |
| $\beta_q$ | 2 | 4 | 6 |
| $\sigma_q$ | 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 3: Other case example parameters

| Parameter | Value | Parameter | Value | Parameter | Value |
|---|---|---|---|---|---|
| $fc_c$ | $U(400000, 600000)$ | $fa_a$ | $U(400000, 600000)$ | $fm_m$ | $U(700000, 900000)$ |
| $fb_b$ | $U(400000, 600000)$ | $fg_g$ | $U(400000, 600000)$ | $fd_d$ | $U(200000, 400000)$ |
| $Pk_k$ | $U(600, 1300)$ | $Ps_p$ | $U(50, 70)$ | $Pw_l$ | $U(100, 120)$ |
| $Pe_r$ | $U(20, 30)$ | $cx_{xl}$ | $U(70, 90)$ | $cz_{zp}$ | $U(30, 50)$ |
| $cu_{ur}$ | $U(10, 20)$ | $ci_i$ | $U(6, 7)$ | $cj_j$ | $U(1, 2)$ |
| $cb_b$ | $U(1.5, 2.5)$ | $cg_{gr}$ | $U(1.5, 2.5)$ | $cd_d$ | $U(1.5, 2.5)$ |
| $\eta_r$ | $U(0.2, 0.3)$ | $\tau_r$ | $U(0.05, 0.15)$ | $\theta$ | 0.7 |

stances as shown in Table 4. These test instances vary according to the number of CLSC facilities as well as the number of the first and secondary markets. Table 5 presents the number of constraints and variables, including binary and continuous ones, in each class.

Table 4: 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 |

### *5.2. Computational results*

The proposed algorithmic scheme was implemented in C++ using Concert Technology with IBM-ILOG CPLEX 12.51. All the experiments were 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

Table 5: 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 |

Table 6: The value of parameters of local branching procedure

| Class | MIP phase | | | | | MP phase | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Sub. | Ndiv. | Time (sec) | $\kappa_v$ | $\epsilon_{\kappa_v}$ | Sub. | Ndiv. | Time (sec) | $\kappa_v$ | $\epsilon_{\kappa_v}$ |
| 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% |

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 6 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 $\kappa_v$, and the optimality gap $\epsilon_{\kappa_v}$% each time this procedure is called in both phases of “MIP” and “MP”.

Table 7 summarizes the computational statistics obtained after solving each test 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 noted that we examined the performance of the classic Benders decomposition as well as the acceleration strategy in [23]. As the convergence behavior of those two algorithms were not desirable, we do not report the corresponding computational results. The most important observations concerning Table 7 are summarized as follows.

- CPLEX is only able to find the optimal solution of the test instances

Table 7: Comparison of both algorithms and CPLEX

| Class | "BD1" | | | | "BD2" | | | CPLEX | |
|---|---|---|---|---|---|---|---|---|---|
| | Time (sec) | Iter. | Profit | GAP | Time (sec) | Iter. | Profit | Time (sec) | Profit |
| 1 | 28 | 40 | 11282100 | $\leq$ 1% | 101 | 16 | 11282100 | 621 | 11282100* |
| | 30 | 54 | 11600600 | $\leq$ 1% | 140 | 19 | 11600600 | 485 | 11600600* |
| | 22 | 35 | 12524000 | $\leq$ 1% | 98 | 15 | 12524000 | 583 | 12524000* |
| 2 | 289 | 128 | 17071000 | $\leq$ 1% | 231 | 35 | 17071000 | 605 | 17071000* |
| | 1359 | 209 | 15568800 | $\leq$ 1% | 814 | 77 | 15568800 | 953 | 15568800* |
| | 70 | 82 | 15072600 | $\leq$ 1% | 185 | 29 | 15072600 | 709 | 15072600* |
| 3 | $\geq$ 3600 | 90 | 22416000 | 1.61% | 924 | 20 | 22494300 | $\geq$ 7200 | 22321100 |
| | $\geq$ 3600 | 77 | 22258400 | 2.65% | 670 | 11 | 22271200 | $\geq$ 7200 | 22196200 |
| | $\geq$ 3600 | 75 | 22868800 | 3.15% | 587 | 14 | 22951700 | $\geq$ 7200 | 22801400 |
| 4 | $\geq$ 3600 | 110 | 28496100 | 2.32% | 772 | 14 | 28531900 | $\geq$ 7200 | 28475000 |
| | $\geq$ 3600 | 91 | 20482300 | 2.42% | 751 | 13 | 20532400 | $\geq$ 7200 | 20260000 |
| | $\geq$ 3600 | 139 | 22473400 | 2.34% | 832 | 12 | 22560200 | $\geq$ 7200 | 22415500 |
| 5 | $\geq$ 3600 | 65 | 22697800 | 3.58% | 1087 | 16 | 22862600 | $\geq$ 7200 | 22601400 |
| | $\geq$ 3600 | 97 | 27909000 | 2.00% | 967 | 17 | 27909900 | $\geq$ 7200 | 27702800 |
| | $\geq$ 3600 | 107 | 20563300 | 2.84% | 1023 | 16 | 20610300 | $\geq$ 7200 | 20530700 |
| 6 | $\geq$ 3600 | 74 | 31624100 | 1.62% | 1835 | 19 | 31682100 | $\geq$ 7200 | 31476600 |
| | $\geq$ 3600 | 96 | 28978500 | 2.34% | 1624 | 17 | 28982300 | $\geq$ 7200 | 28556300 |
| | $\geq$ 3600 | 59 | 31723300 | 2.69% | 1498 | 15 | 31829700 | $\geq$ 7200 | 31711900 |
| 7 | $\geq$ 3600 | 19 | 31700500 | 2.88% | 2510 | 14 | 31741300 | $\geq$ 7200 | 30799800 |
| | $\geq$ 3600 | 16 | 30323400 | 4.10% | 2938 | 15 | 30799100 | $\geq$ 7200 | 29708100 |
| | $\geq$ 3600 | 57 | 30501300 | 2.45% | 1963 | 14 | 30573000 | $\geq$ 7200 | 27530000 |
| Average | - | 82 | 22768348 | - | 1026 | 20 | 22831005 | - | 22485995 |

of the first and the second classes indicated by * symbol in the last column. However, it fails to solve the test instances of other classes 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 those classes is reported in the last column.

- "BD1" demonstrates better performance in terms of CPU time, i.e., in average, four times faster than "BD2" only in solving the test instances of the first class. However, "BD2" implementation reduces the number of iterations, averagely, 2.5 times. This algorithm also indicates an improved convergence behavior when applied to solve the test instances of the second class. In particular, "BD2" is in 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 rest of five classes, "BD1" is unable to solve the test instances to optimality even in 1 hour. Columns GAP and profit indicate, respectively, the relative difference between lower and upper bounds and the best profit 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 classes is 1.82%. It indicates the extent to which the best feasible solution identified by CPLEX within 2 hours is far from the optimal solution obtained by "BD2". The average solution time of "BD2" is 18 minutes when applied to solve the test instances of classes 3 to 6. In addition, the average solution time of "BD2" is 42 minutes in solving the test instances of the last class, i.e., the largest class (see Figure 3).

The average number of iterations when solving each class with both decomposition algorithms is illustrated in Figure 4. In summary, it can be observed that local branching search considerably improves the performance of the Benders decomposition algorithm.

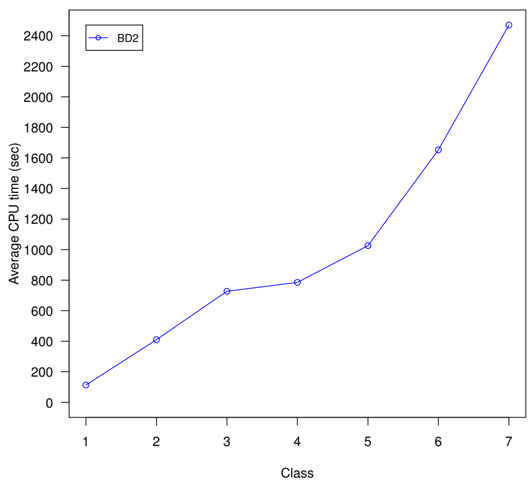


Figure 3: CPU time vs. Class

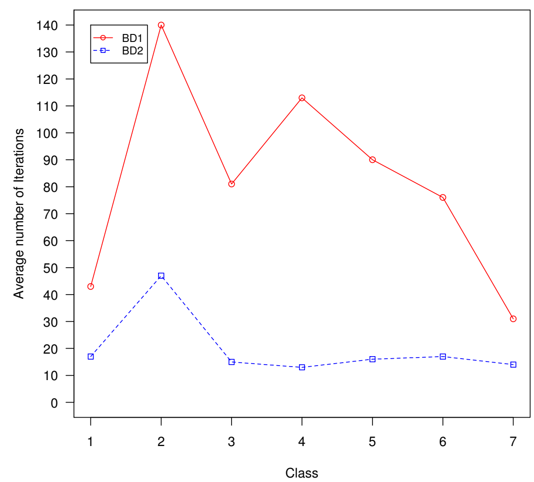


Figure 4: Iterations vs. Class

### *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. Hence, it imposes the environmentally friendly constraint (13) to the CLSC network design model. We examine the effect of increasing the recovery target $\theta$ by 7% and 15% on quantity of the acquired used washing machines in disassembly centers for one test instance. As expected, the OEM favors the recovery of high quality returns as they are more profitable. In other words, 100% of the high quality returns is procured by the OEM in all levels of recovery targets, i.e., 0.7, 0.77, and 0.85. On the contrary, the company is not willing to use all of potential recovery capacities according to low profitability of medium and poor quality returns. More specifically, 50%, 65%, and 80% of poor quality used washing machines are desirable for recovery purposes with the recovery targets of 0.7, 0.77, and 0.85, respectively. As the legislations become more restrictive, the OEM will be forced to take a larger amount of poor quality returns. Consequently, the total costs of designing the CLSC will increase mainly due to opening more facilities in the reverse network, which incurs 15% additional infrastructural cost as well as higher processing costs. It is worth mentioning that increasing the recovery target to 0.77 and 0.85 will reduce the net profit, respectively, by 1.27% and 1.37% compared to the base case of 0.7.

## 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 complete 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 the 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 performances 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 solve only 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.

A natural extension of the setting considered in this article is to extend the model into a multi-product setting. The model can also be further validated in the context of more complex durable products, such as aircraft or automobiles which include thousands of components in their disassembly tree. Finally, the locations and capacities of facilities can be adjusted over a planning horizon which gives rise to a multi-period CLSC network design problem in order to reflect the dynamic behavior of demand and return ratio over time.

## Acknowledgments

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

## References

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

[2] 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.

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

[4] 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.

[5] M. Pishvaee, J. Razmi, S. 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

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

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

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

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

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

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

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

[24] 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.