DECOMPOSITION BASED SOLUTION APPROACHES FOR MULTI-PRODUCT CLOSED-LOOP SUPPLY CHAIN NETWORK DESIGN MODELS

A Dissertation

by

GOPALAKRISHNAN EASWARAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2008

Major Subject: Industrial Engineering

DECOMPOSITION BASED SOLUTION APPROACHES FOR MULTI-PRODUCT CLOSED-LOOP SUPPLY CHAIN NETWORK DESIGN MODELS

A Dissertation

by

GOPALAKRISHNAN EASWARAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Co-Chairs of Committee,	Sıla Çetinkaya
	Halit Üster
Committee Members,	Elif Akçalı
	Brett Peters
	Jennifer L Welch
Head of Department,	Brett A. Peters

August 2008

Major Subject: Industrial Engineering

ABSTRACT

Decomposition Based Solution Approaches for Multi-product Closed-Loop Supply Chain Network Design Models. (August 2008)

Gopalakrishnan Easwaran, B.E., PSG College of Technology, Coimbatore, India; M.S., Texas A&M University

Co–Chairs of Advisory Committee: Dr. Sıla Çetinkaya
Dr. Halit Üster

Closed-loop supply chain (CLSC) management provides opportunity for cost savings through the integration of product recovery activities into traditional supply chains. Product recovery activities, such as *remanufacturing*, reclaim a portion of the previously added value in addition to the physical material.

Our problem setting is motivated by the practice of an Original Equipment Manufacturer (OEM) in the automotive service parts industry, who operates a well established forward network. The OEM faces customer demand due to warranty and beyond warranty vehicle repairs. The warranty based demand induces part returns. We consider a case where the OEM has not yet established a product recovery network, but has a strategic commitment to implement remanufacturing strategy. In accomplishing this commitment, complications arise in the network design due to activities and material movement in both the forward and reverse networks, which are attributed to remanufacturing. Consequently, in implementing the remanufacturing strategy, the OEM should simultaneously consider both the forward and reverse flows for an optimal network design, instead of an independent and sequential modeling approach. In keeping with these motivations, and with the goal of implementing the remanufacturing strategy and transforming independent forward and reverse supply chains to CLSCs, we propose to investigate the following research questions:

1. How do the following transformation strategies leverage the CLSC's overall cost performance?
    - Extending the already existing forward channel to incorporate reverse channel activities.
    - Designing an entire CLSC network.
2. How do the following network flow integration strategies influence the CLSC's overall cost performance?
    - Using distinct forward and reverse channel facilities to manage the corresponding flows.
    - Using hybrid facilities to coordinate the flows.

In researching the above questions, we address significant practical concerns in CLSC network design and provide cost measures for the above mentioned strategies. We also contribute to the current literature by investigating the optimal CLSC network design. More specifically, we propose three models and develop mathematical formulations and novel solution approaches that are based on decomposition techniques, heuristics, and meta-heuristic approaches to seek a solution that characterizes the configuration of the CLSC network, along with the coordinated forward and reverse flows.

To My Family

ACKNOWLEDGMENTS

I express my sincere gratitude to my co-advisors, Dr. Sıla Çetinkaya and Dr. Halit Üster, for their guidance and encouragement during my graduate studies at Texas A&M University. I am indebted for their support and advice for my professional and personal development.

I am grateful to Dr. Elif Akçalı, Dr. Brett A. Peters and Dr. Jennifer L. Welch for serving as members of my advisory committee and providing valuable comments on my dissertation. I would like to thank my officemates: Fatih, Burcu, Homarjun, Hui, Ken, Joaquin, Liqing, Xinghua and Su. Their friendship and support have been invaluable to me. I wish to thank Judy Meeks, Claudia Samford, Michele Bork, Lesly Bell, Letty Benning, and Katherine Edwards for administrative help, and Mark Henry, Mark Hopcus and Dennis Allen for their prompt support in technical assistance.

I am grateful to my family and friends for all their affection, encouragements and patience. I could not have completed this work without their support and love. I thank them for extending their unconditional love to me.

TABLE OF CONTENTS

LIST OF TABLES

TABLE Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Closed-loop supply chain management provides ample opportunity for cost savings through the integration of product recovery activities into traditional supply chains (a.k.a. forward supply chain). It has recently received research focus in the context of *reverse logistics* and recovery activities such as *remanufacturing*. Remanufacturing processes *used* products to reclaim a portion of the previously added value in the form of cost of energy, labor, and manufacturing operations in addition to the physical material (Parkinson and Thompson, 2003).

A number of firms have focused on remanufacturing initiatives over the past years. Dell, GM, Caterpillar, and HP are a few prominent examples. Kodak and Xerox became pioneers in their industries by adopting successful remanufacturing practices for single-use cameras and refillable toner cartridges, respectively. The financial success, coupled with the environmental benefits, attained by these companies has been instrumental to the current interest in remanufacturing practices. In the United States, automotive parts remanufacturing market was estimated to be $36 billion, which accounts for a significant share of the $56 billion remanufacturing sector (Lund, 1996; Giuntini, 2001). However, widespread adoption and successful integration of remanufacturing strategy into closed-loop supply chains (CLSC) still encounter obstacles, which include a lack of quantitative decision-making tools to address the unique challenges of the underlying CLSC network.

The business decisions pertaining to the issues and challenges in CLSC networks follow a three-part hierarchy consisting of the *strategic*, *tactical* and *opera-*

This dissertation follows the style and format of *Operations Research*.

tional level decisions (Langevin and Riopel, 2005). Guide et al. (2003) report the unique challenges facing the CLSC networks and they emphasize the need for building quantitative business models to address the larger strategic issues. Within the above decision hierarchy, the *logistic network design* is considered a strategic issue of prime importance, since it impacts the performance and economic viability of CLSCs (Fleischmann, 2001).

The goal of the network design is to facilitate an appropriate logistics infrastructure for the underlying CLSC by examining the alternate network structures, outsourcing strategies, locations and capacities of supply chain facilities, product flow patterns, and transportation strategies. An optimal network design improves the competitive advantage of a firm through increased supply chain performance in terms of higher customer service and reduced operational costs. In general, optimal network design for the traditional supply chains involves complex optimization problems and requires advanced technology and solution approaches (Simchi-Levi et al., 2004). In the case of CLSCs, further complications arise due to the simultaneous consideration of forward and reverse flows in the underlying network and these complicating factors are primarily attributed to the product recovery and reuse activities.

I.1. Motivation and Scope of the Dissertation

Our problem of interest is motivated by the setting where an original equipment manufacturer (OEM) produces and distributes products, which are characterized by high durability, long life cycles, and high recovery value, via an established forward channel with new product plants and distribution centers (DCs) to satisfy the demand at the retail locations. Due to the economic incentives and the environmental benefits, the OEM has a strategic commitment to implement remanufacturing practices and establish a reverse channel network. In accomplishing this commitment, complications arise in the network design due to different kinds of flows in the network, and these are primarily attributed to remanufacturing.

Remanufacturing extends the scope of traditional manufacturing and logistics to include not only *forward*, but also *reverse*, *flows* along with the corresponding *forward* and *reverse channel activities.* Although adopting a remanufacturing strategy requires a significant initial investment to establish and manage a reverse channel in addition to a forward channel, in general, this investment can be justified by the effective recovery of high value components via an *integrated* CLSC network.

Since remanufacturing processes returned parts, the efficiency with which the OEM collects these returned parts has a direct impact on the profitability of the remanufacturing operations. Moreover, the remanufactured products can be used to satisfy a portion of the customer demand in the forward channel. This impacts the forward channel flows and introduces a strong interdependence between the forward and the reverse flows in the underlying CLSC network. Consequently, in implementing the remanufacturing strategy, the OEM should simultaneously consider both the forward and reverse flows for an optimal network design, instead of an independent and sequential modeling approach to the forward and reverse network design. In keeping with these motivations, and with the goal of implementing the remanufacturing strategy and transforming independent forward and reverse supply chains to *closed-loop supply chains*, in this dissertation, we investigate the following research questions:

(i) How do the following transformation strategies leverage the CLSC's overall cost performance?

- Extending the already existing forward channel to incorporate reverse channel activities.
- Designing an entire CLSC.

(ii) How do the following network flow integration strategies influence the CLSC's overall cost performance?

- Using distinct forward and reverse channel facilities to manage the forward and reverse flows, respectively.
- Using hybrid facilities to coordinate the forward and reverse flows.

In researching the above questions, our main goal is to address significant practical concerns in the CLSC network design (in terms of facility location and forward and reverse flow integration) and provide cost measures (in terms of total cost of facility location, processing and transportation) for different transformation strategies. We also contribute to the current literature by investigating the optimal network design for CLSCs. More specifically, we propose three models and develop mathematical formulations and novel solution approaches that are based on decomposition techniques, heuristics, and meta-heuristic approaches to seek a solution that characterizes

(i) the configuration of the CLSC, that is, the locations of forward and reverse channel facilities, and

(ii) the integrated and coordinated forward and reverse flows in this network.

I.2. Description of Problem Settings

In our network design problem the OEM operates a forward channel for producing and distributing multiple types of products. Specifically, we focus on the network design for remanufacturable/refurbishable durable products/parts, i.e., consumer, commercial, and industrial equipment such as automotive parts, photocopying equipment, ships, and aircraft engines. Products in this category are characterized by their *high recoverable value*, *long product life cycles*, and *well-established forward networks*. For example, consider a photocopying equipment manufacturer managing a well-established forward network with manufacturing facilities, DCs, and customer/retail locations. Due to increasing popularity of leasing practices in this industry, the manufacturer faces two streams of demand:

(i) new demand, i.e., new equipment acquisitions, and

(ii) replacement demand, i.e., leased equipment renewals.

Adopting a remanufacturing strategy, the manufacturer can satisfy both streams of demand using new or remanufactured products, i.e., the forward flows. In this setting, the customers do not distinguish between the two types of products. The replacement demand generates a return stream, i.e., the reverse flow, which, in turn, can be transformed into remanufactured products.

Likewise, in the automotive industry, the OEMs operate well-established service parts networks that consist of part suppliers, DCs, and retail locations. The spare parts that are required for vehicle maintenance and repair operations are sold through service shops at car dealerships and warehousing distributors, which we refer to as the retailers. The OEM faces two streams of service parts demand:

(i) warranty based vehicle repair demand, and

(ii) beyond warranty vehicle repair demand.

Both streams of demand are satisfied by new or remanufactured service parts, due to the part warranties offered by the manufacturer. The main distinction between warranty and beyond warranty vehicle repairs is that the former generates a part return–the failed part that should be replaced due to warranty obligations–whereas the latter may or may not generate a return. We refer to the stream that generates part returns as the *induced demand* stream, and the one that does not generate part returns as the *new demand* stream. The new parts are either produced by the OEM or purchased from an outside supplier. Each supplier provides a particular type of new part, but the OEM may purchase a particular type of part from multiple suppliers. The DCs, which act as breakbulk and packaging locations between the manufacturing and the retail locations, perform operations that facilitate the economies-of-scale in transportation. Typically, a DC receives new parts from multiple suppliers and serves multiple retailers within its geographical proximity.

At the retailer locations, the induced demand stream generates parts returns that can be used for remanufacturing. In establishing a reverse channel, the OEM can locate the collection centers (CCs) between the retail locations and the product recovery plants to perform sorting and consolidation of parts returns. Similar to the DCs, the CCs provide opportunities to benefit from economies-of-scale in transportation in the reverse channel. Although remanufacturing is a popular practice in the automotive industry, the reverse channel activities such as the collection of used parts, disassembly of recoverable components, and remanufacturing are often performed by small or medium size remanufacturing firms, on an ad-hoc basis. Also, since the remanufacturable parts are sold and purchased as commodities, the retailers may sell some of the parts returns directly to independent remanufacturers and, hence, return only a fraction of the parts returns, which we refer to as the *return fraction*, to the CCs managed by the OEM. At a CC location, the returned parts received from the retailers are cleaned, sorted, and consolidated into individual parts streams.

In practice, there are two common alternatives for retailer assignments. The *single-sourcing* assignment is where a retailer works with only one CC (i.e., it sends all of the returns it receives to its dedicated CC) and with only one DC (i.e., it receives all of its demand from its dedicated DC). Whereas, the *multi-sourcing* assignment is where a retailer works with many CCs and DCs. A single-sourcing requirement may be preferable by the DCs, CCs and the retailers due to operational ease of its implementation. However, the advancements in the information technology to effectively track shipments and deliveries, which are available as a software module in enterprise resource planning software applications, provide opportunities to benefit from possible cost savings of the multi-sourcing strategy without additional operational burden. Moreover, in practice, retailers may prefer to procure service parts from multiple locations to alleviate the difficulties during unforeseen supply interruptions. Furthermore, in practice, a combination of single-sourcing and multi-sourcing strategies may be preferred by the retailers.

After sorting the retailer returns into individual parts stream, the CC locations send the returns stream to the associated remanufacturing locations. The facilities, namely the remanufactured product plants (RPPs), reclaim the components and subassemblies from the consumer returned parts. The remanufacturing process deems some of the parts as unrecoverable, and only a fraction of the parts returns received at a remanufacturing facility can be remanufactured. We refer to this fraction as the *recovery fraction*. The OEM in the automotive industry can invest in remanufacturing facilities or outsource the remanufacturing for a particular type of part. In doing so, the OEM can select a single remanufacturing location to take advantage of learning-by-doing effects, core competencies, quality assurance, and lower unit costs due to consolidated volume. Alternatively, the OEM can select multiple remanufacturing locations for a particular type of part. These alternatives essentially constitute the outsourcing strategies for the remanufacturing operations. The RPPs reclaim the components and sub-assemblies from the consumer returned parts, which, in turn, are used in the remanufactured parts.

The problem setting associated with the CLSC network described above, can be characterized by inclusion or exclusion of finite capacity restrictions on the facilities. The restrictions can be excluded when we have sufficient processing and storage capacities at all the facilities in the CLSC network. Also, the exclusion of the capacity restrictions enables us to develop mathematical models and solution methods that provide insights for the relatively hard-to-solve problems that assume capacity restrictions. Alternatively, the inclusion of the capacity restrictions enables us to consider a significant practical assumption where there are finite processing capacities in production, distribution, and collection activities. In practice, we have capacity limitations due to either the storage space limitation, the finite resources for processing and handling, the manpower, or a combination of these factors.

Based on the above description, we identify the following important features that characterize the network design problems for CLSCs.

(i) Transformation strategies

- Extending the already existing forward channel to incorporate reverse channel activities.
- Designing an entire CLSC.

(ii) Network flow integration strategies

- Using hybrid facilities such as hybrid plants (that are capable of producing new service parts in addition to performing remanufacturing operations) and hybrid centers (that are capable of handling both forward and reverse flow of service parts) to coordinate the forward and reverse flows.
- Using distinct forward and reverse channel facilities to manage the forward and reverse flows, respectively.

(iii) Retailer assignment strategies

- Single-sourcing, where each retailer works with a unique DC to receive all types of service parts and a unique CC to send all types of product returns.
- Multi-sourcing, where a retailer can work with multiple DC and CC locations to receive and send service parts, respectively.

(iv) Remanufacturing location selection strategies

- Selecting a single remanufacturing location for each type of service part.
- Selecting multiple remanufacturing locations for each type of service part.

Based on these features, in this dissertation, we consider three different practical settings for the underlying multi-product CLSC and define the corresponding network design problems, namely an *Uncapacitated Remanufacturing Network Design Problem* (**URP**), *Capacitated Remanufacturing Network Design Problem* (**CRP**), and *Closed-Loop Network Design Problem* (**CLP**). More specifically, in the first two problem settings (i.e., in **URP** and **CRP**), we extend the existing forward channel infrastructure to accommodate distinct reverse channel infrastructure to coordinate the forward and reverse flows. In the latter setting (**CLP**), we design the entire CLSC network by considering hybrid facilities. We next describe these three problem settings.

I.2.1. An Uncapacitated Remanufacturing Network Design Problem

Under this setting, we assume that the manufacturers operate well-established forward networks, i.e., the locations of the new product manufacturing facilities, DCs, and retailers are known. It is worthwhile to note that this is the case for an OEM who has not yet established a reverse network for remanufacturing but has a strategic commitment to do so. In order for the OEM to realize the full potential of remanufacturing, the interdependence between reverse and forward networks should be considered explicitly and the OEM should modify the forward network operations and accommodate reverse flows to transform the existing supply chain into CLSC. As a result, there is a strong motivation to develop quantitative modeling tools that consider the existing infrastructure while designing the reverse network and modifying the flows on the forward network accordingly. Moreover, in order to realize the full potential of remanufacturing, the interdependence between reverse and forward networks should be considered explicitly. The network design issues associated with designing the reverse network, while simultaneously coordinating the forward and reverse flows on the CLSC network, pertain to the following questions:

(i) How should the existing forward network be extended to accommodate the remanufacturing processes, i.e., where should the CCs and RPP facilities be installed?

(ii) How should the forward and reverse flows be routed/coordinated in this extended network?

In this problem, we assume single-sourcing assignments for the retailers in both the forward and reverse channels. This network setting–where the demand at each retailer is satisfied via a single DC (rather than direct shipments) and the returns at each retailer is collected via a single CC (rather than via multiple collection centers per retailer)–captures important practical characteristics of such problems, particularly in the automotive industry.

I.2.2. A Capacitated Remanufacturing Network Design Problem

We consider two important extensions to **URP**. The first one is the inclusion of finite capacity restrictions on the facilities. This inclusion generalizes the CLSC network design problem to consider a significant practical concern associated with finite processing capacities in production, distribution, and collection activities. In practice, we have capacity limitations due to finite resources for processing and handling the forward and reverse flows. Moreover, the inclusion of capacity restrictions influences the network flow of a product relative to the others.

Secondly, we relax the single-sourcing assumption in which a retailer works with only one CC (i.e., it sends all of the returns it receives to its dedicated CC) and with only one DC (i.e., it receives all of its demand from its dedicated DC). More specifically, we consider multi-sourcing for the corresponding relations in the CLSC network. A single-sourcing requirement may be preferable to the DCs, CCs and retailers due to the operational ease of its implementation. However, advancements in information technology for effectively tracking shipments and deliveries (available as a software module in enterprise resource planning software applications) provide opportunities to benefit from possible cost savings from the multi-sourcing strategy without additional operational burdens. Moreover, in practice, retailers may prefer to procure products from multiple locations to alleviate difficulties from unforeseen supply interruptions. Therefore, in the **CRP** problem setting, a retailer can receive shipments from multiple DCs and send the customer returned products to multiple CCs.

I.2.3. A Closed-Loop Network Design Problem

In this problem setting, we generalize the **URP** and **CRP** settings by deciding on the locations of the forward channel facilities, i.e., we determine the optimal locations of the manufacturing/remanufacturing facilities, DCs and CCs. We note that this is the case for an OEM who wishes to establish an entire CLSC network for managing multiple types of service parts. Under this setting, we coordinate the forward and reverse flows using hybrid plants and hybrid centers. The network design issues associated with designing the entire CLSC network, pertain to the following questions:

(i) How should the CLSC network be configured, i.e., where should the hybrid plants and hybrid centers be commissioned?

(ii) How should the forward and reverse flows be routed/coordinated in this CLSC network?

In this problem, we assume single-sourcing assignments for the retailers in both the forward and reverse channels. Moreover, we select a single hybrid plant from a set of candidate locations available for each type of service part.

For the sake of clarity in model development and analysis, we refer to the parts as *products* in the remainder of this document. Also, a supply location that provides new products is referred to as a new product plant (NPP), a supply location where remanufacturing takes place is referred to as a remanufactured product plant (RPP), a supply location where both manufacturing and remanufacturing takes place is referred to as hybrid product plant (HPP), and an intermediate center where both collection and distribution operations are performed is referred to as hybrid center (HC).

I.3. Solution Methodologies

For the three problem settings of interest, we formulate mixed integer linear programs (MILP) to determine the optimal locations of the network facilities along with the integrated forward and reverse flows such that the total cost of facility location, processing and transportation is minimized. These CLSC network design problems dictate large scale MILPs. Furthermore, we can clearly see that these network design problems are generalization of the traditional *Uncapacitated Facility Location Problem* (**UFLP**), which belongs to the class of $\mathcal{NP}$-hard problems (Garey and Johnson, 1979). As a consequence of the combinatorial nature of these problems, obtaining an optimal or near-optimal solution, in general, requires very high computational runtimes, specifically for large scale problem instances. However, the network flow structures underlying our models make them amenable for Benders decomposition (BD).

For the first setting (**URP**), we develop an efficient dual solution approach to generate strong Benders cuts. In addition to the classical single Benders cut approach, we propose three different approaches for adding multiple Benders cuts. We present computational results that illustrate the superior performance of the proposed solution methodology with multiple Benders cuts in comparison to the branch-and-cut (B&C) approach and the traditional BD approach with a single cut.

For the second setting (**CRP**), we devise two tabu search heuristics in which we effectively combine simple neighborhood search functions utilizing moves and exchanges to improve the efficiency of exploration. We propose a transshipment heuristic to quickly, but effectively, estimate the objective function value of a feasible solution in the course of a tabu search. We also present a BD approach that incorporates the tabu search heuristics and the strong Benders cuts to facilitate faster convergence and improve computational efficiency, especially for large scale instances. We present our computational results illustrating the superior performance of the solution algorithms developed based on the heuristics and BD approach in terms of both solution quality and computation time.

For the third setting (**CLP**), we propose a BD approach that utilizes the dual solution approach for obtaining strong cuts, developed in the first setting. In addition to this method, we present alternate formulations for this problem and develop alternate strong cuts. We also present different approaches for combining the strong cuts and alternate strong cuts to facilitate improvements in computational efficiency. We present computational results and compare the computational performance among the BD approach with strong, alternate strong and combined strong cuts.

I.4. Organization of the Dissertation

The remainder of this dissertation is organized as follows. In Chapter II, we provide a brief overview of related literature in CLSC network design. In Chapter III, we present the mathematical model and develop the components of the BD approach for **URP**. In Chapter IV, we discuss the model formulation for **CRP** along with the components of Tabu search heuristics and the heuristics-enhanced BD approach. Chapter V focuses on the problem setting for **CLP**. In this chapter, we present alternative formulations for **CLP** and develop a BD approach to obtain strong cuts using these alternative formulations. Conclusions and future research directions are summarized in Chapter VI.

CHAPTER II

LITERATURE REVIEW

The general topic of network design for product recovery has received considerable attention in recent years. For a comprehensive review of the literature, refer to Bloemhof-Ruwaard et al. (1999), Dekker et al. (2004), Fleischmann (2001), Fleischmann et al. (1997), and Fleischmann et al. (2000). A most recent comprehensive review that exclusively focuses on network design for reverse and CLSC systems is presented in Akçalı et al. (2007).

The existing network design models can be classified according to underlying network structure by making a distinction between *reverse supply chain models* and *CLSC models*. Reverse supply chain models *only* consider reverse flows: The source nodes are the collection locations; the sink nodes are the disposal or re-use locations; and, since the forward network is excluded in these models, the transportation links are solely for the reverse flows. CLSC models consider *both* the reverse and forward flows: The locations on the corresponding network may serve as sink and source nodes, and the transportation links are for both the forward and reverse flows. Since our model belongs to the class of CLSC models, we focus our review of the literature on this line of work which consists of case-based (Krikke et al., 2003) and generic (Beamon and Fernandes, 2004; Fleischmann et al., 2001; Lu and Bostel, 2007; Sim et al., 2004) models.

Krikke et al. (2003) propose to integrate product design and CLSC design, and they present a case study analyzing the interaction between the two design aspects. Beamon and Fernandes (2004) consider a generic network structure, which consists of capacitated hybrid manufacturing/remanufacturing facilities, uncapacitated distribution centers, capacitated collection centers, and retailers, with the following flow characteristics:

(i) a single product is produced at several hybrid manufacturing/remanufacturing facilities,

(ii) the known demand at the retailers is satisfied by shipping via distribution centers,

(iii) the known returns at the retailers are shipped via collection centers to hybrid manufacturing and remanufacturing facilities, and

(iv) each node on the network can be supplied by, and can supply, multiple nodes.

Their model seeks the optimal location of distribution and collection centers (between the known locations of manufacturing/remanufacturing facilities and retailers) and the routing of forward and reverse flows throughout the corresponding network. An MILP model is formulated and solved using the B&C approach that is available in commercial software. The network structure considered by Fleischmann et al. (2001) and Sim et al. (2004) is similar to the one considered by Beamon and Fernandes (2004). More specifically, Fleischmann et al. (2001) consider the uncapacitated version of the Beamon and Fernandes (2004) model, where the locations of the hybrid manufacturing/remanufacturing facilities are also decision variables. They use commercial software to obtain the optimal solution. Sim et al. (2004) extend the Fleischmann et al. (2001) model to consider the multi-product case where all nodes of the network are capacitated. They develop a genetic algorithm-based heuristic approach for computing a solution.

Lu and Bostel (2007) consider a different network structure consisting of separate manufacturing and remanufacturing facilities as well as collection centers and retailers. Specifically, they develop a facility location model for remanufacturing with the following flow characteristics:

(i) demand for a single product can be satisfied by new or remanufactured products produced at manufacturing and remanufacturing facilities, respectively,

(ii) the known demand at the retailers is satisfied by direct shipments to each retailer,

(iii) the known returns at the retailers are shipped via collection centers to remanufacturing facilities, and

(iv) each node on the network can be supplied by, and can supply, multiple nodes.

They develop a heuristic approach, based on Lagrangian relaxation, which searches for the location of manufacturing facilities, remanufacturing facilities, and collection centers and routes the forward and reverse flows through the corresponding network.

Recently, Sahyouni et al. (2007) consider an uncapacitated fixed-charge location model that decides on the locations of collection centers and distribution centers, in addition to the retailer assignments. They propose a solution algorithm based on Lagrangian relaxation and provide a comparison between the sequential and integrated decision making approaches.

The traditional production/distribution system design literature is also closely related to network design for CLSC systems. There is a considerable amount of previous work in this area Brown et al. (1987); Geoffrion and Graves (1974); Jayaraman and Pirkul (2001); Pirkul and Jayaraman (1996); Pyke and Cohen (1994) that analyzes the optimal locations of manufacturing facilities and distribution centers. Although a detailed review of this literature is beyond the scope of this dissertation, it is worthwhile to note that the forward network structure we consider is based on the classical production/distribution system design modeled in Geoffrion and Graves (1974); Keskin and Üster (2007).

More specifically, Geoffrion and Graves (1974) consider a network structure, which consists of capacitated manufacturing facilities, capacitated distribution centers, and retailers, with the following flow characteristics:

(i) multiple products are produced at several manufacturing facilities, and

(ii) the known demand for multiple products at the retailers is satisfied by shipping via distribution centers where each retailer is served exclusively by a distribution center, i.e., each retailer is supplied via a single-source.

Their model seeks the optimal location of distribution centers (between the known locations of manufacturing facilities and retailers/customers) and the assignment of retailers to distribution centers under single-sourcing restrictions. Keskin and Üster (2007) employ a multi-sourcing strategy for retailer assignments as opposed to the single-sourcing assumption in Geoffrion and Graves (1974). In our models, the commonality with these traditional production/distribution system design models lies in the consideration of multiple products and the capacitated NPPs and DCs, single-sourcing (as in Geoffrion and Graves (1974)), and multi-sourcing strategy (as in Keskin and Üster (2007)). We consider a general network structure, which consists of manufacturing facilities, remanufacturing facilities, distribution centers, collection centers, and retailers. We develop MILP formulations that seek an optimal solution, which characterizes

(i) the location of facilities,

(ii) the assignment of retailers,

(iii) the assignment of products to remanufacturing/manufacturing facilities, and

(iv) the coordinated forward and reverse flows in the CLSC network.

From the modeling perspective, we extend the previous work by considering multiple products (Beamon and Fernandes, 2004; Fleischmann et al., 2001; Lu and Bostel, 2007), separate manufacturing and remanufacturing facilities (Beamon and Fernandes, 2004; Fleischmann et al., 2001; Sim et al., 2004), and indirect shipments via distribution centers (Lu and Bostel, 2007).

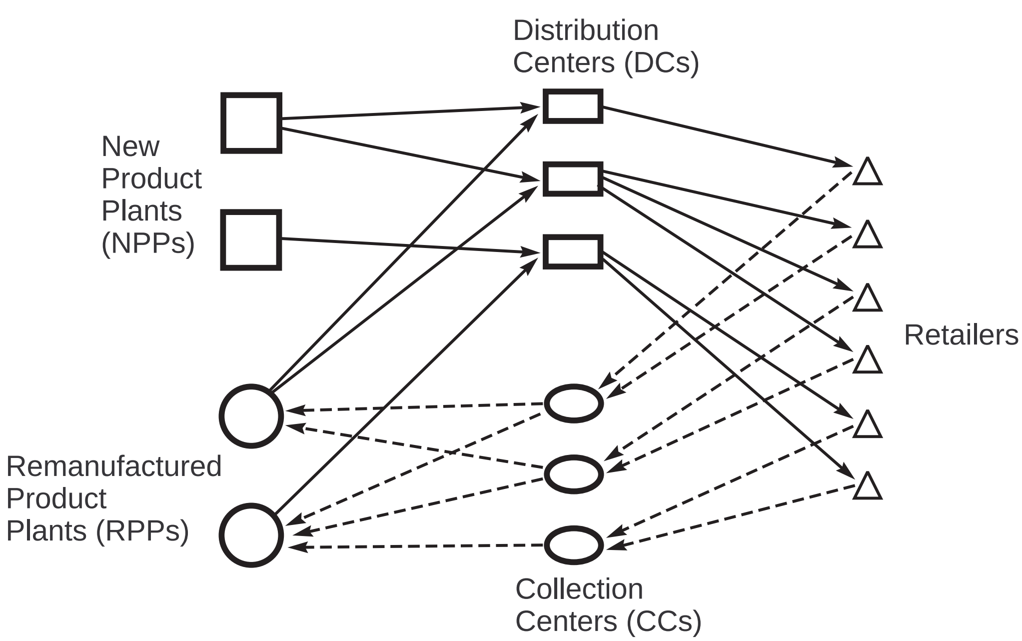
From the methodological perspective, the previous work mainly relies on the use of commercial software (Beamon and Fernandes, 2004; Fleischmann et al., 2001; Krikke et al., 2003) or the development of heuristic approaches (Lu and Bostel, 2007; Sim et al., 2004) for CLSC design, whereas we propose heuristic approaches in addition to exact solution methodologies that build on the BD framework (Benders, 1962).

CHAPTER III

AN UNCAPACITATED REMANUFACTURING NETWORK DESIGN PROBLEM

In the **URP**, given the locations of the suppliers (NPPs) and the distribution centers (DCs), the objective is to determine the locations of the collection centers (CCs) and remanufacturing facilities (RPPs) along with the forward and reverse flows such that the processing costs (associated with manufacturing, remanufacturing, collection, and distribution), the transportation costs (associated with forward distribution and reverse collection), and the facility location costs (associated with establishing collection centers and remanufacturing facilities) are minimized. A general network structure is depicted in Figure 1.

Figure 1 General Structure of the CLSC Network.



III.1. Assumptions and Operational Characteristics

We provide a detailed exposition of the specific underlying assumptions and operational characteristics of the **URP**:

1. Each retailer satisfies the new and the induced demand for all types of products and, hence, may receive all types of product returns. The location of the retailers as well as the new product demand, the induced demand due to product returns, and the returns for each product type at each retailer are known.
2. Each retailer works with one CC and sends product returns to its CC (i.e., single reverse link assignment per retailer).
3. Each retailer returns a fraction of the product returns received to its CC, and the return fraction of a retailer for each product type is known.
4. There is a potential set of locations where the CCs can be opened. Opening a CC incurs a fixed location cost. Each CC sorts and performs other processes on returns, incurring a variable processing cost.
5. Each CC can receive returns from multiple retailers, but is assigned to exactly one RPP per product (as a consequence of having a single RPP per product as we explain next).
6. Exactly one RPP location per product is to be determined from the set of potential product-specific RPP locations. Opening a RPP incurs a fixed location cost. Each RPP remanufactures a single type of product, incurring a variable remanufacturing cost for each product shipped out of the remanufacturing plant. Without loss of generality, we assume that disposal cost associated with each product is zero as this is a sunk cost.
7. The recovery fraction at a RPP is known. Furthermore, all remanufactured products should be used to satisfy the demand (i.e., there is no planned disposal

at the RPPs).

8. Each NPP can supply one type of product, but there may be multiple NPPs for each product. The locations of NPPs for each product are known.
9. Each DC can receive products from either a RPP or a NPP or both. The DC locations are known and the product flow through a DC incurs a variable processing cost.
10. Each DC can supply multiple retailers with any type of product, but each retailer can receive its products from only one DC (i.e., single forward link assignment per retailer).
11. Transportation costs are linear and based on direct shipments on the network illustrated in Figure 1.

III.2. Problem Formulation

We proceed with a discussion of the objective function and constraints of the mathematical formulation of the **URP**. For this purpose, first we introduce the following notation for model development. In Figure 2, we depict the underlying network structure and illustrate the location sets along with the flow, assignment, and location variables in the CLSC under consideration.

Sets and Indices

$\mathcal{P}$ set of products, $p \in \mathcal{P}$.

$\mathcal{R}$ set of retailers, $r \in \mathcal{R}$.

$\mathcal{K}$ set of candidate CC locations, $k \in \mathcal{K}$.

$\mathcal{D}$ set of DCs, $d \in \mathcal{D}$.

$\mathcal{S}_p$ set of candidate RPP locations for $p \in \mathcal{P}, s \in \mathcal{S}_p$.

$\mathcal{T}_p$ set of NPP locations for $p \in \mathcal{P}, t \in \mathcal{T}_p$.

Parameters

D_{pr} induced demand at retailer $r \in \mathcal{R}$ for product $p \in \mathcal{P}$.

D'_{pr} new demand at retailer $r \in \mathcal{R}$ for product $p \in \mathcal{P}$.

F'_k fixed cost of opening a CC at location $k \in \mathcal{K}$.

F_{ps} fixed cost of opening a RPP for product $p \in \mathcal{P}$ at location $s \in \mathcal{S}_p$.

G_{ij} unit transportation cost from a location i to a location j for $i, j \in \mathcal{R} \cup \mathcal{D} \cup \mathcal{K} \cup \mathcal{S}_p \cup \mathcal{T}_p$.

η_{pd} unit processing cost of product $p \in \mathcal{P}$ at DC $d \in \mathcal{D}$.

κ_{pk} unit processing cost of product $p \in \mathcal{P}$ at CC $k \in \mathcal{K}$.

ν_{pt} unit manufacturing cost of product $p \in \mathcal{P}$ at NPP $t \in \mathcal{T}_p$.

ρ_{ps} unit remanufacturing cost of product $p \in \mathcal{P}$ shipped out of RPP $s \in \mathcal{S}_p$.

δ_{pr} return fraction at retailer $r \in \mathcal{R}$ for product $p \in \mathcal{P}$.

α_{ps} recovery fraction for product $p \in \mathcal{P}$ at RPP $s \in \mathcal{S}_p$.

Decision Variables

v_{ps} 1 if RPP $s \in \mathcal{S}_p$ is used for product $p \in \mathcal{P}$, 0 otherwise.

c_k 1 if CC $k \in \mathcal{K}$ is opened, 0 otherwise.

u_{rk} 1 if retailer $r \in \mathcal{R}$ is assigned to CC $k \in \mathcal{K}$, 0 otherwise.

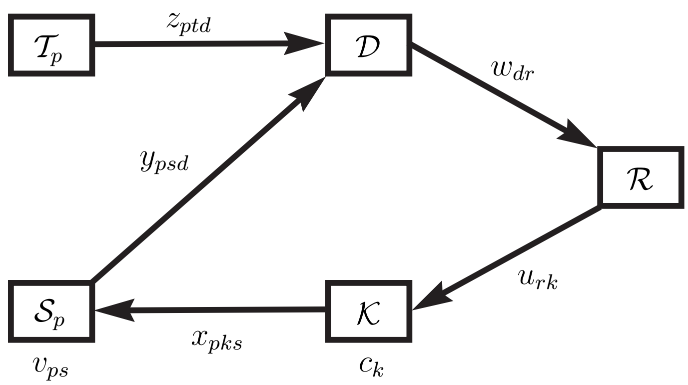
w_{dr} 1 if retailer $r \in \mathcal{R}$ is assigned to DC $d \in \mathcal{D}$, 0 otherwise.

x_{pks} quantity of product $p \in \mathcal{P}$ shipped from CC $k \in \mathcal{K}$ to RPP $s \in \mathcal{S}_p$.

y_{psd} quantity of product $p \in \mathcal{P}$ shipped from RPP $s \in \mathcal{S}_p$ to DC $d \in \mathcal{D}$.

z_{ptd} quantity of product $p \in \mathcal{P}$ shipped from NPP $t \in \mathcal{T}_p$ to DC $d \in \mathcal{D}$.

Figure 2 Underlying Structure of the CLSC Network for **URP**.



Objective Function

$$
\begin{aligned}
\min \quad & \sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p} F_{ps}v_{ps} + \sum_{k\in\mathcal{K}} F'_k c_k + \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\sum_{k\in\mathcal{K}} \left(G_{rk}+\kappa_{pk}\right)\delta_{pr}D_{pr}u_{rk} \\
+ \quad & \sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}_p} G_{ks}x_{pks} + \sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p}\sum_{d\in\mathcal{D}} \left(G_{sd}+\rho_{ps}\right)y_{psd} \\
+ \quad & \sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}_p}\sum_{d\in\mathcal{D}} \left(G_{td}+\nu_{pt}\right)z_{ptd} \\
+ \quad & \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} \left(G_{dr}+\eta_{pd}\right)\left(D_{pr}+D'_{pr}\right)w_{dr}.
\end{aligned} \tag{3.1}
$$

The first term in the objective function represents the fixed costs associated with locating the product-specific RPPs for each type of product. The second term represents the fixed costs of locating the CCs. The third and the fourth terms are associated with the reverse flows, and they represent transportation costs from the retailers to the CCs and from the CCs to the RPPs, respectively. The third term also includes the processing costs at the CCs. The fifth and the sixth terms are associated with the forward flows, and they represent the transportation costs from the RPPs to the DCs and from the NPPs to the DCs along with the associated processing costs at the RPPs and NPPs, respectively. Finally, the last term corresponds to the transportation costs from the DCs to the retailers and the processing costs at the DCs. In order to simplify the expressions representing the flow and processing costs, in this chapter, henceforth we employ the notation G_{prk}, G_{psd}, G_{ptd} and G_{pdr} to represent the sums $(G_{rk} + \kappa_{pk})$, $(G_{sd} + \rho_{ps})$, $(G_{td} + \nu_{pt})$ and $(G_{dr} + \eta_{pd})$, respectively.

Constraints

$$\sum_{k \in \mathcal{K}} u_{rk} = 1 \qquad \forall\, r \in \mathcal{R}, \tag{3.2}$$

$$\sum_{s \in \mathcal{S}_p} v_{ps} = 1 \qquad \forall\, p \in \mathcal{P}, \tag{3.3}$$

$$\sum_{d \in \mathcal{D}} w_{dr} = 1 \qquad \forall\, r \in \mathcal{R}, \tag{3.4}$$

$$u_{rk} \leq c_k \qquad \forall\, r \in \mathcal{R},\, k \in \mathcal{K}, \tag{3.5}$$

$$x_{pks} \leq M\, v_{sp} \qquad \forall\, p \in \mathcal{P},\, k \in \mathcal{K},\, s \in \mathcal{S}_p, \tag{3.6}$$

$$\sum_{s \in \mathcal{S}_p} x_{pks} - \sum_{r \in \mathcal{R}} \delta_{pr} D_{pr}\, u_{rk} = 0 \qquad \forall\, p \in \mathcal{P},\, k \in \mathcal{K}, \tag{3.7}$$

$$\sum_{d \in \mathcal{D}} y_{psd} - \alpha_{ps} \sum_{k \in \mathcal{K}} x_{pks} = 0 \qquad \forall\, p \in \mathcal{P},\, s \in \mathcal{S}_p, \tag{3.8}$$

$$\sum_{s \in \mathcal{S}_p} y_{psd} + \sum_{t \in \mathcal{T}_p} z_{ptd} = \sum_{r \in \mathcal{R}} \left(D_{pr} + D'_{pr}\right) w_{dr} \qquad \forall\, p \in \mathcal{P},\, d \in \mathcal{D}, \tag{3.9}$$

$$x_{pks},\, y_{psd},\, z_{ptd} \geq 0 \qquad \forall\, p \in \mathcal{P},\, k \in \mathcal{K},\, d \in \mathcal{D},\, s \in \mathcal{S}_p,\, t \in \mathcal{T}_p, \tag{3.10}$$

$$v_{ps}, c_k, w_{dr}, u_{rk} \in \{0, 1\} \qquad \forall\, p \in \mathcal{P},\, r \in \mathcal{R},\, k \in \mathcal{K},\, d \in \mathcal{D},\, s \in \mathcal{S}_p. \tag{3.11}$$

Constraint set (3.2) ensures that each retailer r is assigned to exactly one CC k. Constraint set (3.3) guarantees that, for each product p, a RPP location s is established. Constraint set (3.4) represents the requirement that each retailer r is uniquely assigned to one DC d. Constraint set (3.5) forces the creation of a CC k if a retailer r is assigned to that location. Similarly, constraint set (3.6) makes sure that a RPP is established for a product p at location s if there is flow of product p from a CC k to this candidate RPP location. Since x_{pks} represent non-negative continuous flow variables and v_{ps} represent binary location variables, M simply represents a sufficiently large scalar. Constraint sets (3.7) and (3.8) represent the flow conservation (mass balance) for each product type at the CCs and RPPs, respectively. Constraint set (3.8) implicitly guarantees that the remanufactured products are returned to the system in their entirety to satisfy customer demand. Constraint set (3.9) ensures that the customer demand is completely satisfied using the remanufactured products (as implied by constraint set (3.8)) and new products as necessary.

Note that, for product p, the total reverse flow on the links from CCs to the corresponding RPP must be equal to the total amount returned by the retailers. This observation leads us to a more efficient formulation of the problem that lends itself to a decomposition in terms of the forward and reverse *flow* problems. More specifically, we have

$$\sum_{k\in\mathcal{K}} x_{pks} = \sum_{r\in\mathcal{R}} \delta_{pr} D_{pr}\, v_{ps} \qquad \forall\, p\in\mathcal{P},\ s\in\mathcal{S}_p, \tag{3.12}$$

and, hence, constraint set (3.8) can be restated as

$$\sum_{d\in\mathcal{D}} y_{psd} - \alpha_{ps} \sum_{r\in\mathcal{R}} \delta_{pr}\, D_{pr}\, v_{ps} = 0 \qquad \forall\, p\in\mathcal{P},\ s\in\mathcal{S}_p. \tag{3.13}$$

Furthermore, we observe that constraint set (3.12) effectively addresses the RPP location constraints given by (3.6). Thus, by including constraint sets (3.12) and (3.13) and eliminating constraint sets (3.6) and (3.8), we not only obtain a formulation with constraints that are separable in terms of the forward and the reverse flow variables, but also eliminate the M from the formulation and thereby facilitate the availability of stronger linear programming relaxation bounds. In particular, constraint sets (3.9) and (3.13) involve only forward flow variables, and constraint sets (3.7) and (3.12) involve only reverse flow variables. We note that the revised formulation relies on the assumption that there is a single, dedicated RPP for each product, and, as will be clear in the following section, it is is very helpful for developing efficient solution algorithms to solve the subproblems in a BD framework.

III.3. Solution Approach Using Benders Decomposition

BD approach (Benders, 1962) relies on forming two problems, a master problem and a subproblem based on the original formulation, and on solving these problems in an iterative fashion. The master problem typically involves the set of integer variables in the original problem and an auxiliary continuous variable that relates it to the subproblem. On the other hand, the subproblem is created as a relatively easier problem that involves only continuous variables. At each iteration, the master and the subproblem are solved to obtain lower and upper bounds on the objective value of the original problem. The objective function of the *dual subproblem* is utilized to add the so-called Benders cuts to the master problem and, thus, to strengthen the bounds at each iteration. It is exactly these cuts that introduce the auxiliary continuous variables to the master problem.

For our formulation, the master problem mainly includes the binary variables associated with the locations of the RPPs and CCs, the assignment of retailers to CCs, and the assignment of retailers to DCs. Given the values of these binary location and assignment variables, the subproblem includes the continuous variables representing the forward and reverse flows. We start the iterative procedure by solving the master problem without any Benders cuts. Next, we solve the dual subproblem by incorporating the initial master problem solution, and, thus, we obtain the first set of Benders cuts. Afterwards, at each iteration, we add a new set of cuts into the master problem, obtain its solution, and use it to solve the dual subproblem, which provides a new set of Benders cuts. We also update the lower and upper bound values of the objective function value of the original problem in order to decide if the iterations should be terminated. In order to develop the components of this iterative framework, first we provide the underlying Benders reformulation and introduce the subproblem. More specifically, the original problem can be restated as follows.

$$\begin{aligned}\min \quad Z \;=\; & \sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p} F_{ps}v_{ps} + \sum_{k\in\mathcal{K}} F'_k c_k + \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\sum_{k\in\mathcal{K}} G_{prk}\,\delta_{pr}\,D_{pr}\,u_{rk} \\ + \; & \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} G_{pdr}(D_{pr}+D'_{pr})w_{dr} + SP(\mathbf{x},\mathbf{y},\mathbf{z}|\widehat{\mathbf{v}},\widehat{\mathbf{u}},\widehat{\mathbf{w}}) \qquad (3.14)\end{aligned}$$

subject to (3.2), (3.3), (3.4), (3.5) and (3.11).

where $SP(\mathbf{x},\mathbf{y},\mathbf{z}|\widehat{\mathbf{v}},\widehat{\mathbf{u}},\widehat{\mathbf{w}})$ represents the Benders subproblem whose formulation and solution procedure are discussed next. Observe that it is not necessary to include the information on open collection centers, $\widehat{\mathbf{c}}$, in the Benders subproblem since it can be derived from the retailer assignments to collection centers, $\widehat{\mathbf{u}}$. It is thus not included in the Benders subproblem. As we demonstrate in Sections III.3.1 and III.3.2, the Benders subproblem is separable and, hence, can be solved efficiently based on the forward flows, the reverse flows, and the product types. Utilizing this property, in Section III.3.3, we introduce four different alternatives for generating effective Benders cuts.

III.3.1. Benders Subproblem

The subproblem $SP(\mathbf{x}, \mathbf{y}, \mathbf{z} | \widehat{\mathbf{v}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$ is essentially a minimization problem that determines the optimum values of the flow variables for fixed values of the location and assignment variables, and it can be stated as

$$\min Z_{SP} = \sum_{p\in\mathcal{P}} \sum_{k\in\mathcal{K}} \sum_{s\in\mathcal{S}_p} G_{ks}\, x_{pks} + \sum_{p\in\mathcal{P}} \sum_{s\in\mathcal{S}_p} \sum_{d\in\mathcal{D}} G_{psd}\, y_{psd} + \sum_{p\in\mathcal{P}} \sum_{t\in\mathcal{T}_p} \sum_{d\in\mathcal{D}} G_{ptd}\, z_{ptd} \quad (3.15)$$

subject to

$$\sum_{d\in\mathcal{D}} y_{psd} = \sum_{r\in\mathcal{R}} \alpha_{ps}\, \delta_{pr}\, D_{pr}\, \widehat{v}_{ps} \qquad \forall\, p\in\mathcal{P},\ s\in\mathcal{S}_p, \quad (3.16)$$

$$\sum_{s\in\mathcal{S}_p} y_{psd} + \sum_{t\in\mathcal{T}_p} z_{ptd} = \sum_{r\in\mathcal{R}} \left(D_{pr} + D'_{pr}\right) \widehat{w}_{dr} \qquad \forall\, p\in\mathcal{P},\ d\in\mathcal{D}, \quad (3.17)$$

$$\sum_{k\in\mathcal{K}} x_{pks} \geq \sum_{r\in\mathcal{R}} \delta_{pr}\, D_{pr}\, \widehat{v}_{ps} \qquad \forall\, p\in\mathcal{P},\ s\in\mathcal{S}_p, \quad (3.18)$$

$$\sum_{s\in\mathcal{S}_p} x_{pks} \leq \sum_{r\in\mathcal{R}} \delta_{pr} D_{pr}\, \widehat{u}_{rk} \qquad \forall\, p\in\mathcal{P},\ k\in\mathcal{K}, \quad (3.19)$$

$$x_{pks},\ y_{psd},\ z_{ptd} \geq 0 \qquad \forall\, p\in\mathcal{P},\ k\in\mathcal{K},\ d\in\mathcal{D},\ s\in\mathcal{S}_p,\ t\in\mathcal{T}_p. \quad (3.20)$$

We note that, without any effect on the final optimal solution, the equality constraints (3.7) and (3.12) are represented by inequalities (3.18) and (3.19). This alternative representation does not affect the solution space, but it does facilitate an easy exposition for the solution of the dual of the subproblem to generate Benders cuts as explained below.

As mentioned above, in the BD framework, we employ the solution and the objective function of the dual subproblem in order to generate the Benders cuts. We define dual variables β_{ps} associated with constraints (3.16), γ_{pd} associated with constraints (3.17), λ_{ps} associated with constraint (3.18), and μ_{pk} associated with constraints (3.19). Then, the dual of the subproblem, $DSP(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\mu} | \widehat{\mathbf{v}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$, can be stated as

$$\begin{aligned} \max \quad Z_{DSP} \;=\; & \sum_{p\in\mathcal{P}} \sum_{s\in\mathcal{S}_p} \sum_{r\in\mathcal{R}} \alpha_{ps}\, \delta_{pr}\, D_{pr}\, \widehat{v}_{ps}\, \beta_{ps} + \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} \sum_{r\in\mathcal{R}} \left(D_{pr} + D'_{pr}\right) \widehat{w}_{dr}\, \gamma_{pd} \\ + \; & \sum_{p\in\mathcal{P}} \sum_{s\in\mathcal{S}_p} \sum_{r\in\mathcal{R}} \delta_{pr}\, D_{pr}\, \widehat{v}_{ps}\, \lambda_{ps} - \sum_{p\in\mathcal{P}} \sum_{k\in\mathcal{K}} \sum_{r\in\mathcal{R}} \delta_{pr}\, D_{pr}\, \widehat{u}_{rk}\, \mu_{pk} \end{aligned} \tag{3.21}$$

subject to

$$\beta_{ps} + \gamma_{pd} \leq G_{psd} \qquad \forall\, p \in \mathcal{P},\ s \in \mathcal{S}_p,\ d \in \mathcal{D}, \tag{3.22}$$

$$\gamma_{pd} \leq G_{ptd} \qquad \forall\, p \in \mathcal{P},\ t \in \mathcal{T}_p,\ d \in \mathcal{D}, \tag{3.23}$$

$$\lambda_{ps} - \mu_{pk} \leq G_{ks} \qquad \forall\, p \in \mathcal{P},\ s \in \mathcal{S}_p,\ k \in \mathcal{K}, \tag{3.24}$$

$$\beta_{ps},\ \gamma_{pd} \text{ unrestricted and } \mu_{pk},\ \lambda_{ps} \geq 0 \qquad \forall\, p \in \mathcal{P},\ s \in \mathcal{S}_p,\ k \in \mathcal{K},\ d \in \mathcal{D}. \tag{3.25}$$

In order to efficiently solve the subproblem and obtain its optimum dual solution for a given set of values for the integer variables, i.e., for given $\widehat{\mathbf{v}}, \widehat{\mathbf{u}}$, and $\widehat{\mathbf{w}}$, we observe that the subproblem $SP(\mathbf{x}, \mathbf{y}, \mathbf{z} | \widehat{\mathbf{v}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$ is separable in terms of the forward flow variables, $\mathbf{y}$ and $\mathbf{z}$, and the reverse flow variables, $\mathbf{x}$. This observation implies that the dual of the subproblem is also separable according to the direction of flow, i.e., *forward* and *reverse* flows, resulting in forward and reverse subproblems. As we have mentioned before, this separability is due to the requirement of a single, dedicated RPP per product. Furthermore, we observe that the forward and reverse subproblems are separable for each product leading to *single product forward* and *single product reverse* subproblems. Next, we state the subproblems and their formulations.

III.3.1.1. Single Product Forward Subproblem

The subproblem $SP(\mathbf{x},\mathbf{y},\mathbf{z}|\widehat{\mathbf{v}},\widehat{\mathbf{u}},\widehat{\mathbf{w}})$ yields $|\mathcal{P}|$ - singe product forward subproblems. Each such subproblem, denoted by $FSP_{\bar{p}}(\mathbf{y},\mathbf{z}|\widehat{\mathbf{v}},\widehat{\mathbf{w}})$, $\bar{p}\in\mathcal{P}$, is given by

$$\min \quad \sum_{s\in\mathcal{S}_{\bar{p}}}\sum_{d\in\mathcal{D}} G_{\bar{p}sd}\, y_{\bar{p}sd} + \sum_{t\in\mathcal{T}_{\bar{p}}}\sum_{d\in\mathcal{D}} G_{\bar{p}td}\, z_{\bar{p}td} \tag{3.26}$$

subject to

$$\sum_{d\in\mathcal{D}} y_{\bar{p}sd} = \sum_{r\in\mathcal{R}} \alpha_{\bar{p}s}\delta_{\bar{p}r}D_{\bar{p}r}\widehat{v}_{\bar{p}s} \qquad \forall\, s\in\mathcal{S}_{\bar{p}}, \tag{3.27}$$

$$\sum_{s\in\mathcal{S}_{\bar{p}}} y_{\bar{p}sd} + \sum_{t\in\mathcal{T}_{\bar{p}}} z_{\bar{p}td} = \sum_{r\in\mathcal{R}} \left(D_{\bar{p}r} + D'_{\bar{p}r}\right)\widehat{w}_{dr} \qquad \forall\, d\in\mathcal{D}, \tag{3.28}$$

$$y_{\bar{p}sd},\ z_{\bar{p}td} \geq 0 \qquad \forall\, d\in\mathcal{D},\ s\in\mathcal{S}_{\bar{p}},\ t\in\mathcal{T}_{\bar{p}}. \tag{3.29}$$

As a result, the dual of the single product forward subproblem above, denoted by $DFSP_{\bar{p}}(\boldsymbol{\beta},\boldsymbol{\gamma}|\widehat{\mathbf{v}},\widehat{\mathbf{w}})$, is given by

$$\max \quad \sum_{s\in\mathcal{S}_{\bar{p}}}\sum_{r\in\mathcal{R}} \alpha_{\bar{p}s}\,\delta_{\bar{p}r}\,D_{\bar{p}r}\,\widehat{v}_{\bar{p}s}\,\beta_{\bar{p}s} + \sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} \left(D_{\bar{p}r} + D'_{\bar{p}r}\right)\widehat{w}_{dr}\,\gamma_{\bar{p}d} \tag{3.30}$$

subject to

$$\beta_{\bar{p}s} + \gamma_{\bar{p}d} \leq G_{\bar{p}sd} \qquad \forall\, s\in\mathcal{S}_{\bar{p}},\ d\in\mathcal{D}, \tag{3.31}$$

$$\gamma_{\bar{p}d} \leq G_{\bar{p}td} \qquad \forall\, t\in\mathcal{T}_{\bar{p}},\ d\in\mathcal{D}, \tag{3.32}$$

$$\beta_{\bar{p}s},\ \gamma_{\bar{p}d}\ \text{ unrestricted} \qquad \forall\, s\in\mathcal{S}_{\bar{p}},\ d\in\mathcal{D}. \tag{3.33}$$

III.3.1.2. Single Product Reverse Subproblem

Similarly, we have $|\mathcal{P}|$ - single product reverse subproblems, and each such subproblem, denoted by $RSP_{\bar{p}}(\mathbf{x}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$, $\bar{p} \in \mathcal{P}$, is given by

$$\min \quad \sum_{k\in\mathcal{K}} \sum_{s\in\mathcal{S}_{\bar{p}}} G_{ks}\, x_{\bar{p}ks} \tag{3.34}$$

subject to

$$\sum_{k\in\mathcal{K}} x_{\bar{p}ks} \geq \sum_{r\in\mathcal{R}} \delta_{\bar{p}r}\, D_{\bar{p}r}\, \widehat{v}_{\bar{p}s} \qquad \forall\, s \in \mathcal{S}_{\bar{p}}, \tag{3.35}$$

$$\sum_{s\in\mathcal{S}_{\bar{p}}} x_{\bar{p}ks} \leq \sum_{r\in\mathcal{R}} \delta_{\bar{p}r}\, D_{\bar{p}r}\, \widehat{u}_{rk} \qquad \forall\, k \in \mathcal{K}, \tag{3.36}$$

$$x_{\bar{p}ks} \geq 0 \qquad \forall\, k \in \mathcal{K},\; s \in \mathcal{S}_{\bar{p}}, \tag{3.37}$$

whereas its dual, denoted by $DRSP_{\bar{p}}(\boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$, is given by

$$\max \quad \sum_{s\in\mathcal{S}_{\bar{p}}} \sum_{r\in\mathcal{R}} \delta_{\bar{p}r}\, D_{\bar{p}r}\, \widehat{v}_{\bar{p}s}\, \lambda_{\bar{p}s} - \sum_{k\in\mathcal{K}} \sum_{r\in\mathcal{R}} \delta_{\bar{p}r}\, D_{\bar{p}r}\, \widehat{u}_{rk}\, \mu_{\bar{p}k} \tag{3.38}$$

subject to

$$\lambda_{\bar{p}s} - \mu_{\bar{p}k} \leq G_{ks} \qquad \forall\, s \in \mathcal{S}_{\bar{p}},\; k \in \mathcal{K}, \tag{3.39}$$

$$\mu_{\bar{p}k},\; \lambda_{\bar{p}s} \geq 0 \qquad \forall\, s \in \mathcal{S}_{\bar{p}},\; k \in \mathcal{K}. \tag{3.40}$$

III.3.2. Solving the Subproblems

In order to solve $DSP(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$ efficiently, we first examine the solutions of $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$ and $DRSP_{\bar{p}}(\boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$ for all $\bar{p} \in \mathcal{P}$. These solutions lead directly to the solution of $DSP(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$ and, hence, the solution and the objective function value of $SP(\mathbf{x}, \mathbf{y}, \mathbf{z}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$. Next, we discuss in detail how we obtain the solutions of each of these dual subproblems.

III.3.2.1. Solving the Single Product Forward Subproblems

For a given product $\bar{p} \in \mathcal{P}$, the single product forward subproblem, which is represented by $FSP_{\bar{p}}(\mathbf{y}, \mathbf{z}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$, is given by (3.26)-(3.29). This is similar to a transportation problem with side constraints that define specific shipment requirements. More specifically, the set of source nodes includes (i) the specific RPP location $s \in \mathcal{S}_{\bar{p}}$ to which product $\bar{p}$ is assigned according to $\widehat{\mathbf{v}}$, i.e., $s \in \mathcal{S}_{\bar{p}}$ such that $\widehat{v}_{\bar{p}s} = 1$, and (ii) the set of NPP locations $\mathcal{T}_{\bar{p}}$. Similarly, the set of sink nodes is given by the DC locations $d \in \mathcal{D}$ for which at least one retailer $r \in \mathcal{R}$ is assigned according to $\widehat{\mathbf{w}}$, i.e., $d \in \mathcal{D}$ such that $\widehat{w}_{dr} = 1$ for at least one retailer $r \in \mathcal{R}$. Let (s) denote the RPP location $s \in \mathcal{S}_{\bar{p}}$ to which product $\bar{p}$ is assigned according to $\widehat{\mathbf{v}}$, i.e., $\widehat{v}_{\bar{p}(s)} = 1$. By (3.27), the RPP location (s) has a supply that is equal to the $\alpha_{\bar{p}(s)}$ fraction of the total return flow it receives. Moreover, (3.28) ensures that the incoming flow is equal to the outgoing flow at each DC. Clearly, the total outgoing flow at the DCs is greater than the total flow that the RPP location (s) can provide. Thus, each DC may fulfill its additional requirement for product $\bar{p} \in \mathcal{P}$, if any, from a NPP location $t \in \mathcal{T}_{\bar{p}}$, where there is ample supply. That is, we have

$$\sum_{s \in \mathcal{S}_{\bar{p}}} \sum_{r \in \mathcal{R}} \alpha_{\bar{p}s}\, \delta_{\bar{p}r}\, D_{\bar{p}r}\, \widehat{v}_{\bar{p}s} \leq \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}} \left(D_{\bar{p}r} + D'_{\bar{p}r}\right) \widehat{w}_{dr},$$

which verifies the feasibility of the single product forward subproblem, $FSP_{\bar{p}}(\mathbf{y}, \mathbf{z}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$, and the boundedness of its dual problem, $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$, given by (3.30)-(3.33).

We note that $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$ also has a special structure as it resembles the dual of the well-known transportation problem. Specifically, $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$ may have multiple alternative optimal solutions due to inherent degeneracy, and each of these alternative solutions characterizes a cut that can be used in the iterative BD framework. However, to increase the efficiency of this approach, it is worthwhile to identify the dominating solutions for $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$ which, in turn, helps us to work with corresponding strong (i.e., dominating) cuts. The benefits of strong Benders cuts have been investigated and established in other problem settings that arise in the context of capacitated facility location and network design Magnanti and Wong (1981); Van Roy (1986); Wentges (1996).

Magnanti and Wong (1981) define the strongness (or dominance) of a cut (constraint) in a general optimization problem (e.g., in our master problem in which we incorporate the cuts in the BD framework) given by $\min_{y \in Y, z \in \Re}\{z : f(u) + y\, g(u) \leq z \quad \forall\, u \in U\}$ as follows: The cut $f(u_1) + y\, g(u_1) \leq z$ dominates (is stronger than) the cut $f(u_2) + y\, g(u_2) \leq z$, if $f(u_1) + y\, g(u_1) \geq f(u_2) + y\, g(u_2) \quad \forall\, y \in Y$ with a strict inequality for at least one $y \in Y$. Then, clearly, due to the existence of multiple solutions as discussed above, it is important that we determine the solution to the $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$ in such a way that we can incorporate strong cuts in our BD framework. In order to obtain these strong Benders cuts, we propose a two-phase solution approach to $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$. In the first phase, we concentrate on determining the optimal values of the dual variables ($\beta_{\bar{p}s}$ and $\gamma_{\bar{p}d}$) associated with $\widehat{v}_{\bar{p}s}$ and $\widehat{w}_{dr}$ variables whose values are equal to 1. Notice that, for the rest of the dual variables, the associated $\widehat{v}_{\bar{p}s}$ and $\widehat{w}_{dr}$ variable values are 0, and, hence, these can assume any value without affecting the objective function value (3.30). As a result, in order to obtain strong cuts, in the second phase, we solve a maximization problem that produces dual variable values generating the strong cuts with respect to the above definition of strong cuts. We proceed with a detailed discussion of the two-phase approach.

First Phase for Solving $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$

In the first phase, we only consider the variables $\beta_{\bar{p}s}$, $s \in \mathcal{S}_{\bar{p}}$, and $\gamma_{\bar{p}d}$, $d \in \mathcal{D}$, whose associated $\widehat{v}_{\bar{p}s}$ and $\widehat{w}_{dr}$ (for at least one $r \in \mathcal{R}$) values, respectively, are equal to 1 in (3.30)-(3.33), i.e., according to $\widehat{\mathbf{v}}$ and $\widehat{\mathbf{w}}$. Recall that subscript (s) denotes the RPP location assigned to product $\bar{p}$ according to $\widehat{\mathbf{v}}$, i.e., $\widehat{v}_{\bar{p}(s)} = 1$. Also, let $\mathcal{D}^u \subseteq \mathcal{D}$ denote the set of DCs that are utilized according to $\widehat{\mathbf{w}}$, i.e., $\mathcal{D}^u = \{d \in \mathcal{D}|\ \widehat{w}_{dr} = 1,\ \exists\, r \in \mathcal{R}\}$. Finally, let $\mathcal{R}^{(d)} \subseteq \mathcal{R}$ denote the set of retailers assigned to DC location $(d) \in \mathcal{D}^u$ according to $\widehat{\mathbf{w}}$, i.e., $\mathcal{R}^{(d)} = \{r \in \mathcal{R}|\widehat{w}_{(d)r} = 1\}$. Noting that constraint set (3.32) can be rewritten as in (3.42), we have the following first phase problem:

$$\max \quad \sum_{r\in\mathcal{R}} \alpha_{\bar{p}(s)}\, \delta_{\bar{p}r}\, D_{\bar{p}r}\, \beta_{\bar{p}(s)} + \sum_{(d)\in\mathcal{D}^u} \sum_{r\in\mathcal{R}^{(d)}} \left(D_{\bar{p}r} + D'_{\bar{p}r}\right) \gamma_{\bar{p}(d)}$$

subject to

$$\beta_{\bar{p}(s)} + \gamma_{\bar{p}(d)} \leq G_{\bar{p}(s)(d)} \qquad \forall\, (d) \in \mathcal{D}^u, \tag{3.41}$$

$$\gamma_{\bar{p}(d)} \leq \min_{t\in\mathcal{T}_{\bar{p}}}\{G_{\bar{p}t(d)}\} \qquad \forall\, (d) \in \mathcal{D}^u, \tag{3.42}$$

$$\beta_{\bar{p}(s)},\ \gamma_{\bar{p}(d)} \ \text{ unrestricted} \qquad \forall\, (d) \in \mathcal{D}^u.$$

Examining the structure of the above formulation, we have the following important observations which lead to an efficient solution approach for the first phase problem:

1. Constraint set (3.41) implies that each unit of increase in the value of $\beta_{\bar{p}(s)}$ over and above $G_{\bar{p}(s)(d)}$ for $(d) \in \mathcal{D}^u$ requires a unit decrease in the values of the corresponding $\gamma_{\bar{p}(d)}$ for $(d) \in \mathcal{D}^u$. Thus, an excessively high $\beta_{\bar{p}(s)}$ value would lead to decreases in the $\gamma_{\bar{p}(d)}$ values so that none of the constraints in (3.42) are binding. Such increments in the value of $\beta_{\bar{p}(s)}$, that decrease $\gamma_{\bar{p}(d)}$ values, only result in decreases in the objective function value since for each such unit

change in the value of the variables the objective function value changes by

$$\sum_{r \in \mathcal{R}} \alpha_{\bar{p}(s)}\, \delta_{\bar{p}r}\, D_{\bar{p}r} - \sum_{(d) \in \mathcal{D}^u} \sum_{r \in \mathcal{R}^{(d)}} \left(D_{\bar{p}r} + D'_{\bar{p}r} \right),$$

which is a negative value (note that $\bigcup_{d \in \mathcal{D}} \mathcal{R}^{(d)} = \mathcal{R}$ and $\bigcap_{d \in \mathcal{D}} \mathcal{R}^{(d)} = \emptyset$). Thus, the value of $\beta_{\bar{p}(s)}$ is bounded from above such that at least one of the constraints in set (3.42) is binding.

2. Likewise, constraint set (3.41) implies that each unit decrease in the value of $\beta_{\bar{p}(s)}$ necessitates a unit increase in the values of $\gamma_{\bar{p}(d)}$ for $(d) \in \mathcal{D}^u$, which are bounded by (3.42). Thus, excessive decreases in the value of $\beta_{\bar{p}(s)}$ only result in decreases in the objective function value. Thus, the value of $\beta_{\bar{p}(s)}$ is bounded from below such that all of the constraints in set (3.42) are binding.

These observations suggest an efficient procedure to solve the first phase problem optimally. For product $\bar{p}$ with RPP location (s), let $\beta^{\bar{d}}_{\bar{p}(s)} = \left\{ G_{\bar{p}(s)\bar{d}} - \min_{t \in \mathcal{T}_{\bar{p}}} \{ G_{\bar{p}t\bar{d}} \} \right\}$ for $\bar{d} \in \mathcal{D}^u$. Similarly, for a given product $\bar{p}$ and a fixed DC location $\bar{d}$, let $\gamma_{\bar{p}(d)} = \min \left\{ (G_{\bar{p}(s)(d)} - \beta^{\bar{d}}_{\bar{p}(s)}), \min_{t \in \mathcal{T}_{\bar{p}}} \{ G_{\bar{p}t(d)} \} \right\}$ for $(d) \in \mathcal{D}^u$. The following procedure provides the optimal solution to the first phase problem:

Step 1. For each $\bar{d} \in \mathcal{D}^u$, determine $\beta^{\bar{d}}_{\bar{p}(s)}$. Sort $\beta^{\bar{d}}_{\bar{p}(s)}$ values in non-decreasing order.

Step 2. Starting with the first element on the list, i.e., the smallest $\beta^{\bar{d}}_{\bar{p}(s)}$ value, determine the value of $\gamma_{\bar{p}(d)}$ for all $(d) \in \mathcal{D}^u$ and evaluate the corresponding objective function value and delete $\beta^{\bar{d}}_{\bar{p}(s)}$ from the list. If a decrease in the objective function value occurs, go to Step 3. Otherwise, repeat Step 2.

Step 3. Report the solution with the maximum objective function value, which is the optimal solution to the first phase problem.

Second Phase for Solving $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$

In the second phase, we compute the values of the remaining variables in (3.30)-(3.33) so as to obtain strong Benders cuts for $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$. That is, we now consider the variables $\beta_{\bar{p}s}$, $s \in \mathcal{S}_{\bar{p}}$, and $\gamma_{\bar{p}d}$, $d \in \mathcal{D}$, whose associated $\widehat{v}_{\bar{p}s}$ and $\widehat{w}_{dr}$ (for at least one $r \in \mathcal{R}$) values, respectively, are equal to 0 according to $\widehat{\mathbf{v}}$ and $\widehat{\mathbf{w}}$. To this end, we first note that if $\widehat{v}_{\bar{p}s} = 0$, $s \in \mathcal{S}_{\bar{p}}$, in (3.30)-(3.33) then the corresponding $\beta_{\bar{p}s}$ values do not impact the objective value of $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$. Likewise, if $\widehat{w}_{dr} = 0$, $d \in \mathcal{D}^u$ and $r \in \mathcal{R}$, in (3.30)-(3.33), then the corresponding $\gamma_{\bar{p}d}$ values do not impact the objective value of $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$. In order to obtain the values of the remaining variables in (3.30)-(3.33) that provide strong Benders cuts, we consider the following linear programming problem in the second phase:

$$\begin{aligned} \max \quad & \sum_{s \in \mathcal{S}_{\bar{p}}} \sum_{r \in \mathcal{R}} \alpha_{\bar{p}s}\, \delta_{\bar{p}r}\, D_{\bar{p}r}\, \beta_{\bar{p}s} + \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}} \left(D_{\bar{p}r} + D'_{\bar{p}r} \right) \gamma_{\bar{p}d} \qquad (3.43) \\ \text{subject to} \quad & (3.31) - (3.33) \end{aligned}$$

Here, to ensure feasibility, we have to take into account the values of $\beta_{\bar{p}(s)}$ and $\gamma_{\bar{p}(d)}$ that are fixed in the first phase. After substituting these values, we solve problem (3.43) to find the values of the $\beta_{\bar{p}s}$ and $\gamma_{\bar{p}d}$ variables, i.e., remaining variables in (3.30)-(3.33). We note that the cardinality of these two sets of remaining variables are $|\mathcal{S}_{\bar{p}}| - 1$ and $|\mathcal{D} \setminus \mathcal{D}^u|$, respectively. Thus, the size of the above linear program is considerably smaller than that of $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$.

III.3.2.2. Solving the Single Product Reverse Subproblems

For a given product $\bar{p} \in \mathcal{P}$, now consider the single product reverse subproblem $RSP_{\bar{p}}(\mathbf{x}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$ given by (3.34)-(3.37). Similar to the single product forward subproblem analyzed in Section III.3.2.1, this subproblem also resembles a transportation problem with a special structure. More specifically, the set of source nodes is given by the CC locations $k \in \mathcal{K}$ for which at least one retailer $r \in \mathcal{R}$ is assigned according to $\widehat{\mathbf{u}}$, i.e., $k \in \mathcal{K}$ such that $\widehat{u}_{rk} = 1$ for at least one $r \in \mathcal{R}$. There is a single sink node $s \in \mathcal{S}_{\bar{p}}$ which is the RPP location for product $\bar{p}$ according to $\widehat{\mathbf{v}}$, i.e., $s \in \mathcal{S}_{\bar{p}}$ such that $\widehat{v}_{\bar{p}s} = 1$. The problem $RSP_{\bar{p}}(\mathbf{x}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$ is clearly feasible. Thus, its dual $DRSP_{\bar{p}}(\boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$, given by (3.38)-(3.40), is bounded. In order to obtain strong cuts, we again solve $DRSP_{\bar{p}}(\boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$ in two phases.

First Phase for Solving $DRSP_{\bar{p}}(\boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$

In the first phase, we only consider the variables $\lambda_{\bar{p}s}$, $s \in \mathcal{S}_{\bar{p}}$, and $\mu_{\bar{p}k}$, $k \in \mathcal{K}$, whose associated $\widehat{v}_{\bar{p}s}$ values and $\widehat{u}_{rk}$ values for at least one $r \in \mathcal{R}$, respectively, are equal to 1 in (3.38)-(3.40), i.e., according to $\widehat{\mathbf{v}}$ and $\widehat{\mathbf{u}}$. As before, let (s) denote the RPP location $s \in \mathcal{S}_{\bar{p}}$ to which product $\bar{p}$ is assigned according to $\widehat{\mathbf{v}}$, i.e., $\widehat{v}_{\bar{p}(s)} = 1$. Also, let $\mathcal{K}^u \subseteq \mathcal{K}$ denote the set of CC locations that are open according to $\widehat{\mathbf{u}}$, i.e., $\mathcal{K}^u = \{k \in \mathcal{K} |\, \exists\, r \in \mathcal{R} :\ \widehat{u}_{rk} = 1\}$. Finally, let $\mathcal{R}^{(k)}$ for $\mathcal{R}^{(k)} \subseteq \mathcal{R}$ denote the set of retailers assigned to CC location $(k) \in \mathcal{K}^u$ according to $\widehat{\mathbf{u}}$, i.e., $\mathcal{R}^{(k)} = \{r \in \mathcal{R} | \widehat{u}_{r(k)} = 1\}$. Then, we have the following first phase problem:

$$\max \quad \sum_{r \in \mathcal{R}} \delta_{\bar{p}r}\, D_{\bar{p}r}\, \lambda_{\bar{p}(s)} - \sum_{(k) \in \mathcal{K}^u} \sum_{r \in \mathcal{R}^{(k)}} \delta_{\bar{p}r}\, D_{\bar{p}r}\, \mu_{\bar{p}(k)} \tag{3.44}$$

subject to

$$\lambda_{\bar{p}(s)} - \mu_{\bar{p}(k)} \leq G_{(k)(s)} \qquad \forall\, (k) \in \mathcal{K}^u, \tag{3.45}$$

$$\mu_{\bar{p}(k)},\, \lambda_{\bar{p}(s)} \geq 0 \qquad \forall\, (k) \in \mathcal{K}^u. \tag{3.46}$$

The solution of the above first phase problem is given by $\lambda_{\bar{p}(s)} = \max_{(k) \in \mathcal{K}^u} \{G_{(k)(s)}\}$ and $\mu_{\bar{p}(k)} = \lambda_{\bar{p}(s)} - G_{(k)(s)}$ for all $(k) \in \mathcal{K}^u$. In this solution, all of the $\mu_{\bar{p}(k)}$ variables are positive but one, which is zero. Thus, we observe that if the values of $\lambda_{\bar{p}(s)}$ and

$\mu_{\bar{p}(k)}$, for all $(k) \in \mathcal{K}^u$, are further increased by one unit, the value of the objective function as well as the left-hand side of the constraints in (3.45) do not change.

Second Phase for Solving $DRSP_{\bar{p}}(\boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$

In the second phase, we again focus on the variables $\lambda_{\bar{p}s}$, $s \in \mathcal{S}_{\bar{p}}$, and $\mu_{\bar{p}k}$, $k \in \mathcal{K}$, whose associated $\widehat{v}_{\bar{p}s}$ values and $\widehat{u}_{rk}$ values for at least one $r \in \mathcal{R}$, respectively, are equal to zero according to $\widehat{\mathbf{v}}$ and $\widehat{\mathbf{u}}$. These values provide the coefficients for strong Benders cuts. To this end, we consider the following linear programming problem in the second phase:

$$\begin{aligned} \max \quad & \sum_{s \in \mathcal{S}_{\bar{p}}} \sum_{r \in \mathcal{R}} \delta_{\bar{p}r}\, D_{\bar{p}r}\, \lambda_{\bar{p}s} - \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}} \delta_{\bar{p}r}\, D_{\bar{p}r}\, \mu_{\bar{p}k} \qquad (3.47) \\ \text{subject to} \quad & (3.39) - (3.40) \end{aligned}$$

Here, to ensure feasibility, we have to take into account the values of $\lambda_{\bar{p}(s)}$ and $\mu_{\bar{p}(k)}$ for all $(k) \in \mathcal{K}^u$ that are fixed in the first phase. After substituting these values, we solve the problem (3.47) to find the values of the $\lambda_{\bar{p}s}$ and $\mu_{\bar{p}k}$ variables, i.e., remaining variables in (3.38)-(3.40). We note that the cardinality of these two sets of remaining variables are $|\mathcal{S}_{\bar{p}}| - 1$ and $|\mathcal{K} \setminus \mathcal{K}^u|$, respectively.

III.3.2.3. Solving the Benders Subproblem

Using the solutions of the single product forward subproblems, $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$ for $\bar{p} \in \mathcal{P}$, we set $\beta_{ps} = \beta_{\bar{p}s}$ for all $p \in \mathcal{P}$ and $s \in \mathcal{S}_p$ and $\gamma_{pd} = \gamma_{\bar{p}d}$ for all $p \in \mathcal{P}$ and $d \in \mathcal{D}$. Similarly, using the solutions of the single product reverse subproblems, $DRSP_{\bar{p}}(\boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$ for $\bar{p} \in \mathcal{P}$, we set $\mu_{pk} = \mu_{\bar{p}k}$ for all $p \in \mathcal{P}$ and $k \in \mathcal{K}$ and $\lambda_{ps} = \lambda_{\bar{p}s}$ for all $p \in \mathcal{P}$ and $s \in \mathcal{S}_p$. Hence, we obtain the solution of $DSP(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$ by combining the solutions of $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$ and $DRSP_{\bar{p}}(\boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$.

III.3.3. Benders Master Problem

The master problem in the BD framework is directly based on the formulation given in (3.14). As mentioned before, in a typical BD framework, the master problem includes the integer variables of the original model in addition to an auxiliary continuous variable introduced to incorporate Benders cuts via the solution of the dual subproblem. However, our above discussion on alternative separation schemes, i.e., flow and product separation, for the overall subproblem provides us with alternative representations of Benders cuts in the master problem. In order to explore these alternative Benders cuts, we first state the master problem $MP(\mathbf{v}, \mathbf{u}, \mathbf{w} | \widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}}, \widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}})$ using general cut related terms as follows.

$$\begin{aligned} \min \quad Z_{MP} \quad = & \sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}_p} F_{ps} v_{ps} + \sum_{k \in \mathcal{K}} F'_k c_k + \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} \sum_{k \in \mathcal{K}} G_{prk} \, \delta_{pr} \, D_{pr} \, u_{rk} \\ + & \sum_{p \in \mathcal{P}} \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}} G_{pdr} (D_{pr} + D'_{pr}) w_{dr} + \text{SumLHS(BCuts)} \end{aligned} \tag{3.48}$$

subject to

$$\sum_{k \in \mathcal{K}} u_{rk} = 1 \qquad \forall \, r \in \mathcal{R}, \tag{3.49}$$

$$\sum_{s \in \mathcal{S}_p} v_{ps} = 1 \qquad \forall \, p \in \mathcal{P}, \tag{3.50}$$

$$\sum_{d \in \mathcal{D}} w_{dr} = 1 \qquad \forall \, r \in \mathcal{R}, \tag{3.51}$$

$$u_{rk} \leq c_k \qquad \forall \, r \in \mathcal{R}, \, k \in \mathcal{K}, \tag{3.52}$$

(Constraints for the Set of BCuts) (3.53)

$$v_{ps}, c_k, w_{dr}, u_{rk} \in \{0, 1\} \qquad \forall \, p \in \mathcal{P}, \, r \in \mathcal{R}, \, k \in \mathcal{K}, \, d \in \mathcal{D}, \, s \in \mathcal{S}_p. \tag{3.54}$$

***Alternative Benders Cuts* (BCuts)**

Since the Benders subproblem is separable according to the direction of flow leading to single product forward and reverse subproblems, it is possible to generate different sets of cuts employing the solutions obtained for these individual subproblems. In our work we consider four different types of Benders cuts. The first set of cuts are obtained using *flow and product* separation. We use *only product* separation for the second set of cuts. The third set of cuts are obtained using *only flow* separation. For the fourth set of cuts, we do not employ any separation scheme. We discuss in detail below how we obtain the cuts, i.e., the Set of BCuts in (3.53), and the term SumLHS(BCuts) that is included in the objective function of the Benders master problem. We note that the use of separation schemes is similar to the use of multiple cuts in the multicut L-shaped method developed in the context of two-stage stochastic linear programs (Birge and Louveaux, 1988).

Type 1: For each product type $p \in \mathcal{P}$, we derive a forward and a reverse cut using the solutions of $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$ and $DRSP_{\bar{p}}(\boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$, respectively. To this end, we define two new decision variables, $\psi_p^F \geq 0$ and $\psi_p^F \geq 0$, for each $p \in \mathcal{P}$. Then, the constraints that correspond to the $|\mathcal{P}|$ *single product forward cuts* and $|\mathcal{P}|$ *single product reverse cuts* are given by

$$\begin{aligned} \psi_p^F &\geq \sum_{s\in\mathcal{S}_p}\sum_{r\in\mathcal{R}} \alpha_{ps}\delta_{pr}D_{pr}\widehat{\beta}_{ps}v_{ps} + \sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} \left(D_{pr} + D'_{pr}\right)\widehat{\gamma}_{pd}w_{dr} \quad \forall\, p \in \mathcal{P}, \quad \text{and} \\ \psi_p^R &\geq \sum_{s\in\mathcal{S}_p}\sum_{r\in\mathcal{R}} \delta_{pr}D_{pr}\widehat{\lambda}_{ps}v_{ps} - \sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}} \delta_{pr}D_{pr}\widehat{\mu}_{pk}u_{rk} \quad \forall\, p \in \mathcal{P}, \end{aligned}$$

respectively. In order to formulate the master problem that is based on this type of Benders cuts, the above two sets of constraints and $\psi_p^F,\ \psi_p^R \geq 0$, $\forall\, p \in \mathcal{P}$ are considered in constraint set (3.53) of $MP(\mathbf{v}, \mathbf{u}, \mathbf{w}|\widehat{\boldsymbol{\beta}},\ \widehat{\boldsymbol{\gamma}},\ \widehat{\boldsymbol{\lambda}},\ \widehat{\boldsymbol{\mu}})$ given by (3.48)-(3.54). Also, the SumLHS(BCuts) term in the objective function (3.48) of

$MP(\mathbf{v}, \mathbf{u}, \mathbf{w}|\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}}, \widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}})$ is replaced with $\sum_{p\in\mathcal{P}} \psi_p^F + \sum_{p\in\mathcal{P}} \psi_p^R$.

Type 2: Using the solutions of $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$ for $\bar{p} \in \mathcal{P}$ $DRSP_{\bar{p}}(\boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$ for $\bar{p} \in \mathcal{P}$, we derive a single cut for each product $p \in \mathcal{P}$. To this end, we define a new decision variable $\psi_p \geq 0$ for each $p \in \mathcal{P}$. Then, the constraints that correspond to the $|\mathcal{P}|$ *single product cuts* are given by

$$\begin{aligned} \psi_p \ \geq\ & \sum_{s\in\mathcal{S}_p}\sum_{r\in\mathcal{R}} \delta_{pr} D_{pr} \left(\alpha_{ps}\widehat{\beta}_{ps} + \widehat{\lambda}_{ps}\right) v_{ps} + \sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} \left(D_{pr} + D'_{pr}\right) \widehat{\gamma}_{pd} w_{dr} \\ - & \sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}} \delta_{pr} D_{pr} \widehat{\mu}_{pk} u_{rk} \quad \forall\, p \in \mathcal{P}. \end{aligned}$$

In order to formulate the master problem that is based on Type 2 Benders cuts, the above set of constraints and $\psi_p \geq 0, \forall\, p \in \mathcal{P}$ are considered in constraint set (3.53). Finally, the SumLHS(BCuts) term in the objective function (3.48) is replaced with $\sum_{p\in\mathcal{P}} \psi_p$.

Type 3: Using the solutions of $DFSP_{\bar{p}}(\boldsymbol{\beta}, \boldsymbol{\gamma}|\widehat{\mathbf{v}}, \widehat{\mathbf{w}})$ for $\bar{p} \in \mathcal{P}$, we derive a *forward flow* cut. Likewise, using the solutions of $DRSP_{\bar{p}}(\boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}})$ for $\bar{p} \in \mathcal{P}$, we derive a *reverse flow* cut. Defining two new decision variables, $\psi^F \geq 0$ and $\psi^R \geq 0$, the constraints that correspond to the *forward* and *reverse flow cuts* can be expressed as

$$\begin{aligned} \psi^F \ &\geq\ \sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p}\sum_{r\in\mathcal{R}} \alpha_{ps}\delta_{pr} D_{pr} \widehat{\beta}_{ps} v_{ps} + \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} \left(D_{pr} + D'_{pr}\right) \widehat{\gamma}_{pd} w_{dr}, \quad \text{and} \\ \psi^R \ &\geq\ \sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p}\sum_{r\in\mathcal{R}} \delta_{pr} D_{pr} \widehat{\lambda}_{ps} v_{ps} - \sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}} \delta_{pr} D_{pr} \widehat{\mu}_{pk} u_{rk}, \end{aligned}$$

respectively. Consequently, in the master problem that is based on Type 3 Benders cuts, constraint set (3.53) consists of the above two constraints along with $\psi^F,\ \psi^R \geq 0$ whereas the SumLHS(BCuts) term in the objective function (3.48) is given by $\psi^F + \psi^R$.

Type 4: We derive a single cut using the solution of $DSP(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$. We note that this approach corresponds to the typical use of the BD framework where there is only one cut added at each iteration. In order to include the cut in the master problem, we define a new decision variable, $\psi \geq 0$. The constraint that corresponds to the cut is then given by

$$\begin{aligned} \psi \geq & \sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p}\sum_{r\in\mathcal{R}} \delta_{pr} D_{pr} \left(\alpha_{ps}\widehat{\beta}_{ps} + \widehat{\lambda}_{ps}\right) v_{ps} + \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} \left(D_{pr} + D'_{pr}\right) \widehat{\gamma}_{pd} w_{dr} \\ - & \sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}} \delta_{pr} D_{pr} \widehat{\mu}_{pk} u_{rk}. \end{aligned}$$

In the master problem that is based on this cut type, we consider the above constraint and $\psi \geq 0$ in constraint set (3.53) and substitute ψ for the SumLHS(BCuts) term in the objective function (3.48).

In Display 1, we present the general procedure we use to solve the CLSC network design problem of interest using particular types of cuts. We note that, in Display 1, $\varepsilon > 0$ is the tolerance for stopping criterion; $IterNo$ is the iteration counter; and `MaxIter` is the maximum number of Benders iterations considered. Also, UB and LB denote the current upper and lower bounds on the objective function value, respectively. Recall that Z_{DSP} and Z_{MP} correspond to the objective function values of the dual of the Benders subproblem and the Benders master problem as defined in (3.21) and (3.48), respectively. Finally, we note that, at each iteration of the procedure, the use of Type 1, 2, 3, and 4 cuts adds $2|\mathcal{P}|$ cuts, $|\mathcal{P}|$ cuts, two cuts, and a single cut, respectively.

Display 1 Pseudo-code of the BD Approach.

Set $UB = \infty$, $IterNo = 0$, and $\widehat{\boldsymbol{\beta}}=\widehat{\boldsymbol{\gamma}}=\widehat{\boldsymbol{\mu}}=\widehat{\boldsymbol{\lambda}}=0$. Initialize `MaxIter`.
Solve the master problem $MP(\mathbf{v}, \mathbf{u}, \mathbf{w}|\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}}, \widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}})$ to obtain the values for $\widehat{\mathbf{v}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}}$, and Z_{MP}. Set $LB = Z_{MP}$.
while $(((UB - LB)/LB \geq \varepsilon)$ and $(IterNo<$`MaxIter`$))$ **do**
  $IterNo=IterNo + 1$
  Solve the $DSP(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\mu}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$ to obtain the values for $\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}}, \widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}}$, and Z_{DSP}.
  if $((Z_{MP} + Z_{DSP}$ - SumLHS(BCuts)$) < UB)$ **then**
    $UB = (Z_{MP} + Z_{DSP}$ - SumLHS(BCuts)$)$
  end if
  Add the (Set of BCuts) to the master problem using the $\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}}, \widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}}$ values.
  Solve the master problem $MP(\mathbf{v}, \mathbf{u}, \mathbf{w}|\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}}, \widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}})$ to obtain the values for $\widehat{\mathbf{v}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}}$, and Z_{MP}. Set $LB = Z_{MP}$.
end while
Find the values for $\mathbf{x}, \mathbf{y}, \mathbf{z}$ corresponding to $\widehat{\mathbf{v}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}}$ (i.e., solve $SP(\mathbf{x}, \mathbf{y}, \mathbf{z}|\widehat{\mathbf{v}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$).
Report $\mathbf{v}, \mathbf{u}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}$ and the objective function value for (3.1).

III.4. Computational Experiments

In order to examine the computational performance of the proposed BD approach using alternative types of cuts and benchmark it against the B&C approach, we carry out a computational study. In order to solve the test instances and the master problem using the B&C approach, we employ CPLEX 9.0 and Concert Technology [1] with default settings for cut generation, preprocessing, and upper bound heuristics. The cuts that are utilized by CPLEX include clique, cover, disjunctive, flow cover, flow path, Gomory fractional, generalized upper bound cover, implied bound, and mixed integer rounding cuts. We implement our solution approaches using the C++ programming language and perform the runs on a machine with a 3 GHz Intel XEON

[1]CPLEX and Concert Technology are trademark of ILOG, Inc.

processor and 6 GB RAM.

III.4.1. Random Test Instance Generation

In order to develop a set of test instances that are of realistic size, we vary the number of products $|\mathcal{P}|$, the number of retailers $|\mathcal{R}|$, and the number of potential CC locations $|\mathcal{K}|$. In particular, we consider two levels for $|\mathcal{P}|$ (5 and 10), three levels for $|\mathcal{R}|$ (60, 90, and 120), and two levels for $|\mathcal{K}|$ (25 and 35). Hence, we have $2 \times 3 \times 2 = 12$ different problem classes as shown in Table 1. We generate 10 random instances for each of these problem classes. For each instance, we generate the number of NPP locations $|\mathcal{T}_p|$ and the number of potential RPP locations $|\mathcal{S}_p|$ for all of the products using uniform distributions as shown in Table 2. In generating the number of DC locations $|\mathcal{D}|$, we induce correlation between $|\mathcal{R}|$ and $|\mathcal{D}|$ as in Table 2. For each retailer, we randomly generate the new demand D_{pr}, the induced demand D'_{pr}, and the return fraction δ_{pr} for each product p using uniform distributions as shown in Table 2. We randomly generate the recovery fraction α_{ps} at each potential RPP location for each product also as in Table 2.

Table 1 **URP**: Problem Classes Used in Computational Testing.

Class	$\|\mathcal{P}\|$	$\|\mathcal{R}\|$	$\|\mathcal{K}\|$	Class	$\|\mathcal{P}\|$	$\|\mathcal{R}\|$	$\|\mathcal{K}\|$
C1	5	60	25	C7	10	60	25
C2	5	60	35	C8	10	60	35
C3	5	90	25	C9	10	90	25
C4	5	90	35	C10	10	90	35
C5	5	120	25	C11	10	120	25
C6	5	120	35	C12	10	120	35

The **URP** has three major cost categories that represent processing, transportation, and facility location costs. Associated with each major cost category, we have several cost components. More specifically, we have four cost components associated

Table 2 **URP**: Distributions for Sets, Demand, Return Fraction, and Recovery Fraction Values.

Parameter	Value
$\lvert\mathcal{T}_p\rvert$	Uniform[1, 5]
$\lvert\mathcal{S}_p\rvert$	Uniform[2, 10]
$\lvert\mathcal{D}\rvert$	$\lfloor \lvert\mathcal{R}\rvert/12 \rfloor$
D_{pr}	Uniform[250, 300]
D'_{pr}	Uniform[450, 500]
δ_{pr}	Uniform[0.7, 0.9]
α_{ps}	Uniform[0.8, 0.98]

with processing (corresponding to manufacturing, remanufacturing, DC processing, and CC processing activities), five cost components associated with transportation (corresponding to NPP-DC, RPP-DC, CC-RPP, DC-Retailer, and Retailer-CC flows on the network), and two cost components associated with facility location (corresponding to collection center and remanufacturing facility location decisions), resulting in a total of eleven cost components. In our preliminary computational experiments, we observed that the relative contribution of each major cost category to the overall cost in the optimal solution has an impact on the difficulty of the test instance. Therefore, we distinguish between two settings where (i) each cost category contributes significantly to the overall cost, i.e., *balanced* case, and (ii) a cost category dominates over the others in the overall cost, i.e., *unbalanced* case. It is worthwhile to note that in the balanced case the trade-offs involved in realistic decision-making situations are better reflected in the data, and, in fact, this case corresponds to more difficult test instances as will be clear while we present our computational results in the following discussion.

For the purpose of generating test instances corresponding to balanced and unbalanced cases, we follow a specific procedure that is guided by the *estimated* percentage contribution (EPC) of the above mentioned eleven cost components in the optimal overall objective function value. More specifically, we utilize the EPC of the cost components to induce correlation between different cost components so that we exercise some control over the percentage contribution of each cost category in the optimal overall objective function value for each randomly generated test instance. We begin by generating the unit manufacturing cost for each product randomly and obtain an estimate of the total manufacturing cost for each product. Then, we use the total manufacturing cost for each product in conjunction with the EPC of the cost components to randomly generate other processing cost parameters for each product, transportation cost parameters for each transportation link on the network, and fixed location cost parameters for potential RPP and CC locations. The complete test bed includes a total of 120 instances with balanced costs and a total of 60 instances with unbalanced costs (see Table 3 for details).

Our main objective in considering the balanced and unbalanced cases is to examine the performance of both the B&C and BD approaches under these scenarios for the same size problems, and thus, to highlight the impact of the input data on algorithmic performance. For each problem instance we first obtain the optimal objective function value using the B&C approach and evaluate the percentage contribution of the individual cost components. Table 3 reports the *average realized* percentage contribution (ARPC) of the cost components where we use the following notation. We note that each cell for the balanced case is the average of 120 observations, each cell for each unbalanced case is the average of 20 observations.

$\widehat{\varphi}_\nu$ ARPC of manufacturing cost of the products,

$\widehat{\varphi}_\rho$ ARPC of remanufacturing cost of the products,

$\widehat{\varphi}_\eta$ ARPC of DC processing cost of the products,

$\widehat{\varphi}_\kappa$ ARPC of CC processing cost of the products,

$\widehat{\varphi}_F$ ARPC of fixed cost of opening the RPP locations,

$\widehat{\varphi}_{F'}$ ARPC of fixed cost of opening the collection centers,

$\widehat{\varphi}_{TD}$ ARPC of transportation cost associated with the NPP-DC flows,

$\widehat{\varphi}_{SD}$ ARPC of transportation cost associated with the RPP-DC flows,

$\widehat{\varphi}_{KS}$ ARPC of transportation cost associated with the CC-RPP flows,

$\widehat{\varphi}_{DR}$ ARPC of transportation cost associated with the DC-Retailer flows, and

$\widehat{\varphi}_{RK}$ ARPC of transportation cost associated with the Retailer-CC flows.

Table 3 **URP**: Average Realized Percentage Contribution of Cost Components in the Test Instances.

Processing costs					Transportation costs						Location costs		
$\widehat{\varphi}_\nu$	$\widehat{\varphi}_\rho$	$\widehat{\varphi}_\eta$	$\widehat{\varphi}_\kappa$	**Total**	$\widehat{\varphi}_{TD}$	$\widehat{\varphi}_{SD}$	$\widehat{\varphi}_{KS}$	$\widehat{\varphi}_{DR}$	$\widehat{\varphi}_{RK}$	**Total**	$\widehat{\varphi}_F$	$\widehat{\varphi}_{F'}$	**Total**
Balanced Case (based on 120 instances–10 in each of the 12 problem classes in Table 1.)													
18	4	2	1	**25**	4	2	1	19	8	**34**	23	18	**41**
Unbalanced Case 1: Processing cost dominant (based on 20 instances–10 in each of C11 and C12)													
39	14	8	6	**67**	2	2	1	11	8	**24**	5	4	**9**
Unbalanced Case 2: Transportation cost dominant (based on 20 instances–10 in each of C11 and C12)													
7	2	1	1	**11**	1	1	1	48	21	**72**	8	9	**17**
Unbalanced Case 3: Facility location cost dominant (based on 20 instances–10 in each of C11 and C12)													
6	2	2	1	**11**	1	1	1	2	2	**7**	60	22	**82**

III.4.2. Computational Results

As we have noted earlier, we solve each instance using the B&C approach and the BD approach with the alternative types of cuts developed in Section III.3. In obtaining the optimal solution to an instance using the B&C approach for the test instances, we observe that reducing the optimality gap below 1 percent requires considerable computational effort. Thus, to avoid this tail-off effect, we set the tolerance for stopping criterion to 1 percent gap between the incumbent and the best lower bound. As mentioned earlier, while solving the master problem in the BD approach, we use the B&C approach as implemented in CPLEX. In order to quickly generate initial Benders cuts, at the first iteration of BD, we employ an early stopping criterion for the master problem. For this purpose, we set the optimality gap to 30% at this iteration. In successive iterations, we set the stopping criterion to 0.009% optimality gap. This way of handling the master problem also helps us to circumvent the tail-off effect with only a small compromise in the quality of the bound values. We use the corresponding upper and lower bounds of the master problem (obtained upon termination of the B&C) to calculate the upper (Z_{MP} in line 7 of Display 1) and lower (Z_{MP} in lines 2 and 10 of Display 1) bounds in the BD approach. Furthermore, we limit the maximum number of iterations (`MaxIter`) in the BD approach to five and set the the tolerance for stopping criterion (ε) to 1 percent as in the B&C approach (line 3 of Display 1). In the following, we compare the B&C and BD approaches with alternative types of cuts in terms of the solution quality and the time required to obtain the solution as well as the number of iterations required to obtain the optimal solution.

III.4.3. Balanced Costs

Considering the 120 instances corresponding to the case of balanced costs, we summarize the average and the maximum optimality gaps ($100(UB - LB)/LB$) upon termination of the approaches in Table 4. We note that, in Tables 4, 5, and 6, row minimums for the BD related results are listed in bold. As expected, in the B&C approach, the maximum optimality gap for almost all instances are at most 1% except for one instance in problem class C12. In this instance, the run was terminated due to memory limitations with an optimality gap of 2.8%. The overall average optimality gap for the B&C approach is 1% (which is the stopping criterion) whereas the average gaps for the BD approach with Type 1, 2, 3, and 4 cuts are 0.6%, 0.8%, 0.8%, and 1.0%, respectively. Although we use the termination criteria of 1% optimality gap or 5 iterations for the BD approach, we observe that the optimality gap is still smaller than 1% in many instances. Notably, the use of Type 1 cuts in the BD approach appears to be the most effective, as the BD with Type 1 cuts provides the lowest average optimality gap in all of the problem classes and the lowest maximum optimality gap in most classes. This provides empirical evidence as to the potential benefit of using disaggregated cuts in the BD framework.

Table 4 **URP**: Comparison of the Optimality Gaps upon Termination for Balanced Instances.

Class	Average Optimality Gap (%)					Maximum Optimality Gap (%)				
	B & C	Type 1	Type 2	Type 3	Type 4	B & C	Type 1	Type 2	Type 3	Type 4
C1	1.0	**0.3**	0.7	0.7	0.8	1.0	**0.7**	1.0	1.0	1.0
C2	1.0	**0.4**	0.8	0.6	1.0	1.0	**0.8**	1.3	1.0	1.8
C3	0.9	**0.5**	0.6	0.7	0.7	1.0	**0.9**	1.0	1.0	0.9
C4	1.0	**0.6**	0.8	0.8	0.7	1.0	**1.0**	**1.0**	**1.0**	**1.0**
C5	1.0	**0.6**	0.7	0.8	0.8	1.0	1.0	**0.9**	1.0	1.0
C6	1.0	**0.7**	**0.7**	**0.7**	0.8	1.0	**1.0**	**1.0**	**1.0**	1.1
C7	1.0	**0.6**	0.9	0.7	1.0	1.0	1.0	1.2	**0.9**	1.7
C8	1.0	**0.7**	1.0	0.9	1.0	1.0	**1.0**	1.1	**1.0**	1.5
C9	1.0	**0.7**	0.8	**0.7**	0.9	1.0	**1.0**	**1.0**	**1.0**	**1.0**
C10	1.0	**0.8**	1.3	**0.8**	1.6	1.0	**1.0**	2.1	1.7	2.8
C11	1.0	**0.6**	0.7	0.9	1.1	1.0	**1.0**	**1.0**	1.1	1.3
C12	1.2	**0.7**	**0.7**	1.1	1.2	2.8	**1.0**	**1.0**	2.4	2.3

In Table 5, we present a comparison of the time required to obtain the solution by the B&C approach and the BD approach with alternative types of cuts. The results show that the BD approach with Type 1 or Type 2 cuts performs better in terms of solution times than the BD approach with Type 3 and Type 4 cuts. Except in C12, the BD approach with these types of cuts provides either the shortest, or the second shortest, solution time in all of the problem classes. We note that the time required to solve the *master problem* with Type 1 and Type 2 cuts is marginally longer than the time required with Type 3 and Type 4 cuts. This can clearly be attributed to the higher number of inequalities being added to the master problem at each iteration due to the disaggregation of cuts, i.e., the BD approach with Type 1 cuts adds $2|\mathcal{P}|$ inequalities whereas the approach with Type 2 cuts adds $|\mathcal{P}|$ inequalities as opposed to two inequalities and one inequality added in the BD with Type 3 and Type 4 cuts, respectively. For even larger size instances (especially ones with larger $|\mathcal{K}|$), the solution times with the BD approach are expected to increase mainly due to the solution time of the master problem. On the other hand, the results reported in Table 5 indicate that the solution times with the B&C approach will increase very dramatically for those instances.

Table 5 **URP**: Comparison of the Solution Times for Balanced Instances.

Class	Average of Solution Times (sec.)					Maximum of Solution Times (sec.)				
	B & C	Type 1	Type 2	Type 3	Type 4	B & C	Type 1	Type 2	Type 3	Type 4
C1	36.2	**13.6**	16.1	16.1	18.3	75.5	24.7	**23.9**	23.9	38.2
C2	164.2	47.7	**42.1**	82.0	58.5	642.2	**157.9**	160.3	293.0	190.5
C3	134.8	29.2	**26.1**	52.8	52.7	251.1	56.4	**41.4**	100.3	116.9
C4	447.2	**96.8**	207.0	207.0	291.2	828.1	**143.4**	458.0	458.0	925.0
C5	166.5	**29.8**	80.2	58.0	58.0	304.5	**50.3**	286.2	164.3	164.3
C6	1194.9	**132.7**	419.0	295.3	377.4	3224.3	**263.6**	1592.7	1131.0	1014.5
C7	162.4	18.3	**13.6**	34.0	24.5	404.0	39.8	**29.5**	59.9	38.6
C8	475.0	46.5	**38.1**	96.5	70.7	924.6	105.8	**74.4**	371.8	126.2
C9	1006.0	**51.9**	56.7	129.7	106.7	6223.8	**134.4**	194.0	251.3	274.2
C10	9629.2	480.1	**355.2**	1362.9	569.8	42763.8	1356.1	**738.2**	3063.1	1419.5
C11	8080.8	**181.7**	472.7	249.2	399.1	34073.7	**374.4**	1175.4	918.8	1404.8
C12	42614.4	1800.8	3810.6	**785.9**	1769.5	120226.0	7318.4	19066.6	**2059.3**	3359.5

Table 6 reports the average and the maximum number of iterations required by the BD approach with alternative types of cuts. We observe that Type 1 cuts provide the smallest values for the average and the maximum number of iterations.

Table 6 **URP**: Comparison of the Number of Iterations for Balanced Instances.

Class	Average Number of Iterations				Maximum Number of Iterations			
	Type 1	Type 2	Type 3	Type 4	Type 1	Type 2	Type 3	Type 4
C1	**1.7**	**1.7**	**1.7**	2.3	**2**	**2**	**2**	4
C2	**1.9**	2.4	2.2	2.4	**3**	5	4	5
C3	**1.1**	1.2	1.3	1.5	**2**	**2**	**2**	3
C4	**1.1**	1.5	1.5	2.2	**2**	**2**	**2**	4
C5	**1.0**	1.4	1.4	1.4	**1**	2	2	2
C6	**1.2**	1.9	1.5	2.6	**2**	4	3	5
C7	**1.6**	2.3	2.3	2.8	**3**	5	4	5
C8	**1.7**	2.6	2.2	3.3	**2**	5	4	5
C9	**1.4**	2.2	2.1	2.7	**2**	5	3	5
C10	**2.6**	4.3	3.8	4.7	**5**	**5**	**5**	**5**
C11	**1.6**	2.2	2.4	3.4	**2**	3	5	5
C12	**2.1**	2.6	3.1	3.9	**5**	**5**	**5**	**5**

We also note that a detailed analysis of the upper and lower bounds obtained by the BD approach upon termination reveals that the use of Type 1 cuts provides the best lower bound in 75 of the total 120 random test instances. This is followed by Type 3, Type 2 and Type 4 cuts with 29 instances, 18 instances, and 1 instance, respectively. The B&C approach provides the best lower bound in only one instance. Furthermore, although the upper bounds obtained by all of the approaches are very close, we observe that the B&C approach provides the lowest upper bound in 94 of the instances followed by 15, 5, 6, and 7 instances with the BD approach using Type 1, 2, 3, and 4 cuts, respectively.

III.4.4. Unbalanced Costs

Considering the 60 instances corresponding to the case of unbalanced costs, we present a summary of our experimental results in In Table 7. Interestingly, the experimental results show that the time required to obtain the optimal solution for all of the approaches decreases considerably when one of the cost components is dominant. This observation can be directly attributed to the performance of the upper and lower bounds generated both in the B&C and BD approaches since the bounds are determined primarily by the solution related only to the dominant cost category. We note that, as opposed to the balanced cost case, the results in Table 7 do not provide any evidence that the larger problem size instances cause increased solution times, independent of the solution approach. It appears that, once the cost data is unbalanced, the instance becomes very easy to solve with both the BD and the B&C approaches.

Table 7 URP: Comparison of the Solution Times for Unbalanced Instances.

Processing Cost Dominant (on average, 67% of the total cost)										
Class	Average of Solution Times (sec.)					Maximum of Solution Times (sec.)				
	B & C	Type 1	Type 2	Type 3	Type 4	B & C	Type 1	Type 2	Type 3	Type 4
C11	1.43	0.42	0.51	0.36	0.37	3.05	0.75	1.17	0.63	0.59
C12	1.11	0.53	0.53	0.43	0.43	1.55	0.59	0.56	0.48	0.45
Transportation Cost Dominant (on average, 72% of the total cost)										
Class	Average of Solution Times (sec.)					Maximum of Solution Times (sec.)				
	B & C	Type 1	Type 2	Type 3	Type 4	B & C	Type 1	Type 2	Type 3	Type 4
C11	1.44	0.48	0.49	0.42	0.43	1.66	0.50	0.52	0.44	0.44
C12	1.55	0.50	0.58	0.47	0.48	2.50	0.78	0.75	0.63	0.63
Facility Location Cost Dominant (on average, 82% of the total cost)										
Class	Average of Solution Times (sec.)					Maximum of Solution Times (sec.)				
	B & C	Type 1	Type 2	Type 3	Type 4	B & C	Type 1	Type 2	Type 3	Type 4
C11	1.25	0.48	0.49	0.41	0.43	1.34	0.50	0.52	0.42	0.44
C12	1.62	0.72	0.74	0.64	0.65	1.80	0.75	0.77	0.67	0.66

III.5. Concluding Remarks

In this chapter, we considered the **URP**, which is a multi-product CLSC network design problem where we locate the collection centers and remanufacturing facilities while determining the material flows in the whole network so as to minimize the processing, transportation, and fixed location costs. Although our work was primarily motivated by the practice of an OEM in the automotive service parts industry, we developed a generic model that can be used in establishing CLSC networks for other consumer, commercial, and industrial products/parts, where the products/parts have high recoverable value, long product life cycles, and well-established forward networks. We developed an efficient mathematical formulation that models the flow variables separately for each stage in the network. In our preliminary computational studies, we found that this type of formulation performs better with the B&C approach as opposed to a formulation that uses four-index multi-stage variables, similar to the multi-stage flow variables given in Geoffrion and Graves (1974), that specify product flow from the retailers to the RPPs and from NPPs and RPPs to the retailers. Furthermore, the formulation considered here also lends itself to an efficient Benders reformulation and solution approach. Therefore, on the modelling side, we presented an efficient mathematical formulation for a generic model for CLSC network design.

On the methodological side, we provided an exact solution approach based on BD, which performs faster than the B&C approach. In this context, we provided efficient dual problem solution methods that generate *strong* Benders cuts. Furthermore, we determined that, in our problem setting, the use of multiple Benders cuts, as opposed to the classical single Benders cut approach, generated stronger lower bounds and promoted faster convergence.

On the empirical side, we used the model and solution approach that we developed to understand the impact of problem data characteristics on solution performance. For this purpose, we generated a test-bed of problem instances with cost structures underlining the trade-off considerations. In this case, our tests clearly showed the efficiency of the BD approach with strong and, also with disaggregated, cuts. Interestingly, we also determined that if the input parameters are such that the different cost components are not balanced, but, rather, are biased towards one of the major cost categories, the time required to obtain the optimal solution decreased considerably when using the B&C approach and the proposed BD approach as compared to problem instances of the same size with balanced costs.

CHAPTER IV

A CAPACITATED REMANUFACTURING NETWORK DESIGN PROBLEM

In this chapter, we first extend the problem setting of **URP** to the **CRP** by incorporating capacity constraints and multi-sourcing requirements for retailer assignments. As a consequence of these two extensions, each retailer receives incoming product shipments from multiple DCs and sends product returns to multiple CCs, both of which operate under certain capacity restrictions. Moreover, in the forward channel, a DC may receive shipments from multiple capacitated product-specific NPPs, in addition to the open product-specific RPPs. We note that the capacities at the DCs and the CCs represent aggregate capacities that can be shared by all products. Thus, for the purpose of incorporating the non-uniformity in capacity usage, we utilize a coefficient specified separately for each product as a modifier to one capacity use unit. Moreover, since we can estimate the remanufacturing capacity for each product by using the estimated return quantity and return fraction as well as the recovery fraction, and we furthermore assume a single RPP per product, we can identify the feasible RPP candidate sites for each product (and consider these the only candidates) before attempting to solve any specific instance. Thus, we do not consider any capacity limitations on candidate RPP sites in this model.

In **CRP**, as in **URP**, we are interested in determining the best locations of the RPPs and the CCs (out of their respective set of candidate sites) with respect to the NPPs, DCs and the retailers at known locations, and the best flow of products in both the forward and reverse channels so that the total cost of location, processing (at NPPs, RPPs, CC, and DCs) and transportation is minimized.

IV.1. Problem Formulation

We next give the additional notation and the model, referred in this chapter henceforth as **MP**. Figure 3 depicts the underlying network structure with the flow, assignment, and location variables in the CLSC network.

Additional Parameters

C_d aggregate processing/storage capacity at DC $d \in \mathcal{D}$.

B_k aggregate processing/storage capacity at CC $k \in \mathcal{K}$.

Q_{pt} production/supply capacity to produce/supply $p \in \mathcal{P}$ at NPP $t \in \mathcal{T}_p$.

γ_p processing/storage capacity coefficient at the DCs for product $p \in \mathcal{P}$.

β_p processing/storage capacity coefficient at CCs for product $p \in \mathcal{P}$.

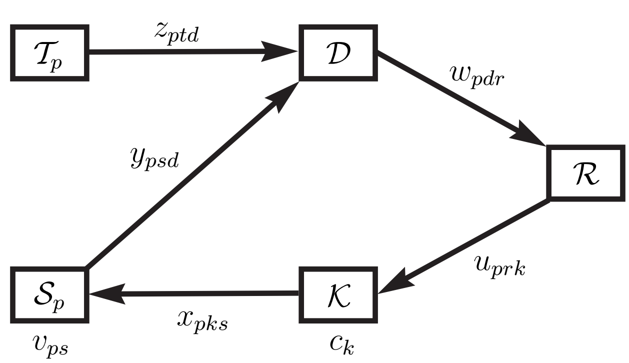
M_{prk} constraint parameters whose values are $\min\{\delta_{pr}D_{pr}, B_k\}$ for $p \in \mathcal{P}, r \in \mathcal{R}, k \in \mathcal{K}$.

Additional Decision Variables

u_{prk} quantity of product $p \in \mathcal{P}$ shipped from retailer $r \in \mathcal{R}$ to CC $k \in \mathcal{K}$.

w_{pdr} quantity of product $p \in \mathcal{P}$ shipped to retailer $r \in \mathcal{R}$ from DC $d \in \mathcal{D}$.

Figure 3 Underlying Structure of the CLSC Network for **CRP**.



Objective Function

$$
\begin{aligned}
\min \quad & \sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p} F_{ps}v_{ps} + \sum_{k\in\mathcal{K}} F'_k c_k + \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\sum_{k\in\mathcal{K}} \left(G_{rk} + \kappa_{pk}\right) u_{prk} \\
+ \; & \sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}_p} G_{ks}x_{pks} + \sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p}\sum_{d\in\mathcal{D}} \left(G_{sd} + \rho_{ps}\right) y_{psd} \\
+ \; & \sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}_p}\sum_{d\in\mathcal{D}} \left(G_{td} + \nu_{pt}\right) z_{ptd} + \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} \left(G_{dr} + \eta_{pd}\right) w_{pdr} \qquad (4.1)
\end{aligned}
$$

The first two terms in the objective function represent the fixed costs associated with locating the RPPs and the CCs, respectively. The next two terms give the reverse channel cost including the retailer-to-CC and the CC-to-RPP transportation costs, respectively, with the third term also including the processing costs at the CCs. The following fifth and the sixth terms represent the transportation cost in the forward channel that is on the RPP-to-DC and the NPP-to-DC links, along with the processing costs at the RPPs and NPPs, respectively. The last term calculates the transportation cost on the DC-to-retailer links and the processing cost at the DCs. In the following, we assume that, for the sake of simplicity, the notation G_{prk}, G_{psd}, G_{ptd} and G_{pdr} correspond to the terms $(G_{rk}+\kappa_{pk})$, $(G_{sd}+\rho_{ps})$, $(G_{td}+\nu_{pt})$, and $(G_{dr}+\eta_{pd})$, respectively.

Constraints

$$\sum_{s\in\mathcal{S}_p} v_{ps} = 1 \qquad \forall\, p \in \mathcal{P}, \tag{4.2}$$

$$\sum_{k\in\mathcal{K}} u_{prk} = \delta_{pr} D_{pr} \qquad \forall\, p \in \mathcal{P},\ r \in \mathcal{R}, \tag{4.3}$$

$$\sum_{s\in\mathcal{S}_p} x_{pks} - \sum_{r\in\mathcal{R}} u_{prk} = 0 \qquad \forall\, p \in \mathcal{P},\ k \in \mathcal{K}, \tag{4.4}$$

$$\sum_{k\in\mathcal{K}} x_{pks} - \sum_{r\in\mathcal{R}} \delta_{pr}\, D_{pr}\, v_{ps} = 0 \qquad \forall\, p \in \mathcal{P},\ s \in \mathcal{S}_p, \tag{4.5}$$

$$\sum_{d\in\mathcal{D}} y_{psd} - \alpha_{ps} \sum_{r\in\mathcal{R}} \delta_{pr}\, D_{pr}\, v_{ps} = 0 \qquad \forall\, p \in \mathcal{P},\ s \in \mathcal{S}_p, \tag{4.6}$$

$$\sum_{s\in\mathcal{S}_p} y_{psd} + \sum_{t\in\mathcal{T}_p} z_{ptd} - \sum_{r\in\mathcal{R}} w_{pdr} = 0 \qquad \forall\, p \in \mathcal{P},\ d \in \mathcal{D}, \tag{4.7}$$

$$\sum_{d\in\mathcal{D}} w_{pdr} = \left(D_{pr} + D'_{pr}\right) \qquad \forall\, p \in \mathcal{P},\ r \in \mathcal{R}, \tag{4.8}$$

$$\sum_{p\in\mathcal{P}} \sum_{r\in\mathcal{R}} \beta_p\, u_{prk} \le B_k\, c_k \qquad \forall\, k \in \mathcal{K}, \tag{4.9}$$

$$u_{prk} \le M_{prk}\, c_k \qquad \forall\, p \in \mathcal{P},\ r \in \mathcal{R},\ k \in \mathcal{K}, \tag{4.10}$$

$$\sum_{d\in\mathcal{D}} z_{ptd} \le Q_{pt} \qquad \forall\, p \in \mathcal{P},\ t \in \mathcal{T}_p, \tag{4.11}$$

$$\sum_{p\in\mathcal{P}} \sum_{r\in\mathcal{R}} \gamma_p\, w_{pdr} \le C_d \qquad \forall\, d \in \mathcal{D}, \tag{4.12}$$

$$u_{prk},\ w_{pdr},\ x_{pks},\ y_{psd},\ z_{ptd} \ge 0 \qquad \forall\, p \in \mathcal{P},\ k \in \mathcal{K},\ d \in \mathcal{D},\ r \in \mathcal{R},\ s \in \mathcal{S}_p,\ t \in \mathcal{T}_p, \tag{4.13}$$

$$v_{ps}, c_k \in \{0, 1\} \qquad \forall\, p \in \mathcal{P},\ s \in \mathcal{S}_p. \tag{4.14}$$

Constraint set (4.2) guarantees that a single RPP location is established for each product p. Constraint set (4.3) ensures that the return quantity for product p at each retailer r is shipped to the CCs. Constraint sets (4.4), (4.5), (4.6) and (4.7) represent the flow conservation (mass balance) for each product type at the CCs, RPPs, and the DCs. Constraint set (4.5) also makes sure that, for a product p, a RPP is established at a location s if there is flow of this product from a CC k to the location s. Constraint set (4.6) enforces that the remanufactured products are totally returned into the system. Constraint set (4.8) ensures that the customer demand is satisfied using new and/or remanufactured products. Constraint set (4.9) serves to create a CC k if that location is set to receive some product returns from the retailers, and it also ensures that the aggregate processing capacity limitation at k is honored. Constraint set (4.10) is redundant; however, it provides a tighter formulation as in the case of capacitated facility location problem Van Roy (1986). Constraint sets (4.11) and (4.12) enforce the capacity restrictions at the NPP locations and the DCs, respectively. Constraint sets (4.13) and (4.14) are the restrictions on the decision variables.

As mentioned earlier, this formulation can be implemented and solved using standard MILP optimizers that are readily available. However, since the computational time to obtain optimal or near optimal solutions increases prohibitively as the size of the problem instance increases, we utilize such an optimizer only in order to obtain benchmark results, and we focus our efforts on devising efficient heuristic approaches that provide *good* feasible solutions (upper bounds) in *reasonable* runtimes. For this purpose, we propose two tabu search meta-heuristics, Sequential and Random Neighborhood Search procedures (SNS() and RNS(), respectively) in which we combine search functions utilizing three simple neighborhoods in particular ways to improve efficiency in exploring the solution space. The neighborhood functions include *moves* on CCs, and *exchanges* on CCs and RPPs. A Tabu search framework has been employed and proved itself a powerful approach for solving various combinatorial optimization problems Glover (1989, 1990); Glover and Laguna (1997). In addition, we also propose an effective transhipment heuristic to quickly estimate the goodness (objective function value) of a feasible solution so that our search procedures execute with improved solution times with little impact on final solution quality.

We then apply the BD approach presented in Section III.3 to the current problem setting and observe that, although the BD approach exhibits superior performance for small scale instances, it becomes highly impractical for large scale instances. Thus, to improve its computational efficiency, we incorporate a tabu search heuristic to provide an initial solution to use while solving the initial subproblem in the Benders decomposition scheme. Such an heuristic-enhanced Benders decomposition (denoted by HBD) framework greatly facilitates a faster convergence of the Benders decomposition, which is otherwise impossible for larger instances. On the other hand, the lower bounds thus obtained provide benchmarks that illustrate the high quality of the heuristic solutions. We present computational results to illustrate the efficient performance of the tabu search heuristics as well as the HBD framework in comparison to an exact B&C approach as implemented in CPLEX.

IV.2. Heuristic Solution Methods

In this section, we first describe the main components of the improvement heuristics, including a solution representation, a method for evaluating the goodness of a solution quickly and effectively, the initial solution construction heuristics, and the neighborhood functions used to explore the solution space.

IV.2.1. Solution Representation

We characterize a solution by a pair of integer vectors $(\mathcal{S}^u, \mathcal{K}^u)$, where $\mathcal{S}^u$ and $\mathcal{K}^u$ represent the set of open RPPs and open CCs, respectively. Since there is a single and dedicated RPP for each type of product, as stated in constraint (4.2), the set $\mathcal{S}^u$ consists of $|\mathcal{P}|$ elements, where the p^{th} element is referred to as $s^{[p]}$ in $\mathcal{S}^u$, and it represents the open RPP that processes the product p. On the other hand, the set $\mathcal{K}^u$ consists of elements that refer to the open CCs; thus, it is a subset of the set $\mathcal{K}$. We note that a feasible solution $(\mathcal{S}^u, \mathcal{K}^u)$ should satisfy the following conditions.

$$\sum_{k \in \mathcal{K}^u} B_k \geq \sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} \beta_p \, \delta_{pr} \, D_{pr} \tag{4.15}$$

$$v_{ps^{[p]}} = 1 \qquad \text{and} \qquad \sum_{s \in \mathcal{S}_p \setminus \{s^{[p]}\}} v_{ps} = 0 \qquad\qquad \forall\, p \in \mathcal{P}. \tag{4.16}$$

The inequality (4.15) states that a feasible solution must imply a network configuration with enough aggregate storage/processing capacity at the open CCs to accommodate the reverse flow of products from the retailers. The equalities in (4.16) dictate that, for each product p, the RPP location $s^{[p]}$ is the only open RPP location for processing returns. This, in turn, satisfies the restrictions represented by the constraints (4.2).

IV.2.2. Objective Function Evaluation

A given feasible solution $(\mathcal{S}^u, \mathcal{K}^u)$ implies fixed binary values for the location variables in **MP**. Specifically, for each product p, we set $v_{ps^{[p]}}$ to 1 for the candidate RPP location $s^{[p]}$ included in $\mathcal{S}^u$, and for all $s \in \mathcal{S}_p \setminus \{s^{[p]}\}$, we have a v_{ps} value of 0. Similarly, for each $k \in \mathcal{K}^u$, we have a c_k value equal to 1, and for all other candidate CC locations given by $\mathcal{K} \setminus \mathcal{K}^u$, the c_k is zero. Then, the resulting linear program, which we denote by $\mathbf{SP}(\mathbf{u},\ \mathbf{w},\ \mathbf{x},\ \mathbf{y},\ \mathbf{z}|\mathcal{S}^u, \mathcal{K}^u)$, can be solved using any linear programming (LP) solver (e.g. CPLEX) to obtain the optimal value for the rest of the variables ($\mathbf{u},\ \mathbf{w},\ \mathbf{x},\ \mathbf{y},\ \mathbf{z}$) and the objective function. However, in a heuristic neighborhood search framework, since the goodness of a solution needs to be evaluated many times (specifically, for each solution encountered in the process), the use of an LP solver for this purpose results in excessively high runtimes for the heuristics, especially for large problem instances. Hence, we next describe a transshipment heuristic to quickly estimate the objective value or goodness of a given feasible solution in the course of the solution improvement procedure. We note that a similar procedure was used previously for the same purpose while solving a production/distribution system problem in (Keskin and Üster, 2007).

Given a feasible solution $(\mathcal{S}^u, \mathcal{K}^u)$, the $\mathbf{SP}(\mathbf{u},\ \mathbf{w},\ \mathbf{x},\ \mathbf{y},\ \mathbf{z}|\mathcal{S}^u, \mathcal{K}^u)$ can be solved in five stages as shown in Display 2. In the first stage (lines 1-12), we solve for the flow variables u_{prk} in a greedy fashion, based on the unit costs θ^1_{prk}, and the capacity of the open CCs. In the second stage (lines 13-14), we solve for the flow variables x_{pks}. For each product p, once we know the quantities collected at the CC locations from the previous stage, the CCs should ship the entire quantity of returns to the open RPP location. This can be easily calculated as shown on line 14. In the third stage (lines 15-27), we solve the transportation problem between the DCs and the retailers. We assign the flows w_{pdr} in a greedy fashion, based on the unit cost θ^2_{pdr}, and the capacities at the DCs. The last two stages solve a transportation problem from NPP-RPP locations to the DCs for each product p separately. In such a problem for a fixed p, the open RPP location and the NPP locations constitute the set of supply locations. We represent this set by $\mathcal{T}'_p = \mathcal{T}_p \cup \{s^{[p]}\}$, where the first $|\mathcal{T}_p|$ elements of the set $\mathcal{T}'_p$ correspond to the NPPs and the $(|\mathcal{T}_p| + 1)$-st element corresponds to the open RPP $s^{[p]}$. We define an index $T \in \mathcal{T}'_p$ that refers to the open RPP $s^{[p]}$. The

RPP can use all of the remanufactured products to cater to the requirements at the DCs, and hence, we define its capacity as $Q_{pT} = \alpha_{ps^{[p]}} \sum_{r \in \mathcal{R}} \delta_{pr} D_{pr}$. Associated with the index T, we also define flow variables z_{pTd}, and unit costs θ^3_{pTd} (line 31). We assign the flows z_{ptd} in a greedy fashion, based on the unit costs θ^3_{ptd}, and the supply capacities Q_{pt}. At the end of these assignments, the RPP may have some unused remanufactured products, in which case $Q_{pT} > 0$. To ensure the usage of the entire quantity of remanufactured products to satisfy the DC demands, in the fifth stage (lines 42-57), we increase the flow from the RPP $s^{[p]}$ while simultaneously reducing the flows from the NPP locations by an equivalent amount, in a greedy fashion based on the unit cost θ^4_{ptd}, until constraint (4.6) is satisfied. Finally, the values of the flow variables $\mathbf{u, w, x, y, z}$, together with the values of the binary variables $\mathbf{c}$ and $\mathbf{v}$, are used to estimate the objective function value $Z(\mathcal{S}^u, \mathcal{K}^u)$ of a feasible solution $(\mathcal{S}^u, \mathcal{K}^u)$.

We also implement a variation to the above heuristic by replacing stages I and II with a single stage consisting of all the reverse channel flow variables and by replacing stages III and IV with a single stage consisting of all the forward channel flow variables. In this variation, instead of solving flow variables related to a given channel in two distinct stages, we include all of the channel locations for consideration at once. In this alternative implementation, stage V of the above heuristic is also modified to include the flow variables $\mathbf{w}$ with the corresponding unit costs and the capacities. This variation provides better objective function estimates, but at the expense of higher computational times.

Display 2 Procedure ObjectiveEval()

STAGE I - FLOWS FROM RETAILERS TO CCS
Calculate $RD_{pr} = \delta_{pr} D_{pr}$ and $\theta^1_{prk} = G_{prk}\ \forall p \in \mathcal{P}, r \in \mathcal{R}, k \in \mathcal{K}^u$
List θ^1_{prk} in non-decreasing order
for each θ^1_{prk} in this list **do**
    if $RD_{pr} > 0$ **then**
        if $B_k \geq \beta_p\ RD_{pr}$ **then**
            $u_{prk} \leftarrow u_{prk} + RD_{pr}$; $B_k \leftarrow B_k - (\beta_p\ RD_{pr})$; $RD_{pr} = 0$
        else if $B_k > 0$ **then**
            $u_{prk} \leftarrow u_{prk} + (B_k/\beta_p)$; $RD_{pr} \leftarrow RD_{pr} - (B_K/\beta_p)$; $B_k = 0$
        end if
    end if
end for
STAGE II - FLOWS FROM CCS TO RPPS
$x_{pks} \leftarrow \sum_{r\in\mathcal{R}} \delta_{pr} D_{pr}\, v_{ps}\, c_k\ \forall p \in \mathcal{P}, k \in \mathcal{K}, s \in \mathcal{S}_p$
STAGE III - FLOWS FROM DCS TO RETAILERS
Calculate $FD_{pr} = D'_{pr} + D_{pr}$ and $\theta^2_{pdr} = G_{pdr}\ \forall p \in \mathcal{P}, r \in \mathcal{R}, d \in \mathcal{D}$
List θ^2_{pdr} in non-decreasing order
for each θ^2_{pdr} in this list **do**
    if $FD_{pr} > 0$ **then**
        if $C_d \geq \gamma_p\, FD_{pr}$ **then**
            $w_{pdr} \leftarrow w_{pdr} + FD_{pr}$; $C_d \leftarrow C_d - (\gamma_p\, FD_{pr})$; $FD_{pr} = 0$
        else if $C_d > 0$ **then**
            $w_{pdr} \leftarrow w_{pdr} + (C_d/\gamma_p)$; $FD_{pr} \leftarrow FD_{pr} - (C_d/\gamma_p)$; $C_d = 0$
        end if
    end if
end for
for each product $p \in \mathcal{P}$ **do**
STAGE IV - FLOWS FROM NPP/RPP TO DCS
    Set $\mathcal{T}'_p = \mathcal{T}_p \bigcup \{s^{[p]}\}$ and let $T = |\mathcal{T}_p| + 1$ represent the RPP location $s^{[p]}$ in set $\mathcal{T}'_p$
    Calculate $DCD_{dp} = \sum_{r\in\mathcal{R}} w_{pdr} \forall p \in \mathcal{P}, d \in \mathcal{D}$ and $\theta^3_{ptd} = G_{ptd}\ \forall p \in \mathcal{P}, d \in \mathcal{D}, t \in \mathcal{T}_p$;
    Set $\theta^3_{pTd} = G_{ps^{[p]}d}$; $Q_{pT} = \alpha_{ps^{[p]}} \sum_{r\in\mathcal{R}} \delta_{pr}\, D_{pr}$; $z_{pTd} = y_{ps^{[p]}d}$;
    List θ^3_{ptd} in non-decreasing order, where $t \in \mathcal{T}'_p$
    for each θ^3_{ptd} in this list **do**
        if $DCD_{dp} > 0$ **then**
            if $Q_{pt} \geq DCD_{dp}$ **then**
                $z_{ptd} \leftarrow z_{ptd} + DCD_{dp}$; $Q_{pt} \leftarrow Q_{pt} - DCD_{dp}$; $DCD_{dp} = 0$
            else if $Q_{pt} > 0$ **then**
                $z_{ptd} \leftarrow z_{ptd} + Q_{pt}$; $DCD_{dp} = Q_{pt}$; $Q_{pt} = 0$
            end if
        end if
    end for
STAGE V - FLOW ADJUSTMENTS
    if $Q_{pT} > 0$ **then**
        Calculate $\theta^4_{ptd} = G_{ps^{[p]}d} - G_{ptd}\ \forall p \in \mathcal{P}, d \in \mathcal{D}, t \in \mathcal{T}_p$
        List θ^4_{ptd} in non-decreasing order, where $t \in \mathcal{T}_p$
        for each θ^4_{ptd} in this list **do**
            if $Q_{pT} > 0$ **then**
                if $Q_{pT} > z_{ptd}$ **then**
                    $y_{ps^{[p]}d} \leftarrow y_{ps^{[p]}d} + z_{ptd}$; $Q_{Tp} \leftarrow Q_{pT} - z_{ptd}$; $z_{ptd} = 0$
                else
                    $y_{ps^{[p]}d} \leftarrow y_{ps^{[p]}d} + Q_{Tp}$; $z_{ptd} \leftarrow z_{ptd} - Q_{pT}$; $Q_{pT} = 0$
                end if
            else
                break;
            end if
        end for
    end if
end for
return $Z(\mathcal{S}^u, \mathcal{K}^u)$

IV.2.3. Construction Heuristics

Due to our solution representation, the construction of a feasible solution corresponds to finding a set of locations for the CCs and the product RPPs. For this purpose, we utilize a partially randomized algorithm, given in Display 3, that aims to identify the most favorable set of RPP locations for each product by randomly sampling over the possible selections of the CC locations (i.e., a subset of $\mathcal{K}$). More specifically, first, we pick a set of CCs at random to form the set $\mathcal{K}^u$, such that the condition (4.15) is satisfied. Then, for each $k \in \mathcal{K}^u$, we fix the values of the location variables c_k to 1 in the formulation **MP**. Similarly, for each $k \in \mathcal{K} \setminus \mathcal{K}^u$, we fix the values of the location variables c_k to 0. We then relax the binary restrictions on the location variables v_{ps} to obtain a LP which we denote by $\mathbf{SPK}(\mathbf{u},\ \mathbf{v},\ \mathbf{w},\ \mathbf{x},\ \mathbf{y},\ \mathbf{z}|\mathcal{K}^u)$. We solve this LP, and for each RPP location, we record a $\texttt{score}(p, s)$ parameter value determined by the the value of the associated v_{ps} variables in the solution. We repeat this process to a preset number of iterations, represented by `CTR`. After performing `CTR` iterations, for each product $p \in \mathcal{P}$, we choose the RPP with the maximum cumulative $\texttt{score}(p, s)$ and use it in the construction of the set $\mathcal{S}^u$ of the initial feasible solution. Furthermore, while performing the iterations, we record a set $\mathcal{K}^{u*}$ corresponding to the least cost objective value $Z^*(\mathcal{K}^u)$ for the $\mathbf{SPK}(\mathbf{u},\ \mathbf{v},\ \mathbf{w},\ \mathbf{x},\ \mathbf{y},\ \mathbf{z}|\mathcal{K}^u)$. We then use the set $\mathcal{K}^{u*}$ as the $\mathcal{K}^u$ in the initial feasible solution.

We note that alternatively to the above implementation, we can construct a feasible solution by fixing the RPP locations instead of the open CC locations. In this case, for each product p, we randomly choose one RPP location to be opened. We set the values of the binary variables v_{ps} accordingly to 1 or 0. We then relax the binary restrictions on the c_k variables to obtain a LP $\mathbf{SPS}(\mathbf{c},\ \mathbf{u},\ \mathbf{w},\ \mathbf{x},\ \mathbf{y},\ \mathbf{z}|\mathcal{S}^u)$. We measure the popularity of the CCs using a parameter $\texttt{score}(k)$ that cumulatively holds the values of the c_k variables obtained at each iteration. After a number of iterations over the sets of open RPP locations, we construct the set $\mathcal{K}^u$ as follows. We include the CC location with the highest $\texttt{score}(k)$ value in the set $\mathcal{K}^u$. If the capacity constraint (4.15) is violated, then we pick the CC with the next highest $\texttt{score}(k)$ value, and repeat this procedure until the constraint is satisfied. The resulting set $\mathcal{K}^u$ contains the selected CC locations.

Based on our computational testing of both of the construction procedures, we find that the former procedure exhibits superior performance, both in terms of computational time and the quality of the solution (objective function value). Hence, we employ the former procedure which is given in Display 3.

Display 3 Procedure Construct()

initialize $\texttt{CTR}$, $i = 1$, $\mathcal{S}^u = \emptyset$, $\mathcal{K}^{u*} = \emptyset$, $Z_{best} = \infty$,
    $\texttt{score}(p, s) = 0, \; \forall p \in \mathcal{P}, \; s \in \mathcal{S}_p$
while $\texttt{itr} \leq \texttt{CTR}$ **do**
  $\mathcal{K}' = \mathcal{K}$ and $\mathcal{K}^u = \emptyset$
  while $(\sum_{k \in \mathcal{K}^u} B_k < \sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} \beta_p \, \delta_{pr} \, D_{pr})$ **do**
    Randomly pick a CC $k \in \mathcal{K}'$
    $\mathcal{K}^u = \mathcal{K}^u \cup \{k\}$ and $\mathcal{K}' = \mathcal{K}' \setminus \{k\}$
  end while
  Solve $\mathbf{SPK}(\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z} | \mathcal{K}^u)$ to find $Z^*(\mathcal{K}^u)$ and $\mathbf{v}^*$
  $\texttt{score}(p, s) = \texttt{score}(p, s) + v^*_{ps}, \; \forall p \in \mathcal{P}, \; s \in \mathcal{S}_p$
  if $Z^*(\mathcal{K}^u) < Z_{best}$ **then**
    $Z_{best} = Z^*(\mathcal{K}^u)$ and $\mathcal{K}^{u*} = \mathcal{K}^u$
  end if
  $\texttt{itr} = \texttt{itr} + 1$
end while
For each p, $s^{[p]} = \arg\max\{\texttt{score}(p, s) \colon s \in \mathcal{S}_p\}$
Construct $\mathcal{S}^u$ using the $s^{[p]}$, $\forall p \in \mathcal{P}$
Solve $\mathbf{SP}(\mathbf{u}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z} | \mathcal{S}^u, \mathcal{K}^{u*})$
return $(\mathcal{S}^u, \mathcal{K}^{u*})$ and $Z(\mathcal{S}^u, \mathcal{K}^{u*})$.

IV.2.4. Neighborhood Functions

We employ three neighborhood functions which generate three distinct sets of neighboring solutions.

CC-exchange neighborhood (ccx): Given a feasible solution $(\mathcal{S}^u, \mathcal{K}^u)$, the CC-exchange neighborhood function considers only the set $\mathcal{K}^u$ to construct its neighboring solutions. For each open CC in the set $\mathcal{K}^u$, we randomly choose a preset number ENL of CCs from the set $\mathcal{K} \setminus \mathcal{K}^u$ to form the exchange pairs. That is, an *exchange* simply refers to closing an open CC while opening a new one. If an exchange implies a violation of the aggregate capacity constraint (4.15), we simply ignore it and proceed to the next possible exchange. We denote the set of neighboring solutions by $\mathcal{V}_{\texttt{ccx}}$ which includes $\texttt{ENL}|\mathcal{K}^u|$ solutions. Note that the $|\mathcal{V}_{\texttt{ccx}}|$ can be at most $|\mathcal{K}^u||\mathcal{K}\backslash\mathcal{K}^u|$.

CC-move neighborhood (ccm): Given a feasible solution $(\mathcal{S}^u, \mathcal{K}^u)$, the CC-move neighborhood function considers the set $\mathcal{K}^u$ similarly to the previous neighborhood. However, in this case, a *move* operation corresponds to opening a closed CC *or* closing an open CC. We randomly pick a preset number MNL of CCs to perform move operations and ignore a move that implies a solution violating the constraint (4.15). We denote the the set of MNL solutions thus obtained by $\mathcal{V}_{\texttt{ccm}}$ which can have a cardinality of at most $|\mathcal{K}|$.

RPP-exchange neighborhood (rppx): Given a feasible solution $(\mathcal{S}^u, \mathcal{K}^u)$, the RPP-exchange neighborhood considers the set $\mathcal{S}^u$ to generate neighboring solutions. In order to generate neighboring solutions by considering each product, we first fix a product type p for which some RPP-exchanges are to be made. Then, for the open RPP $s^{[p]}$, we randomly select a preset number $\texttt{RNL}_\texttt{p}$ of RPPs from the set $\mathcal{S}_p \backslash \{s^{[p]}\}$ to form the exchange pairs and the corresponding neighboring solutions. We repeat this process to form neighboring solutions for all of the products. We denote the complete set of $\mathtt{RNL_p}|\mathcal{P}|$ neighboring solutions by $\mathcal{V}_{\mathtt{rppx}}$. Note that, for each product p, the $\mathtt{RNL_p}$ can have a maximum value of $|\mathcal{S}_p| - 1$.

Note that, in the sequel, we use $(\mathcal{S}^u, \mathcal{K}^u)_{cur}$, $(\mathcal{S}^u, \mathcal{K}^u)_{new}$, and $(\mathcal{S}^u, \mathcal{K}^u)_{best}$ to represent the current, new, and incumbent solutions, respectively. Furthermore, for brevity, we use the notation Z_{new} and Z_{best} to represent the objective function values of the new and incumbent solutions, respectively.

IV.2.5. Sequential Neighborhood Search Procedure

Observe that, in general, the CC-Exchange neighborhood is larger than the CC-Move and RPP-Exchange neighborhoods. Following this observation, while devising a heuristic search procedure, we assign different roles to these neighborhood functions to facilitate the intensification and diversification characteristics. Specifically, we employ the former in a short-term memory tabu search framework where we emphasize intensification, and the latter two successively in a perturb routine to facilitate diversification. The overall search procedure SNS is outlined in Display 4.

First, we find an initial feasible solution using the construction heuristics outlined in Section IV.2.3. We then use this solution in the tabu search framework to explore the neighborhood $\mathcal{V}_{\mathtt{ccx}}$, and find an improvement in its objective value. The tabu-search heuristic parameters include the aspiration level `ASP` and the number of iterations `MaxIter`. We maintain a tabu list $\mathtt{TabuL_{ccx}}$, in which an entry k corresponds to a CC that is *closed* in a previous iteration and is prohibited to be *opened* unless the corresponding CC-exchange yields an objective value better than the aspiration level. Once we find a solution better than the incumbent, we update the aspiration level with its objective value and the tabu list with the current exchange; we also include the solution in a list `EliteL` of *elite solutions*. Upon completion of the the tabu search after `MaxIter` iterations, we use the procedure outlined in Display 5 to *perturb* the incumbent solution for possible improvements.

In the perturbation process, we first examine all of the solutions in the neighborhood $\mathcal{V}_{\texttt{ccm}}$ of the incumbent solution by setting the parameter `MNL` to $|\mathcal{K}|$. We update the incumbent, and, then, we examine all of the solutions in the neighborhood $\mathcal{V}_{\texttt{rppx}}$ of the updated incumbent by setting the parameters $\texttt{RNL}_{\texttt{p}}$ to $|\mathcal{S}_p| - 1$ for each product type p. Following the perturbation procedure, we update the current solution to the new incumbent solution. Using this updated current solution as input, we repeat the tabu search–perturbation search *sequence* in an outer loop. We terminate the outer loop either if it exceeds a preset number of iterations (`ITNLOOP`) or if there is no improvement in the objective value of the incumbent solution. Finally, after the termination of the outer loop, we once again use the perturbation search to explore the neighborhood of the incumbent solution. This final perturbation search often provides improvement over the solution that results from the outer loop of the SNS procedure.

In the above heuristic framework, to estimate the objective value of a solution, we can use either a standard LP solver or the transportation heuristic described in Section IV.2.2. We maintain a list `EliteL` only if we use the latter approach. This is because the objective function value (obtained by the transportation heuristic) at the end of the overall procedure is not necessarily optimal. In this case, the list `EliteL` provides an advantage over a single incumbent solution since it often contains solutions with better *optimal* objective values than that of the incumbent solution. Hence, as a final step in the overall algorithm, we find the optimal objective values for all of the solutions in the list `EliteL` using the LP solver and then pick the solution that provides the least optimal objective value. We refer to the SNS procedure with the transportation heuristic as the SNS-TH and as the SNS if we employ an LP solver for objective function evaluations.

Display 4 Procedure SNS()

Construct an initial feasible solution $(\mathcal{S}^u, \mathcal{K}^u)$ using the procedure Construct()
Set $(\mathcal{S}^u, \mathcal{K}^u)_{cur} = (\mathcal{S}^u, \mathcal{K}^u)$ and $(\mathcal{S}^u, \mathcal{K}^u)_{best} = (\mathcal{S}^u, \mathcal{K}^u)_{cur}$
Initialize neighborhood parameters `ENL`, `MNL`, and $\texttt{RNL}_\texttt{p}$
while No improvement on Z_{best} and $\texttt{otr} \leq \texttt{ITNLOOP}$ **do**
  Initialize $\texttt{EliteL} = \emptyset$, $\texttt{TabuL}_\texttt{ccx} = \emptyset$, $\texttt{ASP} = Z_{best}$, $\texttt{itr} = 0$, and `MaxIter`
  while $\texttt{itr} \leq \texttt{MaxIter}$ **do**
    Generate $\mathcal{V}_{ccx}$ for the $(\mathcal{S}^u, \mathcal{K}^u)_{cur}$ with `ENL`
    Find the best solution $(\mathcal{S}^u, \mathcal{K}^u)_{new}$ in $\mathcal{V}_{ccx}$ and Z_{new}
    if $(\mathcal{S}^u, \mathcal{K}^u)_{new}$ is permitted by $\texttt{TabuL}_\texttt{ccx}$ **then**
      $(\mathcal{S}^u, \mathcal{K}^u)_{cur} = (\mathcal{S}^u, \mathcal{K}^u)_{new}$
      if $Z_{new} < Z_{best}$ **then**
        $(\mathcal{S}^u, \mathcal{K}^u)_{best} = (\mathcal{S}^u, \mathcal{K}^u)_{new}$, $\texttt{ASP} = Z_{best}$, and
        $\texttt{EliteL} = \texttt{EliteL} \cup \{(\mathcal{S}^u, \mathcal{K}^u)_{best}\}$
      end if
      Update $\texttt{TabuL}_\texttt{ccx}$; $\texttt{itr} = \texttt{itr} + 1$
    else
      if $Z_{new} < \texttt{ASP}$ **then**
        $(\mathcal{S}^u, \mathcal{K}^u)_{cur} = (\mathcal{S}^u, \mathcal{K}^u)_{new}$; $(\mathcal{S}^u, \mathcal{K}^u)_{best} = (\mathcal{S}^u, \mathcal{K}^u)_{new}$
        $\texttt{ASP} = Z_{best}$, and
        $\texttt{EliteL} = \texttt{EliteL} \cup \{(\mathcal{S}^u, \mathcal{K}^u)_{best}\}$ and update $\texttt{TabuL}_\texttt{ccx}$; $\texttt{itr} = \texttt{itr} + 1$
      end if
    end if
  end while
  $(\mathcal{S}^u, \mathcal{K}^u)_{best} = \text{PERTURB}((\mathcal{S}^u, \mathcal{K}^u)_{best}, \texttt{EliteL})$
  $(\mathcal{S}^u, \mathcal{K}^u)_{cur} = (\mathcal{S}^u, \mathcal{K}^u)_{best}$
  $\texttt{otr} = \texttt{otr} + 1$
end while
$(\mathcal{S}^u, \mathcal{K}^u)_{best} = \text{PERTURB}((\mathcal{S}^u, \mathcal{K}^u)_{best}, \texttt{EliteL})$
Find the **optimal** objective values of the solutions in `EliteL` and pick best solution as the $(\mathcal{S}^u, \mathcal{K}^u)_{best}$
Return $(\mathcal{S}^u, \mathcal{K}^u)_{best}$ and Z_{best}.

Display 5 Procedure PERTURB($(\mathcal{S}^u, \mathcal{K}^u)_{best}$, `EliteL`)

```
PERTURB USING CC-MOVE
Generate V_ccm for the (S^u, K^u) with MNL = |K|
Find the best solution (S^u, K^u)_new in V_ccm and Z_new
if Z_new < Z_best then
    (S^u, K^u)_best = (S^u, K^u)_new, and EliteL = EliteL ∪ {(S^u, K^u)_best}
end if
PERTURB USING RPP-EXCHANGE
Generate V_rppx for the (S^u, K^u)_best with RNL_p = (|S_p| − 1) ∀p ∈ P
Find the best solution (S^u, K^u)_new in V_rppx and Z_new.
if Z_new < Z_best then
    (S^u, K^u)_best = (S^u, K^u)_new, and EliteL = EliteL ∪ {(S^u, K^u)_best}
end if
Return (S^u, K^u)_best and EliteL.
```

IV.2.6. Random Neighborhood Search Procedure

In the SNS procedure described in the previous section, we combine the three simple neighborhood functions in the tabu search–perturbation search sequence. Whereas, in the RNS procedure (Display 6), we randomly choose a simple neighborhood function each time within a tabu search heuristic framework.

First, we use construction heuristics to generate an initial solution. Using this solution as input, we perform a random neighborhood tabu search procedure. In the tabu search, we maintain three different tabu lists $\texttt{TabuL}_{\texttt{ccx}}$, $\texttt{TabuL}_{\texttt{ccm}}$, and $\texttt{TabuL}_{\texttt{rppx}}$ corresponding to the CC-exchange, the CC-move, and the RPP-exchange neighborhoods, respectively. An entry k in $\texttt{TabuL}_{\texttt{ccx}}$ corresponds to a CC that is *closed* in a previous CC-exchange. An entry k in $\texttt{TabuL}_{\texttt{ccm}}$ corresponds to a CC location, which cannot be *opened* (or *closed*) if it is *closed* (or *opened*) in a previous CC-move. The tabu list $\texttt{TabuL}_{\texttt{rppx}}$ consists of $|\mathcal{P}|$ sub-lists corresponding to different product types. An entry s in the p^{th} sub-list corresponds to a RPP location capable of processing product type p, which was *closed* in a previous iteration.

At any given iteration of the tabu search procedure, we randomly choose a neighborhood function ω from the neighborhood functions `ccx`, `ccm`, and `rppx`. We then explore the corresponding neighborhood $\mathcal{V}_\omega$ to find a new solution with a better objective value. If the new solution has an objective value better than the incumbent value, we update the incumbent solution, the tabu list $\texttt{TabuL}_\omega$ corresponding to the neighborhood ω, and the aspiration level `ASP`. We then include the updated incumbent in the elite list `EliteL`. We consider the new solution for improvement and repeat the process for `MaxIter` times. Following the termination of the tabu search, we use the incumbent solution as input and repeat the search in an outer loop until the stopping criteria are realized. Similarly to the SNS procedure, upon completion of the outer loop, we use the perturbation search to explore the neighborhood of the incumbent solution for further improvement.

Also as in SNS-TH, we use an elite list `EliteL` in conjunction with the transportation heuristic and report the solution with the best *optimally* evaluated objective value in `EliteL` as the final solution. We refer to the RNS procedure with the transportation heuristic as the RNS-TH and as the RNS if we employ an LP solver for objective function evaluations.

IV.2.7. An Alternative Parallel Neighborhood Search and Other Variations

We also implemented the parallel neighborhood search (PNS) procedure, which is very similar to the RNS procedure. At each iteration of the tabu search, instead of randomly choosing a simple neighborhood function to explore the neighborhood, we use all the three simple neighborhood functions to perform the search and pick a solution with the best objective function value over all the CC-exchanges, CC-moves and the RPP-exchanges. Our computational experimentation of the PNS heuristic resulted in a very poor performance, in comparison to the SNS and RNS heuristics, both in terms of computational time and quality of the resulting solution. Moreover, we examined the computational performance of the variations of the SNS procedure, where we varied the sequence of execution of the simple neighborhood functions. We found that the computational time and the quality of the solution in terms of the objective function values for the SNS procedure, are better than those of its variations. Hence for the sake of brevity, we report the computational performance of the SNS and RNS procedures.

Display 6 Procedure RNS()

Construct an initial feasible solution $(\mathcal{S}^u, \mathcal{K}^u)$ using the procedure Construct()
Set $(\mathcal{S}^u, \mathcal{K}^u)_{cur} = (\mathcal{S}^u, \mathcal{K}^u)$ and $(\mathcal{S}^u, \mathcal{K}^u)_{best} = (\mathcal{S}^u, \mathcal{K}^u)_{cur}$
Initialize neighborhood parameters `ENL`, `MNL`, and $\texttt{RNL}_\texttt{p}$
while No improvement on Z_{best} and $\texttt{otr} \leq \texttt{ITNLOOP}$ **do**
  Initialize $\texttt{EliteL} = \emptyset$, $\texttt{TabuL}_\texttt{ccx} = \emptyset$, $\texttt{TabuL}_\texttt{ccm} = \emptyset$, $\texttt{TabuL}_\texttt{rppx} = \emptyset$, $\texttt{ASP} = Z_{best}$, $\texttt{itr} = 0$, and `MaxIter`
  while $\texttt{itr} \leq \texttt{MaxIter}$ **do**
    Let $\omega = \texttt{random}\{\texttt{ccx}, \texttt{ccm}, \texttt{rppx}\}$
    Generate $\mathcal{V}_\omega$ for the $(\mathcal{S}^u, \mathcal{K}^u)_{cur}$
    Find the best solution $(\mathcal{S}^u, \mathcal{K}^u)_{new}$ in $\mathcal{V}_\omega$
    if $(\mathcal{S}^u, \mathcal{K}^u)_{new}$ is permitted by $\texttt{TabuL}_\omega$ **then**
      $(\mathcal{S}^u, \mathcal{K}^u)_{cur} = (\mathcal{S}^u, \mathcal{K}^u)_{new}$
      if $Z_{new} < Z_{best}$ **then**
        $(\mathcal{S}^u, \mathcal{K}^u)_{best} = (\mathcal{S}^u, \mathcal{K}^u)_{new}$, $\texttt{ASP} = Z_{best}$, and $\texttt{EliteL} = \texttt{EliteL} \cup \{(\mathcal{S}^u, \mathcal{K}^u)_{best}\}$
      end if
      Update $\texttt{TabuL}_\omega$; $\texttt{itr} = \texttt{itr} + 1$
    else
      if $Z_{new} < \texttt{ASP}$ **then**
        $(\mathcal{S}^u, \mathcal{K}^u)_{cur} = (\mathcal{S}^u, \mathcal{K}^u)_{new}$; $(\mathcal{S}^u, \mathcal{K}^u)_{best} = (\mathcal{S}^u, \mathcal{K}^u)_{cur}$ $\texttt{ASP} = Z_{best}$, and $\texttt{EliteL} = \texttt{EliteL} \cup \{(\mathcal{S}^u, \mathcal{K}^u)_{best}\}$ and update $\texttt{TabuL}_\omega$; $\texttt{itr} = \texttt{itr} + 1$
      end if
    end if
  end while
  $(\mathcal{S}^u, \mathcal{K}^u)_{cur} = (\mathcal{S}^u, \mathcal{K}^u)_{best}$ and $\texttt{otr} = \texttt{otr} + 1$
end while
$(\mathcal{S}^u, \mathcal{K}^u)_{best} = \text{PERTURB}((\mathcal{S}^u, \mathcal{K}^u)_{best}, \texttt{EliteL})$
Find the **optimal** objective values of the solutions in `EliteL` and pick best solution as the $(\mathcal{S}^u, \mathcal{K}^u)_{best}$
Return $(\mathcal{S}^u, \mathcal{K}^u)_{best}$ and Z_{best}.

IV.3. Benders Decomposition Framework

We extend the BD solution framework presented in Section III.3 for the problem formulation **MP**. In this section, we describe the Benders subproblems, the associated dual subproblems, the master problem, and an integration of the BD with the tabu search heuristics described above.

IV.3.1. Benders Subproblem

The Benders subproblem **BSP**(**u**,**w**,**x**,**y**,**z** $|\widehat{\mathbf{c}}, \widehat{\mathbf{v}}$) is a linear program obtained by fixing the binary decisions pertaining to the CC and RPP locations in the **MP**. In the iterative BD framework, the values of these binary variables are supplied to the subproblem by the Benders master problem (**BMP**) which we describe in Section IV.3.2. The subproblem **BSP**$(\cdot)$ is given by

$$\begin{aligned} \min \quad & \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\sum_{k\in\mathcal{K}} G_{prk}u_{prk} + \sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}_p} G_{ks}x_{pks} + \sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p}\sum_{d\in\mathcal{D}} G_{psd}y_{psd} \\ + \quad & \sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}_p}\sum_{d\in\mathcal{D}} G_{ptd}z_{ptd} + \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} G_{pdr}w_{pdr} \end{aligned} \tag{4.17}$$

subject to

$$\sum_{k\in\mathcal{K}} u_{prk} = \delta_{pr}D_{pr} \qquad \forall\, p\in\mathcal{P},\ r\in\mathcal{R}, \tag{4.18}$$

$$\sum_{s\in\mathcal{S}_p} x_{pks} - \sum_{r\in\mathcal{R}} u_{prk} = 0 \qquad \forall\, p\in\mathcal{P},\ k\in\mathcal{K}, \tag{4.19}$$

$$\sum_{k\in\mathcal{K}} x_{pks} - \sum_{r\in\mathcal{R}} \delta_{pr}\, D_{pr}\, \widehat{v}_{ps} = 0 \qquad \forall\, p\in\mathcal{P},\ s\in\mathcal{S}_p, \tag{4.20}$$

$$\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} \beta_p\, u_{prk} \le B_k\, \widehat{c}_k \qquad \forall\, k\in\mathcal{K}, \tag{4.21}$$

$$u_{prk} \leq M_{prk}\, \widehat{c}_k \qquad \forall\, p \in \mathcal{P},\, r \in \mathcal{R},\, k \in \mathcal{K}, \tag{4.22}$$

$$\sum_{d \in \mathcal{D}} y_{psd} - \alpha_{ps} \sum_{r \in \mathcal{R}} \delta_{pr}\, D_{pr}\, \widehat{v}_{ps} = 0 \qquad \forall\, p \in \mathcal{P},\, s \in \mathcal{S}_p, \tag{4.23}$$

$$\sum_{s \in \mathcal{S}_p} y_{psd} + \sum_{t \in \mathcal{T}_p} z_{ptd} - \sum_{r \in \mathcal{R}} w_{pdr} = 0 \qquad \forall\, p \in \mathcal{P},\, d \in \mathcal{D}, \tag{4.24}$$

$$\sum_{d \in \mathcal{D}} w_{pdr} = \left(D_{pr} + D'_{pr}\right) \qquad \forall\, p \in \mathcal{P},\, r \in \mathcal{R}, \tag{4.25}$$

$$\sum_{d \in \mathcal{D}} z_{ptd} \leq Q_{pt} \qquad \forall\, p \in \mathcal{P},\, t \in \mathcal{T}_p, \tag{4.26}$$

$$\sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} \gamma_p\, w_{pdr} \leq C_d \qquad \forall\, d \in \mathcal{D}, \tag{4.27}$$

$$u_{prk},\, x_{pks}, w_{pdr},\, y_{psd},\, z_{ptd} \geq 0 \qquad \forall\, p \in \mathcal{P},\, d \in \mathcal{D},\, k \in \mathcal{K},\, r \in \mathcal{R},\, s \in \mathcal{S}_p,\, t \in \mathcal{T}_p. \tag{4.28}$$

IV.3.1.1. Dual Subproblem

In order to generate cuts for the master problem $\mathbf{BMP}(\cdot)$, we use the dual linear program of $\mathbf{BSP}(\mathbf{u}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}\ |\widehat{\mathbf{c}}, \widehat{\mathbf{v}})$. For this purpose, we define dual variables λ_{rp}, θ_{pk}, ζ_{ps}, φ_k, ι_{prk} σ_{ps}, μ_{pd}, ε_{rp}, τ_{pt}, and π_d corresponding to the constraints (4.18), (4.19), (4.20), (4.21), (4.22), (4.23), (4.24), (4.25), (4.26), and (4.27), respectively. The dual linear program $\mathbf{DBSP}(\boldsymbol{\varepsilon}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\varphi}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\theta}, \boldsymbol{\zeta}\ |\widehat{\mathbf{c}}, \widehat{\mathbf{v}})$ is stated as

$$\max \quad \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\delta_{pr}D_{pr}\lambda_{rp}+\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p}\sum_{r\in\mathcal{R}}\delta_{pr}D_{pr}\widehat{v}_{ps}\zeta_{ps}-\sum_{k\in\mathcal{K}}B_k\widehat{c}_k\varphi_k$$
$$-\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\sum_{k\in\mathcal{K}}M_{prk}\widehat{c}_k\iota_{prk}+\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p}\sum_{r\in\mathcal{R}}\alpha_{ps}\delta_{pr}\,D_{pr}\,\widehat{v}_{ps}\sigma_{ps}$$
$$+\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\left(D_{pr}+D'_{pr}\right)\varepsilon_{rp}-\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}_p}Q_{pt}\tau_{pt}-\sum_{d\in\mathcal{D}}C_d\pi_d \tag{4.29}$$

subject to

$$\lambda_{rp}-\theta_{pk}-\beta_p\,\varphi_k-\iota_{prk}\leq G_{prk} \qquad \forall\,p\in\mathcal{P},\ r\in\mathcal{R},\ k\in\mathcal{K}, \tag{4.30}$$

$$\theta_{pk}+\zeta_{ps}\leq G_{pks} \qquad \forall\,p\in\mathcal{P},\ k\in\mathcal{K},\ s\in\mathcal{S}_p, \tag{4.31}$$

$$\sigma_{ps}+\mu_{pd}\leq G_{psd} \qquad \forall\,p\in\mathcal{P},\ s\in\mathcal{S}_p,\ d\in\mathcal{D}, \tag{4.32}$$

$$\mu_{pd}-\tau_{pt}\leq G_{ptd} \qquad \forall\,p\in\mathcal{P},\ t\in\mathcal{T}_p,\ d\in\mathcal{D}, \tag{4.33}$$

$$\varepsilon_{rp}-\mu_{pd}-\gamma_p\,\pi_d\leq G_{pdr} \qquad \forall\,p\in\mathcal{P},\ d\in\mathcal{D},\ r\in\mathcal{R}, \tag{4.34}$$

$$\varphi_k,\ \iota_{prk},\ \tau_{pt},\ \pi_d\geq 0 \qquad \forall\,p\in\mathcal{P},\ d\in\mathcal{D},\ k\in\mathcal{K},\ t\in\mathcal{T}_p, \tag{4.35}$$

$$\zeta_{ps},\ \theta_{pk},\ \lambda_{rp},\ \sigma_{ps},\ \mu_{pd},\ \varepsilon_{rp}\text{ - free variables} \qquad \forall\,p\in\mathcal{P},\ d\in\mathcal{D},\ k\in\mathcal{K},\ r\in\mathcal{R},\ s\in\mathcal{S}_p. \tag{4.36}$$

We note that the **DBSP(·)** is separable in terms of the dual variables associated with the reverse flows ($\boldsymbol{\iota}, \boldsymbol{\lambda}, \boldsymbol{\varphi}, \boldsymbol{\theta},$ and $\boldsymbol{\zeta}$), and the dual variables associated with the forward flows ($\boldsymbol{\varepsilon}, \boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\sigma},$ and $\boldsymbol{\tau}$).

IV.3.1.2. Forward Dual Subproblem

The forward dual subproblem, $\mathbf{FDBSP}(\boldsymbol{\varepsilon}, \boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\tau} \mid \widehat{\mathbf{v}})$, is given by

$$\max \quad \sum_{p\in\mathcal{P}} \sum_{s\in\mathcal{S}_p} \sum_{r\in\mathcal{R}} \alpha_{ps}\delta_{pr}\, D_{pr}\, \widehat{\upsilon}_{ps}\sigma_{ps} + \sum_{p\in\mathcal{P}} \sum_{r\in\mathcal{R}} \left(D_{pr} + D'_{pr}\right) \varepsilon_{rp}$$

$$- \sum_{p\in\mathcal{P}} \sum_{t\in\mathcal{T}_p} Q_{pt}\tau_{pt} - \sum_{d\in\mathcal{D}} C_d\pi_d \tag{4.37}$$

subject to (4.32)-(4.34)

$$\sigma_{ps},\, \mu_{pd},\, \varepsilon_{rp} \text{ - free variables, and } \tau_{pt},\, \pi_d \geq 0 \qquad \forall\, p \in \mathcal{P},\, d \in \mathcal{D},\, r \in \mathcal{R},\, s \in \mathcal{S}_p,\, t \in \mathcal{T}_p. \tag{4.38}$$

IV.3.1.3. Reverse Dual Subproblem

The reverse dual subproblem, $\mathbf{RDBSP}(\boldsymbol{\iota}, \boldsymbol{\lambda}, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\zeta} \mid \widehat{\mathbf{c}}, \widehat{\mathbf{v}})$, is given by

$$\max \quad \sum_{p\in\mathcal{P}} \sum_{r\in\mathcal{R}} \delta_{pr} D_{pr}\lambda_{rp} + \sum_{p\in\mathcal{P}} \sum_{s\in\mathcal{S}_p} \sum_{r\in\mathcal{R}} \delta_{pr} D_{pr}\widehat{\upsilon}_{ps}\zeta_{ps}$$

$$- \sum_{k\in\mathcal{K}} B_k\widehat{c}_k\varphi_k - \sum_{p\in\mathcal{P}} \sum_{r\in\mathcal{R}} \sum_{k\in\mathcal{K}} M_{prk}\widehat{c}_k\iota_{prk} \tag{4.39}$$

subject to (4.30) and (4.31)

$$\zeta_{ps},\, \theta_{pk},\, \lambda_{rp} \text{ - free variables, and } \iota_{prk},\, \varphi_k \geq 0 \qquad \forall\, p \in \mathcal{P},\, k \in \mathcal{K},\, r \in \mathcal{R},\, s \in \mathcal{S}_p. \tag{4.40}$$

IV.3.1.4. Solving the Dual Subproblems

The forward and reverse dual subproblems can be solved using a standard LP solver to obtain the corresponding optimal solutions. However, these problems have inherent degeneracy, which results in alternate optimal solutions. Within the set of alternate optimal solutions, some solutions dominate the rest. We utilize the *two phase* solution approach presented in Section III.3.2 to identify the *dominating* optimal solutions, and we add the corresponding *strong cuts* to the Benders master problem. As mentioned earlier, such strong cuts increase the algorithm efficiency by facilitating better lower bounds as shown in various network design problem settings in (Magnanti and Wong, 1981; Van Roy, 1986; Wentges, 1996).

Solving the Forward Subproblem FDBSP(·) - _Phase I_

In the first phase, we reduce the problem size in order to effectively find an optimal solution to the problem. We achieve this by considering only the constraints (4.32) for which the associated $\widehat{v}_{\bar{p}s}$, $p \in \mathcal{P}, s \in \mathcal{S}_p$ value is equal to 1. Recall that for product p, the index $s^{[p]}$ represents the RPP location $s \in \mathcal{S}_p$ whose $\widehat{v}_{ps}$ value is equal to 1. Moreover, we consider only the dual variables $\sigma_{ps^{[p]}}$, $p \in \mathcal{P}$ to obtain the following first phase problem.

$$\begin{aligned} \max \quad & \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} \alpha_{ps^{[p]}} \delta_{pr} \, D_{pr} \, \sigma_{ps^{[p]}} + \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} \left(D_{pr} + D'_{pr} \right) \varepsilon_{rp} \\ & - \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}_p} Q_{pt} \tau_{pt} - \sum_{d \in \mathcal{D}} C_d \pi_d \end{aligned} \tag{4.41}$$

subject to

$$\sigma_{ps^{[p]}} + \mu_{pd} \leq G_{ps^{[p]}d} \qquad \forall \, p \in \mathcal{P}, \; d \in \mathcal{D}, \tag{4.42}$$

$$\mu_{pd} - \tau_{pt} \leq G_{ptd} \qquad \forall \, p \in \mathcal{P}, \; t \in \mathcal{T}_p, \; d \in \mathcal{D}, \tag{4.43}$$

$$\varepsilon_{rp} - \mu_{pd} - \gamma_p \, \pi_d \leq G_{pdr} \qquad \forall \, p \in \mathcal{P}, \; d \in \mathcal{D}, \; r \in \mathcal{R}, \tag{4.44}$$

$$\sigma_{ps^{[p]}}, \; \mu_{pd}, \; \varepsilon_{rp} \text{ - free variables, and } \tau_{pt}, \; \pi_d \geq 0 \qquad \forall \, p \in \mathcal{P}, \; d \in \mathcal{D}, \; r \in \mathcal{R}, \; t \in \mathcal{T}_p. \tag{4.45}$$

Solving the Forward Subproblem **FDBSP(·)** ***- Phase II***

To find a dominating solution in the second phase, we compute the values of the σ_{ps} variables that were eliminated in the first phase. We use the following formulation to obtain a dominating solution for the forward subproblem.

$$\max \quad \sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{R}} \alpha_{ps}\delta_{pr}\, D_{pr}\, \sigma_{ps} \tag{4.46}$$

$$\text{subject to} \qquad (4.32)\text{–}(4.34), \text{ and } (4.38).$$

In the above problem, we fix the values of the dual variables considered in the previous phase to their respective optimal values. This ensures feasibility of the dominating dual solution. Examining the structure of the above problem, we observe that the optimal values for the remaining σ_{ps} variables can be computed as follows.

$$\sigma_{ps} = \min_{d\in\mathcal{D}}\{G_{psd} - \mu_{pd}\} \qquad\qquad \forall\, p \in \mathcal{P},\; s \in \mathcal{S}_p\backslash\{s^{[p]}\}. \tag{4.47}$$

Solving the Reverse Subproblem **RDBSP(·)** ***- Phase I***

Similarly to the forward phase I problem, we consider only the constraints and dual variables for which the associated variables $v_{ps}, p \in \mathcal{P}, s \in \mathcal{S}_p$ and $c_k, k \in \mathcal{K}$ are equal to 1. Recall that we use $\mathcal{K}^u$ to represent all of the CC locations $k \in \mathcal{K}$ whose c_k value is equal to 1. The first phase reverse subproblem is

$$\begin{aligned} \max \quad & \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} \delta_{pr}D_{pr}\lambda_{rp} + \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} \delta_{pr}D_{pr}\zeta_{ps^{[p]}} - \sum_{k\in\mathcal{K}^u} B_k\widehat{c}_k\varphi_k \\ & - \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\sum_{k\in\mathcal{K}^u} M_{prk}\widehat{c}_k\iota_{prk} \end{aligned} \tag{4.48}$$

subject to

$$\lambda_{rp} - \theta_{pk} - \beta_p \varphi_k - \iota_{prk} \leq G_{prk} \qquad \forall\, p \in \mathcal{P},\, r \in \mathcal{R},\, k \in \mathcal{K}^u, \quad (4.49)$$

$$\theta_{pk} + \zeta_{ps^{[p]}} \leq G_{pks^{[p]}} \qquad \forall\, p \in \mathcal{P},\, k \in \mathcal{K}^u, \quad (4.50)$$

$$\zeta_{ps^{[p]}},\, \theta_{pk},\, \lambda_{rp} \text{ - free variables, and } \iota_{prk},\, \varphi_k \geq 0 \qquad \forall\, p \in \mathcal{P},\, k \in \mathcal{K}^u,\, r \in \mathcal{R}. \quad (4.51)$$

Solving the Reverse Subproblem* RDBSP(·) - *Phase II

In the second phase, we compute the values for the ζ_{ps}, ι_{prk}, θ_{pk}, and φ_k variables that were not considered in the first phase. We ensure feasibility by fixing the variables considered in the previous phase to their respective optimal values. The second phase reverse subproblem is given by

$$\max \quad \sum_{p\in\mathcal{P}} \sum_{s\in\mathcal{S}_p} \sum_{r\in\mathcal{R}} \delta_{pr} D_{pr} \zeta_{ps} - \sum_{k\in\mathcal{K}} B_k \varphi_k - \sum_{p\in\mathcal{P}} \sum_{r\in\mathcal{R}} \sum_{k\in\mathcal{K}} M_{prk} \iota_{prk} \quad (4.52)$$

subject to (4.30), (4.31) and (4.40).

IV.3.2. Benders Master Problem

The Benders master problem $\mathbf{BMP}(\mathbf{v},\mathbf{c}|\widehat{\boldsymbol{\varepsilon}},\widehat{\boldsymbol{\lambda}},\widehat{\boldsymbol{\iota}},\widehat{\boldsymbol{\varphi}},\widehat{\boldsymbol{\pi}},\widehat{\boldsymbol{\sigma}},\widehat{\boldsymbol{\tau}},\widehat{\boldsymbol{\zeta}})$ solves the binary decision variables $\mathbf{v}$ and $\mathbf{c}$. In order to incorporate the Benders cuts in the master problem, we can combine the solutions provided by the two dual subproblems $\mathbf{RDBSP}(\boldsymbol{\lambda},\boldsymbol{\iota},\boldsymbol{\varphi},\boldsymbol{\theta},\boldsymbol{\zeta}\ |\widehat{\mathbf{c}},\widehat{\mathbf{v}})$ and $\mathbf{FDBSP}(\boldsymbol{\varepsilon},\boldsymbol{\mu},\boldsymbol{\pi},\boldsymbol{\sigma},\boldsymbol{\tau}\ |\widehat{\mathbf{v}})$ and obtain a *single* cut. Alternatively, we can follow the the *flow separation scheme* described in Section III.3.3, and obtain *two* cuts, namely, the *forward* and *reverse* cuts, corresponding to the solutions of the forward and reverse dual subproblems, respectively.

Our preliminary experimentation reveals a superior performance of the flow separation scheme, both in terms of lower bound quality and solution runtimes. Hence, for brevity, we report only the computational performance of the flow separation scheme in Section IV.4. To this end, we define two continuous variables ψ^F and ψ^R corresponding to the forward and reverse cuts, respectively. Then, the $\mathbf{BMP(\cdot)}$ problem is given by

$$\min \quad \sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p} F_{ps}v_{ps} + \sum_{k\in\mathcal{K}} F'_k c_k + \psi^F + \psi^R \tag{4.53}$$

subject to

$$\sum_{s\in\mathcal{S}_p} v_{ps} = 1 \quad \forall\, p \in \mathcal{P}, \tag{4.54}$$

$$\sum_{k\in\mathcal{K}} B_k\, c_k - \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} \beta_p\, \delta_{pr}\, D_{pr} \le 0 \tag{4.55}$$

$$\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} \delta_{pr} D_{pr} \widehat{\lambda}_{rp} + \sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p}\sum_{r\in\mathcal{R}} \delta_{pr} D_{pr} \widehat{\zeta}_{ps} v_{ps}$$

$$- \sum_{k\in\mathcal{K}} B_k \widehat{\varphi}_k c_k - \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\sum_{k\in\mathcal{K}} M_{prk} \widehat{\iota}_{prk} c_k \le \psi^R \tag{4.56}$$

$$\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}_p}\sum_{r\in\mathcal{R}} \alpha_{ps}\delta_{pr}\, D_{pr}\, \widehat{\sigma}_{ps} v_{ps} + \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} \left(D_{pr} + D'_{pr}\right) \widehat{\varepsilon}_{rp}$$

$$- \sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}_p} Q_{pt} \widehat{\tau}_{pt} - \sum_{d\in\mathcal{D}} C_d \widehat{\pi}_d \le \psi^F \tag{4.57}$$

$$v_{ps}, c_k \in \{0,1\}\ \forall\, p \in \mathcal{P},\ s \in \mathcal{S}_p,\ \text{and}\ \ \psi^F, \psi^R \ge 0. \tag{4.58}$$

Constraint set (4.55) is a surrogate constraint that ensures the availability of minimum aggregate capacity at the CC locations. Specifically, it ensures that the $\mathbf{BMP}(\cdot)$ configures the network with sufficient collection capacity to accommodate the reverse flow of products from the retailers. Thus, the constraint (4.55) ensures the feasibility of the $\mathbf{BSP}(\cdot)$ with respect to constraint set (4.21), which in turn, characterizes the boundedness of the dual subproblem. Constraint sets (4.56) and (4.57) represent the collection of forward and reverse cuts added to the master problem at each iteration. Constraint set (4.58) gives the restriction on the decision variables.

IV.3.3. Heuristic-Enhanced Benders Decomposition

Our objective in considering the BD framework is two-fold. First, the BD yields lower bounds for the formulation **MP**, which can be used to evaluate the quality of the solutions provided by the heuristic methods described in the previous section. Second, especially for larger instances, combining the heuristics and the BD framework promotes faster convergence as we illustrate in our computational results in Section IV.4.

In a typical BD implementation, we initially solve the master problem without any dual Benders cuts. Following this, at any given iteration, we use the values of master problem variables to solve the dual subproblems, and derive the Benders cuts using the optimal dual subproblem solution values. We add these cuts to the master problem and continue the iterative procedure until a preset stopping condition is realized. In general, the aforementioned implementation requires multiple iterations to significantly improve the upper and lower bounds. To improve the computational efficiency of the typical implementation, we combine the BD framework with tabu search heuristics. More specifically, we use the solution obtained from a heuristic method in the first subproblem (instead of initially solving the **BMP** $(\cdot)$ without any cuts as in a typical implementation) to generate the initial set of Benders cuts. For this, we can use any of the heuristic methods developed in Section IV.2; however, based on the relative performances reported in Section IV.4, we suggest the SNS-TH heuristic for integration with the BD framework. The use of heuristic solutions to derive initial Benders cuts in the context of the *capacitated facility location problem* (**CFLP**) was introduced in Wentges (1996).

We outline the overall framework HBD in Display 7. We use the notation `Tol`, `BDMaxItr`, `ItrNo`, `UB`, and `LB` to represent the stopping tolerance, maximum number of iterations, iteration counter, upper bound, and lower bound, respectively. Furthermore, Z(**BMP**) and Z(**DBSP**) represent the objective function values for the **BMP** $(\cdot)$ and **DBSP** $(\cdot)$. In HBD, we first employ the SNS-TH heuristic to obtain a good feasible solution. Recall that the solution vectors $\mathcal{S}^u$ and $\mathcal{K}^u$ of the heuristics represent the RPP and CC location decisions, respectively. We extract the values of the location decision variables from the solution vectors and adopt them to solve the forward and reverse dual subproblems using the method outlined in Section IV.3.1.4. Then, we add the obtained forward and reverse cuts to the master problem **BMP** $(\cdot)$ which we solve to obtain a lower bound. Once the initial Benders cuts are obtained in this fashion, we discontinue the use of heuristics in successive iterations.

The HBD framework significantly improves the computational efficiency of the typical implementation. Furthermore, the initial set of cuts generated using the heuristic solution provides substantial improvements to the lower bound of the typical implementation. We next summarize the computational results of the HBD approach for large scale problem instances.

Display 7 Procedure HBD()

Initialize input parameters `Tol`, `BDMaxItr`

Initialize $(\mathcal{S}^u, \mathcal{K}^u)_{best}$ = SNS-TH(), and obtain $\widehat{\mathbf{v}}, \widehat{\mathbf{c}}$ using $(\mathcal{S}^u, \mathcal{K}^u)_{best}$

Set $\texttt{UB} = Z_{best}$ and $\texttt{ItrNo} = 0$

Solve the $\mathbf{DBSP}(\boldsymbol{\varepsilon}, \boldsymbol{\iota}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\varphi}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\theta}, \boldsymbol{\zeta}\ |\widehat{\mathbf{c}}, \widehat{\mathbf{v}})$ to obtain $\widehat{\boldsymbol{\varepsilon}}, \widehat{\boldsymbol{\iota}}, \widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\varphi}}, \widehat{\boldsymbol{\sigma}}, \widehat{\boldsymbol{\tau}}, \widehat{\boldsymbol{\zeta}}$ and the $Z(\mathbf{DBSP})$

Add forward and reverse strong cuts to $\mathbf{BMP}(\cdot)$ using $\widehat{\boldsymbol{\varepsilon}}, \widehat{\boldsymbol{\iota}}, \widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\varphi}}, \widehat{\boldsymbol{\sigma}}, \widehat{\boldsymbol{\tau}}, \widehat{\boldsymbol{\zeta}}$

Solve $\mathbf{BMP}(\mathbf{v}, \mathbf{c}|\widehat{\boldsymbol{\varepsilon}}, \widehat{\boldsymbol{\iota}}, \widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\varphi}}, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\sigma}}, \widehat{\boldsymbol{\tau}}, \widehat{\boldsymbol{\zeta}})$, and set $\texttt{LB} = Z(\mathbf{BMP})$

while $(\texttt{UB} - \texttt{LB})/\texttt{LB} > \texttt{Tol}$ and $\texttt{ItrNo} \le \texttt{BDMaxItr}$ **do**

  `ItrNo`=`ItrNo`+1

  Solve the $\mathbf{DBSP}(\boldsymbol{\varepsilon}, \boldsymbol{\iota}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\varphi}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\theta}, \boldsymbol{\zeta}\ |\widehat{\mathbf{c}}, \widehat{\mathbf{v}})$ to obtain $\widehat{\boldsymbol{\varepsilon}}, \widehat{\boldsymbol{\iota}}, \widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\varphi}}, \widehat{\boldsymbol{\sigma}}, \widehat{\boldsymbol{\tau}}, \widehat{\boldsymbol{\zeta}}$ and the $Z(\mathbf{DBSP})$

  Calculate $\texttt{UB}$=Z($\mathbf{BMP}$)+Z($\mathbf{DBSP}$)$-\psi^F - \psi^R$

  if $\texttt{UB} < Z_{best}$ **then**

    $Z_{best} = \texttt{UB}$

  end if

  Add forward and reverse strong cuts to $\mathbf{BMP}(\cdot)$ using $\widehat{\boldsymbol{\varepsilon}}, \widehat{\boldsymbol{\iota}}, \widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\varphi}}, \widehat{\boldsymbol{\sigma}}, \widehat{\boldsymbol{\tau}}, \widehat{\boldsymbol{\zeta}}$

  Solve $\mathbf{BMP}(\mathbf{v}, \mathbf{c}|\widehat{\boldsymbol{\varepsilon}}, \widehat{\boldsymbol{\iota}}, \widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\varphi}}, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\sigma}}, \widehat{\boldsymbol{\tau}}, \widehat{\boldsymbol{\zeta}})$, and set $\texttt{LB} = Z(\mathbf{BMP})$

end while

Find $\mathbf{u}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}$ corresponding to $\widehat{\mathbf{v}}, \widehat{\mathbf{c}}$ (e.g. solve $\mathbf{BSP}(\mathbf{u,w,x,y,z}\ |\widehat{\mathbf{c}}, \widehat{\mathbf{v}})$)

Return $\widehat{\mathbf{c}}, \widehat{\mathbf{u}}, \widehat{\mathbf{v}}, \widehat{\mathbf{w}}, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \widehat{\mathbf{z}}$ and the Z_{best}.

IV.4. Computational Experiments

In this section, we first develop a testbed of random data instances and conduct a computational study to establish the performance of the proposed solution methodologies. Since our problem setting is a generalization of the **URP**, while generating our testbed, we utilize a similar approach as the one given in Section III.4. To benchmark the performance of our heuristics, we use the B&C approach for small scale instances and the HBD approach for large scale instances. We employ the B&C implementation in CPLEX with default settings for cut generation, preprocessing, and upper bound heuristics. The default cut generation includes clique, cover, disjunctive, flow cover, flow path, Gomory fractional, generalized upper bound cover, implied bound, and mixed integer rounding cuts. We also employ CPLEX to solve the linear program mentioned in Section IV.2.2, the forward and the reverse dual subproblems mentioned in Section IV.3.1.4, and the Benders master problem presented in Section IV.3.2. We implement the solution approaches using the C++ programming language and perform the runs on a machine with a 3 GHz Intel XEON processor and 6 GB RAM.

We set the construction heuristics parameter `CTR` to 50. For the tabu search, we set parameters `ENL`, `MNL`, and $\texttt{RNL}_\texttt{p}$ to 3, the parameter `MaxIter` to 20, and use a fixed length tabu lists of size 5. We set the parameter `ITNLOOP` to 10.

IV.4.1. Random Test Instance Generation

We generate test instances under two data settings (Setting I - Small instances and Setting II - Large instances) by altering the number of products $|\mathcal{P}|$, the number of retailers $|\mathcal{R}|$, and the number of potential CC locations $|\mathcal{K}|$, as shown in Table 8. We create 10 random instances for each class. We use uniform distributions to randomly create the number of NPP and potential RPP locations for each product p, and calculate the number of DC locations proportional to the number of retailer locations as shown in Table 2. Uniform distributions are also employed, as shown in Table 2, to generate the demands (D_{pr} and D'_{pr}), the return fractions (δ_{pr}), the recovery fractions (α_{ps}). Also as shown in Table 9, using uniform distributions, we randomly generate the storage capacity coefficients (γ_p and β_p) and storage and processing capacities C_d, B_k, Q_{pt} for the DCs, the CCs, and the NPPs, respectively. Note that we use the notation TD_p and TR_p, $\forall p \in \mathcal{P}$, to represent the total demand quantity given by $\sum_{r\in\mathcal{R}}(D_{pr} + D'_{pr})$ and the total return quantity given by $\sum_{r\in\mathcal{R}} \delta_{pr} D_{pr}$, respectively.

Table 8 **CRP**: Problem Classes Used in Computational Testing.

Setting I - Small Instances				Setting II - Large Instances			
Class	$\|\mathcal{P}\|$	$\|\mathcal{R}\|$	$\|\mathcal{K}\|$	Class	$\|\mathcal{P}\|$	$\|\mathcal{R}\|$	$\|\mathcal{K}\|$
CS1	5	60	25	CL1	5	240	25
CS2	5	60	35	CL2	5	240	35
CS3	5	90	25	CL3	5	300	25
CS4	5	90	35	CL4	5	300	35
CS5	5	120	25	CL5	5	360	25
CS6	5	120	35	CL6	5	360	35
CS7	10	60	25	CL7	10	240	25
CS8	10	60	35	CL8	10	240	35
CS9	10	90	25	CL9	10	300	25
CS10	10	90	35	CL10	10	300	35
CS11	10	120	25	CL11	10	360	25
CS12	10	120	35	CL12	10	360	35

IV.4.2. Computational Results

We first summarize the computational results of the two tabu search based meta-heuristics (i.e., SNS, RNS, SNS-TH, and RNS-TH) for Settings I and II. As noted earlier, in the SNS and RNS implementations, we use CPLEX to solve the $\mathbf{SP}(\mathbf{u}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}|\mathcal{S}^u, \mathcal{K}^u)$ whereas in SNS-TH and RNS-TH, we use the transportation heuristic for this purpose. Later, we provide our results for our HBD framework which also incorporates a heuristic to obtain an initial set of Benders cuts.

IV.4.2.1. Heuristic Results for Setting I - Small Instances

In addition to using the SNS, the RNS, the SNS-TH, and the RNS-TH heuristics, in order to obtain some benchmark results as well as to observe the performance of the B&C method, we also solve each problem instance using CPLEX. To this end, we observe that reducing the optimality gap below 1% requires considerable

Table 9 **CRP**: Distributions for Product Capacity Coefficients and Storage Capacity Values.

Parameter	Distribution
γ_p	Uniform[1, 10]
β_p	Uniform[1, γ_p]
C_d	Uniform[0.1, 0.3] * $\sum_{p \in \mathcal{P}} \gamma_p TD_p$
Q_{pt}	Uniform[0.1, 0.4] * $\sum_{r \in \mathcal{R}} TD_p$
B_k	Uniform[0.1, 0.3] * $\sum_{p \in \mathcal{P}} \beta_p TR_p$

computational effort. Moreover, for some of the test instances, CPLEX takes an excessively long time to converge even to an optimality gap of 1.0%. Therefore, to avoid high computational times, we use a stopping criterion of a 1% optimality gap between the incumbent and the best lower bound or a runtime of 36000 sec, whichever comes first.

For each problem class, we summarize the average and the maximum optimality gaps in Table 10. We calculate the optimality gap for a method as $100(Z_{best} - LB)/LB$ where Z_{best} represents the objective value of the incumbent solution obtained using the corresponding method and LB represents the lower bound obtained upon the termination of the B&C method. In the tables presenting the computational results, note that, the row minimums for the heuristics results are listed in bold. We observe that the SNS heuristic is effective as it provides the lowest values for both the average and the maximum optimality gaps in most classes. The use of the transportation heuristic results in a slight increase in the optimality gaps. Also, the performance of the SNS-TH heuristic is superior to the RNS-TH heuristic.

In Table 11, we present a comparison of the solution runtimes for the B&C and the heuristic approaches. The results show that the SNS-TH heuristic performs better than the other approaches. We note that, in general, solution runtime increases

Table 10 **CRP**: Optimality Gaps for Setting I.

Setting I	Average Optimality Gap (%)					Maximum Optimality Gap (%)				
Classes	BC	SNS	RNS	SNS-TH	RNS-TH	BC	SNS	RNS	SNS-TH	RNS-TH
CS1	0.97	**1.32**	1.98	2.82	2.91	1.00	**2.07**	4.69	4.98	4.91
CS2	0.98	**1.79**	2.42	2.69	4.02	1.00	**3.19**	3.70	4.00	5.47
CS3	0.98	**2.02**	2.55	2.47	3.48	1.00	**3.70**	4.60	5.03	4.31
CS4	0.91	**2.10**	2.13	2.78	3.84	1.00	4.03	**3.80**	4.19	6.87
CS5	0.90	**1.88**	2.36	2.07	2.20	1.00	**3.16**	4.71	4.42	6.04
CS6	1.00	**1.63**	2.24	2.34	3.24	1.00	**2.86**	4.39	3.52	5.05
CS7	0.95	**1.55**	2.01	2.12	3.24	1.00	**2.94**	4.94	4.54	5.23
CS8	1.96	2.89	**2.85**	3.38	4.69	3.45	**3.29**	3.51	4.99	7.43
CS9	0.99	1.20	**1.15**	2.07	3.27	1.00	2.61	**1.87**	4.75	6.69
CS10	0.96	**1.49**	2.18	2.10	4.12	1.00	**2.23**	4.20	2.74	7.11
CS11	1.00	**1.46**	1.63	1.92	3.48	1.00	3.17	**2.58**	3.73	4.43
CS12	3.56	**3.86**	4.45	4.38	5.78	4.35	**5.49**	5.87	6.30	9.00
Average	1.26	**1.93**	2.32	2.60	3.69	1.48	**3.23**	4.07	4.43	6.04

with an increasing problem size, especially with increasing number of potential CC locations. However, it is worth noting that with increasing problem size, the B&C approach exhibits excessively high solution runtimes while the runtime increases in all of the heuristic implementations stay at modest levels. In Tables 10 and 11, we also observe that the solution times are significantly lower for heuristics in which solution goodness evaluations are performed using the transportation heuristic (SNS-TH and RNS-TH) while sacrificing solution quality only slightly.

IV.4.2.2. Heuristic Results for Setting II - Large Instances

In this case, we examine the heuristics based on the optimality gaps, the solution times, and the number of times they provide the best objective value (upper bound). We use the same algorithmic parameter values for the heuristics as mentioned before. For the HBD approach, we initialize the parameters `Tol` and `BDMaxItr` to 0.02 and 250, respectively. That is, in HBD, we perform 250 iterations unless the optimality gap reduces below 2%.

In Table 12, we compare the heuristics in terms of the average and the maximum

Table 11 **CRP**: Solution Runtimes for Setting I.

Setting I	Average Runtimes (Secs.)					Maximum Runtimes (Secs.)				
Classes	BC	SNS	RNS	SNS-TH	RNS-TH	BC	SNS	RNS	SNS-TH	RNS-TH
CS1	76	55	120	**4**	4	141	87	198	8	**6**
CS2	447	69	159	**7**	9	718	154	201	**10**	13
CS3	608	139	421	**13**	12	1091	205	653	20	**16**
CS4	2754	128	436	**23**	25	5044	232	665	**28**	33
CS5	602	109	131	**8**	10	1000	138	175	**11**	16
CS6	3197	124	235	**14**	15	7430	193	321	23	**21**
CS7	9917	270	607	**21**	22	31159	439	925	**29**	30
CS8	29084	363	833	**45**	46	36001	620	1376	**59**	60
CS9	1651	64	166	**9**	11	4364	150	267	20	**14**
CS10	11141	263	331	33	**30**	25041	932	759	69	**39**
CS11	15641	313	986	**36**	39	34395	498	1600	**46**	50
CS12	35259	472	1254	**59**	72	36002	526	2286	**71**	87
Average	9198	197	473	**23**	25	15199	348	786	33	**32**

optimality gaps in which the lower bounds are calculated using the HBD approach. The results show that, although the methods with exact objective evaluations (SNS and RNS) perform well, the solution qualities provided by the heuristics are, in general, comparable. When examined in detail by comparing the results with and without the transportation heuristic, we observe that the deterioration in the optimality gaps (both average and maximum) due to the use of the transportation heuristic decreases as the problem size increases. Thus, it appears that the use of heuristic cost evaluations is even more favorable for larger instances. This conclusion is further corroborated in the light of better solution runtimes for the SNS-TH and RNS-TH heuristics. Also in Table 12, we observe that the performance of the SNS-TH heuristic in terms of optimality gaps is better than the RNS-TH heuristic. Similarly, the SNS heuristic performs better than the RNS heuristic. Furthermore, for each class, we also compare the performance of the heuristics using the number of instances for which the best solution is provided. In our testbed, out of the 120 large instances, the SNS, RNS, SNS-TH, and RNS-TH heuristics find the best objective value for 99, 20, 11, and 1 instances, respectively. In general, the SNS heuristics outperforms all the other heuristics as indicated by the bold entries in Table 12.

Table 12 **CRP**: Optimality Gaps and the Number of Instances with the Best Objective Value for Setting II.

Setting II	Average Optimality Gap (%)				Maximum Optimality Gap (%)				No. of Instances with min Z_{best}			
Classes	SNS	RNS	SNS-TH	RNS-TH	SNS	RNS	SNS-TH	RNS-TH	SNS	RNS	SNS-TH	RNS-TH
CL1	**2.03**	2.26	2.31	3.77	**2.18**	3.62	3.62	4.80	**9**	3	2	0
CL2	**2.16**	2.43	2.45	4.33	**2.63**	3.08	3.40	6.08	**8**	1	1	0
CL3	**1.87**	1.96	2.24	3.56	**2.94**	3.09	3.24	5.54	**7**	2	1	0
CL4	**2.19**	2.33	2.62	3.82	**2.62**	3.16	3.40	4.66	**6**	2	2	0
CL5	**1.61**	2.13	2.25	3.16	**2.13**	3.51	3.52	3.75	**9**	3	1	0
CL6	**2.21**	2.39	2.80	3.95	**2.55**	3.29	3.72	5.31	**7**	3	0	0
CL7	**1.79**	1.91	2.11	2.95	**2.07**	2.62	2.75	4.50	**9**	1	0	0
CL8	**2.24**	2.34	2.80	3.83	**2.62**	2.96	3.47	4.26	**8**	1	1	0
CL9	**1.75**	1.91	2.21	2.88	**2.06**	2.14	2.81	3.60	**10**	1	1	0
CL10	**2.07**	2.33	2.39	3.87	**2.63**	2.84	3.48	4.61	**9**	0	1	0
CL11	**1.66**	1.75	2.00	2.96	**2.19**	2.34	2.48	3.52	**10**	2	0	0
CL12	**2.14**	2.21	2.39	3.28	2.57	**2.55**	2.89	4.62	**7**	1	1	1
Average	**1.98**	2.16	2.37	3.53	**2.43**	2.93	3.23	4.60				

Table 13, where the runtimes of the heuristics are summarized, shows that the RNS-TH heuristic outperforms the others in terms of both the average and the maximum values. Clearly, the use of the transportation heuristic results in a significant decrease in computational time. The SNS-TH shows a modest increase over RNS-TH in solution runtimes; however, as opposed to the RNS-TH, it finds better quality solutions with relatively smaller solution gaps. More specifically, it appears that the computational results justify the use of the transportation heuristic rather than optimum objective evaluations in terms of both runtime and solution quality.

IV.4.2.3. HBD Approach Results for Setting II - Large Instances

We present the computational performance of the HBD approach in Table 14. Next to the problem classes in the first two columns, we present the optimality gaps resulting upon the termination of the HBD algorithm. We report the average and the maximum *HBD runtimes* in the following two columns. We note that these runtimes do not

Table 13 **CRP**: Comparison of the Solution Runtimes for Setting II Instances.

Setting II Classes	Average Runtimes (Secs.)				Maximum Runtimes (Secs.)			
	SNS	RNS	SNS-TH	RNS-TH	SNS	RNS	SNS-TH	RNS-TH
CL1	257.30	793.13	89.00	**62.29**	308.95	1095.45	166.39	**108.63**
CL2	342.14	1339.30	150.87	**128.74**	444.18	2036.15	196.24	**170.31**
CL3	979.76	3565.78	277.44	**188.41**	1687.42	7475.97	436.87	**300.73**
CL4	1186.29	4610.46	398.96	**307.88**	1930.85	8395.83	521.98	**375.72**
CL5	360.40	1075.96	139.23	**103.39**	699.94	1758.78	174.30	**115.92**
CL6	701.81	1786.17	241.72	**191.29**	1198.86	3178.14	304.18	**241.58**
CL7	1292.18	4790.67	414.83	**272.50**	1967.61	8152.68	557.76	**300.38**
CL8	2386.18	8835.36	683.71	**441.03**	3322.27	13134.70	1001.77	**571.08**
CL9	534.06	1851.96	176.13	**141.15**	756.85	2918.83	204.13	**177.47**
CL10	870.95	2538.23	266.90	**230.27**	1300.64	4915.02	454.71	**410.69**
CL11	2003.72	6950.21	482.27	**402.71**	2695.54	11448.30	621.98	**468.08**
CL12	2706.61	10382.39	787.56	**645.65**	3673.72	13857.10	1011.52	**852.78**
Average	1135.12	4043.30	342.38	**259.61**	1665.57	6530.58	470.99	**341.11**

include the SNS-TH heuristic runtimes. In the last two columns, we report the average and maximum number of iterations performed by the HBD approach.

Recall that, in HBD, we first employ the SNS-TH heuristic to obtain an initial upper bound as well as the initial Benders cuts by solving the $\mathbf{BSP}(\cdot)$ using binary variables obtained via the heuristic solution. We observe that the optimality gaps for the SNS-TH heuristics (from the HBD lower bound), which are reported in Table 12, are very close to the preset stopping tolerance parameter `Tol` of the HBD. Hence, the runtimes in Table 14 essentially represent the time taken by the HBD approach to tighten the lower bounds.

Finally, we note that an identical BD approach with all of the cut enhancements, but without the initial use of a heuristic, was not able to produce any results in a reasonable time frame. Thus, our above results show that the use of heuristics in a BD framework improves the computational efficiency of the Benders implementation, thereby making the implementation a viable solution procedure for large-scale problem instances. Moreover, the enhanced framework is beneficial in providing good upper and lower bounds in a relatively short time span. The lower bounds thus obtained also present efficient means for evaluating the quality of the heuristic solutions.

Table 14 **CRP**: Computational Performance of the HBD Approach.

Setting II	Optimality Gap (%)		HBD Runtimes (sec.)		No. of Iterations	
Classes	Average	Maximum	Average	Maximum	Average	Maximum
CL1	1.99	2.00	143.74	186.83	95.6	114
CL2	2.32	2.70	1166.27	1470.50	238.2	250
CL3	1.98	2.00	517.50	1124.22	122.8	186
CL4	2.35	3.10	2648.40	3919.78	237.6	250
CL5	2.00	2.00	304.40	415.23	132.9	164
CL6	2.24	2.50	1553.46	3188.89	243.9	250
CL7	2.00	2.00	729.12	1911.13	125.0	202
CL8	2.46	2.91	3076.89	5040.80	235.1	250
CL9	1.99	2.00	337.92	843.41	125.3	226
CL10	2.41	2.83	1537.54	1793.27	243.6	250
CL11	1.99	2.00	680.58	917.94	112.5	143
CL12	2.23	2.57	3222.39	4680.16	237.6	250
Average	2.16	2.38	1326.52	5040.80	179.2	212

IV.5. Concluding Remarks

In this chapter, we consider the **CRP**, for which, we develop a mixed integer linear program formulation to optimally extend an existing forward channel in order to incorporate a reverse channel in the context of product reclamation through remanufacturing. We observe that the B&C implementation (CPLEX) requires very high computation times to solve the test instances. In order to find good feasible solutions, we develop tabu search based meta-heuristics that combine search procedures using three simple neighborhood functions. The heuristics are found to be effective in terms of finding good feasible solutions and are also efficient in terms of the computational time. Moreover, to evaluate the objective function value or goodness of a feasible solution, we devise a transportation heuristic that can be used effectively to replace an exact method for this purpose. The use of the transportation heuristic significantly reduces the computational times but results in a modest deterioration in the quality of the solutions. In addition, we also extend the BD framework presented in Chapter III to this problem setting. We suggest the use of a heuristic within this BD framework to obtain an enhanced approach HBD. We test our solution methods on a testbed that we develop under two data settings that correspond to small and large instances. Our computational results illustrate the superior performance of the heuristic algorithms as well as the integrated HBD approach. In general, the heuristics with sequential neighborhood search (SNS and SNS-TH) perform better than the heuristics with the random neighborhood search (RNS and RNS-TH), which resemble a more typical implementation of a tabu search framework. The value of the integrated HBD approach is two-fold. First, it provides good solutions with low optimality gaps in reasonable runtimes, and second, its lower bound provides an excellent means to measure the quality of the heuristic solutions for large instances.

CHAPTER V

A CLOSED-LOOP NETWORK DESIGN PROBLEM

In this problem setting, we generalize the **URP** and **CRP** settings by deciding on the locations of the forward channel facilities, i.e., we determine the optimal locations of the manufacturing/remanufacturing facilities, DCs and CCs. This setting is applicable for an OEM who wishes to establish a new CLSC network for managing multiple types of products. Under this setting, we coordinate the forward and reverse flows using capacitated hybrid centers (HCs) and product-specific hybrid plants (HPPs), which lead to a common infrastructure for managing the forward and reverse flows. The operational characteristics of the **CLP** setting is similar to the ones considered in the **URP** setting. More specifically, we first assume single-sourcing strategy for retailers assignments. That is, for the reverse flows, each retailer works with a single HC to send all the returned products, and similarly, for the forward flows, each retailer works with a single HC to receive all of its requirements. However, a retailer can be assigned to two different HCs, where each HC can manage the flows associated with different channels. Secondly, we require a single HPP for each type of product, as a consequence of which, each HC is assigned to exactly one product-specific HPP.

We note that the capacities at the HCs represent aggregate capacities that can be shared by all products. Thus, for the purpose of incorporating the non-uniformity in capacity usage, as before, we utilize product-specific coefficients as modifiers to one capacity use unit. Moreover, since we can estimate the required manufacturing/remanufacturing capacities for each product by using the estimated demand for products, return quantity, return and recovery fractions, and we furthermore assume a single HPP per product, we can identify the feasible HPP candidate sites for each product (and consider only these candidates) before attempting to solve any specific instance. Therefore, we do not consider any capacity limitation on the candidate HPP sites. It is worthwhile to note that the inclusion of capacities on HCs induces stronger relation among the forward and reverse flows associated with different types of products.

In the **CLP** setting, we are interested in determining the best locations of the HPPs and the HCs with respect to the known retailer locations, and the best flow of products in the CLSC network such that the total cost of location, processing and transportation is minimized.

V.1. Problem Formulation

We next give the additional notation and the mathematical formulation that is referred henceforth as $\mathbf{MP-CL}$. Figure 4 depicts the underlying network structure with the flow, assignment and location variables in the CLSC network.

Additional Sets and Indices

$\mathcal{H}$ set of candidate HC locations, $h \in \mathcal{H}$.

$\mathcal{M}_p$ set of candidate HPP locations for $p \in \mathcal{P}, m \in \mathcal{M}_p$.

Additional Parameters

F'_h fixed cost of opening a HC at location $h \in \mathcal{H}$.

F_{pm} fixed cost of opening a HPP for product $p \in \mathcal{P}$ at location $m \in \mathcal{M}_p$.

η_{ph} unit distribution processing cost of product $p \in \mathcal{P}$ at HC $h \in \mathcal{H}$.

κ_{ph} unit collection processing cost of product $p \in \mathcal{P}$ at HC $h \in \mathcal{H}$.

ν_{pm} unit manufacturing cost of product $p \in \mathcal{P}$ shipped out of HPP $m \in \mathcal{M}_p$.

ρ_{pm} unit remanufacturing cost of product $p \in \mathcal{P}$ shipped out of HPP $m \in \mathcal{M}_p$.

Q_h aggregate processing/storage capacity at HC $h \in \mathcal{H}$.

α_{pm} recovery fraction for product $p \in \mathcal{P}$ at HPP $m \in \mathcal{M}_p$.

Decision Variables

b_h 1 if HC $h \in \mathcal{H}$ is opened, 0 otherwise.

g_{pm} 1 if HPP $m \in \mathcal{M}_p$ is used for product $p \in \mathcal{P}$, 0 otherwise.

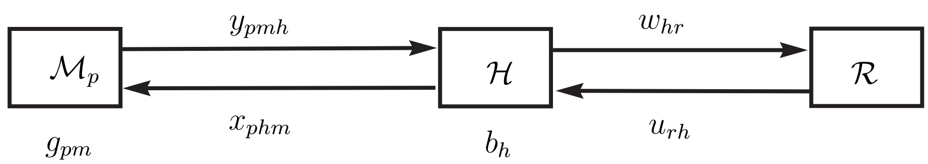
u_{rh} 1 if retailer $r \in \mathcal{R}$ is assigned to HC $h \in \mathcal{H}$
for the reverse flow of products, 0 otherwise.

w_{hr} 1 if retailer $r \in \mathcal{R}$ is assigned to HC $h \in \mathcal{H}$
for the forward flow of products, 0 otherwise.

x_{phm} quantity of product $p \in \mathcal{P}$ shipped from HC $h \in \mathcal{H}$ to HPP $m \in \mathcal{M}_p$.

y_{pmh} total quantity of new and remanufactured product $p \in \mathcal{P}$
shipped from HPP $m \in \mathcal{M}_p$ to HC $h \in \mathcal{H}$.

Figure 4 **CLP**: Underlying Structure of the CLSC Network.



Objective Function

$$
\begin{aligned}
\min \quad & \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p} F_{pm}g_{pm} + \sum_{h\in\mathcal{H}} F'_h b_h + \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\sum_{h\in\mathcal{H}} (G_{rh}+\kappa_{ph})\delta_{pr}D_{pr}u_{rh} \\
+ \; & \sum_{p\in\mathcal{P}}\sum_{h\in\mathcal{H}}\sum_{m\in\mathcal{M}_p} (G_{hm}+\alpha_{pm}\rho_{pm})x_{phm} + \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p}\sum_{h\in\mathcal{H}} \nu_{pm}(y_{pmh}-\alpha_{pm}x_{phm}) \\
+ \; & \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p}\sum_{H\in\mathcal{H}} G_{mh}y_{pmh} + \sum_{p\in\mathcal{P}}\sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}} (G_{hr}+\eta_{ph})\left(D_{pr}+D'_{pr}\right)w_{hr} \qquad (5.1)
\end{aligned}
$$

The first two terms in the objective function represent the fixed costs associated with locating the product-specific HPPs and HCs, respectively. The third term represents the transportation costs from the retailers and collection processing costs at the HCs. The fourth term represents the transportation costs from the HCs to the HPPs, in addition to the remanufacturing costs at the HPPs. Notice that, for each product p, $\sum_{h\in\mathcal{H}} y_{pmh}$ represents the total shipment, which contains both new and remanufactured products, from HPP m. Since we use all the remanufactured products to satisfy the retailer demand, for each product p, the expression $\sum_{h\in\mathcal{H}} y_{pmh} - \alpha_{pm}\sum_{h\in\mathcal{H}} x_{phm}$ represents the quantity of newly manufactured products at HPP m. We use this expression in the fifth term of the objective function to compute the total cost of manufacturing the new products. The sixth term represents the transportation costs from the HPPs to the HCs. The seventh term represents the transportation costs from the HCs to the retailers, in addition to the distribution processing costs at the

HCs. For brevity, we employ the notation G_{prh}, G_{phr}, G_{phm} and G_{pmh} to represent the sums $(G_{rh}+\kappa_{ph})$, $(G_{hr}+\eta_{ph})$, $(G_{hm}+\alpha_{pm}(\rho_{pm}-\nu_{pm}))$, and $(G_{mh}+\nu_{pm})$, respectively.

Constraints

$$\sum_{h\in\mathcal{H}} u_{rh} = 1 \qquad \forall\, r \in \mathcal{R}, \tag{5.2}$$

$$\sum_{m\in\mathcal{M}_p} g_{pm} = 1 \qquad \forall\, p \in \mathcal{P}, \tag{5.3}$$

$$\sum_{h\in\mathcal{H}} w_{hr} = 1 \qquad \forall\, r \in \mathcal{R}, \tag{5.4}$$

$$\sum_{m\in\mathcal{M}_p} x_{phm} = \sum_{r\in\mathcal{R}} \delta_{pr} D_{pr} u_{rh} \qquad \forall\, p \in \mathcal{P},\ h \in \mathcal{H}, \tag{5.5}$$

$$\sum_{h\in\mathcal{H}} x_{phm} = \sum_{r\in\mathcal{R}} \delta_{pr} D_{pr} g_{pm} \qquad \forall\, p \in \mathcal{P},\ m \in \mathcal{M}_p, \tag{5.6}$$

$$\sum_{h\in\mathcal{H}} y_{pmh} = \sum_{r\in\mathcal{R}} \left(D_{pr} + D'_{pr}\right) g_{pm} \qquad \forall\, p \in \mathcal{P},\ m \in \mathcal{M}_p, \tag{5.7}$$

$$\sum_{m\in\mathcal{M}_p} y_{pmh} = \sum_{r\in\mathcal{R}} \left(D_{pr} + D'_{pr}\right) w_{hr} \qquad \forall\, p \in \mathcal{P},\ h \in \mathcal{H}, \tag{5.8}$$

$$\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} \gamma_p \left(D_{pr} + D'_{pr}\right) w_{hr} + \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} \beta_p\, \delta_{pr} D_{pr} u_{rh} \le Q_h\, b_h \qquad \forall\, h \in \mathcal{H}, \tag{5.9}$$

$$x_{phm},\ y_{pmh} \ge 0 \qquad \forall\, p \in \mathcal{P},\ h \in \mathcal{H},\ m \in \mathcal{M}_p, \tag{5.10}$$

$$g_{pm},\ b_h,\ u_{rh},\ w_{hr} \in \{0, 1\} \qquad \forall\, p \in \mathcal{P},\ h \in \mathcal{H},\ m \in \mathcal{M}_p. \tag{5.11}$$

Constraint set (5.2) ensures that a retailer r is assigned to exactly one HC for the reverse flow of products. Constraint set (5.3) guarantees that, for each product p, a single dedicated HPP location m is established. Constraint set (5.4) ensures that a retailer r is assigned to exactly one HC for the forward flow of products. Constraint sets (5.5) and (5.6) represent the conservation (mass balance) of reverse flows at the HCs and HPPs, respectively. Moreover, for each product p, constraint set (5.6) ensures that all the returned products are sent to the open HPP. Constraint sets (5.7) and (5.8) represent the conservation of forward flows at the HPPs and HCs, respectively. We note that, for each product p, the coefficient $\sum_{r \in \mathcal{R}} \left(D_{pr} + D'_{pr} \right)$ represents the total demand at the retailer locations, and, because of the single HPP requirement, the total forward flow originating from an open HPP should be equal to this coefficient, and, hence, we use equality in the constraint sets (5.7) and (5.8). Constraint sets (5.6) and (5.7) forces the creation of a HPP m if a HC h has an associated forward or reverse flow for product p. Constraint set (5.9) forces the creation of a HC h if a retailer r has an associated forward or reverse flow for product p with that location. Moreover, constraint set (5.9) ensures that the total forward and reverse shipments at any HC does not exceed its aggregate processing capacity. Constraint sets (5.10) and (5.11) are the restrictions on the decision variables.

We note that, constraint sets (5.5) and (5.6) are identical to constraint sets (3.7) and (3.12), respectively, and they involve only the reverse flow variables. Furthermore, constraint sets (5.7) and (5.8) involve only the forward flow variables and are identical in structure to the constraint sets (3.7) and (3.12), respectively. This formulation relies on the assumption that there is a single, dedicated HPP for each product, and, it is very helpful for developing efficient solution algorithms to solve the subproblems in the BD framework similar to the one developed in Section III.3.2.

V.2. Solution Approach Using Benders Decomposition

In order to develop the components of this iterative framework, first we present the underlying Benders reformulation and state the subproblem. More specifically, the original problem can be restated as follows.

$$\begin{aligned} \min \quad Z &= \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p} F_{pm}g_{pm} + \sum_{h\in\mathcal{H}} F'_h b_h + \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\sum_{h\in\mathcal{H}} G_{prh}\delta_{pr}D_{pr}u_{rh} \\ &+ \sum_{p\in\mathcal{P}}\sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}} G_{phr}\left(D_{pr}+D'_{pr}\right)w_{hr} + \mathbf{BSP}(\mathbf{x},\ \mathbf{y}|\widehat{\mathbf{g}},\widehat{\mathbf{u}},\widehat{\mathbf{w}}) \end{aligned} \tag{5.12}$$

$$\text{subject to} \qquad (5.2), (5.3), (5.4), (5.9) \text{ and } (5.11).$$

where $\mathbf{BSP}(\mathbf{x},\ \mathbf{y}|\widehat{\mathbf{g}},\widehat{\mathbf{u}},\widehat{\mathbf{w}})$ represents the Benders subproblem whose formulation and solution procedure are discussed below.

V.2.1. Benders Subproblem

The subproblem $\mathbf{BSP}(\mathbf{x},\ \mathbf{y}|\widehat{\mathbf{g}},\widehat{\mathbf{u}},\widehat{\mathbf{w}})$ is essentially a minimization problem that determines the optimum values of the flow variables for fixed values of the location and assignment variables, and it can be stated as

$$\min \quad \sum_{p\in\mathcal{P}}\sum_{h\in\mathcal{H}}\sum_{m\in\mathcal{M}_p} G_{phm}x_{phm} + \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p}\sum_{H\in\mathcal{H}} G_{pmh}y_{pmh} \tag{5.13}$$

subject to

$$\sum_{m \in \mathcal{M}_p} x_{phm} \leq \sum_{r \in \mathcal{R}} \delta_{pr} D_{pr} \widehat{u}_{rh} \qquad \forall\, p \in \mathcal{P},\, h \in \mathcal{H}, \tag{5.14}$$

$$\sum_{h \in \mathcal{H}} x_{phm} \geq \sum_{r \in \mathcal{R}} \delta_{pr} D_{pr} \widehat{g}_{pm} \qquad \forall\, p \in \mathcal{P},\, m \in \mathcal{M}_p, \tag{5.15}$$

$$\sum_{h \in \mathcal{H}} y_{pmh} \geq \sum_{r \in \mathcal{R}} \left(D_{pr} + D'_{pr}\right) \widehat{g}_{pm} \qquad \forall\, p \in \mathcal{P},\, m \in \mathcal{M}_p, \tag{5.16}$$

$$\sum_{m \in \mathcal{M}_p} y_{pmh} \leq \sum_{r \in \mathcal{R}} \left(D_{pr} + D'_{pr}\right) \widehat{w}_{hr} \qquad \forall\, p \in \mathcal{P},\, h \in \mathcal{H}, \tag{5.17}$$

$$x_{phm},\, y_{pmh} \geq 0 \qquad \forall\, p \in \mathcal{P},\, h \in \mathcal{H},\, m \in \mathcal{M}_p, \tag{5.18}$$

We note that, without any effect on the final optimal solution, the equality constraints (5.5), (5.6), (5.7) and (5.8) are represented by the inequalities (5.14), (5.15), (5.16), and (5.17), respectivley. This alternative representation does not affect the solution space, but it does facilitate an easy exposition for the solution of the dual subproblem to generate Benders cuts as explained below.

V.2.1.1. Dual Subproblem

In order to generate Benders cuts for the master problem, we use the dual linear program of $\mathbf{BSP}(\mathbf{x},\, \mathbf{y}|\widehat{\mathbf{g}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$. For this purpose, we define dual variables μ_{ph}, λ_{pm}, τ_{pm}, and σ_{ph} corresponding to the constraints (5.14), (5.15), (5.16), and (5.17), respectively. The dual linear program $\mathbf{DBSP}(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\tau}\ |\widehat{\mathbf{g}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$ is stated as

$$\max \quad \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p}\sum_{r\in\mathcal{R}}\delta_{pr}D_{pr}\widehat{g}_{pm}\lambda_{pm} - \sum_{p\in\mathcal{P}}\sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}}\delta_{pr}D_{pr}\widehat{u}_{rh}\mu_{ph}$$
$$+ \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p}\sum_{r\in\mathcal{R}}\left(D_{pr}+D'_{pr}\right)\widehat{g}_{pm}\tau_{pm}$$
$$- \sum_{p\in\mathcal{P}}\sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}}\left(D_{pr}+D'_{pr}\right)\widehat{w}_{hr}\sigma_{ph} \tag{5.19}$$

subject to

$$\lambda_{pm} - \mu_{ph} \leq G_{phm} \qquad \forall\, p\in\mathcal{P},\ m\in\mathcal{M}_p,\ h\in\mathcal{H}, \tag{5.20}$$

$$\tau_{pm} - \sigma_{ph} \leq G_{pmh} \qquad \forall\, p\in\mathcal{P},\ m\in\mathcal{M}_p,\ h\in\mathcal{H}, \tag{5.21}$$

$$\lambda_{pm},\ \mu_{ph},\ \tau_{pm},\ \sigma_{ph} \geq 0 \qquad \forall\, p\in\mathcal{P},\ m\in\mathcal{M}_p,\ h\in\mathcal{H}. \tag{5.22}$$

We observe that the subproblem $\mathbf{BSP}(\mathbf{x},\,\mathbf{y}|\widehat{\mathbf{g}},\widehat{\mathbf{u}},\widehat{\mathbf{w}})$, and hence, the dual subproblem $\mathbf{DBSP}(\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{\tau}\ |\widehat{\mathbf{g}},\widehat{\mathbf{u}},\widehat{\mathbf{w}})$, are separable in terms of the forward flow variables, $\mathbf{y}$ and the reverse flow variables, $\mathbf{x}$. This separability is due to the requirement of a single, dedicated HPP per product. Furthermore, as in the case of the **URP**, we observe that the forward and reverse subproblems are separable for each product leading to *single product forward* and *single product reverse* subproblems. Next, we state the corresponding dual subproblems and their formulations.

V.2.1.2. Single Product Forward Dual Subproblem

For each product, the dual program of the single product forward subproblem, denoted by $\mathbf{FDBSP}_{\bar{p}}(\boldsymbol{\sigma},\boldsymbol{\tau}|\widehat{\mathbf{g}},\widehat{\mathbf{w}})$, is given by

$$\max \quad \sum_{m\in\mathcal{M}_{\bar{p}}}\sum_{r\in\mathcal{R}}\left(D_{\bar{p}r}+D'_{\bar{p}r}\right)\widehat{g}_{\bar{p}m}\tau_{\bar{p}m}-\sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}}\left(D_{\bar{p}r}+D'_{\bar{p}r}\right)\widehat{w}_{hr}\sigma_{\bar{p}h} \tag{5.23}$$

subject to

$$\tau_{\bar{p}m}-\sigma_{\bar{p}h}\leq G_{\bar{p}mh} \qquad \forall\, h\in\mathcal{H},\, m\in\mathcal{M}_{\bar{p}}, \tag{5.24}$$

$$\tau_{\bar{p}m},\, \sigma_{\bar{p}h}\geq 0 \qquad \forall\, h\in\mathcal{H},\, m\in\mathcal{M}_{\bar{p}}. \tag{5.25}$$

V.2.1.3. Single Product Reverse Dual Subproblem

For each product, the dual program of the single product reverse dual subproblem, denoted by $\mathbf{RDBSP}_{\bar{p}}(\boldsymbol{\lambda},\boldsymbol{\mu}|\widehat{\mathbf{g}},\widehat{\mathbf{u}})$, is given by

$$\max \quad \sum_{m\in\mathcal{M}_{\bar{p}}}\sum_{r\in\mathcal{R}}\delta_{\bar{p}r}D_{\bar{p}r}\widehat{g}_{\bar{p}m}\lambda_{\bar{p}m}-\sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}}\delta_{\bar{p}r}D_{\bar{p}r}\widehat{u}_{rh}\mu_{\bar{p}h} \tag{5.26}$$

subject to

$$\lambda_{\bar{p}m}-\mu_{\bar{p}h}\leq G_{\bar{p}hm} \qquad \forall\, h\in\mathcal{H},\, m\in\mathcal{M}_{\bar{p}}, \tag{5.27}$$

$$\lambda_{\bar{p}m},\, \mu_{\bar{p}h}\geq 0 \qquad \forall\, h\in\mathcal{H},\, m\in\mathcal{M}_{\bar{p}}. \tag{5.28}$$

V.2.2. Solving the Subproblems

Examining the dual subproblems $\mathbf{FDBSP}_{\bar{p}}(\boldsymbol{\sigma},\boldsymbol{\tau}|\widehat{\mathbf{g}},\widehat{\mathbf{w}})$ and $\mathbf{RDBSP}_{\bar{p}}(\boldsymbol{\lambda},\boldsymbol{\mu}|\widehat{\mathbf{g}},\widehat{\mathbf{u}})$, we can clearly see that they are identical in terms of their problem structure. Moreover, the single product reverse dual subproblem formulated in Section III.3.1.2 (for the **URP**) is identical to the one formulated in the previous section. As a consequence of these similarities, we can use the solution method developed in III.3.2.2 for solving the $\mathbf{FDBSP}_{\bar{p}}(\boldsymbol{\sigma},\boldsymbol{\tau}|\widehat{\mathbf{g}},\widehat{\mathbf{w}})$ and $\mathbf{RDBSP}_{\bar{p}}(\boldsymbol{\lambda},\boldsymbol{\mu}|\widehat{\mathbf{g}},\widehat{\mathbf{u}})$.

For each product $\bar{p}$, we use the index $m^{[\bar{p}]}$ to represent the HPP location $m\in\mathcal{M}_{\bar{p}}$ whose $\widehat{g}_{\bar{p}m}$ value is equal to 1. Also, we let $\mathcal{H}^u\subseteq\mathcal{H}$ to denote the set of HC locations that are open according to $\widehat{\mathbf{u}}$, i.e., $\mathcal{H}^u = \{h \in \mathcal{H} |\ \exists\, r \in \mathcal{R} :\ \ \widehat{u}_{rh} = 1\}$ and $\mathcal{H}^w \subseteq \mathcal{H}$ to denote the set of HC locations that are open according to $\widehat{\mathbf{w}}$, i.e., $\mathcal{H}^w = \{h \in \mathcal{H} |\ \exists\, r \in \mathcal{R} :\ \widehat{w}_{hr} = 1\}$.

The first phase optimal solution for the $\mathbf{FDBSP}_{\bar{p}}(\boldsymbol{\sigma}, \boldsymbol{\tau} | \widehat{\mathbf{g}}, \widehat{\mathbf{w}})$ is given by $\tau_{\bar{p}m[\bar{p}]} = \max_{h \in \mathcal{H}^w}\{G_{\bar{p}m[\bar{p}]h}\}$ and $\sigma_{\bar{p}h} = \tau_{\bar{p}m[\bar{p}]} - G_{\bar{p}m[\bar{p}]h}$ for all $h \in \mathcal{H}^w$. Then, the corresponding second phase problem is given by

$$\max \quad \sum_{m \in \mathcal{M}_{\bar{p}}} \sum_{r \in \mathcal{R}} \left(D_{\bar{p}r} + D'_{\bar{p}r}\right) \tau_{\bar{p}m} - \sum_{h \in \mathcal{H}} \sum_{r \in \mathcal{R}} \left(D_{\bar{p}r} + D'_{\bar{p}r}\right) \sigma_{\bar{p}h} \tag{5.29}$$

subject to (5.24) and (5.25).

In the above problem, we fix the values of the dual variables considered in the first phase to their respective optimal values.

Similarly to the $\mathbf{FDBSP}_{\bar{p}}(\boldsymbol{\sigma}, \boldsymbol{\tau} | \widehat{\mathbf{g}}, \widehat{\mathbf{w}})$, the first phase optimal solution for the $\mathbf{RDBSP}_{\bar{p}}(\boldsymbol{\lambda}, \boldsymbol{\mu}\, | \widehat{\mathbf{g}}, \widehat{\mathbf{u}})$ is given by $\lambda_{\bar{p}m[\bar{p}]} = \max_{h \in \mathcal{H}^u}\{G_{\bar{p}hm[\bar{p}]}\}$ and $\mu_{\bar{p}h} = \lambda_{\bar{p}m[\bar{p}]} - G_{\bar{p}hm[\bar{p}]}$ for all $h \in \mathcal{H}^u$. Then, we fix the values of these dual variables to their respective optimal values in the first phase and solve the corresponding second phase problem, given by

$$\max \quad \sum_{m \in \mathcal{M}_{\bar{p}}} \sum_{r \in \mathcal{R}} \delta_{\bar{p}r} D_{\bar{p}r} \sigma_{\bar{p}m} - \sum_{h \in \mathcal{H}} \sum_{r \in \mathcal{R}} \delta_{\bar{p}r} D_{\bar{p}r} \mu_{\bar{p}h} \tag{5.30}$$

subject to (5.27) and (5.28).

Using the optimal dual solutions of the single product forward subproblems, $\mathbf{FDBSP}_{\bar{p}}(\boldsymbol{\sigma}, \boldsymbol{\tau}\ | \widehat{\mathbf{g}}, \widehat{\mathbf{w}})$ for $\bar{p} \in \mathcal{P}$, we set $\tau_{pm} = \tau_{\bar{p}m}$ for all $p \in \mathcal{P}$ and $m \in \mathcal{M}_p$ and $\sigma_{ph} = \sigma_{\bar{p}h}$ for all $p \in \mathcal{P}$ and $h \in \mathcal{H}$. Similarly, using the optimal dual solutions of the single product reverse subproblems, $\mathbf{RDBSP}_{\bar{p}}(\boldsymbol{\lambda}, \boldsymbol{\mu}\ | \widehat{\mathbf{g}}, \widehat{\mathbf{u}})$ for $\bar{p} \in \mathcal{P}$, we set $\mu_{ph} = \mu_{\bar{p}h}$ for all $p \in \mathcal{P}$ and $h \in \mathcal{H}$ and $\lambda_{pm} = \lambda_{\bar{p}m}$ for all $p \in \mathcal{P}$ and $m \in \mathcal{M}_p$. Thus, we obtain an optimal solution of $\mathbf{DBSP}(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\tau}\ | \widehat{\mathbf{g}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$.

V.2.3. Benders Master Problem

We can use the alternative separation schemes, i.e., flow and product separation, for the overall subproblem and utilize the alternative representations of Benders cuts in the master problem (as in Section III.3.3).

Our preliminary experimentation reveals a superior performance of the separation schemes corresponding to Type 1 (flow as well as product separation) and Type 2 (product separation), both in terms of lower bound quality and solution runtimes. Hence, for brevity, we report only the computational performance of the Type 1 and Type 2 cuts, in Section V.4. To this end, we first state the master problem $\mathbf{BMP}(\mathbf{g}, \mathbf{b}|\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\sigma}}, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\tau}})$ using general cut related terms as follows.

$$\begin{aligned}\min \quad & \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p} F_{pm} g_{pm} + \sum_{h\in\mathcal{H}} F'_h b_h + \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\sum_{h\in\mathcal{H}} G_{prh}\delta_{pr} D_{pr} u_{rh} \\ + \quad & \sum_{p\in\mathcal{P}}\sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}} G_{phr}\left(D_{pr} + D'_{pr}\right) w_{hr} + \text{SumLHS(BCuts)} \end{aligned} \tag{5.31}$$

subject to

$$\sum_{h\in\mathcal{H}} u_{rh} = 1 \qquad \forall\, r \in \mathcal{R}, \tag{5.32}$$

$$\sum_{m\in\mathcal{M}_p} g_{pm} = 1 \qquad \forall\, p \in \mathcal{P}, \tag{5.33}$$

$$\sum_{h\in\mathcal{H}} w_{hr} = 1 \qquad \forall\, r \in \mathcal{R}, \tag{5.34}$$

$$\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\gamma_p\,\left(D_{pr}+D'_{pr}\right)\,w_{hr}$$
$$+\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}\beta_p\,\delta_{pr}D_{pr}u_{rh}\leq Q_h\,b_h \qquad \forall\,h\in\mathcal{H}, \tag{5.35}$$

(Constraints for the Set of BCuts) (5.36)

$$g_{pm},\,b_h,\,u_{rh},\,w_{hr}\in\{0,1\} \qquad \forall\,p\in\mathcal{P},\,h\in\mathcal{H},\,m\in\mathcal{M}_p. \tag{5.37}$$

In Display 8, we present the BD implementation procedure. We note that, in Display 8, $\varepsilon(>0)$, `IterNo`, `MaxIter`, `UB`, `LB` represent the stopping criteria, iteration counter, maximum number of Benders iterations, upper bound, and lower bound, respectively. We use $Z(\mathbf{BDSP})$ and $Z(\mathbf{BMP})$ to represent the objective function values of the dual of the Benders subproblem and the Benders master problem as defined in (5.19) and (5.31), respectively.

Display 8 Pseudo-code of the BD Approach.

Set $Z_{best}=\texttt{UB}=\infty$, $\texttt{IterNo}=0$, and $\widehat{\boldsymbol{\lambda}}=\widehat{\boldsymbol{\sigma}}=\widehat{\boldsymbol{\mu}}=\widehat{\boldsymbol{\tau}}=0$. Initialize `MaxIter` and ε.
Solve $\mathbf{BMP}(\mathbf{g},\mathbf{b}|\widehat{\boldsymbol{\lambda}},\widehat{\boldsymbol{\sigma}},\widehat{\boldsymbol{\mu}},\widehat{\boldsymbol{\tau}})$ and set $\texttt{LB}=Z(\mathbf{BMP})$.
while $(\texttt{UB}-\texttt{LB})/\texttt{LB}\geq\varepsilon)$ and $(\texttt{IterNo}<\texttt{MaxIter})$ **do**
    $\texttt{IterNo}=\texttt{IterNo}+1$
    Solve the $\mathbf{DBSP}(\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{\tau}\;|\widehat{\mathbf{g}},\widehat{\mathbf{u}},\widehat{\mathbf{w}})$ to obtain $\widehat{\boldsymbol{\lambda}},\,\widehat{\boldsymbol{\sigma}},\,\widehat{\boldsymbol{\mu}},\,\widehat{\boldsymbol{\tau}}$, and $Z(\mathbf{BDSP})$.
    Calculate $\texttt{UB}=Z(\mathbf{BMSP})+Z(\mathbf{BDSP})-$SumLHS(BCuts)
    if $(Z_{best}>\texttt{UB})$ **then**
        $Z_{best}=\texttt{UB}$
    end if
    Add the (Set of BCuts) to **BMP(.)** using $\widehat{\boldsymbol{\lambda}},\,\widehat{\boldsymbol{\sigma}},\,\widehat{\boldsymbol{\mu}},\,\widehat{\boldsymbol{\tau}}$ values.
    Solve $\mathbf{BMP}(\mathbf{g},\mathbf{b}|\widehat{\boldsymbol{\lambda}},\widehat{\boldsymbol{\sigma}},\widehat{\boldsymbol{\mu}},\widehat{\boldsymbol{\tau}})$ and set $LB=Z(\mathbf{BMP})$.
end while
Find $\mathbf{x},\mathbf{y}$ corresponding to $\widehat{\mathbf{g}},\widehat{\mathbf{u}},\widehat{\mathbf{w}}$ (i.e., solve $\mathbf{BSP}(\mathbf{x},\,\mathbf{y}|\widehat{\mathbf{g}},\widehat{\mathbf{u}},\widehat{\mathbf{w}})$).
Report $\widehat{\mathbf{g}},\widehat{\mathbf{b}},\widehat{\mathbf{u}},\widehat{\mathbf{w}},\widehat{\mathbf{x}},\widehat{\mathbf{y}}$ and the objective function value for (5.1).

V.2.3.1. Alternative Benders Cuts (BCuts)

The alternate multiple cuts for the master problem are given below.

Type 1: We define two new decision variables, $\psi_p^F \geq 0$ and $\psi_p^R \geq 0$, for each $p \in \mathcal{P}$, and add the following constraints that correspond to the $|\mathcal{P}|$ *single product forward cuts* and $|\mathcal{P}|$ *single product reverse cuts*, given by

$$\psi_p^F \geq \sum_{m \in \mathcal{M}_p} \sum_{r \in \mathcal{R}} \left(D_{pr} + D'_{pr}\right) \widehat{\tau}_{pm} g_{pm} - \sum_{r \in \mathcal{R}} \sum_{h \in \mathcal{H}} \left(D_{pr} + D'_{pr}\right) \widehat{\sigma}_{ph} w_{hr} \ \forall\, p \in \mathcal{P},$$

and

$$\psi_p^R \geq \sum_{m \in \mathcal{M}_p} \sum_{r \in \mathcal{R}} \delta_{pr} D_{pr} \widehat{\lambda}_{pm} g_{pm} - \sum_{r \in \mathcal{R}} \sum_{h \in \mathcal{H}} \delta_{pr} D_{pr} \widehat{\mu}_{ph} u_{rh} \quad \forall\, p \in \mathcal{P}.$$

The SumLHS(BCuts) term in the objective function (5.31) is replaced with $\sum_{p \in \mathcal{P}} \psi_p^F + \sum_{p \in \mathcal{P}} \psi_p^R$.

Type 2: We define a new decision variable $\psi_p \geq 0$ for each $p \in \mathcal{P}$. The constraints that correspond to the $|\mathcal{P}|$ *single product cuts* are given by

$$\begin{aligned} \psi_p \geq & \sum_{m \in \mathcal{M}_p} \sum_{r \in \mathcal{R}} \left((D_{pr} + D'_{pr} \widehat{\tau}_{pm}) + \delta_{pr} D_{pr} \widehat{\lambda}_{pm} \right) g_{pm} \\ & - \sum_{r \in \mathcal{R}} \sum_{h \in \mathcal{H}} \left(D_{pr} + D'_{pr}\right) \widehat{\sigma}_{ph} w_{hr} - \sum_{r \in \mathcal{R}} \sum_{h \in \mathcal{H}} \delta_{pr} D_{pr} \widehat{\mu}_{ph} u_{rh} \quad \forall\, p \in \mathcal{P}. \end{aligned}$$

The SumLHS(BCuts) term in the objective function (5.31) is replaced with $\sum_{p \in \mathcal{P}} \psi_p$.

V.3. An Alternative Formulation

In formulation **MP-CL**, constraint sets (5.2) and (5.3), together with constraint set (5.6), provide a special characterization on the reverse flows from the HCs to HPPs. More specifically, constraint set (5.2) ensures, through single reverse-link assignment, that all the returned products at each retailer location are sent to the set of open HCs. Further, for each product, constraint sets (5.3) and (5.6) require all the returned products be sent to the open HPP. As a consequence of these constraints, for each product p, the total quantity of product returns available at all the open HCs is equal to the quantity required at the open HPP. Hence, it suffices to the have the following set of inequalities in formulation **MP-CL** instead of constraint set (5.5).

$$x_{phm} \leq \sum_{r \in \mathcal{R}} \delta_{pr} D_{pr} u_{rh} \qquad \forall\, p \in \mathcal{P},\, h \in \mathcal{H},\, m \in \mathcal{M}_p. \tag{5.38}$$

Using a similar argument for the forward flows, we can replace constraint set (5.8), without affecting the feasible region of **MP-CL**, using the following set of inequalities.

$$y_{pmh} \leq \sum_{r \in \mathcal{R}} \left(D_{pr} + D'_{pr}\right) w_{hr} \qquad \forall\, p \in \mathcal{P},\, h \in \mathcal{H},\, m \in \mathcal{M}_p. \tag{5.39}$$

Replacing the constraint sets (5.5) and (5.8) with the inequalities (5.38) and (5.39), we obtain an alternate formulation of **MP-CL**, which we denote by **MP-CL-G**. This alternate formulation is very helpful for developing efficient solution algorithms to solve the subproblems in the BD framework. We briefly describe the alternative subproblems of the BD framework along with their properties and solution algorithms.

V.3.1. Dual Subproblem for Alternative Formulation

We define dual variables θ_{pm}, φ_{pm}, π_{phm}, and ξ_{phm} corresponding to the constraints (5.6), (5.7), (5.38), and (5.39), respectively. The alternative dual linear program **DBSP-G**$(\boldsymbol{\theta}, \boldsymbol{\xi}, \boldsymbol{\pi}, \boldsymbol{\varphi} \,|\widehat{\mathbf{g}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$ is stated as

$$
\begin{aligned}
\max \quad & \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p}\sum_{r\in\mathcal{R}}\delta_{pr}D_{pr}\widehat{g}_{pm}\theta_{pm} - \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p}\sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}}\delta_{pr}D_{pr}\widehat{u}_{rh}\pi_{phm} \\
+ & \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p}\sum_{r\in\mathcal{R}}\left(D_{pr}+D'_{pr}\right)\widehat{g}_{pm}\varphi_{pm} \\
- & \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p}\sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}}\left(D_{pr}+D'_{pr}\right)\widehat{w}_{hr}\xi_{phm} \qquad (5.40)
\end{aligned}
$$

subject to

$$\theta_{pm} - \pi_{phm} \leq G_{phm} \qquad \forall\, p\in\mathcal{P},\, h\in\mathcal{H},\, m\in\mathcal{M}_p, \quad (5.41)$$

$$\varphi_{pm} - \xi_{phm} \leq G_{pmh} \qquad \forall\, p\in\mathcal{P},\, h\in\mathcal{H},\, m\in\mathcal{M}_p, \quad (5.42)$$

$$\theta_{pm},\, \varphi_{pm} \text{ - free variables and } \xi_{phm},\, \pi_{phm} \geq 0 \qquad \forall\, p\in\mathcal{P},\, h\in\mathcal{H},\, m\in\mathcal{M}_p. \quad (5.43)$$

We observe that the dual subproblem **DBSP-G**$(\boldsymbol{\theta}, \boldsymbol{\xi}, \boldsymbol{\pi}, \boldsymbol{\varphi}\,|\widehat{\mathbf{g}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$, is separable in terms of the forward flow variables ($\boldsymbol{\xi}$ and $\boldsymbol{\varphi}$) and reverse flow variables ($\boldsymbol{\theta}$ and $\boldsymbol{\pi}$). Furthermore, we observe that the forward and reverse dual subproblems are separable for each product-specific HPP locations, which leads to *single HPP forward* and *single HPP reverse* subproblems.

V.3.1.1. Single HPP Forward Dual Subproblem

For each product-specific HPP $\bar{m}$, associated with product $\bar{p}$, the dual program of the single product forward subproblem, denoted by **FDBSP-G**$_{\bar{p}\bar{m}}(\boldsymbol{\xi}, \boldsymbol{\varphi}\,|\widehat{\mathbf{g}}, \widehat{\mathbf{w}})$, is given by

$$\max \quad \sum_{r\in\mathcal{R}}\left(D_{\bar{p}r}+D'_{\bar{p}r}\right)\widehat{g}_{\bar{p}\bar{m}}\varphi_{\bar{p}\bar{m}} - \sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}}\left(D_{\bar{p}r}+D'_{\bar{p}r}\right)\widehat{w}_{hr}\xi_{\bar{p}h\bar{m}} \qquad (5.44)$$

subject to

$$\varphi_{\bar{p}\bar{m}} - \xi_{\bar{p}h\bar{m}} \leq G_{\bar{p}\bar{m}h} \qquad \forall\, h\in\mathcal{H}, \qquad (5.45)$$

$$\varphi_{\bar{p}\bar{m}} \text{ - free variable and } \xi_{\bar{p}h\bar{m}} \geq 0 \qquad \forall\, h\in\mathcal{H}. \qquad (5.46)$$

V.3.1.2. Single HPP Reverse Dual Subproblem

For each product-specific HPP $\bar{m}$, associated with product $\bar{p}$, the dual program of the single product reverse subproblem, denoted by **RDBSP-G**$_{\bar{p}\bar{m}}(\boldsymbol{\theta}, \boldsymbol{\pi} \ |\widehat{\mathbf{g}}, \widehat{\mathbf{u}})$, is given by

$$\max \quad \sum_{r\in\mathcal{R}} \delta_{\bar{p}r} D_{\bar{p}r} \widehat{g}_{\bar{p}\bar{m}} \theta_{\bar{p}\bar{m}} - \sum_{h\in\mathcal{H}} \sum_{r\in\mathcal{R}} \delta_{\bar{p}r} D_{\bar{p}r} \widehat{u}_{rh} \pi_{\bar{p}h\bar{m}} \tag{5.47}$$

subject to

$$\theta_{\bar{p}\bar{m}} - \pi_{\bar{p}h\bar{m}} \leq G_{\bar{p}h\bar{m}} \qquad \forall\, h \in \mathcal{H}, \tag{5.48}$$

$$\theta_{\bar{p}\bar{m}} \text{ - free variable and } \pi_{\bar{p}h\bar{m}} \geq 0 \qquad \forall\, h \in \mathcal{H}. \tag{5.49}$$

V.3.2. Solving the Subproblems

For a given HPP, examining the dual subproblem **FDBSP-G**$_{\bar{p}\bar{m}}(\boldsymbol{\xi}, \boldsymbol{\varphi} \ |\widehat{\mathbf{g}}, \widehat{\mathbf{w}})$, we can clearly see that the coefficient of $\varphi_{\bar{p}\bar{m}}$ variable is equal to the summation of the coefficients of all the $\xi_{\bar{p}h\bar{m}}$ variables. Following this observation, for a given HPP, if $\widehat{g}_{\bar{p}\bar{m}} = 1$, then an optimal solution to this problem is given by, $\varphi_{\bar{p}\bar{m}} = \max_{h\in\mathcal{H}}\{G_{\bar{p}\bar{m}h}\}$ and $\xi_{\bar{p}h\bar{m}} = \varphi_{\bar{p}\bar{m}} - G_{\bar{p}\bar{m}h}$ for all $h \in \mathcal{H}$. Observe that, if the values of $\varphi_{\bar{p}\bar{m}}$ and $\xi_{\bar{p}h\bar{m}}$, for all $h \in \mathcal{H}$, are further increased by one unit, the value of the objective function as well as the left-hand side of the constraints (5.45) do not change. However, if $\widehat{g}_{\bar{p}\bar{m}} = 0$, the trivial solution to this problem is given by, $\varphi_{\bar{p}\bar{m}} = 0$ and $\xi_{\bar{p}h\bar{m}} = 0$ for all $h \in \mathcal{H}$. In this procedure, we obtain the values of all the dual variables associated with **FDBSP-G**$_{\bar{p}\bar{m}}(\boldsymbol{\xi}, \boldsymbol{\varphi} \ |\widehat{\mathbf{g}}, \widehat{\mathbf{w}})$, and hence, there is no need for a second phase problem.

Similarly to the **FDBSP-G**$_{\bar{p}\bar{m}}(\boldsymbol{\xi}, \boldsymbol{\varphi} \ |\widehat{\mathbf{g}}, \widehat{\mathbf{w}})$, for a given HPP, if $\widehat{g}_{\bar{p}\bar{m}} = 1$, an optimal solution to the dual subproblem **RDBSP-G**$_{\bar{p}\bar{m}}(\boldsymbol{\theta}, \boldsymbol{\pi} \ |\widehat{\mathbf{g}}, \widehat{\mathbf{u}})$, is given by $\theta_{\bar{p}\bar{m}} = \max_{h\in\mathcal{H}}\{G_{\bar{p}h\bar{m}}\}$ and $\pi_{\bar{p}h\bar{m}} = \theta_{\bar{p}\bar{m}} - G_{\bar{p}h\bar{m}}$ for all $h \in \mathcal{H}$. However, if $\widehat{g}_{\bar{p}\bar{m}} = 0$, the trivial solution to this problem is given by, $\theta_{\bar{p}\bar{m}} = 0$ and $\pi_{\bar{p}h\bar{m}} = 0$ for all $h \in \mathcal{H}$.

We note that, the optimal solutions thus obtained, do not depend on the values of the assignment variables $\widehat{\mathbf{u}}$ and $\widehat{\mathbf{w}}$. As a result, for a given instance, we can solve the dual subproblems associated with each HPP prior to solving the Benders master problem (given by (5.31)-(5.32)) and obtain $\sum_{p\in\mathcal{P}} 2|\mathcal{M}_p|$ Benders cuts, which we refer as *G-cuts* (discussed in the following section). In such an implementation, we can add all the G-cuts *upfront* to the master problem, and perform a single iteration of the Benders algorithm to obtain an optimal solution to the model **MP-CL-G**. However, preliminary test results clearly reveal that, such an implementation leads to a large number of G-cuts (associated with each HPP) being added to the master problem, which causes a steep increase in solution runtimes, especially for large problem instances. Therefore, in order to reduce the number of G-cuts being added to the master problem in each iteration of the algorithm, we add only the G-cuts corresponding to the HPP locations that are associated with the solution ($\widehat{\mathbf{g}}$) provided by the master problem. For this purpose, we modify the pseudo-code in Display 8 as follows. We solve the $\mathbf{DBSP\text{-}G}(\boldsymbol{\theta}, \boldsymbol{\xi}, \boldsymbol{\pi}, \boldsymbol{\varphi} \ |\widehat{\mathbf{g}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$ to obtain the optimal values of the dual variables $\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\xi}}, \widehat{\boldsymbol{\pi}},$ and $\widehat{\boldsymbol{\varphi}}$ corresponding to each HPP, in line 1 of the pseudo-code. We replace line 5 with the following step.

$$\text{Calculate } Z(\mathbf{BSP}) = \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p}\sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}} G_{phm}\delta_{pr}D_{pr}\widehat{u}_{rh}\widehat{g}_{pm} + \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}_p}\sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}} G_{pmh}(D_{pr}+D'_{pr})\widehat{w}_{hr}\widehat{g}_{pm} \quad (5.50)$$

In line 10, instead of the (set of BCuts), at each iteration, we add the alternative G-cuts that are described in the following section.

V.3.2.1. Benders Cuts (G-Cuts) Obtained Using the Alternative Formulation

The alternate multiple cuts obtained using the dual subproblem **DBSP-G**$(\boldsymbol{\theta}, \boldsymbol{\xi}, \boldsymbol{\pi}, \boldsymbol{\varphi} \,|\widehat{\mathbf{g}}, \widehat{\mathbf{u}}, \widehat{\mathbf{w}})$ are described below.

Type GA: For each HPP $m \in \mathcal{M}_p$, $p \in \mathcal{P}$, we derive forward and reverse cuts using the solutions of **FDBSP-G**$_{\bar{p}\bar{m}}(\boldsymbol{\xi}, \boldsymbol{\varphi} \,|\widehat{\mathbf{g}}, \widehat{\mathbf{w}})$ and **RDBSP-G**$_{\bar{p}\bar{m}}(\boldsymbol{\theta}, \boldsymbol{\pi} \,|\widehat{\mathbf{g}}, \widehat{\mathbf{u}})$, respectively. To this end, we define two new decision variables, $\psi_p^F \geq 0$ and $\psi_p^R \geq 0$, for each $p \in \mathcal{P}$. If $\widehat{g}_{\bar{p}\bar{m}} = 1$, then the constraints that correspond to the $|\mathcal{P}|$ *single HPP forward G-cuts* and $|\mathcal{P}|$ *single HPP reverse G-cuts* are given by

$$\psi_p^F \geq \sum_{r\in\mathcal{R}} \left(D_{pr} + D'_{pr}\right) \widehat{\varphi}_{pm} g_{pm} - \sum_{r\in\mathcal{R}}\sum_{h\in\mathcal{H}} \left(D_{pr} + D'_{pr}\right) \widehat{\xi}_{phm} w_{hr} \qquad \forall\, p \in \mathcal{P}, m \in \mathcal{M}_p,$$

and

$$\psi_p^R \geq \sum_{r\in\mathcal{R}} \delta_{pr} D_{pr} \widehat{\theta}_{pm} g_{pm} - \sum_{r\in\mathcal{R}}\sum_{h\in\mathcal{H}} \delta_{pr} D_{pr} \widehat{\pi}_{phm} u_{rh} \qquad \forall\; p \in \mathcal{P}, m \in \mathcal{M}_p.$$

The SumLHS(BCuts) term in the objective function (5.31) is replaced with $\sum_{p\in\mathcal{P}} \psi_p^F + \sum_{p\in\mathcal{P}} \psi_p^R$. However, if $\widehat{g}_{\bar{p}\bar{m}} = 0$, then the right-hand side of the aforementioned G-Cuts evaluates to a negative value, and, thus, these cuts become redundant. Moreover, utilizing the corresponding optimal solution (trivial) values of the subproblems for the case where $\widehat{g}_{\bar{p}\bar{m}} = 0$, we obtain G-Cuts ($\psi_p^F \geq 0$ and $\psi_p^R \geq 0$, for each $p \in \mathcal{P}$, $m \in \mathcal{M}_p$) that are redundant.

Type GB: Similarly, we derive a single cut for each HPP $m \in \mathcal{M}_p$, $p \in \mathcal{P}$. To this end, we define a new decision variable $\psi_p \geq 0$ for each $p \in \mathcal{P}$. Then, if $\widehat{g}_{\bar{p}\bar{m}} = 1$, the constraints that correspond to the $|\mathcal{P}|$ *single HPP G-cuts* are given by

$$\begin{aligned}\psi_p \;\; \geq \;\; & \sum_{r \in \mathcal{R}} \left(D_{pr} + D'_{pr}\right) \widehat{\varphi}_{pm} g_{pm} - \sum_{r \in \mathcal{R}} \sum_{h \in \mathcal{H}} \left(D_{pr} + D'_{pr}\right) \widehat{\xi}_{phm} w_{hr} \\ & + \sum_{r \in \mathcal{R}} \delta_{pr} D_{pr} \widehat{\theta}_{pm} g_{pm} - \sum_{r \in \mathcal{R}} \sum_{h \in \mathcal{H}} \delta_{pr} D_{pr} \widehat{\pi}_{phm} u_{rh} \qquad \forall \; p \in \mathcal{P}, m \in \mathcal{M}_p.\end{aligned}$$

The SumLHS(BCuts) term in the objective function (5.31) is replaced with $\sum_{p \in \mathcal{P}} \psi_p$. However, if $\widehat{g}_{\bar{p}\bar{m}} = 0$, as before, the right-hand side of the aforementioned G-Cuts evaluates to a negative value, and, these cuts become redundant. Moreover, utilizing the corresponding optimal solution (trivial) values of the subproblems, we obtain G-Cuts ($\psi_p \geq 0$ for each $p \in \mathcal{P}$, $m \in \mathcal{M}_p$) that are redundant.

Since the G-cuts corresponding the case where $\widehat{g}_{\bar{p}\bar{m}} = 0$ are redundant, we only consider the G-cuts obtained using the optimal solution values corresponding to the case where $\widehat{g}_{\bar{p}\bar{m}} = 1$.

V.3.2.2. Alternative Implementations of the G-Cuts

At each iteration of the Benders algorithm, similar to the Type 1 or Type 2 cuts (presented in Section V.2.3.1), we can use either the Type GA or Type GB cuts in the master problem. Another option is to use *both* the Type GA and Type 1 cuts (or both the Type GB and Type 2 cuts), which we denote by Type GA&1 (or Type GB&2).

On the other hand, we can further aggregate the Type GA or Type GB cuts to obtain *single product forward G-cuts* and *single product reverse G-cuts*, *single product G-cuts*, *forward G-cut* and *reverse G-cut*, and *single G-cut*.

V.3.3. Another Alternative Formulation

Another way to modify the formulation **MP-CL**, utilizing the previously mentioned characterization of the flows between HCs and HPPs, is to replace constraint set (5.6) and (5.7) with the following inequalities, in the formulation **MP-CL**.

$$x_{phm} \leq \sum_{r \in \mathcal{R}} \delta_{pr} D_{pr} g_{pm} \qquad \forall\, p \in \mathcal{P},\ h \in \mathcal{H},\ m \in \mathcal{M}_p, \tag{5.51}$$

$$y_{pmh} \leq \sum_{r \in \mathcal{R}} \left(D_{pr} + D'_{pr} \right) g_{pm} \qquad \forall\, p \in \mathcal{P},\ h \in \mathcal{H},\ m \in \mathcal{M}_p. \tag{5.52}$$

Replacing the constraint sets (5.6) and (5.7) with the inequalities (5.51) and (5.52), we obtain another alternative formulation, which we denote by **MP-CL-K**. We can apply the BD framework on formulation **MP-CL-K**, and, similar to the G-cuts, we can obtain the *K-cuts* (for each product $p \in \mathcal{P}$ and HC location $h \in \mathcal{H}$).

Our preliminary computational testing, in terms of both solution quality and runtimes, shows poor performance of all types of the K-cuts, all combinations of the K-cuts with the Type 1, Type 2, Type GA, Type GB, Type GA&1, and Type GB&2 cuts, as well as the single product forward G-cuts, single product reverse G-cuts, single product G-cuts, single G-cut. Hence, for brevity, we report the computational results of the algorithm implementations using only the Type 1, Type 2, Type GA, Type GB, Type GA&1, and Type GB&2 cuts.

V.4. Computational Experiments

In this section, we first develop a testbed of random data instances and conduct a computational study to establish the performance of the proposed solution approaches. Since our problem setting is a generalization of the **URP** setting, while generating our testbed, we utilize a similar approach as the one given in Section III.4. To benchmark the performance of the Benders implementation, we use the B&C approach (CPLEX). We also employ CPLEX to solve the Benders master problem presented in Section V.2.3. We implement the solution approaches and perform the runs on a machine with a 2.66 GHz Intel XEON processor and 24 GB RAM.

V.4.1. Random Test Instance Generation

In order to develop a set of test instances that are of realistic size, we vary the number of products $|\mathcal{P}|$, the number of retailers $|\mathcal{R}|$, and the number of potential HC locations $|\mathcal{H}|$. As in Section III.4, we consider two levels for $|\mathcal{P}|$ (5 and 10), three levels for $|\mathcal{R}|$ (60, 90, and 120), and two levels for $|\mathcal{K}|$ (25 and 35) to obtain 10 different problem classes as shown in Table 15. We generate 10 random instances for each of these problem classes.

We use uniform distributions to randomly create the number of HPP for each product p as shown in Table 16. Uniform distributions are also employed, as shown in Table 16, to generate the demands (D_{pr} and D'_{pr}), return fractions (δ_{pr}), recovery fractions (α_{ps}), and storage capacity coefficients (γ_p and β_p). Also as shown in Table 16, we randomly generate capacities Q_h for the HCs. Note that we use the notation TC, to represent the total capacity requirement, given by $\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}}(\gamma_p D_{pr}+\gamma_p D'_{pr}+\beta_p\delta_{pr}D_{pr})$.

V.4.2. Computational Results

As we have noted earlier, we solve each instance using the B&C and the BD approaches with the alternative types of strong cuts developed in Sections V.2.3 and V.3.2. As in the previous computational settings, we avoid the tail-off effect in the B&C approach (CPLEX), by setting the tolerance for stopping criterion to 1 percent gap. Also, as before, while solving the master problem in the BD approach, we employ

Table 15 **CLP**: Problem Classes Used in Computational Testing.

Class	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{H}\rvert$
C1	5	60	25
C2	5	60	35
C3	5	90	25
C4	5	90	35
C5	5	120	25
C6	5	120	35
C7	10	60	25
C8	10	60	35
C9	10	90	25
C10	10	90	35

an early stopping criterion of 30% for the initial iteration. In successive iterations, we set the stopping criterion to 0.009% optimality gap. Furthermore, we limit the maximum number of iterations in the BD approach to five and set the the tolerance for stopping criterion to 1 percent as in the B&C approach. In the following, we compare the B&C and BD approaches with Type 1, Type 2, Type GA, Type GB, Type GA&1, and Type GB&2 cuts.

Considering the 100 instances, we summarize the average and the maximum optimality gaps upon termination of the approaches in Table 17. We note that, in Tables 17, 18, and 19, row minimums for the BD related results are listed in bold. Notably, the use of GB&2 in the BD approach appears to be the most effective, as it provides the lowest average and maximum optimality gaps in all of the problem classes. This provides empirical evidence as to the potential benefit of using Type GB&2 in the BD framework.

Table 16 **CLP**: Distributions for Demand, Return Fraction, Recovery Fraction, Product Capacity Coefficients and Storage Capacity Values.

Parameter	Value
$\lvert\mathcal{M}_p\rvert$	Uniform[2, 15]
D_{pr}	Uniform[250, 350]
D'_{pr}	Uniform[450, 550]
δ_{pr}	Uniform[0.7, 0.9]
α_{pm}	Uniform[0.8, 0.98]
γ_p	Uniform[1, 10]
β_p	Uniform[1, γ_p]
Q_h	Uniform[0.1, 0.3] * TC

Table 17 **CLP**: Comparison of the Optimality Gaps Upon Termination.

Class	Average Optimality Gap (%)							Maximum Optimality Gap (%)						
	B&C	Cuts Type						B&C	Cuts Type					
		1	2	GA	GB	GA&1	GB&2		1	2	GA	GB	GA&1	GB&2
C1	1.0	**0.5**	**0.5**	1.0	1.0	**0.5**	**0.5**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
C2	1.0	**0.0**	**0.0**	0.3	0.3	**0.0**	**0.0**	1.0	**0.1**	**0.1**	0.8	0.8	**0.1**	**0.1**
C3	1.0	**0.0**	**0.0**	0.4	0.4	0.1	**0.0**	1.0	**0.1**	**0.1**	1.0	1.0	0.3	**0.1**
C4	1.0	0.1	0.1	0.5	0.5	**0.0**	**0.0**	1.0	0.2	**0.1**	1.0	1.0	**0.1**	**0.1**
C5	0.7	**0.1**	**0.1**	0.4	0.4	**0.1**	**0.1**	1.0	**0.3**	**0.3**	0.9	0.9	**0.3**	**0.3**
C6	1.0	**0.1**	**0.1**	0.4	0.5	**0.1**	**0.1**	1.0	**0.4**	**0.4**	1.0	1.0	**0.4**	**0.4**
C7	1.0	**0.1**	**0.1**	0.6	0.6	**0.1**	**0.1**	1.0	**0.2**	**0.2**	0.9	0.9	**0.2**	**0.2**
C8	1.2	**0.1**	**0.1**	0.6	0.6	**0.1**	**0.1**	3.1	**0.2**	**0.2**	1.0	1.0	**0.2**	**0.2**
C9	1.0	**0.1**	**0.1**	0.6	0.6	**0.1**	**0.1**	1.0	**0.2**	**0.2**	0.9	0.9	0.3	**0.2**
C10	4.3	**0.1**	**0.1**	0.4	0.4	**0.1**	**0.1**	5.9	**0.1**	**0.1**	0.7	0.7	**0.1**	**0.1**

In Table 18, we present a comparison of the time required to obtain the solution by the B&C approach and the BD approaches. The results show that the BD approach with Type GA&1 or Type GB&2 cuts performs better in terms of solution times than the BD approach with the other types of alternative cuts and the B&C approach. Also, the results reported in Table 18 indicate that the solution times for the B&C approach increases drastically for those larger instances.

Table 18 **CLP**: Comparison of the Solution Times.

Average of Solution Times (sec.)							
Class	B&C	Cuts Type					
		1	2	GA	GB	GA&1	GB&2
C1	299	45	46	51	47	28	**26**
C2	936	93	93	118	109	**67**	71
C3	857	142	142	99	110	**90**	91
C4	2367	346	343	502	459	**216**	225
C5	3927	836	760	630	636	435	**415**
C6	2428	299	329	681	876	264	**218**
C7	1467	79	85	244	197	**58**	68
C8	4198	184	179	444	364	184	**151**
C9	4955	305	352	643	694	257	**242**
C10	6480	2803	3356	5747	5984	**866**	914
Maximum of Solution Times (sec.)							
Class	B&C	Cuts Type					
		1	2	GA	GB	GA&1	GB&2
C1	877	102	106	170	159	**58**	59
C2	1540	152	152	223	204	103	**98**
C3	1312	207	207	170	175	**150**	159
C4	5082	548	531	939	826	411	**371**
C5	6291	1848	2494	1591	**1017**	1675	1229
C6	4239	489	468	2430	3233	626	**328**
C7	2494	246	364	699	382	**139**	180
C8	10800	584	579	782	596	362	**348**
C9	9430	1079	1364	2353	3256	**669**	689
C10	10801	9387	10801	10801	10801	**6544**	8803

Table 19 reports the average and the maximum number of iterations required by the BD approach with alternative types of strong cuts. Recall that, at each iteration of the BD approach, we add $2|\mathcal{P}|$ cuts and $|\mathcal{P}|$ cuts using the Type 1 and Type 2 cuts, respectively. In the BD approaches using Type GA and Type GB cuts, at each iteration, we add at most $2|\mathcal{P}|$ cuts and $|\mathcal{P}|$ cuts, respectively. Moreover, in the case of BD implementations using Type GA and Type GB cuts, the maximum number of iterations are limited by the total number of G-cuts, which are $\sum_{p\in\mathcal{P}} 2|\mathcal{M}_p|$ and $\sum_{p\in\mathcal{P}} |\mathcal{M}_p|$, respectively. In the BD approaches using Type GA&1 and Type GB&2 cuts, at each iteration, we add at most $4|\mathcal{P}|$ cuts and $2|\mathcal{P}|$ cuts, respectively. In Table 19, we observe that both the Type GA&1 and Type GB&2 cuts provide the smallest values for the average and the maximum number of iterations.

Table 19 **CLP**: Comparison of the Number of Iterations.

Class	Average Number of Iterations						Maximum Number of Iterations					
	Cuts Type						Cuts Type					
	1	2	GA	GB	GA&1	GB&2	1	2	GA	GB	GA&1	GB&2
C1	**1.0**	**1.0**	2.1	2.1	**1.0**	**1.0**	**1**	**1**	3	3	**1**	**1**
C2	**1.0**	**1.0**	1.5	1.5	**1.0**	**1.0**	**1**	**1**	2	2	**1**	**1**
C3	**1.0**	**1.0**	1.2	1.2	**1.0**	**1.0**	**1**	**1**	2	2	**1**	**1**
C4	1.2	1.7	2.1	2.1	**1.0**	**1.0**	3	4	3	3	**1**	**1**
C5	1.1	1.3	1.8	1.8	**1.0**	**1.0**	2	3	2	2	**1**	**1**
C6	1.4	1.3	2.0	2.0	**1.0**	**1.0**	3	3	4	4	**1**	**1**
C7	1.3	1.5	1.9	1.9	**1.0**	**1.0**	3	4	3	3	**1**	**1**
C8	1.3	1.4	2.1	2.1	**1.0**	**1.0**	3	3	2	2	**1**	**1**
C9	1.4	1.2	2.0	2.0	**1.0**	**1.0**	4	3	2	2	**1**	**1**
C10	1.5	1.7	2.2	2.3	**1.0**	**1.0**	3	4	3	3	**1**	**1**

V.5. Concluding Remarks

In this chapter, we consider the **CLP**, which is a multi-product CLSC network design problem, where we locate the network facilities while determining the material flows in the whole network so as to minimize the processing, transportation, and fixed location costs. We develop alternative mathematical formulations that models the flow variables separately for each stage in the network. These formulations lend themselves to efficient Benders reformulation.

On the methodological side, we provided exact solution approaches based on the BD approach using alternative formulations of the **CLP**, which perform better than the B&C approach. In this context, we provided efficient dual problem solution methods that generate strong Benders cuts for the alternative formulations. Furthermore, we determined that, in this problem setting, different combinations of alternative multiple Benders cuts, in comparison to the use of strong Benders cuts, generated stronger lower bounds and promoted faster convergence.

CHAPTER VI

CONCLUSIONS AND FUTURE DIRECTIONS

An optimal network design for the CLSCs requires simultaneous consideration of both the forward and reverse flows, instead of an independent and sequential modeling approach to the forward and reverse network design. This integrated approach is a key to the network design as it impacts the economic viability and cost performance of the underlying CLSC.

VI.1. Contributions

The models and solution approaches in this dissertation aim at developing quantitative decision-making tools to evaluate different transformation strategies and provide cost effective solutions to the integrated CLSC network design. From the modeling perspective, this dissertation extends the previous work by considering multiple products, separate manufacturing and remanufacturing facilities, and indirect shipments via distribution centers.

More specifically, in this dissertation, we consider three different practical settings for the underlying multi-product closed-loop supply chain, namely an Uncapacitated Remanufacturing Network Design Problem (**URP**), Capacitated Remanufacturing Network Design Problem (**CRP**), and Closed-Loop Network Design Problem (**CLP**). In the **URP** and **CRP** settings, we extend the existing forward channel infrastructure to accommodate distinct reverse channel infrastructure to coordinate the forward and reverse flows, where as, in the **CLP** setting, we design the entire CLSC network by considering hybrid facilities.

For these three problems settings, we formulate MILPs to determine the optimal locations of the network facilities along with the integrated forward and reverse flows such that the total cost of facility location, processing and transportation is minimized. The network flow structures underlying these models make them amenable for efficient solution approaches using Benders decomposition (BD) framework.

For the first setting (**URP**), we develop an efficient dual solution approach to generate strong Benders cuts. In addition to the classical single Benders cut approach, we propose three different approaches for adding multiple Benders cuts. We present computational results that illustrate the superior performance of the proposed solution methodology with multiple Benders cuts in comparison to the B&C approach and the traditional BD approach with a single cut.

For the second setting (**CRP**), we devise two tabu search heuristics in which we effectively combine simple neighborhood search functions utilizing moves and exchanges to improve the efficiency of exploration. We propose a transshipment heuristic to quickly, but effectively, estimate the objective function value of a feasible solution in the course of a tabu search. We also present a BD approach that incorporates the tabu search heuristics and the strong Benders cuts to facilitate faster convergence and improve computational efficiency, especially for large scale instances. We present our computational results illustrating the superior performance of the solution algorithms developed based on the heuristics and BD framework in terms of both solution quality and computation time.

For the third setting (**CLP**), considering different alternative formulations, we present BD framework that utilizes the dual solution approach for obtaining strong cuts. We also present different approaches for combining these alternate strong cuts to increase computational efficiency. We present computational results and compare the computational performance among the alternative strong cuts.

VI.2. Foundation for Future Research

Immediate extensions of our work, from the modeling perspective, would be to consider the variants of our models that account for dynamic and stochastic demand at the retailer locations, vehicle routing strategies instead of direct shipments, and inclusion of inventory decisions in the design of CLSCs. The development of efficient solution approaches for such variants would be an important contribution to the literature by expanding the set of quantitative decision-making tools available for the design of CLSC networks.

In the same vein, investigating the efficiency of a Lagrangian heuristic–which is known to be effective for forward facility location problems–for the CLSC network design problems considered in this dissertation and a comparison of this approach with BD framework may be informative.

A promising direction for future research is to concentrate on the methodological contribution of our work and study the generalized use of multiple Benders cuts on other optimization problems in which the subproblem is separable. We can also examine the combined-use of simple neighborhoods in heuristic search and alternative ways of integrating heuristics within a BD framework as well as alternative cut disaggregation schemes.

Another interesting direction for future research is to focus on the test instance generation scheme we developed, and examine how it can be generalized for other problem settings in which cost trade-offs are instrumental in decision-making.

REFERENCES

Akçalı, E., S. Çetinkaya, H. Üster. 2007. Network design for reverse and closed-loop supply chains: An annotated bibliography of models and solution approaches. Forthcoming in *Networks*.

Beamon, B.M., C. Fernandes. 2004. Supply-chain network configuration for product recovery. *Production Planning and Control* **15** 270–281.

Benders, J.F. 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* **4** 238–252.

Birge, J. R., F. Louveaux. 1988. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research* **34** 384–392.

Bloemhof-Ruwaard, J.M., M. Fleischmann, J.A.E.E. van Nunen. 1999. Reviewing distribution issues in reverse logistics. M.G. Speranza, P. Stähly, eds., *New Trends in Distribution Logistics*. Springer-Verlag, New York, NY.

Brown, G.G., G.W. Graves, M. D. Honczarenko. 1987. Design and operation of a multicommodity production/distribution system using primal goal decomposition. *Management Science* **33** 1469–1480.

Dekker, R., M. Fleischmann, K. Inderfurth, L.N. Van Wassennhove, eds. 2004. *Reverse Logistics: Quantitative Models for Closed-Loop Supply Chains*. Springer-Verlag, Berlin.

Fleischmann, M. 2001. Reverse logistics network structures and design. ERIM Report Series ERS-2001-52-LIS Erasmus Research Institute of Management. Available at http://ideas.repec.org/p/dgr/eureri/2001109.html.

Fleischmann, M., P. Beullens, J.M. Bloemhof-Ruwaard, L.N. Van Wassenhove. 2001. The impact of product recovery on logistics network design. *Production and Operations Management* **10** 156–173.

Fleischmann, M., J.M. Bloemhof-Ruwaard, R. Dekker, E. van der Laan, J.A.E.E. van Nunen, L. N. Van Wassenhove. 1997. Quantitative models for reverse logistics: A review. *European Journal of Operational Research* **103** 1–17.

Fleischmann, M., H.R. Krikke, R. Dekker, S.D.P. Flapper. 2000. A characterization of logistics networks for product recovery. *Omega* **28** 653–666.

Garey, M. R., D.S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA.

Geoffrion, A. M., G. W. Graves. 1974. Multicommodity distribution system design by Benders decomposition. *Management Science* **20** 822–844.

Giuntini, R. 2001. *The US Market Size of Capital Goods Remanufacturing Process Expenditures*. OEM Product-Services Institute, Lewisburg PA.

Glover, F. 1989. Tabu search - Part 1. *ORSA Journal on Computing* **1** 190–206.

Glover, F. 1990. Tabu search - Part II. *ORSA Journal on Computing* **2** 4–32.

Glover, F., M. Laguna. 1997. *Tabu search*. Kluwer Academic Publishers, Norwell, MA.

Guide, V. D. R., T. P. R. Harrison, L. N. Van Wassenhove. 2003. The challenge of closed-loop supply chains. *Interfaces* **33** 3–6.

Jayaraman, V., H. Pirkul. 2001. Planning and coordination of production and distribution facilities for multiple commodities. *European Journal of Operational Research* **133** 394–408.

Keskin, B. B., H. Üster. 2007. Meta-heuristic approaches with memory and evolution for a multi-product production/distribution system design problem. *European Journal of Operational Research* **127** 663–682.

Krikke, H., J. Bloemhof-Ruwaard, L.N. Van Wassenhove. 2003. Concurrent product and closed-loop supply chain design with an application to refrigerators. *International Journal of Production Research* **41**(16) 3689–3719.

Langevin, A., S. Riopel. 2005. *Logistics Systems: Design and Optimization*. Springer, New York, NY.

Lu, Z., N. Bostel. 2007. A facility location model for logistics systems including reverse flows: The case of remanufacturing activities. *Computers and Operations Research* **34** 299–323.

Lund, R.T. 1996. *The Remanufacturing Industry: Hidden Giant*. Boston University, Boston, MA.

Magnanti, T. L., R. T. Wong. 1981. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research* **29** 464–484.

Parkinson, H. J., G. Thompson. 2003. Analysis and taxonomy of remanufacturing industry practice. *Part E-Journal of Process Mechanical Engineering* **217** 243–256.

Pirkul, H., V. Jayaraman. 1996. Production, transportation, and distribution planning in a multi-commodity tri-echelon system. *Transportation Science* **30** 291–302.

Pyke, D. F., M. A. Cohen. 1994. Multiproduct integrated production-distribution systems. *European Journal of Operational Research* **74** 18–49.

Sahyouni, K., R. C. Savaskan, M. S. Daskin. 2007. A facility location model for bidirectional flows. *Transportation Science* **41** 484–499.

Sim, E., S. Jung, H. Kim, J. Park. 2004. A generic network design for a closed-loop supply chain using genetic algorithm. Kalyanmoy D., Riccardo P., W. Banzhaf, H.-G. Beyer, E.K. Burke, P.J. Darwen, D.Dasgupta, D. Floreano, J.A. Foster, M. Harman, O. Holland, P.L. Lanzi, L. Spector, A. Tettamanzi, D. Thierens, A.M. Tyrrell, eds., *GECCO (2)*, *Lecture Notes in Computer Science*, vol. 3103. Springer, Berlin. 1214–1225.

Simchi-Levi, D., P. Kaminsky, E. Simchi-Levi. 2004. *Managing the Supply Chain*. McGraw-Hill, New York, NY.

Van Roy, T. J. 1986. A cross decomposition algorithm for capacitated facility location. *Operations Research* **34** 145–163.

Wentges, P. 1996. Accelerating Benders' decomposition for the capacitated facility location problem. *Mathematical Methods of Operations Research* **44** 267–290.

VITA

Gopalakrishnan Easwaran earned his Bachelor of Engineering degree in mechanical engineering (sandwich curriculum) from PSG College of Technology in 2000. He joined the industrial engineering graduate program at Texas A&M University in 2001. In August 2003, he earned his Master of Science degree in industrial engineering with a thesis titled *"Design and Development of a Vehicle Routing System under Capacity, Time-windows and Rush-order Reloading Considerations,"* under the guidance of his advisor, Dr. Sıla Çetinkaya. In September 2003, Easwaran enrolled in the doctoral program in the Department of Industrial and Systems Engineering at Texas A&M University. He pursued research in the areas of supply chain logistics and applied optimization.

During his studies, Easwaran has worked as a research assistant/associate on research projects for a variety of firms including PSG Industrial Institute (India), Pricol Inc. (India), Hindustan Aeronautics Limited (India), DRDL (India), Master Halco (San Antonio and Dallas, TX, USA), Frito-Lays and PepsiCo (Plano, TX, USA).

In August 2008, after completing his Ph.D., Easwaran will join the faculty in the Department of Engineering at the St. Mary's University, San Antonio, TX. As a faculty member of the Industrial Engineering program, he will teach and perform research in supply chain management and applied optimization. Easwaran can be reached at the following address:

1. P. N. Street, S. S. Colony

Madurai 625 010.

India.